

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Getting Started on the MSACCESS Forum

Microsoft Forums on CompuServe. A variety of forums are facilitated by Microsoft and provide interactive dialog with a worldwide community of Microsoft customers. You can exchange information with peer users and Microsoft support engineers. There is no support charge, but CompuServe connect charges do apply. For sales information, call CompuServe at (800) 848-8199, representative 230.

Genie access requires a personal computer, an asynchronous modem (300, 1200, or 2400 baud), and a communications package (such as the one in Microsoft Works). For information about GENie, call GENie Client Services at (800) 638-9636.

[Membership Announcement](#)

[Introduction](#)

[Hard and Fast Rules](#)

[Manners and Ambience](#)

- 3.1 Manners
- 3.2 Who's Microsoft and who isn't
- 3.3 What to do if you're upset

[Emoticons and Acronyms](#)

- 4.1 Emoticons
- 4.2 Acronyms

[Forum Organization -- Messages, Threads, Sections](#)

- 5.1 Message Traffic and Scroll Rate
- 5.2 What the Heck All This Means to You

[Asking Questions](#)

- 6.1 Phrasing Your Question
- 6.2 Uploading Files
- 6.3 Responding to Answers
- 6.4 Acknowledgments
- 6.5 If You Don't Get an Answer

[Tips for the Active Participant](#)

- 7.1 Thread Integrity and Continuity
- 7.2 Message Etiquette
- 7.3 On-line vs Off-line

[Sysops](#)

[Libraries](#)

- 9.1 Compressed Files
- 9.2 Uploading files

[Afterword - The Spirit of Contribution](#)

Membership Announcement

To join the MS Access Forum, select option #8 from the Forum menu or type JOIN at any Forum command prompt or choose Join from the appropriate menu. Your first and last name is all that is required. There are no special fees or dues other than your normal CompuServe connect charges.

Membership privileges are enabled automatically, and allow you full participation within the forum. All privileges remain in effect unless specifically deleted by the SYSOPS as long as you access the Forum periodically. The membership file is regularly (90 days) purged of inactive members to improve system operation. If this should occur, all you need to do is sign up again and your membership privileges will be enabled.

Forum members may access the Conferencing area, Libraries, and Message Board, which has been broken down into various sections. To see a listing of the Message section names, Library names, or Conference Rooms available, please type "NAMES" upon entry to the Forum.

Online help is available at virtually every level of the Forum. When in doubt, try entering a "?" or the word "HELP". If you are new to the CompuServe Forum environment, please be sure to visit the Forum INSTRUCTIONS area.

INTRODUCTION

Welcome to the Microsoft Access Support Forum! We look forward to your questions and contributions.

The MSACCESS Forum is an extraordinary resource for people who use, or are considering using, Microsoft products. Forum members are knowledgeable, unselfish, and available virtually 24 hours a day, 7 days a week to assist you. They will not only provide you with the information and help you need, but they'll do so with a generosity of spirit that truly means "support."

The forum is a place to get your questions answered -- Microsoft tech support is here (Monday through Friday) as well as many people who are very knowledgeable in Microsoft products, hardware, other software, and just about everything else (e.g. music, Star Trek, medieval literature, the list goes on and on).

Yet the forum also offers more than a place to come for help...

- you can learn about new trends in the industry and the philosophy of database approach in general
- you can share your thoughts about Microsoft and our line of products, request changes and enhancements, and gripe if you need to
- you can contribute your own knowledge and have the satisfaction of helping others
- and you can just hang out and chat with people who might become some of your best friends.

Our main purpose here is to provide information, guidelines and tips to help you get "up to speed" as quickly as possible.

Although we have a responsibility to watch this forum and see that questions are answered, problems resolved, and concerns are addressed, we hope that you will play an active part participating in the process. The forum works best when we all contribute and help each other. Not only does this make for good technical support, it also helps us to determine where the product can stand some improvement in order to prevent questions in the first place. We also get a good idea of who our customers are and what they need to be more productive.

HARD AND FAST RULES

These are simple.

This forum requests that you use your full name in your CompuServe (CIS) identification. If you have omitted your first name or last name, please change it.

To change your name, get on-line in CompuServe and type GO MSACCESS to get to the CompuServe prompt. Then type OPTIONS which will show you a menu of options you can change. Choose option 4 (name), enter the new name, press <RETURN> and then tell CIS to make the change permanent. You can also do this quickly by typing "OPT;N <newname>;P" which will bypass the menus.

If you normally use a company CompuServe account with the company name but plan to be a regular participant, please change the identification (at least locally in this forum) to your name and enclose the company name (or abbreviation) in brackets after your name. If several people will be using the *same* account, each individual should include his or her complete name in the body of the message.

Please refrain from using profanity, being abusive, advocating illegal actions, or performing any act that may injure another. Microsoft reserves the right to deny access to the forum to anyone posting offensive messages.

This isn't a "hard and fast rule" but it is highly recommended -- please do not type your messages all in upper case (use mixed case as you would in a letter). Upper case is hard to read and is considered "SHOUTING" on CompuServe -- you'll see what I mean if you read enough messages.

There are no other "rules" but there are some conventions of behavior that will ease your life in the forum. Please see the "Manners and Ambience" section, next.

MANNERS AND AMBIENCE

This is the MSACCESS forum, so naturally what brings people together is an interest in Microsoft products -- most of the people here are working with Microsoft Access. These include developers, people using the products to manage their own data, and hobbyists. The range of skill and expertise is very wide -- from the loftiest Access Basic guru to the absolute beginner. Experts and beginners are equally valued -- beginners sometimes ask the best questions!

In addition to the people who write messages on the forum, there is another group we (affectionately) call "lurkers" -- people who monitor the forum (reading messages, downloading files) but for one reason or another haven't chosen to write any messages to us. We might tease lurkers from time to time but we value them as well.

Although intended as a "working" forum for sharing information and professional expertise, this forum is also a friendly and informal place. People here can become quite close -- and a newcomer will hopefully find him/herself quickly accepted. When you join the forum, "old timers" will treat you as a friend as long as you behave yourself (smile).

As with any group of friends, you'll find that there are lots of "in" jokes, many teasing comments, heated debates, occasional spats and feuds, people having a bad day who may be cranky -- the usual stuff of human interaction. You'll find the reason we're friends is because we share common interests, are working in generally the same field, and appreciate the opportunity to communicate. For some of us who are isolated, the forum members are the only people we know who understand what we're talking about (grin).

So, when you "walk" into the forum, it will be more like walking into a large party with people who share professional interests than like walking into an office or a professional association. It's kind of like a user's group with no specific agenda. As with all CompuServe forums, forum members range widely in personality, geography, race, sex, religion, etc. and this is **definitely** an equal opportunity forum.

3.1 Manners

So, now to manners. Basically all we ask is that you be considerate of everybody else. When I say "considerate", I mean **consider** that the person on the other end of your message is a human being with ups and downs, strong and weak points, good days and bad days. Consider that a person may reside in a non-English speaking country and use English as a second language. Consider that a person might be handicapped or that a person may not be the sex you **think** he or she is based on his or her name.

Also, **please** consider that people here represent different nationalities, races, ethnic groups, religions, abilities/disabilities, and sexes, and **respect** these differences. In this forum, it is important to us that we provide an environment where we **all** feel safe. If any one of us is ridiculed or insulted, we all lose since we lose their enthusiasm to participate with us and contribute to the work we do.

Consideration can be most important when you are upset. Within the bounds of consideration, you can complain, whine or throw tantrums, if you need to -- I think we all have at one time or another. But it takes a certain amount of skill to complain, whine or throw a tantrum politely and with consideration for others. If you're upset, consider how your message might affect **you** if you were to receive it.

If you're considerate, you shouldn't run into too many problems here. But I must warn you that you are subject to teasing (because you're considered a **friend** from the beginning -- please don't take this personally) and if you should stray beyond the bounds of what's considered "appropriate" behavior, you may receive a gentle admonition. If you are really out of line, a Sysop may decide to step in -- a Sysop is a final authority on behavior, and even long-term forum members quake before a Sysop's pronouncements (smile).

As a rule of thumb, a belligerent attitude is frowned upon -- it tends to make us uncomfortable because it upsets the balance of friendliness and consideration, retards the free flow of communication, and, frankly, we seem to be able

to have very "loud" arguments without resorting to this approach.

3.2 Who's Microsoft and who isn't

People on the forum who work for Microsoft tech support have [MSFT] or "Sysop" after their names in their identification. If somebody doesn't have [MSFT] or "Sysop" after his or her name, s/he is *not* an employee of the company and is just being helpful. Some forum members are so helpful, people sometimes assume they work for Microsoft even if they don't.

You may be interested to know that senior managers and many of the staff at Microsoft review the forum messages every day. They are all paying attention to what you have to say and their products reflect this interest. Also various Microsoft staff members will contribute to the forum from time to time. They will either have (Microsoft) after their name or identify themselves in their signature.

3.3 What to do if you're upset

It's not uncommon for people to log onto the forum for the first time because they're upset

- because they've spent hours and hours working on a problem and can't solve it
- because something about Access doesn't work the way it should or the way they think it should (often there's a difference)
- because they've tried calling tech support and haven't received an acceptable answer yet
- because they had something working in one product that suddenly doesn't work in an upgrade.

We understand and sympathize -- and it just may be that we can help you if you give us a chance. If this is true for you, here's what we recommend: first, *consider* the other forum members (see Manners) and second, start your message with "I'm upset" (this lets us know that you're upset and we'll be sympathetic - that's a promise). Then try to explain your problem as succinctly as possible (for further guidelines -- see Asking Questions, section 6.0).

EMOTICONS AND ACRONYMS

You may have noticed that humorous comments have so far been expressed with such things as (grin) or (smile). One of the things that most puzzles newcomers is the use of acronyms and what are called "emoticons". Here is an explanation of the important ones.

4.1 Emoticons

In an electronic medium such as the forum, it's pretty hard to get across certain elements of face to face communications that would otherwise be expressed with tone of voice, inflection or body language -- elements like humor, irony, sadness and various non-verbal "comments" on what is being discussed. Consequently, certain conventions have arisen to express this kind of thing.

Technically, an emoticon is an "icon designating emotions" depicted with symbols. Perhaps the most common of which is:

`:-)` a smiling face on its side (tilt your head to the left to read it).

grin either the previous comment was meant humorously or one is responding with a laugh to somebody else's humorous comment -- perhaps the most widely used symbol on this forum.

smile usually means that one is smiling at somebody else's comment or could refer to one's own comment that is slightly humorous.

`<gd&r>` grinning, ducking and running -- the previous comment was a "tease" and the person issuing the tease is now hiding out from his or her target. Some variations of this are a simple `<gd>` (grinning and ducking) or `<gd&rvvf>` (grinning, ducking and running, very very fast).

`<rofl>` or `<rof,l>` rolling on the floor laughing -- the ultimate acknowledgement of somebody else's humor or an indication that you think you just said something yourself that is pretty ridiculous.

`* or _` as in `*emphasize*` or `_emphasize_` equivalent to italics or underlining.

(There are literally hundreds of others.) However, the term is also being used to apply to text-based equivalents which are more commonly used on this forum.

Creative people on the forum will sometimes make up their own emoticons from time to time but they almost always explain them since nobody would otherwise know what they're talking about `<g>`.

4.2 Acronyms

Many acronyms have also become standard -- mainly to save typing. Some of the common ones are explained here. Any others you encounter, you'll have to figure out for yourself, but once you've got the "trick" down, it's kind of fun.

General CIS-wide acronyms:

BTW	By the way
CIS	CompuServe Information Service
FWIW	For what it's worth
IAC	In any case (also IAE -- in any event)
IANAL	I am not a lawyer
IMO	In my opinion

IMHO	In my honest/humble opinion (in the latter case a sure sign the opinion is not going to be humble at all <g>)
IOW	In other words
JIC	Just in case
KOW	Knock on wood
OIC	Oh, I see!
OTOH	On the other hand
PITA	Pain in the "acronym"
POV	Point of view
PPN	Programmer project number, i.e. a CIS user's id#
RSN	Real soon now (computer lingo for "don't hold your breath")
RTFM	Read the f* manual (variously interpreted for public consumption as read the fine manual, read the fabulous manual, you figure it out <g>).
TIA	Thanks in advance
TSR	Terminate and stay resident program
WYSIWYG	What you see is what you get
YA...	Yet another (as in YAA -- yet another acronym)

Here are some other product-related acronyms:

API	Application programming interface
DD	Data dictionary
ER	Enhancement request (request for an enhancement to the product or documentation)
GUI	Graphical user interface
MU	Multi-user
OO, OOP	Object oriented, object oriented programming
SU	Single user
UDF	User defined function
UG	Users Guide

In addition there are a large number of platform specific acronyms and abbreviations such as QEMM, DV, WIN, INIT, CDEV, DA, which you'll either know or can ask about.

FORUM ORGANIZATION -- MESSAGES, THREADS, SECTIONS

The forum is comprised of individual messages from one forum member to another. These messages are organized by thread -- a "thread" is simply a string of related messages and replies brought together under a "thread topic", which serves as the "message header" for every message in that thread. Here's an example of a message header:

```
#: 83761 S1/Non Tech Services      * message number, section
   06-Dec-92  11:38:01            * date and time
Sb: #83591-#Access is Way Cool... * responding to message#, topic
Fm: Kevin Costner 72230,2664      * from forum member
To: Phil Collins 72261,640        * to forum member
```

Threads begin when an individual creates a new message (not replying to a previous message) and grow as other individuals reply. A new thread may also be created when replying to a previous message by changing the topic or posting the reply to a different section.

Threads are further organized by forum section (such as Queries or Forms -- a list or sections is included in Message Etiquette, section 7.2. New threads should be posted to the section appropriate to the topic.

Messages themselves may be either public or private (see below) and cannot exceed about 2,000 characters in length. For assistance in composing and posting messages, please refer either to the CIS help files or to the documentation provided with your communications software. Further information as relates specifically to this forum will be included in this document in appropriate places.

5.1 Message Traffic and Scroll Rate

Message traffic is the number of messages which get posted to a forum in a day. The volume on the MSACCESS is very high -- ranging from about 100 messages a day when a lot of folks are at Comdex to as many as 500 messages a day when something really interesting happens. Message traffic probably averages about 200-300 per day.

A CIS forum is limited to a certain number of message "slots" or total number of messages at any one time. As messages are posted, the message slots fill up. When they are full, messages begin to "scroll off" (disappear) on a First In, First Out basis (early messages scroll off before later ones). With message traffic (also called the scroll rate) of 300 messages a day, the first messages in the queue will disappear in about 6 days.

5.2 What the Heck All This Means to You

The nature of threads and the high scroll rate in this forum frequently have consequences for how you ask questions, how you formulate replies, and how you can help the forum function efficiently. Details will be included in later discussions where appropriate.

ASKING QUESTIONS

Forum members are very helpful, patient and forgiving -- no matter what level of expertise you have. You will almost certainly get an answer to your questions, if you are aware of certain facts of life.

Due to the high message volume on this forum, forum members who read all the messages (and there are many) are trying to keep up with forum traffic, be helpful, earn a living, and live something like normal lives -- all at the same time.

It's recommended that you check in again within two or three days after you leave a message and that you read all the messages in your thread -- otherwise you might 1) miss responses that have disappeared due to the high scroll rate and 2) miss responses that may not have been addressed **directly** to you -- people tend to reply to the most recent message in the thread rather than going back to your original inquiry. Any message sent **directly** to you and not read before it scrolls will automatically be sent to you via CompuServe Mail.

6.1 Phrasing your question

The way you phrase your question can make a big difference in the kind of response you get.

Here's the "worst case" question -- one that forum members will want to avoid (tech support will answer, of course, but you may be missing an opportunity to get a speedy reply from someone else):

"I just got Access and it doesn't work. Can anybody help?"

Obviously, the first thing anybody has to ask is: What do you mean it "doesn't work"? Just finding out the basics could take quite a bit of time.

So, here are some guidelines:

- Be SPECIFIC. (and be sure to post your message in the appropriate section of the forum -- a list of forum sections is included later in Message Etiquette, section 7.2).
- Tell us whether you are writing a program or using Access interactively.
- Tell us exactly what error messages you are getting, if any.
- Tell us what you are trying to do, what you did just before the problem occurred, and **exactly what happened**.
- Please include specific **step-by-step** instructions that will help use to reproduce your problem.
- If it's a programming problem, post the actual bit of code or macro group that you think is causing the problem or a close approximation.

In addition:

Post your question to the proper forum section (see Message Etiquette for a list). Some forum members only read messages in specific sections -- if you post in the wrong place, the person with the answer might not see it. And choose a suitable, informative, and specific **topic** for your message.

"Report Sum Question" in the Reports section will almost certainly get a better response than "HELP!".

Post any program code or formatted messages as UNFORMATTED. CompuServe normally "formats" or "word wraps" code that's uploaded to the forums -- i.e. it strings all the lines together if they are not separated by a completely blank line. This will turn your carefully indented code into a garbled mess which is very difficult to read.

You can avoid this when you're on-line with CIS by using the "POST UNFORMATTED" command. With various programs such as TAPCIS or Navigator (that allow you to read and respond to messages off-line -- explained later in this document), use their "Unformatted" option. You may also insure proper indentation by preceding each line of code with one or more spaces or a tab.

Roughly, if you're having a problem which appears to involve the way Access interacts with your hardware such as loading or operating slowly, not loading at all, running out of memory or disk space, video display problems, running on a network, etc., please be as specific as possible about your operating environment, perhaps including your operating system (Windows 3.1, DOS 5.0), your DOS CONFIG.SYS, AUTOEXEC.BAT, type of network, network settings, and so on and so on <g>.

6.2 Uploading files

If you have a very difficult or complex problem, it may eventually be resolved only by sending it to someone else for review. There are two approaches:

1) If tech support **SPECIFICALLY** requests that you send them your problem (and at no other time), upload it to the forum libraries. Problems sent this way will be treated as private communications -- however, if you want to have the problem reviewed by non-Microsoft folks, you may specifically ask the sysops to post it publicly to the libraries. Uploading to forum libraries is free of connect charges so this is definitely the most efficient way to send things to Microsoft. More information about uploading files to libraries can be found in Uploading Files, section 9.2.

2) If a non-Microsoft forum member has been assisting you, you can send your problem to him or her via CompuServe mail. However, there is something like a "rule" about this:

DO NOT UPLOAD ANYTHING TO ANYBODY WITHOUT THEIR **SPECIFIC PRIOR APPROVAL.**

Downloading something from CompuServe costs the recipient money in phone and CIS connect charges. Many active forum members use automated programs to download their messages and files and these programs do not give them the chance to refuse a download -- anything in their box is downloaded automatically. It's simply not fair to impose this burden on them without their consent. The above rule also means that if somebody has been kind enough to look at something before, it doesn't necessarily mean they want to look at it again, no matter how great it is now <g>. It's a simple matter to request and receive permission before you upload -- do it!

In EITHER CASE, when you upload something to somebody, be considerate again -- send them the minimum that's necessary to "get the job done", i.e. give them the information they need to assist you and compress the file (using PKZIP -- more on this later) to reduce transmission time.

6.3 Responding to answers

If you have asked a question and someone has replied with an answer, read what they have to say carefully. If they've solved your problem, great! If they haven't, but are trying to help, **answer their questions**. Remember they may be in a hurry, or have misunderstood your question or situation, or may even be telling you something incorrect (not on purpose, of course), so if you don't understand something they say, don't be afraid to say "I don't understand."

6.4 Acknowledgments

If somebody helps you out, a simple "Thanks!" is appreciated.

If you've had a particularly knotty problem that appears to have been a challenge to resolve and/or if many people have made a lot of different and possibly conflicting suggestions -- when you've finally got it working, we'd appreciate knowing what exactly you did that worked. The reason we appreciate this kind of summary is because many of us may have or anticipate having the same kind of problem, and we'd like to know the answer too! In addition, it may be a valuable addition to Microsoft's Knowledge Base - a tool that we

use to help you solve your problems.

6.5 If you don't get an answer

If you leave a message and nobody responds, it may be because it "just happens" that nobody has time to respond, because everybody simultaneously decides that "somebody else will answer", or because your message was unclear and nobody knew *how* to respond.

You can increase your chances of having a response to your message by:

- Posting your message again -- and saying that your original message was not addressed.
- Posting your message directly to a sysop or to "all".
- Rephrasing your message.

TIPS FOR THE ACTIVE PARTICIPANT

As mentioned earlier, the message traffic on the MSACCESS is very high. It can be quite a chore to manage this volume of communication and keep CIS connect charges to a minimum. Here are a variety of hints that may prove useful.

7.1 Thread Integrity and Continuity

All forum members, experienced and novice, benefit from the forum threads. Often a thread will address a problem that many other forum members have or anticipate having. Consequently, it is important that all messages that contribute to the thread remain available during its lifetime (thread integrity). In addition, it's extremely useful if relevant messages are contained within the thread and in the proper order (continuity). With this in mind:

- public messages should not be deleted if they contribute to the thread.
- relevant responses to existing messages should be generated by **replying** to the message rather than starting a new message.
- multiple part messages, as much as possible, should be kept together (more on this a little later in Message Etiquette).

Related to this issue is the "other side of the coin" where messages are no longer relevant to the topic. It is possible on CIS to scan available messages by "message header" or thread topic. This can be a useful approach if you're trying to save on connect charges. However, this is not **guaranteed** to be a foolproof approach. Threads on this forum tend to wander, and strict adherence to a thread topic is not rigorously enforced. Consequently if you're downloading a thread whose topic looks interesting, you may discover:

- the messages have wandered off the thread topic
- the messages have deteriorated into a series of jokes

At a rough guess, some of us estimate that the really relevant messages in a thread are the first 1 to 10, although that is not necessarily the case -- when a topic is complicated, messages may still be useful during the thread's entire life or, OTOH, a thread may deteriorate into jokes almost immediately. Unfortunately, due to the high message traffic, the beginning of a thread may scroll off so that all that remains in the forum are the less relevant messages. This is not to say that the messages which aren't relevant to the topic aren't useful <g> -- indeed, they are frequently useful -- and the jokes are often good, too.

- Please use upper and lower-case for your name and messages. Upper case words indicate EMPHASIS and a message in all capitals looks as though you are "SHOUTING."
- All Message Subtopics correspond to the same Libraries and Conference Rooms.
- If you upload a file in the public libraries, please upload it into New Uploads Library. Files uploaded into New Uploads will remain for about one month and will be moved to their permanent location
- We do NOT make any guarantees regarding freeware or shareware available for downloading from this forum.

Section Leaders cannot provide the following:

1. Sample code or macros with responses; however, Microsoft may request sample code or macros for bug verification
2. Private responses to questions or private phone calls
3. Source code or macro debugging, design review, in-depth discussion of Windows internals, or bench marking of code

4. A guarantee of question RESOLUTION
5. Any proprietary, philosophical, or third-party recommendation information.

7.2 Message Etiquette

In consideration of forum members who do not read all the messages (those who download the threads whose topics sound interesting) and to support thread integrity and continuity, please observe the following suggestions:

Please post your message to the most applicable section. MSACCESS sections are (note that these sections will change from time to time):

1. Non Tech Services
2. Tables/DB Design
3. Queries
4. Forms
5. Reports/Printing
6. Macors/Modules
7. Import/Export
8. Multi-user/Networks
9. OLE/DDE
10. SQL Server/ODBC
11. Security
12. Setup
13. Suggestions/Mktg
14. User's Groups
15. Third Party

When starting a new thread, try to name it something that captures the essence of your message and be as specific as possible -- unless it's the Chatter section where you can name it anything you like <g>.

When replying to a message, if your reply is substantially changing the subject, please post it under a different topic and to a different section if applicable.

If your reply is directly relevant to the current topic, please DO NOT post it to a different topic or section. Use CIS or your off-line software to REPLY to the message.

If you want to make sure that a specific person sees your message and sees it as quickly as possible, make sure you address it specifically to him or her and *be sure to include his/her CIS id number*. This is important because:

Some people only read messages addressed directly to them and will miss your message entirely. (Note: reading only personal messages is not recommended practice in this forum because you will almost certainly miss further possibly relevant messages in a thread.)

Many active participants will read their personal messages during the day and download everything else in the evening -- so they might not see a particular message until later.

If the message scrolls off before the addressee sees it, CompuServe can

not forward it to CompuServe mail without an id number.

Long messages: If you have a particularly long message, remember that individual messages are limited to about 2,000 characters. When a message exceeds that limit, you can either send it via CIS Mail to an individual or you can split it into parts and post it on the forum. CIS Mail has a very high message length limit, but no one on the forum will see messages sent that way.

In the interests of thread integrity and continuity:

Post long messages that are relevant to an existing thread, or that are of general interest, to the forum not to CIS mail so we can all benefit. Other long messages may of course be posted as well. If you find that you seem to be engaged in a topic that is not particularly Access related or is really a conversation between you and one other person, consider moving that conversation to CIS mail (you might want to ask first to see if anybody has been following the conversation and would like to see it continue).

There are two approaches to splitting messages which are a little contradictory.

First, there is an option called /SPLIT (please refer to CIS help or your CIS communications software documentation). By inserting /SPLIT at the bottom of a message, the next message you send will automatically continue the message. The benefit of /SPLIT is that it keeps the parts of your message *together* when they are posted. The drawback is that subsequent messages are addressed to *you* rather than the person to whom you were originally replying. Since (as mentioned earlier) some people only read messages addressed directly to themselves, these people will not see the continued parts (or may not see them until later).

The alternative approach is to send continued messages by simply continuing to reply to a message and manually inserting "more" and "continued" in your messages as appropriate. The benefit of this is that the intended recipient will be sure to receive all parts of the message in a timely fashion. The drawback is that messages split this way may not get posted together and can be interrupted by other replies (which impairs continuity).

Rule of thumb: Use /SPLIT unless you want to guarantee that the recipient will see all parts or all parts as soon as possible. In the latter case, continue to reply.

7.3 On-Line vs Off-Line

CIS connect charges are high and MSACCESS message traffic is high -- the two can combine to deliver whopping credit card bills (not, unfortunately, much to joke about). If you plan to frequent the forum or other CIS forums, it is highly recommended that you investigate several programs that download and upload forum messages in a more efficient and less costly way.

Note: CompuServe offers a product called CIM (CompuServe Information Manager) which is useful to manage your work on CIS -- however, it mainly works on-line and will not reduce your connect charges as much as the products I will discuss here.

A couple of programs we would recommend are OzCIS and TAPCIS. These programs will log onto CIS for you, upload and download messages and files, and provide you with other utilities. Their major advantage is that they allow you to read and respond to messages *off-line*. Not only is this *considerably* cheaper, but also it allows you to review and respond at your leisure. If you want to get up to check a manual or break off for a moment to test something in a program, you can do so without the terrible pressure of connect charges eating away at your pocketbook. Moreover, it allows you reflect on what you say, change your mind, correct your mistakes, and improve the quality of your messages in general.

Another advantage is that these programs can be used in a variety of ways to capture only the messages you really want to see and further cut down your connect charges. For example, with TAPCIS, you can read only certain forum sections, you can read all messages or download the thread topics (which you

can mark off-line to be downloaded the next time you log on), or you can read only messages addressed to you personally.

These programs offer a further advantage in that they provide an option for retaining past messages (yours and others) in a file on disk for future reference. These files can be archived from time to time and transferred to floppies or otherwise managed. Forum members have a variety of strategies for managing these files. As an example you can archive (using PKZIP) the message file about once a week and keep about a month's worth of zipped files on your hard disk.

Some forum members keep **all** their messages. Some rely on the forum archive files to capture the "best parts".

In addition to the basic programs, the offline communications programs often have many additional utility programs.

More information about downloading files from libraries can be found later in this document.

OzCIS

OzCIS is freeware. Files to download from CIS, from the IBMCOM forum (GO IBMCOM), Library 12, are:

OZCIS.BRO...7868 (<-file size)

This file outlines many of the capabilities and features of OzCIS version 1.2 and later. Plain ASCII text file; can be REAd online or downloaded.

OZCIS.REQ...5940

This file describes the minimum hardware and software requirements in order to run OzCIS version 1.2a and later. OzCIS is a big program and requires considerable system resources; reviewing this file before downloading the main program files could save you considerable online time. Plain ASCII text file; can be REAd online or downloaded.

OZCIS1.EXE...279454

OzCIS is a auto-navigator to automate CompuServe access for Mail, Forums and other areas of the service. This is Part 1, the Main Program file.

OZCIS2.EXE...167057

This is Part 2, the Online Processor file.

OZCIS3.EXE...131049

This is Part 3, the Accessories file.

OZCIS4.EXE...138820

This is Part 4, the Documentation file. Note: Changes/additions to the documentation since version 1.0 are in the README.1ST file included in the Part 1 and 2 files.

OZHELP.ZIP...159173

The files included in this archive are intended to provide the ability to reference the OzCIS documentation from within OzCIS as a supplement to the on-line help. This on-line documentation is intended to be used with

the shareware documentation reader, AUTOREAD.EXE, installed as an EXTERNAL under OzCIS. AUTOREAD may be found in IBMSYS LIB 6 (BRO /KEY:AUTOREAD).

TAPCIS

TAPCIS is a shareware program. Files to download from CIS, from the TAPCIS forum (GO TAPCIS), Library 1, are listed below. You can also order the program by calling Support Group, Inc. at 800-USA-GROUP.

TAPCIS.INF...13963 (<-file size)

This file describes TAPCIS in detail along with its requirements and pricing. You should download and read this file before downloading the TAP.EXE program and TAPDOC.EXE documentation files from Library 1.

TAP.EXE...211812

TAPCIS(tm) Version 5.4, The Access Program for the CompuServe Information Service. TAPCIS completely automates CompuServe Mail and Forum messages and libraries. TAPCIS cuts online time to a minimum and frees you to do other things while it does all online actions without user input. We guarantee you'll love TAPCIS with a 21 day shareware trial and a 90-day money back guarantee. Copyright (c) 1991 Support Group Inc. Manual in TAPDOC.EXE. Prev downloads: 43,430+

TAPDOC.EXE...154832

TAPCIS(tm) 5.4 Documentation in a self extracting file. This program will expand into all ten sections of the TAPCIS docs plus appendixes, table of contents, and index. See file README.DOC included in this file for more information. You will need this documentation to set up TAPCIS properly. Program is in TAP.EXE.

LIBRARIES

The forum libraries contain many informative and useful files. An exploration of the libraries is well worth the trouble. The MSACCESS libraries are the same as the MSACCESS sections. You should post your file to the section that it relates to most closely. If you don't know what that is, you could post it to Non Tech Services.

Files in the libraries come in two basic forms: ASCII and binary. ASCII files are text files which will be readable without further attention. Binary files are *either* files (including text files) that have been compressed to save space and downloading connect charges *or* other binary files like programs or graphics files.

9.1 Compressed Files

There are several types of compressed files:

.ZIP files -- Compressed using PKZIP -- these are uncompressed by using PKUNZIP.EXE. PKZIP/UNZIP are shareware programs available for downloading from the MSACCESS forum.

.EXE files -- Mostly this extension indicates that the file is a "self-extracting" compressed file. This means that the file will unpack itself when you type the file name. *Usually* these files are also compressed using PKZIP -- however since they're self-extracting, they do not require PKUNZIP.

.ARC files -- Compressed using Arc utilities or such commonly available compatible programs as ARC-A.COM. Can be uncompressed with ARC-E.COM, available in the IBMCOM forum, library 2, and elsewhere (look for ARC-E.COM and ARC-E.DOC).

9.2 Uploading Files

Most files in the libraries have been uploaded by forum members for use by other forum members and we encourage you to do the same. If you have developed a utility or a function or a bit of code or have some particular information to share that appears to spark general interest, please send it on up! Remember, uploading to forum libraries is free of connect charges.

If tech support *SPECIFICALLY* requests it (and at no other time), you can also upload code that you're having a problem with and which appears to be too lengthy or complex to post on the forum itself.

The mechanics of uploading a file can be fairly tricky. Please refer either to CIS Help or your CIS communications software documentation. If you want additional help, CIS has a practice forum (GO PRACTICE) where you can practice uploading a file, writing messages, etc. Time spent in the practice forum is free of connect charges.

Here are some guidelines for uploading:

- Combine and compress any files you are sending into one archived file using PKZIP or the self-extracting options of other compression schemes.
- For some reason, CIS limits file names in libraries to only 6 characters (not including extension). It's better if you limit your file name to 6 characters yourself -- otherwise CIS may truncate the file name in a way that you don't really want. File names should begin with a letter and contain *only* letters and numbers. They should not contain spaces or punctuation (including underlines and hyphens), or miscellaneous characters like * or ^.

- When uploading, please include a brief but complete description of the files included and what they do. If space allows, I always like to see the name of the person who's sending it and a note about how large the file will be when uncompressed (disk space is sometimes tight <g>).
- The sysop will check the file for viruses and perform some other maintenance. Consequently, it may take a few days before the file becomes available to others.
- Let people on the forum know that you've uploaded the file, what its name is, and where you sent it.

NOTE: If you are uploading a file with the **same** name as a file you previously uploaded, the old file will be replaced with the new one. However, if you are uploading a revision and want to use a new name, please help with library housekeeping by deleting the old file yourself using the CIS Library Menu.

AFTERWORD - The Spirit of Contribution

We need all the help we can get! <g> We need people to be more specific and precise asking questions (and it would certainly help if people check the manuals and other product documentation resources first). We need people to be more careful about thread continuity and integrity. We need people to be more considerate (tempers are sure to fray <g>). And we need people to help others more than ever before.

One of the things we want to distinguish the MSACCESS Forum from others is the very unselfish way forum members help each other day after day. We feel there is an underlying intent on the part of most forum folk to make a contribution to others on the same path. We also feel as long as this "spirit of contribution" remains a force, the forum will work well for all of us.

You can really help maintain the value of the forum by logging on with an **intent to contribute**. If you want to ask a question, ask it with an eye toward contributing to our knowledge as well as yours. Share experiences that you feel will be of value to others. Answer the questions of others. And try, especially, to keep to the public agenda and avoid self-promotion.

There's certainly opportunity on the forum for vendors of products and services to offer their wares (in moderation), and vendors are **just the kind of people we need** to help us. Yet it's always a little irritating when we see a vendor log on **only** to promote something while so many others are helping for free. If you are a vendor, hang around a little and contribute; give back a little to the community that supports you. (And from a purely practical point of view, establishing a helpful presence on the forum should serve to enhance your marketability.)

Although making a contribution is always a source of personal satisfaction, we believe that there's more at stake here -- we are building a community that can support us in our work and in our lives. That's something worth investing in -- contribution is truly an investment in our future.

See you on the forum!

Compuserve Sysops

Below is a list of the present Microsoft Sysops that perform daily duties on Compuserve.

Customer Service:

. Charisa Martin	(Domestic)	76701,245
. Kara McKenzie	(Int'l)	72200,1641

ACCESS (MSACCESS FORUM)

.Steve Alboucq	(Support Manager)	72350,3370
.Kevin Kennedy	(Support Manager)	75010,2456
.MariEsther Burnham	(SL)	70363,152
.Monte Slichter	(SL)	70621,1344
.Kim Hightower	(SL)	70761,655
.Kim Abercrombie	(SL)	70761,650
.Ryan LaBrie	(SL)	72360,3371
.Joe Howard	(SL)	72350,3367
.Don Funk	(SL)	72350,3372
.Gary Yukish	(SL)	72420,1356
.Jim Hance	(SL)	75010,2433
.Brian Blackman	(SL)	75010,2461
.Michael Patten	(SL)	71742,1175
.Roger Harui	(SL)	71742,1177
.Tim O'Brien	(SL)	71742,1201
.Tad Orman	(SL)	72370,625

BEETHOVEN (MSAPP FORUM)

.Kris Piccolo	(SL)	76711,166
---------------	------	-----------

BOOKSHELF (MSAPP FORUM)

.Kris Piccolo	(SL)	76711,166
---------------	------	-----------

CINEMANIA (MSAPP FORUM)

.Kris Piccolo	(SL)	76711,166
---------------	------	-----------

DOS (MSDOS FORUM)

.Jeff Armstrong	(SL)	75300,3130
.Nat Bowman	(SL)	75300,3127

EXCEL (MSEXCEL FORUM)

.MAC - Melissa Frix	(SL)	72420,562
.WIN/PM - John LaTour	(SL)	72560,2425

EXCEL SDK (PROGMSA)

.Frank Beeson (SL)	71075,3616
--------------------	------------

FLIGHT SIMULATOR (MSAPP FORUM)

.DOS/MAC - Rich Noren	(SL)	75000,1356
-----------------------	------	------------

MAIL (MSAPP FORUM)

.DOS - August Hahn	(SL)	72360,1177
.MAC - Mark Adcock	(SL)	76701,243

MONEY (MSAPP FORUM)

.Kiska Gifford	(SL)	76711,361
----------------	------	-----------

MOUSE (MSAPP FORUM)

.Dan Brown	(SL)	72410,2415
------------	------	------------

MULTIPLAN (MSAPP FORUM)

.DOS/MAC - Jeff Jorgenson	(SL)	76701,152
---------------------------	------	-----------

POWERPOINT (MSAPP FORUM)

.MAC - Rosie Khoo	(SL)	76711,165
.WIN - John Hobgood	(SL)	76711,167

PROJECT (MSAPP FORUM)

.Jeff Jorgenson	(SL)	76701,152
-----------------	------	-----------

PUBLISHER (MSAPP FORUM)

.Rolf Nelson	(SL)	76701,267
--------------	------	-----------

SOUNDBITS (MSAPP FORUM)

.Kris Piccolo	(SL)	76711,166
---------------	------	-----------

VIDEO FOR WINDOWS (MSAPP)

.John Langhans	(SL)	76711,167
----------------	------	-----------

WINDOWS (WINNEW and WINADV FORUMS)

.Debbie Simmons	(SL) WINNEW	76711,360
.Susie Bernard	(SL) WINNEW	76711,416
.Angela Trusch	(SL) WINNEW	76702,1370
.Gene Whitley	(SL) WINNEW	71333,547
.Jordan Montgomery	WINADV	76702,1371
.Wayne Cook	(SL) WINADV	71075,2433
.Craig Sommerville	WINADV	75300,3125
.Gene Whitley	(SL) WINADV	71333,547
.Jamie Emmanuelli	WINADV	76711,202

WINDOWS FOR WORKGROUPS (MSWRKGRP)

.Jeff Williams	(SL)	76702,1501
.Val Peterson	(SL)	71333,545
.Lisa Childers	(SL)	71333,546

WORD (MSWORD FORUM)

.DOS - John Guerin	(SL)	76701,266
.MAC - David Carson	(SL)	76701,56
.OS/2- Mark Bishop	(SL)	75300,3600
.WINDOWS/PM - Kate Edson	(SL)	76701,54
.WINDOWS/PM - Sandy Weil	(SL)	76701,153
.WINDOWS/PM - Luann Vodder	(SL)	76711,203

WORDBASIC (PROGMSA)

.Kate Edson	(SL)	76701,54
-------------	------	----------

WORKS (MSAPP FORUM)

.DOS - Kris Piccolo	(SL)	76711,166
.MAC - Rosie Khoo	(SL)	76711,165

.WIN - Robert Williams

(SL)

75300,503

THE INFORMATION PROVIDED IN THE MICROSOFT KNOWLEDGE BASE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. MICROSOFT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL MICROSOFT CORPORATION OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER INCLUDING DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, LOSS OF BUSINESS PROFITS OR SPECIAL DAMAGES, EVEN IF MICROSOFT CORPORATION OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES SO THE FOREGOING LIMITATION MAY NOT APPLY.

COPYRIGHT Microsoft Corporation, 1992.

Common problems articles are articles perceived to be problems by the customer (for example, common user errors). Unconfirmed bugs and documentation error articles also fall into this category. These articles are easily identified by the "PRB:" code in the title of the article.

Information articles describe undocumented features and sample programs, or elaborate on topics that are already documented. These articles are easily identified by the "INF:"code in their titles.

Bug articles are articles that are confirmed to be problems . These articles are similar in format to the common problems articles and are easily identified by the "PRACC:" code in the title of the article.

Microsoft FastTips is an automated fax and voice service that provides technical product support information about Microsoft Access at the following phone number 24 hours a day, 7 days a week (including holidays): (206) 635-7051. You can use FastTips with your touch-tone phone. The Microsoft Access FastTips service includes a comprehensive map for easy navigation and a Technical Library catalog, which contains the most popular customer support information. For your convenience, we recommend that you request a map and catalog during your first call. If you do not have direct access to a fax machine, many convenience stores and private mailbox companies provide local FAX services that can be used to receive information from FastTips.

Useful Resources to Make the Most Out of Access

User Groups

Many groups are already established and more are starting. They are a great way to learn more about the features and power of Microsoft Access. Microsoft keeps in close contact with these groups, sending them sample applications, code, tips and tricks, as well as other information. Plus, representatives from Microsoft make visits and speak at user group events around the country. To join or start a Microsoft Access user group or special interest group, call (206) 936-3729.

Training

If you're interested in digging deeper into Microsoft Access or want to come up to speed quickly, training may be the answer. To find a Microsoft Access training site near you, call (800) 426-9400. If you're interested in becoming a trainer for Microsoft Access, call (800) 227-4679 to request an application.

Periodicals

[Access Advisor](#)

Data Based Advisor
DBMS Client/Server Computing

Data Based Solutions, Inc.
M&T Publishing Inc.

[Inside Microsoft Access](#)

[Smart Access](#)

Books

ABC's of Microsoft Access	Sybex
Absolute Beginner's Guide To Access	Sams Publishing
Access Consulting	Prima Publishing
Access for Windows Hot Tips	Que
Access for Windows Power Programming	
Access from the Ground Up	Prima Publishing
Access Programmer's Reference	Que
Access Programming Cookbook	Addison-Wesley
Access Workshop	Brady Books
Critical Information Network	
Guide to Access	
Help! Microsoft Access	Ziff-Davis Press
Inside Microsoft Access	New Riders Publishing
Microsoft Access 1.1	Bantam Computer Books
Microsoft Access Bible	IDG Books Worldwide
Microsoft Access Developer's Guide	Sams
Microsoft Access Inside & Out	Osborne McGraw Hill
Microsoft Access Instant Reference	SYBEX
Microsoft Access Programming	Windcrest/McGraw Hill
Microsoft Access Running Start	SYBEX
Microsoft Access Step by Step	Microsoft Press
MS Access Handbook	Bantam Books
PC Learning Labs Teaches Access	Ziff-Davis Press
PC World Access Handbook	IDG Books
Running Microsoft Access	Microsoft Press
Teach Yourself Access	MIS Press
Understanding Microsoft Access	SYBEX
Using Access 1.1 for Windows	Que
Using Access for Windows, Special Edition	Que Publishing
Your Access Consultant	

Access for Windows Power Programming

Advanced

by Persohke & Liozbanski

Access for Windows Power Programming is aimed at the advanced user, developer and network administrator who is interested in developing powerful and effective applications for Microsoft Access. The book contains advice on the planning and design of databases and applications, as well as tips and techniques for creating effective reports, forms and queries. Coverage of programming in the Access Basic language and creating macros is included, along with detailed information about variables, arrays, fields and other programming structures in Microsoft Access. An Xbase programmers survival guide, a quick command and function reference, and a complete sample application are a few key elements in this book. All code discussed in the book is included on a disk, plus you get a disk containing valuable utility software. Price: \$44.95 (book with two disks).

Order Information: Available April 1993. Call (800) 528-5331. Que Publishing.

Access from the Ground Up

Beginning/Intermediate

by Martin S. Matthews with Edward Hartmann and William E. Lloyd

This fast-paced introduction to Microsofts database program for Windows is the perfect starting point for anyone who is learning to use Microsoft Access for the first time. An expert team of authors begins by teaching readers basic database design concepts. Then, readers learn to customize forms, perform queries, print reports and labels, use macros, and more. Highly illustrated. Price: \$19.95.

Order Information: Available February 1993. Prima Publishing, Inc.

Critical Information Network

Beginner/Advanced (Application developers, MIS)

by Thomas P. McAuliffe & Carolyn S. Shamlin

Critical Information Network is a new book that dispels the myths of online management reporting and offers a complete set of guidelines to avoid failed applications. Based on 20 years of both technical and non-technical experience, the book provides the authors unique view of the relationship between critical information and the management process. Many long-standing industry issues are clearly addressed:

1. The real added value of information technology for management reporting.
2. The keys to sustained management interest.
3. The elusive link between reporting applications and corporate priorities.
4. How groupware will impact the executive world.
5. A blueprint for successful computer use by management.

Price \$29.95 plus \$2.00 for shipping

Order Information: Call (617) 247-2460. McAuliffe & Company, Inc.

Guide to Access

Beginning/Intermediate/Advanced

by Miriam Liskin

Now users at all skill levels can gain access to the complete feature set of Microsoft's leading-edge database program by following best-selling author Miriam Liskin's clear explanations. In 20 exhaustive chapters Liskin covers just about every aspect of Microsoft Access from beginning to advanced subjects. Topics covered include designing and creating databases, defining simple queries, performing calculations, working with external data, using Access on a network, building menu-driven applications, and more.

1. Covers all aspects of database use and introduces database programming to nonprogrammers.
2. Turns beginning and intermediate users into masters.
3. Includes a disk containing the sample databases presented in the text, saving readers hours of input.

Price: \$29.95.

Order Information: Available April 1993. Call (800) 688-0448 or fax (510) 601-2155. Ziff-Davis Press.

Inside Microsoft Access

Beginning/Intermediate

by Richard Wagner, Helen Feddema, Michael Groh, Greg Reddick

Inside Microsoft Access provides a complete description of Microsoft Access database version 1.0 from the conceptual and implementation perspectives. The book covers the fundamentals of database design and creation using Microsoft Access. Everything from using the Microsoft Access environment to constructing macros and Access Basic modules are discussed in detail. Throughout *Inside Microsoft Access*, practical examples illustrate each topic and concept.

The bonus disk contains HomeFinder, a sophisticated application produced with Microsoft Access. HomeFinder comes complete with tables, forms, queries, and macros. HomeFinder, a highly interactive application, demonstrates the many advanced features of Microsoft Access, including query by example, OLE embedding, image handling, and report generation.

Order Information: Call (800) 428-5331. New Riders Publishing, Prentice-Hall Computer Publishing.

Microsoft Access Inside & Out

Beginner/Intermediate

by Mary Campbell

Microsofts exciting new database for the IBM PC and compatibles is sure to be a challenger for the lead database for Windows. Find out all about this revolutionary software in Mary Campbells new book. Youll get an insiders introduction to this powerful, feature-loaded relational database management system (DBMS), from fundamentals to intermediate-level skills and then on to some advanced topics. If youre a beginning database user, youll soon learn how to set up a database and use it productively. If you already know database software, youll learn all about the unique features of Microsoft Access, such as three-dimensional color graphics, object linking and embedding (OLE), multimedia and network capabilities, and other features that make this software a powerful new contender. Price: \$27.95.

Order Information: Available now at most booksellers, or by calling (800) 227-0900. Osborne McGraw-Hill.

Microsoft Access Programming

Beginner/Intermediate/Advanced

by Namir Shamas

Microsoft's first database for Windows, Microsoft Access, is a program powered by easy-to-use visual programming techniques. It appeals to seasoned programmers and users who want to create their own customized database applications.

Namir Shamas's latest book is an accessible, hands-on introduction to database programming for anyone who is familiar with the Basic language. Shamas's relaxed, friendly style leads the reader gently but thoroughly through the fundamentals of Access Basic programming to using the host of powerful database extensions. It's a practical tutorial approach loaded with ready-to-use program code and peppered with professional tips, tricks, and warnings that has proven successful in the author's previous books. Price: \$32.95.

Order Information: Call (800) 822-8138 or fax (717) 794-2080. Tab/McGraw-Hill.

PC Learning Labs Teaches Access

Beginner

by PC Learning Labs

PC Learning Labs, world leader in hands-on computer instruction, has now added Microsoft Access to their successful book/disk learning series. This innovative learning tool will aid first-time users in learning this powerful Windows-based database program. Years of classroom teaching and lab research into how people best learn to use computers are the basis of this integrated package. No other book is better suited to the needs of beginning users of Microsoft Access.

1. Covers start-up and basic operations as well as forms, query techniques, reports, and much more.
2. Assumes no prior computer experience.
3. Includes a Practice Examples Disk with exercises that work hand in hand with instructions in the text.

Price: \$22.95.

Order Information: Available February 1993. Call (800) 688-0448 or fax (510) 601-2155. Publisher Ziff-Davis Press.

PC World Microsoft Access Handbook

Beginner/Intermediate

by Cary Prague and Michael Irwin

Discover the full power and ease of use of Microsoft Access from expert authors. This book/disk combination is a hands-on tutorial reference that walks you through the process of creating powerful applications for Microsoft Access. Key topics include installation, database design, creating tables, creating and printing professional-looking reports, using logic and math functions, data validation, database publishing, using pictures, graphs and OLE objects, creating switchboards, dialogs, and menus, and more! Bonus 3.5-inch disk contains customizable examples. Price: \$34.95.

Order Information: Available (first quarter 93) at all major booksellers or order direct at (800) 762-2974. IDG Books Worldwide, Inc.

Running Microsoft Access

Intermediate

by John Viescas

Running Microsoft Access the Microsoft guide to the long-awaited Windows-based relational database software is now available from Microsoft Press.

This example-packed, hands-on tutorial and users guide incorporates tips and strategies not covered in the product demonstration.

Author John Viescas provides database concepts and terminology, a comprehensive product overview, and step-by-step instructions. Anyone who needs access to data from a small business manager to an enterprise-wide information system administrator will learn how to use the powerful capabilities of Microsoft Access to build database applications from scratch or to make existing data more accessible. Price: \$29.95.

Order Information: Available wherever books and software are sold or by calling toll-free (800) MSPRESS.

teach yourself... Access

Beginning/Intermediate

by Steve Davis

Teach yourself everything you need to know about Microsofts new database software, Microsoft Access. Take the Weekend Tour of Microsoft Access and jump right into:

1. understanding and creating databases and reports
2. indexing
3. creating screen forms, queries, and labels.

With *teach yourself...Access*, youll also learn to modify databases, customize forms and reports, use queries, filters, and macros, and even program in Access Basic. Price: \$21.95.

Order Information: Available February 1993. Call (800) 628-9658 (Telemarketing) or (800) 488-5233 (Individual Sales). MIS:Press.

Using Access for Windows, Special Edition

Beginning/Intermediate/Advanced

by Roger Jennings and Ron Person

Using Access for Windows, Special Edition addresses the needs of all those using Microsoft Access, whether secretaries or small business owners or application developers. The book is formatted as a tutorial for beginners, but also explains advanced database concepts and programming techniques to intermediate users, and provides material on the Access Basic macro and programming language for advanced users.

Using Access for Windows, Special Edition combines step-by-step lessons with real-world power-user techniques and examples. Features covered include: queries, graphics, forms, reports, multiuser capabilities, networking features, macros and programming. Information on SQL, dynamic data exchange (DDE), and object linking and embedding (OLE) is also presented. Other key features of the book include the QuickStart Tutorial format with use of tips, notes, and cautions throughout the text. Price: \$29.95.

Order Information: Call (800) 528-5331. Que Publishing.

Your Access Consultant

Intermediate/Advanced

by Martin S. Matthews with Edward Hartmann and William E. Lloyd

When youre ready for the next step, let the experts at Matthews Technology become your Access consultants. This object-oriented approach to database design with Microsoft Access illustrates general principles and tricks that can be customized and applied to an infinite variety of situations. All code in the book is included on a high-density disk, along with fully operational database systems that have been developed with Microsoft Access. An incredible bargain! Price: \$39.95.

Order Information: Available June/July 1993. Prima Publishing Inc.

Access Advisor

Access Advisor is the magazine that teaches you how to use all the power of Microsoft Access. Each issue is crammed with information for users of all types, from novice to expert developer. As a Charter subscriber, you'll receive a one-year subscription (6 issues) for only \$29 (plus shipping charges outside the U.S.). This is 25% off the cover price. A companion disk is available for an additional \$60 (50% savings). Don't miss an issue!

Order Information: Call (800) 336-6060 or (619) 483-6400. Advisor Communications International, Inc.

Inside Microsoft Access

If you want to gain a complete understanding of Microsoft Access, *Inside Microsoft Access*, a monthly publication from The Cobb Group, will provide you with the skills you need. The articles in *Inside Microsoft Access* will give you vital information you can't find in books and manuals, tips and techniques you can use right away to work smarter and faster with Microsoft Access.

The Cobb Group publishes a wide-range of respected software journals including *Inside Visual Basic™ for Windows*, *The Microsoft C/C++ Developers Journal*, and *Inside Microsoft Windows*. A one-year subscription to *Inside Microsoft Access* (12 issues) costs \$49.

Order Information: To request a free issue of any publication, call (800) 223-8720. The Cobb Group.

Smart Access

Smart Access, Pinnacle Publishing's monthly technical journal for Microsoft Access users, will help you make the most of your Microsoft Access databases from day one. Our expert authors will show you the ins and outs of queries, macros, forms, reports, and Access Basic code (ABC), as well as how to make the move to Microsoft Access from other database platforms. You won't have to spend tedious hours digging through the manuals for solutions and you'll save enough time and money to pay for your subscription many times over!

Order Information: Subscribe to *Smart Access* today! 12 monthly issues for just \$79 you'll receive the accompanying source code disks (a \$60 value) FREE! Call (206) 251-1900, (800) 231-1293, or fax (206) 251-5057. Pinnacle Publishing, Inc., P.O. Box 888, Kent, WA 98035-0888.

action

The basic building block of a macro; a self-contained instruction that can be combined with other actions to automate tasks. This is sometimes called a command in other macro languages.

action argument

Additional information required by some macro actions for example, the object affected by the action or special conditions under which the action is executed. You enter action arguments in the lower portion of the Macro window.

action list

A list of all actions you can use with a macro. The list appears when you click the down arrow in the Action column of the Macro window.

action query

A category of queries that includes append, delete, make-table, and update queries. Delete and update queries change existing data; append and make-table queries move existing data. In contrast, *select queries* don't move or change data; they ask a question and return a dynaset.

append query

An action query that adds the records in a query's dynaset to the end of an existing table.

attached table

A table stored in a file outside the open database but from which Microsoft Access can access records. You can add, delete, and edit records in an attached table, but you can't change its structure.

bound control

A control on a form or report that is tied to a particular field in the underlying table, query, or SQL statement. For example, a text box that displays an employee's last name is bound to the Last Name field in the Employees table.

bound object frame

A control used to display and manipulate OLE objects that are stored in tables on forms and reports.

breakpoint

A line of code in a Function or Sub procedure at which Access Basic automatically halts execution. After execution has been halted, you can use debugging tools, continue execution, edit a procedure, or reinitialize the code. Microsoft Access displays the line in bold type.

caption

The name displayed on a label or the name displayed in the caption bar at the top of a form in Form view or Design view.

case-sensitive

A requirement that text be an exact match of uppercase and lowercase letters.

cell

The intersection of a row and column in a datasheet or grid. In the Categories datasheet, for example, Seafood is the value in the cell located at the intersection of the third row and the Category Name column.

check box

A control that indicates whether an option is selected or cleared. An X appears in the box when the option is selected.

Clipboard

The temporary storage area used by the operating environment to store text, graphics, and other data. You transfer the data to and from the Clipboard using Copy and Paste commands.

column

The visual representation of a field in a datasheet, the QBE grid, or the Filter window grid. In a datasheet, a column is a stack of values showing the value in that field for every record. In the QBE and Filter window grids, a column is a stack of information, such as sort order, about each of the fields included in the query or filter.

combo box

A control, similar to a list box and text box combined, in which you enter a value or select an item from a list.

command button

A control that runs a macro or calls an access Basic function. This is sometimes called a push button in other applications.

comment

Text that you can add to a macro or table to describe or annotate your work. In Access Basic, a comment is text embedded in the code (but ignored by Access Basic) that explains how the code works or describes what the code does. Comments in Access Basic must be preceded by an apostrophe (') or the reserved word **Rem**.

continuous form

A form that displays more than one record on the screen in Form view. You can create a continuous form by setting a form's `DefaultView` property to `Continuous Forms`.

control

A graphical object, such as a text box, an unbound object frame, a rectangle, or a command button, that you place on a form or report to display data, perform an action, or decorate the form or report.

Control menu

The menu in the upper-left corner of an application or a window. It contains commands to move, resize, and close the window and is designated by a short horizontal line rather than by a menu name.

control properties

Attributes of a control on a form or report that affect the appearance or behavior of the control. `BorderStyle` is an example.

Counter data type

A field data type. Such fields store sequential numbers automatically inserted by Microsoft Access.

criteria

A set of limiting conditions, such as Denmark (meaning equal to Denmark) or > 30000, used in creating a query or filter to show a specific set of records.

crosstab query

A query that computes a grid of summary totals based on values for each row and column. For example, a monthly sales-by-region query is a crosstab query.

crosstab report

A report in which you calculate totals for fields you select and group them by date year, quarter, month, week, or day. The data is displayed in a row-and-column format with the date at the top of each column.

Cue Cards

An online coach that walks you through the most common Microsoft Access tasks as you work with your own data. Choose Cue Cards from the Help menu.

Currency data type

A field data type and an Access Basic data type. The Currency data type is useful for calculations involving money or for fixed-point calculations in which accuracy is extremely important. This data type is used to store numbers with up to 15 digits to the left of the decimal point and 4 digits to the right. The type-declaration character in Access Basic is @.

data type

The attribute of a variable or field that determines what kind of data it can hold. Supported data types include field data types for end-user tasks and Access Basic data types for programming tasks.

database

A collection of data related to a particular topic or purpose. A database file can contain tables, queries, forms, reports, macros, and modules.

database objects

Tables, queries, forms, reports, macros, and modules.

Database window

The window that appears when you open a Microsoft Access database. It contains Table, Query, Form, Report, Macro, and Module buttons that you can click to display a list of all objects of that type in the database.

datasheet

A window that displays data from a table, form, or query in a row-and-column format.

Datasheet view

A window that displays data from a table, form, or query in a row-and-column format.

DDE

An established protocol for exchanging data between applications that run with Microsoft Windows.

delete query

An action query that deletes a set of rows that match the criteria you specify.

Design view

A window in which you design tables, queries, forms, and reports.

dialog box

A window in which you select options before Microsoft Access carries out a command.

disabled field

A field that appears dimmed on a form, can't be tabbed to, and won't respond to typing or clicking. You disable a field by setting its Enabled property to No.

DLL

A set of routines that extend the capabilities of Microsoft Access.

dynaset

The set of records that results from running a query or applying a filter. Microsoft Access updates the data in the underlying table or tables when you make changes to a dynaset.

Access Basic

The programming language used by Microsoft Access.

equi-join

A join that combines records from two tables whenever there are matching values in a field common to both tables.

event

A user action, such as a mouse click or key press, recognized by Microsoft Access. Using certain properties, you can tell Microsoft Access to run a macro in response to an event.

expression

Any combination of operators, constants, functions, and names of fields, controls, and properties that evaluates to a single value. You can use expressions as settings for many properties and action arguments, to set criteria or define calculated fields in queries, and to set conditions in macros. You also use expressions in Access Basic.

external table

A table outside the open Microsoft Access database.

field

A category of information, such as last name, that is stored in a table. A field is represented as a column in a datasheet and as a control on a form.

field data types

The set of data types you can choose from for a field in a Microsoft Access table. Microsoft Access has eight field data types: Counter, Currency, Date/Time, Memo, Number, OLE Object, Text, and Yes/No.

field list

A small window that lists all the fields in an underlying table or query. You can display field lists in forms, reports, and queries; Microsoft Access automatically displays the appropriate field lists in the Filter window.

field properties

Attributes of a field that affect its appearance or behavior. FieldSize is one example. The list of field properties for a field appears in the lower half of a table's Design view when you select the field.

filter

A set of criteria applied to records in order to show a subset of the records or to sort the records.

footer

Text and/or graphics appearing at the bottom of each page, at the end of a form or report, or after a group of records. Microsoft Access uses three kinds of footers: page footers, form and report footers, and group footers.

foreign key

One or more table fields that refer to the primary key field or fields in another table.

form

A Microsoft Access object on which you place controls for entering, displaying, and editing data in fields.

form footer

Text and/or graphics that appear at the bottom of a form on screen or at the end of the form in print.

form header

Text and/or graphics that appear at the top of a form on screen or at the beginning of the form in print.

form properties

Attributes of a form that affect its appearance or behavior. `DefaultView` is one example.

Form view

A window that displays data one record at a time in fields on a form. Using Form view is the primary method for adding and modifying data in a table.

Form window

A window in which you work with forms in Design view, Form view, or Datasheet view.

function

An Access Basic procedure that returns a value. It can be used in Access Basic programming or in Microsoft Access expressions.

graph

A control used to display a graph in a form or report.

graph report

A report that displays a graph.

GraphWizard

An AccessWizard you use to create a form containing a graph.

group footer

Text and/or graphics that appear at the bottom of a group in a report.

group frame

A box that encloses a group of controls that provide a limited set of alternatives for users to choose among. Use the option group control to create a group frame that encloses a set of check boxes, option buttons, or toggle buttons.

group header

Text and/or graphics that appear at the top of a group in a report.

group level

Separates groups of data in a report. It is sometimes referred to as a break level in other database products.

group-level report

A report that organizes data into groups and displays it in a row-and-column format.

header

One or more identifying lines of text that appear at the top of each page, at the top of a form or report, or at the beginning of a group of records. Microsoft Access includes three kinds of headers: page headers, form and report headers, and group headers.

Help

An online reference in Microsoft Access that contains information about commands and dialog boxes, Microsoft Access procedures, the interface, properties, keyboard equivalents, and Access Basic.

horizontal scroll bar

A scroll bar that appears along a horizontal axis. You use it to scroll left and right within a form, query, datasheet, or report.

Immediate window

A window in which you can execute individual lines of Access Basic code, usually for the purpose of debugging. You can use this window anytime a module is open.

index

A Microsoft Access feature that speeds data access by ordering the values in a field. You can specify an index by using the Indexed property (a field property) or, for multiple-field indexes, by using the Index1...Index5 properties (table properties).

join

A database operation that combines some or all records from two tables. Microsoft Access supports several kinds of joins:

An *equi-join*, which combines records from two tables whenever there are matching values in the joined fields.

An *outer join*, which includes all the records from one table and only those records from the second table in which the joined fields are equal. In the Join Properties dialog box, you can specify which table will contribute all of its records.

A *self-join*, which is a table joined to itself. A self-join is useful when you ask hierarchical questions, such as a query that shows all the managers and the people who report to them.

label

A control that displays descriptive text, such as a title, a caption, or instructions, on a form or report.

link

A connection between a source document and a destination document. A link inserts a copy of information from the source document into the destination document while maintaining the connection between the two documents. When the information changes in the source document, the changes are reflected in the destination document.

list box

A control that provides a list of choices. A list box consists of a list and an optional label.

list index

The sequence of numbers for items in a list, starting with 0 for the first item, 1 for the second item, and so forth.

macro

An action or a set of actions you can use to automate tasks.

macro group

A collection of related macros that are stored together. It is often referred to simply as a macro.

Macro window

The window in which you create and modify macros.

make-table query

An action query that creates a new table from the dynaset of an existing query.

Memo data type

A field data type. Such fields can contain up to 32,000 characters.

menu

A list of commands, available at the top of a form or window, that is displayed when you click a menu name on a menu bar.

menu bar

The horizontal bar below the title bar that contains the names of Microsoft Access menus. You can click a menu name to display the list of commands.

modal

A type of form that retains the focus until you close it. Most modal forms are dialog boxes, warnings, or pop-up forms.

navigation buttons

The four arrows in the lower-left corner of a datasheet and in Form view. You click the arrows to move through the records as follows:

The navigation buttons are also available during previewing so that you can move through the pages of your document.

network

A group of computers and other devices that can interact by means of a shared communications link.

normalize

To minimize the duplication of information in a relational database through effective table design.

null field

A field containing no characters or values. A null field isn't the same as a field with a value of 0.

Number data type

A field data type designed for numerical data that will be used in mathematical calculations. Use the *Currency* data type, however, to display or calculate currency values.

object

One of the following types of entities: (1) controls and database objects tables, queries, forms, reports, macros, and modules; (2) special system objects, such as Debug and Screen; and (3) in OLE terminology, information such as a graph, a drawing, or a table that can be linked or embedded.

OLE

A protocol by which an object, such as a graph, in a source application or document can be linked to or embedded in a destination document, such as a form or a report.

OLE object

An object in an application that supports object linking and embedding (OLE) and that can be linked and/or embedded.

option button

A control that can be selected or cleared like a check box. It's typically used as part of a group in which the user can select one and only one option button. This button is sometimes called a radio button in other applications.

option group

A control used to create a field on a form or report that frames a set of check boxes, option buttons, or toggle buttons and provides a limited set of alternatives for users to choose among. For example, to enable users to choose a method of payment, you could use an option group with three option buttons: Cash, Check, or Credit Card.

outer join

An outer join includes all the records from one table and only those records from the second table in which the joined fields are equal. In the Join Properties dialog box, you can specify which table will contribute all of its records by choosing option 2 or option 3. (In database terminology, option 2 is called a left outer join and option 3 is called a right outer join.)

page break

A control you use to insert page breaks in printable objects, such as reports and forms. You also use this control to insert breaks in multiple-screen forms.

page footer

Text and/or graphics that appear at the bottom of a page in a form or report.

page header

Text and/or graphics that appear at the top of a page in a form or report.

Palette

A dialog box containing choices for color, special effect (raised or sunken), border, and width that you use when designing a form or report.

parameter query

A query that asks you to enter one or more criteria values, such as "Nairobi" for City, before the query is run. A parameter query is not, strictly speaking, a separate kind of query; rather, it extends the flexibility of other queries (except for crosstab queries).

permissions

A set of attributes that specify whether or not a user has access to objects in a database and what kind of access is allowed. For example, Read Data indicates that a user can view but not edit data.

pixel

An abbreviation for "picture element"; a dot that represents the smallest graphic unit of display on the screen.

point size

The height of a printed character. A point equals approximately $\frac{1}{72}$ of an inch, or $\frac{1}{28}$ of a centimeter.

pop-up form

A window, dialog box, or submenu that stays on top of other windows. For example, the toolbar and Palette are pop-up forms containing tools to help you do your work. In Help, pop-up windows are used to display definitions and lists of topics.

primary key

One or more fields whose value or values uniquely identify each record in a table.

preview

To view data or modules as they will appear when printed. There are two preview windows: Print Preview and Sample Preview.

The Print Preview window is available for forms, reports, datasheets, and modules.

The Sample Preview window is available only for reports in Design view. Sample Preview provides a quick view of the basic layout of a report but may not include all the data in the report.

In either window, you can choose the Zoom button to switch between a close-up view of a portion of the page and the full view of the page. Click the navigation buttons in the lower-left corner of the windows to page through the document.

When you preview a report, Microsoft Access displays the report initially in close-up view as shown below.

property sheet

The window in which you can view or modify the properties of a selected object.

QBE

A technique for designing queries. With graphical QBE, which Microsoft Access uses, you create queries visually by dragging the fields you want to include in the query from the upper portion of the Query window to the QBE grid in the lower portion of the window.

query

With a query you ask a question about the data stored in your tables or perform an action on the data.

Select queries ask questions about the data stored in your tables and return a dynaset without changing the data. Once the dynaset is displayed, you can view and change the data in the underlying tables.

Action queries change data or move data. Append, delete, make-table, and update queries are action queries.

Crosstab queries compute summary totals based on values for each row and column.

Parameter queries, which are not a separate type of query, extend the flexibility of other queries by prompting you to change certain criteria each time the query is run.

Query window

A window in which you work with queries in Design view or Datasheet view.

record

A collection of data about a person, a place, an event, or some other item. A record is represented as a row in Datasheet view of a table, query, or form.

record selector

A small box or bar that you can click to select an entire record in Datasheet view of a table, query, or form and in Form view.

refresh

To redisplay the records in a form or datasheet to reflect changes and indicate deletions that other users have made (in a multiuser environment) or that you have made elsewhere. You can't use the Refresh command to display newly added records. Also, Refresh won't rerun a query or reapply a filter to display new records that meet the criteria or eliminate records that no longer meet the criteria.

relational database

A database in which data is stored in tables, allowing for more efficient and nonredundant data storage and manipulation.

relationship

A connection established between common fields (columns) in two tables. A relationship can be one-to-one, one-to-many, or many-to-many.

report

A Microsoft Access database object that presents information formatted and organized according to your specifications. Examples of reports are sales summaries, phone lists, and mailing labels.

report footer

Text and/or graphics that appear at the end of a report and typically contain summaries.

report header

Text and/or graphics that appear at the beginning of a report and typically contain a title, the print date, and a company logo.

report properties

Attributes of a report that affect the appearance or behavior of the report. For example, you can set the PageHeader property so that a page header appears on every page in a report or on every page except the first.

Report window

A window in which you work with reports in Print Preview or Design view.

requery

To update records in the active form or datasheet to reflect changes to the records, display newly added records, and eliminate deleted records.

row

The visual representation of a record in Datasheet view of a table, query, or form. In a table's Design view, a row contains information about a field. In a query's Design view and in the Filter window, a row contains information about queries or filters. In the Macro window, a row contains information about an action.

row selector

A small box or bar that you can click to select an entire row in a table's Design view or in the Macro window.

rulers

On-screen rulers that you can display by choosing the Ruler command from the View menu. You use the rulers to align controls horizontally or vertically on a form or a report.

run time

The time during which an application is running. During run time, you can interact with an application.

run-time error

An error that Microsoft Access can detect only when an application is running.

section

A part of a form or report, such as a header, footer, or detail section.

section properties

Attributes of a form or report section that affect the appearance or behavior of that section. For example, you can set the SpecialEffect property to specify a three-dimensional effect or color for a form or report header.

security

A set of features used to specify or restrict data access to prescribed users or user groups.

select query

A query that asks a question about the data stored in your tables and returns a dynaset in the form of a datasheet without changing the data. Once the dynaset is displayed, you can view and make changes to the data in the underlying tables. In contrast, *action queries* do make changes to your data.

self-join

A join of a table to itself. Use a self-join to create a query that includes two copies of the same table.

separator

A character that separates units of text or numbers. It is set in the International section of the Microsoft Windows Control Panel.

A *list separator* separates items in a list (0,3,2).

A *thousand separator* separates groups of three digits in numbers (2,000 or 456,871,029).

A *decimal separator* separates integer parts of a number from its fractional parts (9.35).

A *field separator* separates items in imported or exported text files.

Note In Microsoft Access, a semicolon is the default list separator and does not have to be set in the International section of the Windows Control Panel. This is not true for Access Basic, however, which generally uses a comma.

setting

The value of a property that determines how an object or a control looks or behaves. For example, if you set the `SpecialEffect` property of a text box to `Raised`, the box will look three-dimensional.

shortcut key

A function key or a key combination, such as F5 or Ctrl+A, that enables you to carry out a command. In contrast, an *access key* enables you to open a menu or select a control by typing a designated letter while pressing the Alt key.

single stepping

A debugging technique that enables you to trace macro execution one action at a time or Access Basic code execution one line at a time.

sort order

The order in which records are displayed either ascending (A-Z or 0-100) or descending (Z-A or 100-0).

Sorting and Grouping box

The workspace in which you specify the order of your data and define the group levels for a report.

source

A document or application that supplies data in a DDE conversation or in an OLE operation.

SQL

A language used in querying, updating, and managing relational databases. You can view or write Microsoft Access queries as SQL statements using the SQL command in the Query window. You can use SQL SELECT statements anywhere Microsoft Access accepts a table name, query name, or field name. For example, you can use an SQL SELECT statement in the RecordSource property of a list box on a form to generate a list of items.

string

A collection of characters that can include both numbers and text.

subform

A form contained within another form or report.

submenu

A menu that opens when a command in a higher-level menu is chosen.

subreport

A report contained within another report.

tab order

The order in which the focus moves from one field or button to the next as you press the Tab key.

table

The fundamental structure of a relational database management system. In Microsoft Access, a table is an object that stores data in records (rows) and fields (columns). The data is usually about a particular category of things, such as employees or parts.

table properties

Attributes of a table that affect the appearance or behavior of the table as a whole. One example is the PrimaryKey property.

text

Data consisting of ASCII characters that represent words and symbols.

text box

A control that provides a place to enter or view text in a form or report. This is sometimes called an edit field in other applications.

Text data type

A field data type. Such fields can contain up to 255 characters or the number of characters specified by the FieldSize property, whichever is less.

title bar

The bar that appears at the top of the Microsoft Access window and displays its name.

toggle button

A control that acts as an on/off button. A toggle button can display either text or a picture and can stand alone or be part of an option group.

toolbar

A bar at the top of the Microsoft Access window containing a set of buttons that you can click to carry out common menu commands. The items displayed on the toolbar depend on which window has the focus.

toolbox

The set of tools you use at design time to place controls on a form or report. The toolbox usually appears on the left side of the screen. You can show or hide the toolbox using the View Toolbox command.

Note To get information on any of the tools in the toolbox while working in Microsoft Access, display the toolbox, click a tool, and press F1 for Help.

topic

The subject of a dynamic-data exchange (DDE) conversation between two applications. For example, in the syntax Excel|System!Status, System is the topic.

totals query

A query that displays totals or other kinds of summary calculations for values in various fields from a table or tables. Strictly speaking, a totals query is not a separate kind of query; rather, it extends the flexibility of select queries.

twip

Unit of measurement used by Microsoft Access and implemented as 1/1440 of an inch. (A twip is also 1/20 of a point.) There are 1440 twips to an inch and 567 twips to a centimeter.

unbound control

A control that's not connected to a field or an expression. You can use unbound controls to display informational text, such as instructions about using a form, or graphics and pictures from other applications. For example, lines and rectangles are unbound controls, and text boxes can be either bound or unbound controls.

unbound object frame

A control you place on a form to contain an unbound object, which is an object, such as a picture, whose value isn't derived from data stored in a table.

update query

An action query that changes a set of records according to criteria you specify.

validation

The process of checking whether entered data meets certain conditions or limitations.

validation rule

A rule that sets limits or conditions on what can be entered in a particular field. Microsoft Access displays an error message whenever the rule is violated. Rules are set with the ValidationRule property in the property sheet of a table or form.

value

The contents of a field, such as a name in a First Name field.

vertical scroll bar

A scroll bar that appears along a vertical axis. You use it to scroll up and down through a form, query, datasheet, or report.

view

A window that allows you to work with a Microsoft Access object in a certain way.

In Design view, you can design an object the way you want it to look.

In Datasheet view, you can modify and view data in a row-and-column format.

In Form view, you can modify and view data in fields on a form.

In Print Preview, you can see how an object will appear when printed.

WHERE clause

The part of an SQL statement that specifies which records to retrieve.

The asterisk (*), question mark (?), number sign (#), exclamation point (!), hyphen (-), and brackets ([]) are wildcard characters. You can use these characters in queries, commands, and expressions to include all records, file names, or other items that begin with specific characters or match a certain pattern. You can also use wildcard characters and matching characters to further refine a search.

Symbo l	Example	Usage
*	<i>wh*</i> finds what, white, and why <i>*at</i> finds cat, bat, and what	Like the MS-DOS asterisk (*) wildcard character, this asterisk matches any number of characters. But unlike MS-DOS, it can be used as the first or last character in the character string.
?	<i>b?ll</i> finds ball, bell, and bill	Like the MS-DOS ? wildcard character, this symbol matches any single character.
#	<i>l#3</i> finds 103, 113, 123	Matches any single digit.
[]	<i>b[ae]ll</i> finds ball and bell but not bill	Matches any single character within the brackets.
!	<i>b[!ae]ll</i> finds bill and bull but not bell	Matches any character not in the list.
-	<i>b[a-c]d</i> finds bad, bbd, and bcd	Matches any one of a range of characters.

aggregate function

A function, such as **Count**, **Avg**, and **Var**, that you use in creating a query that calculates totals. You can select aggregate functions in the Total row of the QBE grid. In writing expressions and in programming, you can use SQL aggregate functions (including the three listed above) and domain aggregate functions to determine various statistics.

calculated control

A control on a form or report that displays the result of an expression rather than stored data. The value is recalculated each time a value in the expression changes. A *calculated field* is a field defined in a query that displays the result of an expression rather than stored data.

delimited text file

A file containing values separated by commas, tabs, semicolons, or other characters.

exclusive

A way of opening a database that restricts other users or programs from accessing the database. You can use the Exclusive and Read Only options in the Open Database command dialog box to prevent others from opening the database or to limit their access to viewing but not modifying the database.

fixed-width text file

A file containing values arranged so that each field has a certain width.

group

A collection of user accounts in a secure system, identified by group name and PIN (personal identification number). Permissions assigned to a group apply to all users in the group.

In a report, group refers to a collection of records. You can separate records into groups in a report and display introductory and summary data for each group.

ODBC

A standard protocol for SQL database servers, such as Microsoft SQL Server. You can install ODBC drivers that allow Microsoft Access to connect to these SQL database servers and access the data in the SQL databases.

one-to-one relationship

A relationship between tables in which each record in the primary table can be associated with only one record in the related table. Additionally, the related fields must be the primary keys of both tables. When you create a one-to-one relationship, Microsoft Access lets you select only primary key fields as matching fields.

one-to-many relationship

A relationship between tables in which each record in the primary table can be associated with many records in the related table (each primary key value can appear many times in the related table).

PIN

A four-digit number that Microsoft Access uses in combination with the account name to identify a user account or group of accounts in a secure system. You enter the PIN and the account name when creating the account or group.

read-only

A type of access to a database whereby you can view but not modify data and database objects. Read Only is one of the dialog box options for the Open Database command on the File menu. You can also make a particular form or record read-only by setting the appropriate properties.

referential integrity

Rules that you follow to preserve the defined relationships between tables when you enter or delete records. If you enforce referential integrity, Microsoft Access prohibits you from adding records to a related table for which there is no primary record, changing values in a primary table that would result in orphan records in a related table, and deleting records from a primary table when there are matching related records.

update

To accept changes to data in a control or record. Until the record is saved, these changes are stored in a temporary record. The changes are saved in the database when you move to another record on a form or datasheet or when you explicitly save the record.

user account

An account identified by a user name and PIN (personal identification number) that is created to manage access to database objects in a secure system.

alias

Access Basic: An alternate name you give to an external procedure to avoid conflict with an Access Basic reserved word, global variable, constant, or a name not allowed by the standard naming conventions.

SQL: An alternate name you give to a field or expression in a SELECT statement, often shorter or more meaningful.

command line

The string of characters used to invoke an application. For example, you enter a command line when you create a program item in the Microsoft Windows Program Manager, when you start Microsoft Access by choosing Run from the File Menu in the Windows File Manager, or when you use the RunApp action.

command prompt

At the MS-DOS command line, the character or characters that appear at the beginning of the line, indicating that the computer is ready to receive input. Also known as MS-DOS prompt.

installable ISAM

A driver you can install that allows access to external database formats such as Btrieve, dBASE, and Paradox.

library database

A database that contains a collection of database objects that can be loaded automatically when you start Microsoft Access. You specify that a library database is to be loaded by adding its name to the [Libraries] section of MSACCESS.INI.

Memo object

Access Basic: An object you can use to hold the contents of a table field that has a Memo field data type. Memo objects can be up to 32K characters long.

parse

To identify the parts of a statement or expression and then validate those parts against the appropriate language rules for correctness.

system administrator

A member of the group Admins who is responsible for managing and maintaining a database system. A system administrator can create user accounts and groups and can assign permissions for any database object in the system.

work group

A collection of users in a multiuser environment who share data and the same system database.

action QueryDef

Access Basic: A variable that contains the definition of an action query. An action **QueryDef** must be declared as an object variable, can't be used with a **Dynaset** or **Snapshot**, and is used with the **Execute** method.

aliasing

Access Basic: A way of assigning an alternate name to an external procedure in a declaration to avoid conflict with an Access Basic reserved word, global variable, constant, or a name not allowed by Access Basic naming conventions (for example, a name that begins with an underscore).

Append mode

Access Basic: A sequential file output mode that sets the file pointer to the end of a file so that you can add records to the file using the **Print #** and **Write #** statements. The **Open** statement specifies the mode of access.

The other file input/output modes are **Binary**, **Input**, **Output**, and **Random**.

argument

A constant, variable, or expression that is passed to an Access Basic procedure. You use an argument with an operand in an expression or in a macro to supply additional information required by some actions.

bitmap

An image made up of pixels on the screen and stored as a collection of bits, one or more of which correspond to one pixel. When stored as files, bitmaps usually have the file extension .BMP.

Bookmark

Access Basic: A property of a recordset object (**Table**, **Dynaset**, or **Snapshot**) that contains a string identifying the current record. If you assign the Bookmark value to a variable and then move to another record, you can make the earlier record current again by setting the Bookmark property to that string variable.

Boolean

Access Basic: An expression that can be evaluated either true (nonzero) or false (0). You can use the reserved words **True** and **False** to supply the values of -1 and 0, respectively. The field data type Yes/No is Boolean and has the value of -1 for Yes and 0 for No.

Several property sheet settings are Boolean, including Yes/No, True/False, and On/Off.

constant

A constant represents a numeric or string value that doesn't change. **True**, **False**, and **Null** are constants. You can use constants in expressions in tables, forms, reports, queries, and macros. In an Access Basic module, you can use the **Const** statement to declare your own constants.

Microsoft Access also provides intrinsic constants that supply fixed values for action arguments and function values (such as the values returned by the **VarType** function).

current database

The database opened using the OpenDatabase command from the File menu. In Access Basic, use the **CurrentDB()** function to return the current database, which you can then assign to a database variable using the **Set** statement.

current index

For an indexed **Table**, the index most recently set with the **Index** property. It is the basis for ordering records in the **Table** and is used by the **Seek** method to locate records. A **Table** can have more than one index.

current record

Access Basic: The record in a recordset that you can modify or retrieve data from. Use a Move methods to move to the first, last, next, or previous record in a recordset. Use the Find methods (for a **Dynaset** or **Snapshot**) or the **Seek** method (for a **Table**) to change the current record position according to specific criteria.

debugging

Access Basic: The process of finding errors in your program. Debugging usually involves running specific portions of your program and analyzing code at various points. You can use breakpoints to stop code execution so that you can examine variables.

Declarations section

Access Basic: The topmost level of a module that appears before any **Sub** or **Function** procedures. This section contains definitions of user-defined data types, global constants, and global variables.

domain

Access Basic: A set of records defined by a table, a query, or an SQL expression. Domain aggregate functions return statistical information about a specific domain.

domain function

Access Basic: A function used to aggregate data over a certain set of records (the domain). The domain functions are **DAvg**, **DCount**, **DFirst**, **DLast**, **DLookup**, **DMax**, **DMin**, **DStDev**, **DStDevP**, **DSum**, **DVar**, and **DVarP**.

Double data type

Access Basic: A fundamental data type that holds double-precision floating point numbers. A **Double** variable is stored as a 64-bit (8-byte) number ranging in value from -1.79769313486232E308 to -4.94065645841247E-324 for negative values, from 4.94065645841247E-324 to 1.797693134862325E308 for positive values, and 0. The type-declaration character is **#** (ANSI character 35).

Double is also a FieldSize property setting.

Dynaset object

Access Basic: A recordset variable that identifies a **Dynaset** created using the **CreateDynaset** method.

Empty

Access Basic: The state of an uninitialized **Variant** variable. **Variants** that are Empty return a **VarType** of 0.

Empty is not the same as **Null**, which is a variable state indicating invalid data. Variables containing zero-length strings ("") are not Empty, nor are numeric variables having a value of 0.

explicit declaration

Access Basic: A way to declare variables using a **Dim**, **Global**, or **Static** statement. If you place an **Option Explicit** statement in the Declarations section of a module, using implicitly declared variables in the module produces an error.

global variable

Access Basic: A variable you declare with the **Global** reserved word in the Declarations section of a module. A global variable can be shared by all the procedures in every module in a database.

Integer data type

Access Basic: A fundamental data type that holds integer numbers. An **Integer** variable is stored as a 16-bit (2-byte) number ranging in value from -32,768 to 32,767. The type-declaration character is % (ANSI character 37).

You can use **Integers** to store Boolean values.

Integer is also a FieldSize property setting.

logic error

A programming error that can cause code to produce incorrect results or halt execution. A logic error may be caused by using incorrect or similar variable names, incorrect variable types, endless loops, or array problems or by flaws in logical comparisons.

Other kinds of programming errors are syntax errors, compile-time errors, and run-time errors.

Long data type

Access Basic: A fundamental data type that holds long-integer numbers. A **Long** variable is stored as a 32-bit (4-byte) number ranging in value from -2,147,483,648 to 2,147,483,647. The type-declaration character is **&** (ANSI character 38).

method

Access Basic: An operation on a report, recordset, or the **Debug** or **Screen** object. Like properties, methods apply to specific objects and have names that are reserved words in Access Basic.

For example, you can apply the **Print** method to the **Debug** object to transfer printed output to the Immediate window.

module

Access Basic: A collection of declarations, statements, and procedures stored together as one named unit. Unless explicitly made private to the module in which they appear, procedures in one module are recognized and can be called by procedures in other modules in the same database.

module level

Access Basic: The scope of a variable you declare in the Declarations section of a module. The variable can be shared by all the procedures in that module but not by procedures in other modules.

Null

Access Basic: A reserved word that indicates invalid data. You can perform validation by assigning **Null** to a **Variant** field and then later validating it with the **IsNull** function.

Variants that are **Null** return a **VarType** of 1.

Null is not the same as **Empty**, which is a variable state indicating uninitialized data. Zero-length strings ("") are not **Null**, nor are numeric variables having a value of 0.

page

Access Basic: A portion of the .MDB (database) file in which record data is stored. Depending on the size of the records, a page may contain more than one record.

procedure

Access Basic: A sequence of declarations and code statements that define a **Sub** or **Function** procedure.

QueryDef

Access Basic: An object that contains the SQL statement that describes a query.

reserved word

A word that is part of the Access Basic or Microsoft Access SQL language. Reserved words include the names of statements, predefined functions and data types, methods, operators, and objects. Don't use reserved words for variable or object names.

scope

Access Basic: The attribute of a variable or procedure that determines which sections of which modules recognize it. There are three levels of scope: global, module, and procedure. Variables that you declare globally can be accessed by any module, whereas variables that you declare in a specific module can be used only within that module. Further, variables you declare in a **Sub** or **Function** procedure can be used only in that particular procedure.

Single data type

Access Basic: A fundamental data type that holds single-precision floating point numbers. A **Single** variable is stored as a 32-bit (4-byte) number ranging in value from -3.402823E38 to -1.401298E-45 for negative values, from 1.401298E-45 to 3.402823E38 for positive values, and 0. The type-declaration character is ! (ANSI character 33).

Single is also a FieldSize property setting.

Snapshot

Access Basic: A static image of a set of data, such as the records displayed as the result of a query. **Snapshots** are recordsets that can be created from a base table, a query, or another recordset.

Snapshot object

Access Basic: A recordset variable that identifies a **Snapshot** created using the **CreateSnapshot** method.

statement

Access Basic: A syntactically complete unit that expresses one specific kind of operation, declaration, or definition. Usually, a statement consists of one line in a procedure or Declarations section. In most cases, however, you can include more than one statement on a line by separating the statements with a colon (:).

String data type

Access Basic: A fundamental data type that holds character information. A **String** variable can contain approximately 65,535 bytes (64K), is either fixed-length or variable-length, and contains one character per byte. Fixed-length strings are declared to be a specific length. Variable-length strings can be any length up to 64K, less a small amount of storage overhead.

The type-declaration character is **\$** (ANSI character 36).

Sub procedure

Access Basic: A procedure that carries out an operation. Unlike a **Function** procedure, a **Sub** procedure doesn't return a value. You declare a **Sub** procedure with the **Sub** reserved word and end it with an **End Sub** statement.

syntax

The prescribed order and punctuation for placing reserved words, variables, and constants in a statement or expression.

syntax checking

Access Basic: A Module Design option that, when set to Yes, checks the syntax of the code each time you enter a line.

syntax error

An error in the grammatical structure of the code or expression. If syntax checking is enabled, errors are highlighted when you type a line and press Enter.

system object

An object defined by Microsoft Access. The system objects are:

The **Forms** object and the **Reports** object.

The **Debug** object and the **Screen** object.

Recordset objects (**Dynasets**, **Tables**, and **Snapshots**).

System-defined database objects, such as the table MSysIndexes. You can display or hide the system-defined database objects by selecting the General options setting after choosing the Options command from the View menu.

Table object

Access Basic: A recordset variable that identifies a table opened using the **OpenTable** method.

transaction

Access Basic: A series of changes made to a recordset. You mark the beginning of a transaction with the **BeginTrans** statement. If you want to commit the transaction, use the **CommitTrans** statement. If you want to undo all your changes since **BeginTrans**, use the **Rollback** statement.

Transactions are optional and can be nested.

variable

Access Basic: A placeholder that stores data that can change as Access Basic code runs. Each variable has a unique name that identifies it within its scope.

Variant

Access Basic: The default fundamental data type for variables that don't have type-declaration characters when a **DefType** statement is not in effect. A **Variant** can store numeric, string, date/time, **Null**, or Empty data.

Variables of other data types convert automatically to a **Variant**, and **Variants** attempt conversion to other data types. You can use the **VarType**, **IsEmpty**, **IsNull**, **IsNumeric**, and **IsDate** function to determine the underlying data type of a **Variant**.

VarType

Access Basic: A function that indicates the underlying data type of a **Variant**. **VarType** values include: 0 (Empty), 1 (**Null**), 2 (**Integer**), 3 (**Long**), 4 (**Single**), 5 (**Double**), 6 (**Currency**), 7 (Date), and 8 (**String**).

Prtacc
PrtaccOthr

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Generic Text Only](#)

[HP II series \(PCL 4\)](#)

[HP IV](#)

[IBM 4019/4029 native mode](#)

[Other Laser Pinters](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Other](#)

[Print Preview not Match Output](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Installing printer drivers](#)

[Troubleshoot Printing Other](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Gnrith
Gnrithr

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[DDE Server Topics](#)

[Graphing](#)

[Interoperability - Other](#)

[OLE Registration Problem](#)

[Print Merge with Word](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[FastTips: Setup Questions & Answers](#)

[Hangs on disk x](#)

[Network Installation](#)

[ODBC](#)

[Other](#)

[Problems](#)

[Step Build disk](#)

[StepWizard](#)

[Window Setup - Other](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[Colors/Color Palette](#)

[Control Properties](#)

[Default Properties/Templates](#)

[Toolbox](#)

[Wizards](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[Control Panel](#)

[File Manager](#)

[Operating Environment Other](#)

[UAE/GP Fault caused by Windows](#)

[Usage Other](#)

[Windows Program Manager](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Crosstab](#)

[FastTips: Querying Questions & Answers](#)

[Joins](#)

[Make Table/Update/Append/Delete](#)

[Other](#)

[Parameters/Referencing](#)

[Query not updatable/Row Fix-Up](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Btrieve](#)

[Excel/Lotus](#)

[Fox/Xbase](#)

[Other Access .MDB's](#)

[Other ODBC/ISAM Drivers](#)

[Other](#)

[Paradox](#)

[SQL Server/SPT.DLL](#)

[Text \(Delimited\)](#)

[Text \(Fixed Width\)](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Action Arguments](#)

[Actions](#)

[Condition Column](#)

[Macro Name](#)

[Other](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[API/DLL Calls](#)

[Error Handling](#)

[FastTips: Access Basic and Macros Questions & Answers](#)

[Help Compiler](#)

[Libraries \(MDA\)](#)

[Other](#)

[VB Compatibility](#)

[VT Objects](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Data Types](#)

[Database Design](#)

[FastTips: Tables Questions & Answers](#)

[Field Properties](#)

[Modify](#)

[Other](#)

[Primary Key/Indexes](#)

[Validation](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[FastTips: Reports Questions & Answers](#)

[Layout](#)

[Mailing Labels](#)

[Other](#)

[Report Properties](#)

[Sorting and Grouping](#)

[SubReports](#)

[Totaling](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Application Design](#)

[Compact/Repair](#)

[Data Editing](#)

[Data Entry](#)

[Database Analyzer](#)

[Documentation/Cue Cards/Help](#)

[Expressions](#)

[Find/Replace](#)

[Formatting](#)

[Global Settings \(View Options\)](#)

[Multiuser/Locking](#)

[NWIND/Sample Apps](#)

[Other](#)

[Referencing Objects](#)

[Referential Integrity](#)

[Relationships](#)

[SQL](#)

[User Interface](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on both [CompuServe](#) and GENIE. This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[Combo/List Box](#)

[Events \(Calling Macro/Mods\)](#)

[FastTips: Forms Questions & Answers](#)

[Form Properties](#)

[Other](#)

[Requery/Refresh](#)

[SubForm](#)

[Validation](#)

[Forms/Reports \(Common\)](#)

[General](#)

[Import/Export/Attach](#)

[Interoperability](#)

[Macros](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[Programming](#)

[Queries](#)

[Reports \(Specific\)](#)

[Security](#)

[Setup](#)

[Tables](#)

[Windows Issue](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Hangs on disk x

[PRB: Microsoft Access Issues with MS-DOS 6.0](#)

Printing - Access Other

[PRB: Slow reports or large temporary print files](#)

[PRACC9308: Printer Fonts Cause Text to Shift Down and Right](#)

File Manager

[INF: Another Method of Quitting Windows](#)

Help Compiler

[PRB: "Help Topic Does not Exist" Invoking Custom Help File](#)

Othr Laser Pstscript (Not Lvl 2)

[PRB: Tops of Letters May Be Cut Off with Some PostScript Fonts](#)

Totaling

INF: Expressions to Count Yes/No and Other Responses

Setup Wizard

[INF: SUFILES Utility Appears to Hang](#)

[PRACC9307: Cannot Include Comma in App Name Using SetupWizard](#)

[PRB: ADK Custom Setup Hangs During Installation](#)

[PRB: Background Bitmap in Setup Says 'Setup'](#)

[PRB: Custom Setup Creates too many Program Groups](#)

Stepup Build Disk

[INF: SUFILES Utility Appears to Hang](#)

[PRACC9307: Cannot Include Comma in App Name Using SetupWizard](#)

[PRACC9308: Setup Ignores \\$\(AppPath\); Puts Files in MS-SETUP.T](#)

[PRB: ADK Custom Setup Hangs During Installation](#)

[PRB: Background Bitmap in Setup Says 'Setup'](#)

[PRB: Custom Setup Creates too many Program Groups](#)

[PRB: Input to Field not Capitalized](#)

Validation

[INF: Validation Rule to Require all Upper Case in a Field](#)

[INF: A Macro to Check for Duplicate Values in a Field](#)

ODBC

[INF: Upper Limit for QueryTimeout and LoginTimeout Parameters](#)

PwinHpiv

[PRACC9304: TrueType Fonts Shift Vertically on HP LJ4 Printer](#)

IBM 4019/4029 native mode

[PRACC9304: TrueType Fonts Shift Vertically on HP LJ4 Printer](#)

Windows Program Manager

[PRB: Microsoft Access Cannot Find DESKEDIT.EXE File](#)

Query not updatable/Row Fix-Up

[PRB: Cannot Edit Access Data Using Query with Attached dBASE](#)

Print Merge with Word

[INF: Importing Word Print Merge Data Files to Microsoft Access](#)

[INF: Exporting WordPerfect Secondary Files to Microsoft Access](#)

VB Compatibility

[INF: How to Convert to and from Unsigned Integers](#)

[INF: Visual Basic 3.0 and Microsoft Access 1.x](#)

Requery/Refresh

[INF: Requery, Repaint, and F9/Records Refresh](#)

[PRACC9307: Slow Repaint When Saving Form with Many Controls](#)

Installing printer drivers

[PRB: Error Occurs in Print Preview Without Driver](#)

Find/Replace

PRB: Current Record Not Immediately Updated with Replace All

Referential Integrity

[PRB: Referential Integrity Error Message with Shared Databases](#)

[INF: Updating a Separate Table When a Value Changes on a Form](#)

Database Analyzer

[INF: Cannot Use ANALYZER.MDB Filename for Output Database](#)

[INF: How to Install the Database Analyzer on the Help Menu](#)

[PRB: Error in Database Analyzer: DTSysQuery Already Exists](#)

[PRB: Must Have Database Open to Use Database Analyzer Utility](#)

HP II series (PCL 4)

[INF: Printing Microsoft Access Reports on HP Laserjet](#)

SubReports

[PRACC9212: Datasheet Form Saved as Subreport Causes GP Fault](#)

[PRACC9304: Calculated Control on Subreport Displays Wrong Data](#)

[PRACC9304: GPF Printing Subreport with Datasheet Default View](#)

[PRACC9304: Invalid Results in Subreport When CanGrow = Yes](#)

[PRACC9308: Error Previewing Subreports that Call Filter Macro](#)

[PRB: "#ERROR" Referencing Null Subreport Values in Main Report](#)

[PRB: Layout Changes to Subform in Datasheet View Not Saved](#)

Text (Delimited)

[INF: How First Line of Data is Used to Import Delimited Text](#)

[PRACC9302: Importing Delimited Text Files Strips Off Decimals](#)

[PRB: Access Adds Date Delimiters When Importing Text](#)

[PRB: Cannot Import YYMMDD Dates in Fixed-Width Text Files](#)

[PRB: Error Message When Importing Informix Delimited Text File](#)

[PRB: Out of Memory Error Message when Exporting](#)

Referencing Objects

[INF: Passing Values from a Form to a Parameter Query](#)

[PRB: 'Couldn't Find Object \[Object Name\]'](#)

Btrieve

[INF: Btrieve 6.0 Data Type Does Not Work with Access](#)

[INF: Definitions for Settings in WIN.INI Btrieve Section](#)

[INF: Export Btrieve Table Without Existing Xtrieve FILE.DDF](#)

[PRB: "Btrieve-Out of Resources" Error Msg with Attached Table](#)

[PRB: "Internal Database Error \(-5001\)" Msg Using Btrieve NLM](#)

[PRB: 'Couldn't find object <tablename>' Error with Btrieve](#)

[PRB: Btrieve Files Created with Btrieve 6.0 NLM Not Supported](#)

[PRB: Empty Table List When Attaching Btrieve Table](#)

[PRB: Error Message When Attaching Btrieve Table](#)

[PRB: Error Message: Too Many Memo or OLE Fields](#)

[PRB: Exclusively Locked Error When Attaching to Btrieve Table](#)

[PRB: Incorrect Btrieve Error Message, \[Btrieve ISAM\] section](#)

[PRB: Out of Memory Error Message when Exporting](#)

[PRB: Unexpected Error from External Database Driver \[20\]](#)

[PRB: Unexpected Error from External Database Driver \[22\]](#)

Text (Fixed Width)

[PRB: Import/Export Setup Cancel Button Cancels Unsaved Changes](#)

[PRB: Creating Multiple New Import/Export Specifications](#)

[PRB: Cannot Import YYMMDD Dates in Fixed-Width Text Files](#)

[PRB: Access Adds Date Delimiters When Importing Text](#)

[INF: Importing Fixed-Width Text Files](#)

[PRB: Unable to Import Portion of Fixed-Width Text Files](#)

[PRACC9302: Importing Delimited Text Files Strips Off Decimals](#)

[PRB: Exporting to Fixed-Width Text File Left Justifies Numbers](#)

Joins

[INF: Join Type Mismatch Errors Caught at Runtime](#)

[INF: Row Fix-up in the NWIND.MDB Order Form](#)

[INF: Use Row Fix-up Technique to Look Up Info Automatically](#)

[INF: Using a Subquery to Simulate a SQL Sub-SELECT Statement](#)

[INF: Using a Subtract Query to Find Unmatched Records](#)

[PRACC980: DISTINCTROW Keyword Ignored in a Cartesian Product](#)

[PRB: Join Is Broken by Value\(s\) in Fields '\[Field Name\]'](#)

[PRB: One Side of Join Line Not Attached to Table](#)

[PRB: Troubleshooting Out of Memory Errors in Queries](#)

Generic Text Only

[INF: Using the Generic Print Driver with Forms/Reports](#)

[PRB: 'Generic/Text Only' Driver Print Quality List in Error](#)

[PRB: Overlapped Printing an Access Query Using TTY.DRV](#)

Macro Name

[INF: All Duplicate Macro Names in Macro Group Are Executed](#)

[PRB: BeforeUpdate Macro Executing Twice](#)

[PRB: Two Accelerator Keys in Custom Menu Show Only First Key](#)

Other

[INF: Improved Print Merge Feature In Microsoft Access 1.1](#)

[INF: Using SendKeys to Change Option Menus](#)

[PRACC9307: GP Fault: Printing Large Report with Low Disk Space](#)

[PRB: Choosing a New Printer Resets Some Printer Settings](#)

[PRB: Moving Backward in Report Print Preview Window Is Slow](#)

[PRB: Tops of Letters May Be Cut Off with Some PostScript Fonts](#)

Compact/Repair

[INF: Defragment and Compact Database to Improve Performance](#)

[INF: Recovering MDB File Larger Than the 128-Megabyte Limit](#)

[INF: Sample Macro for Repair/Compact Operations](#)

[PRB: Microsoft Access Appears to Hang During Database Compact](#)

SubForm

[INF: Four Ways to Move to a Record from a Combo Box Selection](#)

[INF: How to Reference Controls or Fields on a Subform](#)

[INF: How to Requery Combo Box on Subform](#)

[INF: Making a Subform Uneditable and Unenterable](#)

[INF: Using the .Parent Property of a Subform](#)

[PRACC9303: Orders Form Doesn't Print/Preview SubTotal & Total](#)

[PRACC9306: Help Context ID Property Does Not Work in Subform](#)

[PRACC9306: Hidden Subform Maximized If Main Form Is Maximized](#)

[PRB: #NAME? Referring to Combo Box Column in Subform](#)

[PRB: Changes in Subform Design do not Update in Main Form](#)

[PRB: Empty Subforms Are Not Visible in Print Preview](#)

[PRB: Layout Changes to Subform in Datasheet View Not Saved](#)

[PRB: Navigation Button Unavailable in Main/Subform Wizard](#)

[PRB: NWIND Order Form Updates Order Amount Field with Zero](#)

[PRB: Scroll Bars Displayed for a Subform in Datasheet View](#)

[PRB: Subform not Updated](#)

[PRB: Unwanted Record Separator at the Bottom of a Subform](#)

[PRB: Using CTRL+SHIFT+HOME to Move to Main Form](#)

Multiuser/Locking

[INF: "Record Lock Threshold Exceeded" with Large Action Query](#)

[INF: Error Message 'Unable to Recover'](#)

[INF: Five Parameters to Set in ISAM Section of MSACCESS.INI](#)

[INF: How to Retrieve Windows For Workgroups User Information](#)

[INF: Page Locking, No Cluster Indexing Support](#)

[INF: Two Error Messages with Database on Workgroups Server](#)

[PRACC9303: Multiple Users Adding Fields Causes Cross-Linking](#)

[PRB: Changes to Shared Module Are Kept Local Until MDB Closed](#)

[PRB: Err Msg: 'Couldn't Update: Currently Locked by User...'](#)

[PRB: Error Message: Couldn't Open File](#)

[PRB: Inaccurate Message When Opening MDB Shared vs. Exclusive](#)

[PRB: Low Number of Share Locks Can Cause Error Message](#)

[PRB: Open Object Can Be Deleted by Another User](#)

[PRB: Paste Unavailable After Record Is Copied to Clipboard](#)

OLE-Create/Updt Embd Objects

[INF: Bound OLE Object Blank in Form View](#)

[INF: Calling RegQueryValue\(\) to get an OLE Object Class Name](#)

[INF: Finding the OLE ClassKey of an Embedded Object](#)

[INF: Users Can Edit but Not Commit Read-Only OLE Objects](#)

[INF: Using Sendkeys to Change an OLE Link](#)

[PRACC9307: GP Fault with MS Packager Object in Bound OLE Field](#)

[PRACC9308: Open OLE Objects Not Saved When Record Changes](#)

[PRB: Error Message: Too Many Memo or OLE Fields](#)

[PRB: How the Scaling Property Affects Embedded MS Draw Object](#)

[PRB: How the Scaling Property Affects Embedded MS Draw Object](#)

[PRB: Insert Object Dialog Box Has No 'All Files \(*.*\)' Option](#)

[PRB: Insert Object Dialog Box Has No 'All Files \(*.*\)' Option](#)

[PRB: Insert Object Dirties Record](#)

[PRB: New OLE Field is Created as Text Box in a New Form](#)

[PRB: New OLE Field is Created as Text Box in a New Form](#)

[PRB: No Error Given When Linked OLE Object Can't be Found](#)

[PRB: OLE Excel and WinWord Do Not Close When Form Is Closed](#)

[PRB: OLE Excel Format Changes when Updated in Microsoft Access](#)

Print Preview not Match Output

[INF: Screen Fonts May Not Match Printer Output Fonts](#)

[PRB: Text Label Truncated Printing Form on Epson 9-Pin Printer](#)

[PRB: Two Labels on Top of Each Other Both Print](#)

Crosstab

[INF: Displaying Totals for Each Row in Crosstab Queries](#)

[INF: How to Create a Dynamic Crosstab Report](#)

[INF: Sample Table Design to Support Questionnaire Applications](#)

[PRB: Incorrect Characters in Crosstab Query Column Headings](#)

[PRB: No Implicit Parameters on CrossTab Queries](#)

SQL

[INF: Creating Result Equivalent to SQL UNION Operator Result](#)

[INF: Empty Table Created by SELECT INTO with No Matching Rows](#)

[INF: Limit of AND's in SQL Select Statement](#)

[INF: Queries Sort by Order of IN\(\) Parameters](#)

[INF: Using a Subquery to Simulate a SQL Sub-SELECT Statement](#)

[INF: Using a Subtract Query to Find Unmatched Records](#)

[PRB: 'Too Many Expressions in Distinct Output' Error](#)

Other

[INF: Commonly Asked Questions About Access Version 1.0](#)

Window Setup - Other

[INF: Older Versions of Shared DLLs Cause Problems with Access](#)

[PRACC9309: Custom Setup Gives Wrong Low Disk Space](#)

Colors/Color Palette

[INF: ComboBox On Toolbar Do Not Use Window Background Color](#)

[INF: Transparent Text Boxes Not Supported](#)

[PRB: Bitmap May Not Display Correctly](#)

[PRB: Changing Caption on Title Label Leaves Shadow Unmodified](#)

[PRB: Fonts May Change Zooming Out in Print Preview](#)

[PRB: Option Button Appears Fuzzy in Datasheet View](#)

[PRB: Raised and Sunken Check Box and Option Button Controls](#)

Other Access .MDB's

[INF: Attached Table Must Be Opened as Dynaset from Modules](#)

[PRB: "Maximum Size Reached" Error Attempting to Import File](#)

[PRB: Error Message If Path Info for Attached Databases Changes](#)

Other

[INF: Access 1.0 \(rel. 1/14/93\) Disks 1-3 Contents 3.5" 1.4 meg](#)

[INF: ADK Setup installs to subdirectory of specified](#)

[INF: Contents: MS Access 1.0 Disks 4-6 \(1/14/93, 3.5-Inch\)](#)

[INF: Custom Setup May Show Zero or Negative Disk Space Needed](#)

[INF: Hourglass in Setup Changes to Arrow During Copy](#)

[INF: How to Make Access Opening Screen Display](#)

[INF: List of Articles Relating to MSACCESS.INI File](#)

[INF: Microsoft Access 1.0 Disk Contents \(3.5 inch, 1.44 MB\)](#)

[INF: Microsoft Access 1.0 Disks 1-4 Contents \(5.25 Inch 1.2MB\)](#)

[INF: Microsoft Access 1.0 Disks 5-9 \(5.25 inch, 1.2 MB\)](#)

[INF: ODBC Disk Contents \(1/14/93, 3.5-Inch, MS Access 1.0\)](#)

[INF: Online Help and Text Files Installed By 1.1](#)

[INF: Online Help and Text Files Installed By 1.1](#)

[INF: Setup Options Dialog Box Lists Disk Space Info Twice](#)

[INF: Troubleshooting Access Setup and Installation](#)

[PRACC9306: Windows Setup Ignores Applications Created in ADK](#)

[PRACC9307: Application Admin Setup \(/A\) to Empty Floppy Drive](#)

[PRACC9308: Setup Program Not Handling All Palette Messages](#)

[PRB: Cannot Install Cue Cards Without Installing Help](#)

[PRB: Input to Field not Capitalized](#)

[PRB: Microsoft Access Cannot Find DESKEDIT.EXE File](#)

[PRB: Microsoft Access Issues with MS-DOS 6.0](#)

[PRB: Microsoft Access Issues with MS-DOS 6.0](#)

Condition Column

[INF: A Macro to Check for Duplicate Values in a Field](#)

[INF: How to Create Do While/Do Until Loops in a Macro](#)

[INF: Temporarily Disabling a Macro Action](#)

[INF: Using a Conditional Macro to Confirm Changes to a Field](#)

[PRB: Macro Condition Evaluates True if Control Is Null](#)

Documentation/Cue Cards/Help

[INF: Additions to User's Guide Appendix A Specifications](#)

[INF: Correct Procedure for Inserting New Field Above Existing](#)

[INF: Correct Syntax for TransferDatabase to dBASE and Paradox](#)

[INF: CueCards Started Outside Access Does Not Work](#)

[INF: Example Has Wrong Number for Dlookup Ship Postal Code](#)

[INF: Freestanding Labels Do Not Appear in Datasheet View](#)

[INF: Microsoft Access ADK Legal Agreement](#)

[INF: MoveSize Arguments in Wrong Order in Language Reference](#)

[INF: Online Help for Sigma Button Is Incorrect](#)

[INF: OpenQuery is not a QueryDef Method](#)

[PRACC9303: Access Cue Cards Don't Use Standard Windows Menu](#)

[PRB: Cannot Install Cue Cards Without Installing Help](#)

[PRB: Error in Column Property Help Menu Topic](#)

[PRB: Macro Condition Example Using COUNT\(\) Is Incorrect](#)

[PRB: Memory Block Not Freed If Cue Cards Not Custom Installed](#)

[PRB: Page Hdr/Ftr Is Not Under View Menu](#)

[PRB: Query Window Does not Always Scroll to Record](#)

Other

[INF: Consecutively Numbering Pages for Two or More Reports](#)

[INF: Creating Reports to Mail Merge Microsoft Access Data](#)

[INF: Expressions to Count Yes/No and Other Responses](#)

[INF: How to Disable the Menu Bar When Previewing a Report](#)

[INF: How to Print a Single Record from a Form in a Report](#)

[INF: How to Specify a Custom Starting Page Number for a Report](#)

[INF: Print a Sequential Number for Each Record/Group on Report](#)

[INF: Reporting the Median Value of a Group of Records](#)

[INF: Sample Preview Ignores Query](#)

[PRACC9308: Cannot Use First\(\) and Last\(\) Functions in a Report](#)

[PRB: DoCmd OpenReport Opens Rpt in 3 Views or Prints to File](#)

[PRB: Don't have Permission to Modify Secured Report](#)

[PRB: Toolbar Disappears When Printing a Document](#)

[PRB: Troubleshooting Tips, Reports with "Out of Memory" Errors](#)

Paradox

[INF: 'Couldn't Open File' Error Message Using Paradox Table](#)

[INF: 'Err Msg: Operation Not Supported on Paradox Table...'](#)

[INF: Access Cannot Attach or Import Paradox 4.0 Tables](#)

[INF: Paradox Tables, Indexes, and Multiple Connections](#)

[INF: Trapping SQL Server RAISERROR\(\) Function Values](#)

[PRACC9210: Fast Search on Indexed Paradox Column Fails](#)

[PRACC9210: MS Access Ignores Paradox Private Directory Rules](#)

[PRACC9211: Cannot Export Paradox/dBase Tables](#)

[PRACC9211: Cannot Overwrite Corrupt Paradox/dBASE Files](#)

[PRACC9303: Bad Results Searching Multicolumn Paradox Indexes](#)

[PRACC9303: Error Msg: Data Has Changed, Operation Aborted](#)

[PRACC9304: Ordered Query on Attached Paradox Table](#)

[PRACC9306: Exporting Nulls to Paradox Actually Exports a Space](#)

[PRB: Can't Attach to Paradox Table on Read-Only Drive](#)

[PRB: Exporting Table with OLE Object or Memo Field to Paradox](#)

[PRB: Internal Database Error \(-5207\) When Attaching Tables](#)

[PRB: Out of Disk Space Error Makes Paradox Table Unusable](#)

[PRB: Out of Memory Error Message when Exporting](#)

[PRB: Paradox Character Translation Different in Win 3.0 & 3.1](#)

Report Properties

[INF: Access Saves Printer Information with Each Form/Report](#)

[INF: Detailed Explanation of the KeepTogether Property](#)

[INF: How to Simulate Changing RecordSource Property of Report](#)

[INF: Shrinking Empty or Null OLE Object on Report](#)

[INF: Two Methods to Refer to a Field in a Previous Record](#)

[PRACC9303: Error When Use SQL in Report RecordSource Property](#)

[PRACC9303: GP Fault When RunningSum Tries to Evaluate #ERROR](#)

[PRB: Cannot Process Close Action in Report's OnOpen Property](#)

[PRB: Report Cannot be Opened or Previewed](#)

Troubleshoot Printing Other

[PRB: Limited to 65535 Pages in Print Dialog Box](#)

[PRB: Two Labels on Top of Each Other Both Print](#)

[PRB: Data Dropped when Printing or White Lines on Reports](#)

Default Properties/Templates

[INF: Default AutoLabel and AddColon Properties](#)

[INF: Form Sizing Grid Accepts Non-Integer Values](#)

[INF: How to Change Default Template for Forms and Reports](#)

[PRB: 'No Default Printer' Error Message When Printing](#)

[PRB: Buttons Pasted into Option Groups have Option Value](#)

[PRB: Grid Disappears When Measurement Changed to Metric](#)

[PRB: Select All Not Grayed Even When No Items Are Available](#)

UAE/GP Fault caused by Windows

[PRACC1405: System Hangs when Minimizing Access Window](#)

[PRACC9301: HPPCL5A Printer in DEBUG Windows Causes GP Fault](#)

[PRACC9303: GPF Declaring New Object Variable During Break Mode](#)

[PRACC9303: GPF Using Empty String with CVDate\(\) or IsDate\(\)](#)

[PRACC9303: GPF Using OpenForm Action to Hide Modal Pop-Up Form](#)

[PRACC9304: GPF Printing Subreport with Datasheet Default View](#)

[PRB: Hewlett-Packard Dashboard GP Faults in Microsoft Access](#)

[PRB: Microsoft Access GP Fault with ATI Ultra Video Card](#)

Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file. This help system includes 1000 articles that relate to the Microsoft Access for Windows. These articles were developed by Microsoft's Product Support Engineers. This help file is a snapshot of the Knowledge Base as of September 22, 1993. New and updated articles can be found on several electronic services such as [CompuServe and GENIE](#). This help file contains helpful tips, programming examples([INF](#)), known problems([PRB](#), [PRACC](#)), [References](#), [FastTips](#) and other information that may not be addressed in the manuals.

Please feel free to distribute this information in compliance with the [terms](#) of Microsoft.

[Forms \(Specific\)](#)

[General](#)

[Reports \(Specific\)](#)

[Tables](#)

[Programming](#)

[Macros](#)

[Import/Export/Attach](#)

[Queries](#)

[Windows Issue](#)

[Forms/Reports \(Common\)](#)

[Security](#)

[Setup](#)

[Interoperability](#)

[General Other](#)

[Gnrthr](#)

[Printing: General Win Trblshoot](#)

[Printing: Specific Win3.1 Drvrs](#)

[Printing](#)

[FastTips: Nontechnical and Marketing Questions & Answers](#)

Interoperability - Other

[INF: How to Create Tasks in MS Project for Windows Using DDE](#)

[INF: How to Pass Information to Microsoft Access Using DDE](#)

[INF: How to Use DDE to Fill a List Box](#)

[INF: Links to Formatted DDE Items Result in Text](#)

[INF: OpenDatabase and CloseDatabase Pseudo Actions](#)

[INF: Quit Macro Action](#)

[INF: Sending a MAPI Mail Message from Microsoft Access](#)

[INF: Using Data from Access in a Word for Windows Print Merge](#)

[INF: Using Microsoft Access as a DDE Server](#)

[PRB: Cannot Auto Update an OLE Linked Object in a Table](#)

[PRB: Cannot DDEPoke > 255 Characters of Text from Access](#)

[PRB: DDEPoke To WinWord Contain Incorrect Character Format](#)

[PRB: Random Characters in Text Box Bound to OLE Object Field](#)

[PRB: Wait Argument in SendKeys May Not Stop Macro Processing](#)

Control Panel

[PRACC9308: Setup Program Not Handling All Palette Messages](#)

[PRB: Minimizing, Maximizing the Desktop Flashes Slightly](#)

[PRB: Property List Boxes Show Two-Word Options on Two Lines](#)

Problems

[INF: Couldn't Find SYSTEM.MDA or UTILITY.MDA File](#)

[INF: SUFILES Utility Appears to Hang](#)

[PRACC9303: Setup Leaves MS-SETUP.T Directory and Files on Disk](#)

[PRACC9304: Cue Cards Can't Be Started from Workstation](#)

[PRACC9305: Setup Writes Over Newer CTL3D.DLL File](#)

[PRACC9307: Setup \(STFSETUP\) GP Faults in Windows Standard Mode](#)

[PRACC9308: Setup Cannot Recognize Forward Slash in Path](#)

[PRACC9309: Custom Setup Gives Wrong Low Disk Space](#)

[PRB: 'Not Enough Memory to Run SETUP.EXE' Error](#)

[PRB: ADK Custom Setup Hangs During Installation](#)

[PRB: Cannot Write README.TXT Error Msg During Installation](#)

[PRB: ErrMsg: Couldn't Open Table MSysAccounts](#)

[PRB: Error Message, Cannot Change Working Directory to Default](#)

[PRB: Outdated COMMDLG.DLL Error Message During Setup Program](#)

[PRB: Setup Program STFSETUP.EXE Causes GP Fault](#)

Field Properties

[INF: How to Abort Creating a Field in Table Design](#)

[INF: Snapshot and Underlying Table Updated Until Snapshot Full](#)

[INF: System Objects Retain System Object Properties](#)

[PRACC9309: Formatted Memo Truncates Data At 255 Characters](#)

[PRB: Cannot Undo Property Settings in Table Design Mode](#)

[PRB: Error Message: Query Must Have at Least One Output Field](#)

[PRB: Nulls Not Concatenated with '+' Operator](#)

Other ODBC/ISAM Drivers

[INF: Filenames/Version Numbers, ODBC Files for MS Access V1.1](#)

[INF: Force MS Access to Use "Snapshot" Mode for Attached Data](#)

[INF: Upper Limit for QueryTimeout and LoginTimeout Parameters](#)

[INF: Using SQL Server ODBC Driver with Sybase Servers](#)

[INF: Windows for Workgroups, Novell, SQL Server, and Access](#)

[PRACC9302: Error When Import dBASE, FoxPro, or Paradox Table](#)

[PRACC9309: SQORA.DLL Does Not Allow Lengthy SQL Statements](#)

[PRACC9309: SQORA.DLL Does Not Allow Lengthy SQL](#)

[PRB: 'Couldn't Find Installable ISAM' Error](#)

[PRB: GP Fault If MSACCESS.INI Installable ISAM Path Uses '\.'](#)

[PRB: SQL Server Tables Are Read-Only Without a Unique Index](#)

[PRB: TransferDatabase Macro Accepts Invalid File Names](#)

[PRB: Troubleshooting 'ODBC - couldn't find ODBC.DLL' Error](#)

Other

[INF: Attached Table Must Be Opened as Dynaset from Modules](#)

[INF: Cannot Create Relationships with Attached Tables](#)

[INF: Database Normalization Basics](#)

[INF: Date/Time Data Type Provides Current Year Automatically](#)

[INF: Equivalent of Excel's Fill Down Functionality](#)

[INF: Function Deletes Table Without Using DoMenuItem](#)

[INF: Hiding Rows and Columns in Tables](#)

[INF: How NWIND.MDB Employee Field Photographs Were Created](#)

[INF: How to Find the Difference Between 2 Time Values](#)

[INF: Snapshot and Underlying Table Updated Until Snapshot Full](#)

[INF: System Objects Retain System Object Properties](#)

[PRB: Error When Opening Attached Table in Shared .MDB File](#)

[PRB: Report or Form Bound to a Table Locks the Table](#)

[PRB: Sizing Rows and Columns to Zero Works Differently](#)

Wizards

[INF: AccessWizards Button Grayed Out](#)

[INF: Mail Label Wizard Shows Blank Line in Label Size List Box](#)

[PRACC9307: Options in Form and Report Wizards Do Not Display](#)

[PRB: Changing Caption on Title Label Leaves Shadow Unmodified](#)

[PRB: Clicking Mouse in AccessWizard Background Changes Focus](#)

[PRB: Form Objects Not Placed on Grid by Wizard](#)

[PRB: FormWizard Labels Placed on Left Margin Grow Down](#)

[PRB: Grid Lines Not Visible in Access When Using Wizards](#)

[PRB: Mail Label Wizard Cuts Off First Field at Top of Page](#)

[PRB: Report Wizard Defaults To 8 1/2 by 11 Inch Page](#)

[PRB: ReportWizard Select No Fields for Form Still Shows All](#)

Layout

[INF: Currency Format Causes Report Alignment Problems](#)

[INF: Eliminating White Space in Reports with CanShrink and](#)

[INF: How to Create a Dynamic Crosstab Report](#)

[INF: How to Print Odd or Even Pages of a Report](#)

[INF: How to Specify a Custom Starting Page Number for a Report](#)

[INF: Page Header Will Span Full Width Of MultiColumn Report](#)

[INF: Print Total Number of Pages on Each Page of Report](#)

[PRB: Default Width in Report Design Does Not Reflect Margins](#)

[PRB: Four Problems/Solutions to Blank Pages in Printed Reports](#)

[PRB: Margins Do Not Relate to Printable Area](#)

Toolbox

[PRB: ESC While Creating Control Does not Deselect Control](#)

[PRB: Toolbox Selection Lost When Closed](#)

Action Arguments

- [INF: Close Action has Blank Line in Object Type Argument](#)
 - [INF: Form Controls to Increment and Decrement a Date](#)
 - [INF: How to Force a New Line in a MsgBox Statement](#)
 - [INF: OpenTable Action Argument Does Not List Attached Tables](#)
 - [INF: Requery Control Err Msg: There Is No Control Called...](#)
 - [INF: RunApp Action Command has 255 Character Argument Limit](#)
 - [INF: Using TransferDatabase to Export to Btrieve Format](#)
-
- [PRB: Cannot Use DoMenuItem for Open Database or New Database](#)
 - [PRB: Echo Without Arguments Causes Error](#)
 - [PRB: Filtering Form with OpenForm Action Causes "#Name?" Error](#)
 - [PRB: Invalid Reference to Field '<Value>' Using SetValue](#)
 - [PRB: Not Enough Stack Memory When Opening Form](#)
 - [PRB: Opening a Database Using a Sendkeys Macro](#)
 - [PRB: Out of Stack Memory Error Message with SendKeys Macro](#)
 - [PRB: Sendkeys Infinite Loop Macro](#)
 - [PRB: TransferDatabase Macro Accepts Invalid File Names](#)
 - [PRB: TransferSpreadsheet Action Does not Add File Extension](#)

Usage Other

[INF: Microsoft Access and Scanners](#)

[INF: Tile/Cascade Menu Options Inconsistent with Prog Manager](#)

[PRB: Arrange Icons do Not Locate Icons in Correct Place](#)

[PRB: Cursor Position Unchanged in Undo Paste](#)

[PRB: Out Of Memory When Pasting Text Into Module](#)

[PRB: SCSI Drives can Generate "corrupt" Error Message](#)

[PRB: Undo Cut Does Not Restore Previous Clipboard Contents](#)

Formatting

[INF: Flexible Formatting of Phone Numbers](#)

[INF: Screen Fonts May Not Match Printer Output Fonts](#)

[PRACC9301: Memo Field with Format Property Truncates Data](#)

[PRB: MS Access Short and Medium Time Formats Have Leading Zero](#)

[PRB: Weekday Disappears While Editing Long Date Format Field](#)

Data Editing

[INF: Cut/Paste for Immediate Window and SQL Window](#)

[INF: How to Change the Font for the Zoom Box](#)

[INF: How to Take a Button Out of Tab Order](#)

[INF: Prevent Adding New Records To Editable Form](#)

[PRB: Weekday Disappears While Editing Long Date Format Field](#)

Mailing Labels

[INF: Creating mailing labels for Dot Matrix printers](#)

[INF: Formatting a Label for Each Axis of a Graph](#)

[INF: Hints for Printing Mailing Labels](#)

[INF: How to Print Duplicate Mailing Labels for Each Record](#)

[INF: How to Skip Used Mailing Labels and Print Duplicates](#)

[INF: No Custom Printer Setup for Mailing Labels ReportWizard](#)

[PRB: Avery Labels 4143, 4160, and 5096 Don't Work on Laser Ptr](#)

[PRB: Printing on Avery 5267 labels](#)

[PRB: Two Labels on Top of Each Other Both Print](#)

Other

[INF: Attach R:BASE Table by Exporting dBASE Format in R:BASE](#)

[INF: Attached Table Must Be Opened as Dynaset from Modules](#)

[INF: Attaching Clipper Tables to Microsoft Access Databases](#)

[INF: Cannot Create Relationships with Attached Tables](#)

[INF: Default Properties do not Carry over on Attached Table](#)

[INF: Differences Between Native and Attached Tables](#)

[INF: Exporting Queries to ASCII](#)

[INF: Five Parameters to Set in ISAM Section of MSACCESS.INI](#)

[INF: How to Export the Results of a Query](#)

[INF: How To Get Microsoft Works Files into Access](#)

[INF: How to Import a Microsoft Windows Cardfile Data File](#)

[INF: How to Remove Attached Tables](#)

[INF: Import/Export Accepts Only 24-Hour Time Format](#)

[INF: Importing DataEase Data Files](#)

[INF: Marking Records Having Unmatched or Invalid Values](#)

[INF: Using Data From Access In A Word For Windows Print Merge](#)

[INF: Using Data from Access in a Word for Windows Print Merge](#)

[INF: Using DataPerfect data in Microsoft Access](#)

[PRACC9210: '#' Appears in Attached Table Name in Error Message](#)

[PRACC9211: Cannot Export Paradox/dBase Tables](#)

[PRACC9306: Import/Append Actions Return Errors for Each Record](#)

[PRACC9307: Invalid Characters Added to IN Clause Arguments](#)

[PRB: Case Sensitivity is Different with Attached Tables](#)

[PRB: Creating Multiple New Import/Export Specifications](#)

[PRB: Erroneous Characters Appear in Imported dBASE IV Database](#)

[PRB: Error When Opening Attached Table in Shared .MDB File](#)

[PRB: Export Table to Other Database Type Does not Create Index](#)

[PRB: Exporting Date Fields to Text Includes Time Format](#)

[PRB: GP Fault Using an Out of Date ODBC Driver](#)

[PRB: Out of Memory Error Message when Exporting](#)

[PRB: System Defaults Not Showing in Imp/Exp Setup](#)

[PRB: TransferSpreadsheet Action Does not Add File Extension](#)

Graphing

[INF: Graphing Current Sales Compared To Accumulated Sales](#)

[INF: How to Create a Chart Without a Title](#)

[INF: Microsoft Graph Changes May Not Be Retained](#)

[INF: OLE: New Graph Object Frames are Blank](#)

[INF: Ordering Data in a Graph](#)

[PRB: 'FastForward' Button Is Unavailable in Graph FormWizard](#)

[PRB: Graph Object Is Dimmed in Form View](#)

[PRB: GraphWizard Does Not Run or 'Syntax Error' Is Displayed](#)

[PRB: Unable to Add All Fields in Graph FormWizard](#)

[PRB: Unable to Specify Label Order in Graph Form Wizard](#)

Form Properties

[INF: Bitmaps Hide the Caption of a Push or Option Button](#)
[INF: Border Widths Not Supported with Raised or Sunken Effects](#)
[INF: Bound OLE Object Blank in Form View](#)
[INF: Cannot Change Borders of Up or Pressed Down Controls](#)
[INF: Control Names >40 Chars in Access Basic Require Brackets](#)
[INF: Controlling Design Mode at Runtime to Change Properties](#)
[INF: Form Must Close for Modal & Popup Settings to Take Effect](#)
[INF: Form View Allowed When ViewsAllowed Set to Datasheet](#)
[INF: Headers On Non-Continuous Forms Disable Scroll Bar](#)
[INF: How to Change Read-Only Properties at Run Time](#)
[INF: How to Create a Query By Form](#)
[INF: How to Determine Whether a Form Is Minimized](#)
[INF: How to Get Record Position Number from Record Selector](#)
[INF: How to Position Microsoft Access on the Screen](#)
[INF: How to Remove the Caption Bar from a Form](#)
[INF: How to Simulate Changing the Record Source of a Form](#)
[INF: Minimized Popup Forms Cannot be Easily Restored](#)
[INF: Placing the User Login Name on a Form or in the Title Bar](#)
[INF: Two Methods to Refer to a Field in a Previous Record](#)
[INF: Where Condition Refers to Fields not Controls](#)

[PRB: Allow Editing Set to Unavailable Does Not Work](#)
[PRB: Border Properties Ignored with SpecialEffect](#)
[PRB: Disabled Buttons Not Visible with ATI 1280x1024](#)
[PRB: Fields on Popup not Editable when Access is Minimized](#)
[PRB: Form Opened in Form View Is Completely Blank](#)
[PRB: Giving Controls Duplicate Name Does not Generate Error](#)
[PRB: HelpContextID Does Not Validate Negative Values](#)
[PRB: Minimizing Popup/Modal Form Loses Application's Focus](#)
[PRB: New Label Controls Inherit FormWizards' Label Text Align](#)
[PRB: No Arrows on Scroll Bars if Window Is Larger than Form](#)
[PRB: No Field List When Form Based on SQL Statement](#)
[PRB: Pressing CTRL+ALT+SPACEBAR Enters Control's Default Value](#)
[PRB: Unfreeze All Columns Does Not Restore Column Order](#)
[PRB: ValidationRule Parses Function Name w/o Parens to String](#)

Data Entry

[INF: Using Scroll Bars Doesn't Change Record Pointer](#)

[INF: How to Take a Button Out of Tab Order](#)

[INF: Converting Julian Dates with Access Basic](#)

[INF: A Macro to Check for Duplicate Values in a Field](#)

[INF: Prevent Adding New Records To Editable Form](#)

Global Settings (View Options)

[INF: END Key Behavior in Edit Boxes](#)

[INF: ENTER in Text Box Does Not Move to Next Line](#)

[INF: Making ENTER Add Lines in a Text Box](#)

[PRB: Arrow Keys Inconsistent Between Forms and Datasheets](#)

[PRB: Change Option Affects Only One Instance of Many of Access](#)

[PRB: Pressing HOME Key in Text Box Doesn't Move to Beginning](#)

DDE Server Topics

[INF: Communicating with Microsoft Access using NetDDE](#)
[INF: DDE Link Can't Find Data; Opens Second Copy of Server App](#)
[INF: Err "The other Application Quit" Closing Excel 4.0](#)
[INF: Executing an Access Macro from Visual Basic using DDE](#)
[INF: Getting the Data Types of Fields in a Table through DDE](#)
[INF: How to Buffer SQL Strings Using DDE](#)
[INF: How to Execute Macro Actions Using DDE](#)
[INF: How to Get SQL Pass-Through Functionality Using Q+E](#)
[INF: MS Access Does Not Support Parameter Queries as DDE Topic](#)
[INF: Part 1: DDE in Visual Basic to Request Data from Access](#)
[INF: Part 2, DDE in Visual Basic to Request Data from Access](#)
[INF: Sample DDE Macro Communicates with Microsoft Excel 4.0](#)
[INF: Sending the Current Record to Word for Windows with DDE](#)
[INF: Topics Supported by Access as a DDE Server](#)
[INF: Using DDE to Communicate with a Runtime Application](#)
[INF: Using Microsoft Access as a DDE Server](#)
[INF: Using the DDE Item SQLText to Request the Text of a Query](#)

[PRACC9306: Timeout While Waiting for DDE Response](#)

[PRB: Bound SubReport Does Not Show DDE Data](#)
[PRB: DDE App Name for Access Is Incorrect in Manual](#)
[PRB: DDE Statements Invalid in List Box or Combo Box Control](#)
[PRB: Paste Link to MS Excel Displays First Field Name Only](#)

Validation

[INF: Validating Data with Values From Another Table](#)

[INF: Validation Rule to Require all Upper Case in a Field](#)

[PRB: Table Design Doesn't Check Validity of Default Value](#)

Other

- [INF: Checking for Last Or First Record in a Macro](#)
- [INF: Cut and Paste Removes Blank Lines in Macro's Design View](#)
- [INF: Deciding When to Use Macros and When to Use Access Basic](#)
- [INF: How to Buffer SQL Strings Using DDE](#)
- [INF: How to Execute Macro Actions Using DDE](#)
- [INF: Sample Macro for Repair/Compact Operations](#)
- [INF: Tips on How to Debug Microsoft Access Macros](#)

- [PRB: Cannot Add Security Items to Custom Menu](#)
- [PRB: CTRL+HOME Navigation Problem in Macro Design Mode](#)
- [PRB: Out of Stack Memory Error Message with SendKeys Macro](#)
- [PRB: SendKeys Action doesn't work on Toggle Keys](#)
- [PRB: Sendkeys Infinite Loop Macro](#)

Other

[INF: "Form is read only" error](#)

[INF: "Form is read only" error](#)

[INF: '#Name?' or Control is Blank on Form Based on a Query](#)

[INF: Concatenating Fields in a Text box to Remove Blank Lines](#)

[INF: Creating Custom Navigation \(VCR\) Buttons on a Form](#)

[INF: Creating Floating Popup Menus](#)

[INF: Double-Clicking on Control Moves Control Unintentionally](#)

[INF: Four Ways to Move to a Record from a Combo Box Selection](#)

[INF: How to 'Freeze' a Record on a Form](#)

[INF: How to Check for Duplicate Values in Primary Key Fields](#)

[INF: How to Create a "Please Wait" Message](#)

[INF: How to Create a Running Sum on a Form](#)

[INF: How to Determine the View in Which a Form Is Displayed](#)

[INF: How to Determine Whether a Menu Item Is Selected](#)

[INF: How to Dim \(Gray out\) Menu Items with Access Basic](#)

[INF: How to Find the Difference Between 2 Time Values](#)

[INF: How to Get the Last Two Digits of the Year](#)

[INF: How to Make Forms Always Stay on Top of Other Windows](#)

[INF: How to Prevent Users from Adding New Records to a Form](#)

[INF: How to Print a Single Record from a Form in a Report](#)

[INF: How to Take a Button Out of Tab Order](#)

[INF: How to Use Graphic Images As Wallpaper For Forms](#)

[INF: Inconsistent Save and Save As Menus with Forms](#)

[INF: Recent File List Displays Deleted .MDB Files](#)

[INF: Round or Truncate Values to Desired Number of Decimals](#)

[INF: Sample Table Design to Support Questionnaire Applications](#)

[INF: Setting Form's Dynaset Property Puts Bookmark at EOF](#)

[PRACC1115: ApplyFilter Action Causes Form to be Read-Only](#)

[PRACC9210: Cannot Use ReportWizard Without Default Win Printer](#)

[PRACC9303: GPF Using OpenForm Action to Hide Modal Pop-Up Form](#)

[PRACC9305: GP Fault when Creating a Dynaset from Form Dynaset](#)

[PRACC9305: List Box Controls on Forms Painted Incorrectly](#)

[PRACC9307: Slow Repaint When Saving Form with Many Controls](#)

[PRB: 'Show All Records' Menu Choice not Available](#)

[PRB: 'Table Exclusively Locked' Error Switching to Form View](#)

[PRB: Can't Change Combo Box Value If ControlSource Is Counter](#)

[PRB: Cannot Add Security Items to Custom Menu](#)

[PRB: Cannot open multiple instances of the same form](#)

[PRB: Error Using Edit Find with Yes/No Data Type](#)

[PRB: ESC While Creating Control Does not Deselect Control](#)

[PRB: Extra Unwanted Information Appears in List Box](#)

[PRB: Filtering Form with OpenForm Action Causes "#Name?" Error](#)

[PRB: Focus Doesn't Shift When Find Accelerators Are Depressed](#)

[PRB: Form Displays Data from Deleted Field in Query](#)
[PRB: Form May Lose the Focus When SHIFT+F8 Is Pressed](#)
[PRB: Form Opened in Form View Is Completely Blank](#)
[PRB: Form Opens Blank Except for Header](#)
[PRB: Formatting Yes/No Data Type Example](#)
[PRB: Load From Query Dialog Title Does Not Match Menu](#)
[PRB: Macro on Subform Control Executed by Main Form](#)
[PRB: Navigation Button Unavailable in Main/Subform Wizard](#)
[PRB: Query by Form Returns No Rows When All Rows Expected](#)
[PRB: Switching Modes Disables Undo](#)
[PRB: Trailing Spaces Automatically Truncated During Data Entry](#)

Expressions

[INF: Create a Function to Capitalize Single Words and Phrases](#)

[INF: Displaying Start, Finish, and Elapsed Times on a Form](#)

[INF: Error Messages May Be Caused by International Setting](#)

[INF: Function to Get Date of Monday Prior to Current Day](#)

[INF: Functions to Left and Right Pad Character Strings](#)

[INF: Sorting/Grouping a Query Using a Portion of Field's Value](#)

[PRACC9303: GPF Using Invalid Iif\(\) as ControlSource](#)

[PRACC9308: Cannot Use First\(\) and Last\(\) Functions in a Report](#)

[PRB: #Error Appears Using TRIM\(\) in Forms or Reports](#)

[PRB: #Error When ControlName Used in ControlSource Expression](#)

[PRB: 'Circular Reference Caused by Alias' Error Message](#)

[PRB: Concatenation of Memo Fields Creates Text Field](#)

[PRB: Zoom Box Does not Appear](#)

Totals/Agg. Functions

[INF: Cannot Group By Memo or OLE Object Fields in Query](#)

[INF: How to Use a Query to Filter Unique Data](#)

[INF: How to use Aggregate Functions in a Query Without Errors](#)

[INF: Number of Times a Custom Function Executes in a Query](#)

[INF: Totals/Agg. Functions](#)

[PRB: <Expression> Not Part of Aggregate Function or Grouping](#)

[PRB: Properties Do not Reset Between Crosstab and Make Table](#)

Network Installation

[INF: Access Databases Truncated w/ LAN Manager 2.1a & Earlier](#)

[INF: Access Workstation Setup \(/N\) Instructions Are Confusing](#)

[INF: Installing Microsoft Access on a Diskless Workstation](#)

[INF: Running MS-DOS SHARE with Windows for Workgroups](#)

[INF: Two Error Messages with Database on Workgroups Server](#)

[INF: Windows for Workgroups, Novell, SQL Server, and Access](#)

[PRACC9303: File Truncation on Password-Protected Servers](#)

[PRB: "Error Writing to File 'MSAIN100.DLL'" During Install](#)

[PRB: Error Message: Must Run SETUP /A Before Running SETUP](#)

[PRB: Error When Installing Access on DEC Pathworks Network](#)

Parameters/Referencing

[INF: How to Create a Parameter In\(\) Statement](#)

[INF: How to Create a Query By Form](#)

[INF: Parameter Queries, Wildcards, and SQL Server](#)

[INF: Passing Values from a Form to a Parameter Query](#)

[INF: Query with Parameters to Evaluate Complex Criteria](#)

[INF: Returning Case Sensitive Match in Queries](#)

[INF: Showing All Records \(Including Null\) in a Parameter Query](#)

[INF: Using '<' and '>' and '=' Operators in Query Parameters](#)

[INF: Using the LIKE Operator with Parameter Queries](#)

[PRACC9305: Hard-Coded and Passed Wildcards Different in Query](#)

[PRACC9306: ERR: Undefined Function Referencing Column Property](#)

[PRACC9307: Invalid Characters Added to IN Clause Arguments](#)

[PRB: Access Not Prompting for Second Parameter in Query](#)

[PRB: Can't Exit Extend Mode in Parameters Dialog Box](#)

[PRB: No Implicit Parameters on CrossTab Queries](#)

[PRB: Query Error Caused by Colon in Field Name](#)

[PRB: Query Results in an "Overflow" Error](#)

Combo/List Box

[INF: Base Combo Box on Parameter Query to Filter Values](#)

[INF: Changing the RowSource of a ListBox/ComboBox at Runtime](#)

[INF: Displaying List or Combo Boxes with More Than One Field](#)

[INF: Filling a Multiple-Column List Box by Using Access Basic](#)

[INF: How to Create Synchronized Combo Boxes](#)

[INF: How to Select More than One Item from a List Box](#)

[INF: How to Synchronize Two Combo Boxes on a Form](#)

[INF: Limit to List Set to Yes Needs a Unique Entry](#)

[INF: Use SendKeys Macro Action to Open Combo Box Automatically](#)

[INF: Using Column Property of Combo Box to Update Text Box](#)

[PRACC9301: Wrong RowSourceType for Combo Box Causes GP Fault](#)

[PRB: #NAME? Referring to Combo Box Column in Subform](#)

[PRB: Can't Change Combo Box Value If ControlSource Is Counter](#)

[PRB: Combo Box Displays Only 255 Characters of Memo Field](#)

[PRB: Date/Time Formats Not Used In List or Combo Box Controls](#)

[PRB: DDE Statements Invalid in List Box or Combo Box Control](#)

[PRB: Empty Row of List Box Can Be Selected](#)

[PRB: LimitToList Combo Box Will not Let You Leave After Delete](#)

[PRB: List Box Does Not Scroll when Scroll Box Is Dragged](#)

[PRB: No Error Msg When BoundColumn Greater Than ColumnCount](#)

[PRB: Primary Keys Not Used to Order Data in Combo/List Boxes](#)

[PRB: Scroll Box Does Not Move Down Scroll Bar in Combo Box](#)

[PRB: Undo Property Setting Command Appears in Macro Design](#)

VT Objects

[INF: Performing an SQL Bulk Action Query from Access Basic](#)
[INF: Obtain Name of Current User Database Using Access Basic](#)
[INF: Creating a Snapshot Is Slower Than Creating a Dynaset](#)
[INF: Attached Table Must Be Opened as Dynaset from Modules](#)
[INF: Setting Form's Dynaset Property Puts Bookmark at EOF](#)
[PRB: Cannot Rollback in Transaction without Closing Database](#)
[PRB: Close Dynaset Based on Form Multiple Times](#)
[PRB: Cannot Perform OpenTable on an Attached Table](#)
[INF: Function Deletes Table Without Using DoMenuItem](#)
[PRB: 'Invalid Database Object' Occurs Inexplicably](#)
[INF: Enforcing Transactions on Attached SQL Server Tables](#)
[INF: Microsoft Access Does Not Use Record Numbers](#)
[INF: Criteria Parameters Require Concatenated References](#)
[INF: Changing Area Codes Based on a Telephone Number Prefix](#)
[INF: Short Description of a Dynaset](#)
[INF: Error Messages When Concatenating Variables/Controls](#)
[PRACC9303: GPF When Using ListParameters Method on Database](#)
[PRB: 'Fieldname' Is not an Index in This Table](#)
[PRACC9305: GP Fault when Creating a Dynaset from Form Dynaset](#)
[PRB: Seek Method Is Faster Than Find Method](#)
[INF: How to Prevent Users from Adding New Records to a Form](#)
[INF: How to Avoid Abandoned Group Headers in Reports](#)
[INF: OpenQuery is not a QueryDef Method](#)
[INF: Snapshot and Underlying Table Updated Until Snapshot Full](#)
[PRACC9309: Incorrect Error Message Using OpenTable Method](#)

Relationships

[INF: Changing the Relationship Between Two Existing Tables](#)

[INF: How to List the Related Tables in a Database](#)

[INF: How to Remove Relationships](#)

[INF: Join Type Mismatch Errors Caught at Runtime](#)

[PRB: Can't Create Two Relationships Between Same Tables](#)

Security

[INF: Access Security and RunSQL With Owner's Permission Denied](#)

[INF: Determining Which User You Are Logged](#)

[INF: How to Create a Password Textbox](#)

[INF: How to Give Users the Ability to Change Passwords](#)

[INF: Network Users Can Open Their Own SYSTEM.MDA Exclusively](#)

[INF: No Permission to Modify Data in a Table](#)

[INF: Planning Database Security](#)

[INF: Security Settings for Modifying Data in a Query Dynaset](#)

[PRACC9303: Password Is Case-Sensitive But Verification Is Not](#)

[PRACC9303: User with Full Permissions Cannot Run Query](#)

[PRB: Can't db.OpenTable With Only Read Definition Permission](#)

[PRB: Don't have Permission to Modify Secured Report](#)

[PRB: No System Table Permissions After Admin User Deleted](#)

[PRB: Unable to Change Permissions \(Security\)](#)

[PRB: User with Full Permissions Cannot Change a Query](#)

Control Properties

[INF: Allowing an Ampersand to Appear in a Caption](#)
[INF: Creating a Proper Function for Data Entry](#)
[INF: Default Value Of A Combo Box Equal To The First Row](#)
[INF: Fill New Record with Data from Prev Record Automatically](#)
[INF: General Number Format Ignores Decimal Number Setting](#)
[INF: Height/Width Settings Do Not Affect Size of Objects](#)
[INF: Horizontal Sizing Controls Availability on Controls](#)
[INF: How to Create a Password Textbox](#)
[INF: How to Show Field in Datasheet View But Not Form View](#)
[INF: Resizing Multiple Controls Simultaneously](#)
[INF: Size To Fit a Label Without Using Layout Menu](#)
[INF: Status Bar Text Too Long for Auto Repeat Property](#)
[INF: Using Ampersands in Shortcut Keys, Displaying in Labels](#)
[INF: Using DLookup\(\) to Look Up Values in Forms and Reports](#)

[PRACC9305: GP Fault with Button Caption Text Longer Than 2048](#)
[PRACC9308: GP Fault Referencing Column Property of ComboBox](#)
[PRACC9309: Formatted Memo Truncates Data At 255 Characters](#)

[PRB: Apply Default Doesn't Change Height or Width of Labels](#)
[PRB: Buttons Pasted into Option Groups have Option Value](#)
[PRB: Changes in Subform Design do not Update in Main Form](#)
[PRB: Disabled Buttons Not Visible with ATI 1280x1024](#)
[PRB: Err Msg: Macro " Not Found](#)
[PRB: Incorrect Results Received with DSUM Function](#)
[PRB: Label Placed in, Not to Left of, Command Button on Form](#)
[PRB: Leading Space in Property Sheet Entry May Cause Problems](#)
[PRB: No CanShrink property for Check Boxes](#)
[PRB: Nulls Not Concatenated with '+' Operator](#)
[PRB: Only One Button Accepts Selection in Yes/No Option Group](#)
[PRB: Print Preview Displays Only First Line of Toggle Button](#)
[PRB: Random Characters in Text Box Bound to OLE Object Field](#)
[PRB: Short Label Autosizes to Cursor Size](#)
[PRB: Text Label Truncated Printing Form on Epson 9-Pin Printer](#)
[PRB: Toggle Button with CancelEvent on BeforeUpdate Property](#)
[PRB: Trailing CR/LFs Stripped From Text Box When Retrieved](#)
[PRB: Trailing CR/LFs Stripped from Text Box when Retrieved](#)
[PRB: Unable to Paste Information into Property Sheet](#)
[PRB: Unable to Use Hexadecimal or Octal Values in Properties](#)
[PRB: Valid Sum Function Returns '#Name?' in Calculated Control](#)

Btrieve

[INF: Btrieve 6.0 Data Type Does Not Work with Access](#)

[INF: Definitions for Settings in WIN.INI Btrieve Section](#)

[INF: Export Btrieve Table Without Existing Xtrieve FILE.DDF](#)

[PRB: "Btrieve-Out of Resources" Error Msg with Attached Table](#)

[PRB: "Internal Database Error \(-5001\)" Msg Using Btrieve NLM](#)

[PRB: 'Couldn't find object <tablename>' Error with Btrieve](#)

[PRB: Btrieve Files Created with Btrieve 6.0 NLM Not Supported](#)

[PRB: Empty Table List When Attaching Btrieve Table](#)

[PRB: Error Message When Attaching Btrieve Table](#)

[PRB: Error Message: Too Many Memo or OLE Fields](#)

[PRB: Exclusively Locked Error When Attaching to Btrieve Table](#)

[PRB: Incorrect Btrieve Error Message, \[Btrieve ISAM\] section](#)

[PRB: Out of Memory Error Message when Exporting](#)

[PRB: Unexpected Error from External Database Driver \[20\]](#)

[PRB: Unexpected Error from External Database Driver \[22\]](#)

Fox/Xbase

[INF: Attaching to FoxPro Data in Microsoft Access](#)

[INF: Doc Err: dBASE SET DELETED Characteristic in MS Access](#)

[INF: How Access Variable-Length Fields Export to dBASE](#)

[INF: Importing, Exporting, and Attaching FoxPro 2.0, 2.5 Data](#)

[INF: Importing Several dBASE Databases Simultaneously](#)

[PRACC9211: Cannot Overwrite Corrupt Paradox/dBASE Files](#)

[PRACC9303: Hang Importing .DBF Files Longer Than dBASE IV Max.](#)

[PRB: Irregular Characters in Attached dBASE IV Memo Field](#)

[PRB: Rounding Errors Importing/Exporting dBASE IV Values](#)

[PRB: Two Users Can Open the Same dBASE Database Exclusive](#)

Make Table/Update/Append/Delete

[INF: "Record Lock Threshold Exceeded" with Large Action Query](#)
[INF: Access Security and RunSQL With Owner's Permission Denied](#)
[INF: Changing the Relationship Between Two Existing Tables](#)
[INF: Creating Queries to Execute Cascading Updates](#)
[INF: Empty Table Created by SELECT INTO with No Matching Rows](#)
[INF: Equivalent of Excel's Fill Down Functionality](#)
[INF: How to Force a Cascading Delete](#)
[INF: How to Query for Duplicate Records](#)
[INF: How to Update a Table Based on Values in a Second Table](#)
[INF: MakeTable Query Does not Copy Field Descriptions](#)
[INF: Marking Records Having Unmatched or Invalid Values](#)
[INF: Sample Table Design to Support Questionnaire Applications](#)
[INF: Updating a Separate Table When a Value Changes on a Form](#)
[INF: Updating Summary Information With Data From Second Table](#)
[INF: Use a Query to Parse a Full Name into Two Fields](#)
[INF: Use Append Query to Set Initial Value of Counter Field](#)

[PRACC9301: Incorrect Action Query Message Box Text](#)
[PRACC9302: Running Delete Query Gives Misleading Error Msg](#)
[PRACC9303: GPF on Update Query Due to Query Design Option](#)

[PRB: Append Query Results in Duplicate Counter Numbers](#)
[PRB: Concatenation of Memo Fields Creates Text Field](#)
[PRB: Make-Table Query Causes Existing Table to Be Deleted](#)
[PRB: Update Queries Do Not Check Validation Rules](#)
[PRB: Update Query on More than One Table Gives Inaccurate Data](#)

Libraries (MDA)

[INF: Creating, Debugging, and Using an Access Library](#)

[INF: How to Modify the Toolbar](#)

[INF: Using Undocumented CreateForm\(\),CreateReport\(\) Functions](#)

Error Handling

[INF: Access Basic Error-Handling Supplement](#)

[INF: Creating, Debugging, and Using an Access Library](#)

[INF: DoEvents, SendKeys May Not Be Supported in UDFs](#)

[INF: Error\(\) Function Does Not Return Detailed Information](#)

[PRACC9210: Opening Objects in Transactions Can Cause Rollback](#)

[PRB: Abnormal Action When Debugging Runtime Error](#)

[PRB: Autokeys Macros with the Same Name Are Still Executed](#)

[PRB: Form Handle Not Released When Closed with Basic Code](#)

Application Design

[INF: Additions to User's Guide Appendix A Specifications](#)

[INF: Converting Julian Dates with Access Basic](#)

[INF: Creating, Debugging, and Using an Access Library](#)

[INF: Default Properties do not Carry over on Attached Table](#)

[INF: Disk image of Companion Disk](#)

[INF: How to Use Graphic Images As Wallpaper For Forms](#)

[INF: How to Wait for a Shelled Process to Finish](#)

[INF: INF: Function to Determine Runtime or Retail Version](#)

[INF: Maximum Database Size Increased in Version 1.1](#)

[INF: Nulls Allowed in Foreign Key with Referential Integrity](#)

[PRB: Opening a Database Using a Sendkeys Macro](#)

[PRB: Reordering Rows in Table or Macro Design Window](#)

[PRB: SetValue Does not Trigger Form Events](#)

SQL Server/SPT.DLL

[INF: Connecting to Sybase SQL Server from an ODBC application](#)
[INF: Determining the Version of Catalog Stored Procedures](#)
[INF: Enforcing Transactions on Attached SQL Server Tables](#)
[INF: Exporting to SQL Server Does Not Create Indexes](#)
[INF: Force MS Access to Use "Snapshot" Mode for Attached Data](#)
[INF: How Access Handles Logins to Attached SQL Tables](#)
[INF: How Access Uses SQL Server Connections](#)
[INF: How to Determine Version of INSTCAT.SQL File on Server](#)
[INF: How to Query SQL Server Tables Without Attaching](#)
[INF: Implementing Counter Values on Attached SQL Server Tables](#)
[INF: ODBC Setup for Access and SQL Server](#)
[INF: Parameter Queries, Wildcards, and SQL Server](#)
[INF: Performing an SQL Bulk Action Query from Access Basic](#)
[INF: Query Optimizing Tips for Attached SQL Database Tables](#)
[INF: Requirements to Connect to Sybase SQL Server](#)
[INF: SQL Server Triggers and Key Values to and from Access](#)
[INF: SQL Server Views Read Only When Attached](#)
[INF: Storing SQL Database Login IDs and Passwords Locally](#)
[INF: Strange Stored Procedures Found in SYSOBJECTS Table](#)
[INF: Unsupported SQL Keywords Not Listed in Access](#)
[INF: Upper Limit for QueryTimeout and LoginTimeout Parameters](#)
[INF: Using SQL Server ODBC Driver with Sybase Servers](#)
[INF: Using TransferDatabase Macro to Attach to SQL Server Data](#)
[INF: Using WIN.INI & ODBC.INI When Attaching SQL Server Table](#)

[PRACC9304: SQL 4-Byte Values Incorrectly Mapped to Text](#)

[PRACC9304: SQL Delete Triggers Function Incorrectly](#)

[PRACC9305: Unable to Import a SQL Server View](#)

[PRB: 'Unable to Load Communication Module'](#)

[PRB: Connection Not Open Error Message](#)

[PRB: Data Source Not Found, No Default Driver Specified Error](#)

[PRB: Driver Cannot Be Found When Attaching to SQL Server](#)

[PRB: ErrMsg: CREATE TABLE Permission Denied in Export to SQL](#)

[PRB: Error: Unable to Connect to Data Source](#)

[PRB: Incorrect Catalog Stored Procedures Error Message](#)

[PRB: Microsoft Access Can't Recognize SQL Server Double Index](#)

[PRB: Operation Stops when Editing Attached SQL Tables](#)

[PRB: Query Fails When Linked SQL Server Table Cannot Be Found](#)

[PRB: SQL Data Sources Dialog Box Lists No Servers](#)

[PRB: Stored procedure 'PROCEDURE' not found -- Error Message](#)

[PRB: Unable to Connect to Attached SQL Server Table, Error Msg](#)

Operating Environment Other

[INF: Microsoft Access and Scanners](#)

[INF: Network Users Can Open Their Own SYSTEM.MDA Exclusively](#)

[INF: Norton Desktop 1.x Is Not Compatible with Access](#)

[INF: Older Versions of Shared DLLs Cause Problems with Access](#)

[INF: Opening Databases as Read-Only Causes Error Message](#)

[INF: Using SHARE.EXE and VSHARE.EXE with Microsoft Access](#)

[PRACC9301: Non-TrueType Fonts with 8514.DRV Causes GP Fault](#)

[PRB: '|' Isn't A Valid Path Error Message on Start](#)

[PRB: Arrange Icons Doesn't Work on More Than 40 Windows.](#)

[PRB: GP Fault If MSACCESS.INI Installable ISAM Path Uses '\.'](#)

[PRB: GPF with Diamond Viper VESA Local Bus Video Driver V1.01](#)

[PRB: Help Gives Error Importing File From Floppy Disk](#)

[PRB: Insert Object Dialog Box Has No 'All Files \(*.*\)' Option](#)

[PRB: Low Number of Share Locks Can Cause Error Message](#)

[PRB: Microsoft Access Issues with MS-DOS 6.0](#)

[PRB: Stacker 3.0 CHECK.EXE Reports Database Files as Corrupt](#)

[PRB: Text May be Hard to Read on an EGA Screen](#)

[PRB: Undo Cut Does Not Restore Previous Clipboard Contents](#)

Primary Key/Indexes

[INF: Incrementing the Numeric Portion of a String](#)

[INF: Compound Indexes Must Restrict First Indexed Field](#)

API/DLL Calls

[INF: Another Method of Quitting Windows](#)
[INF: Connecting to the First Available Network Drive](#)
[INF: Conversions for C Data Types in Access Basic](#)
[INF: Creating Floating Popup Menus](#)
[INF: Determine How Many Instances of Application Are Active](#)
[INF: How to Change the 'Microsoft Access' Window Caption](#)
[INF: How to Change the Caption on the Access Window](#)
[INF: How to Check a Menu Item Using Access Basic](#)
[INF: How to Convert Twips to Pixels](#)
[INF: How to Determine if a Specific Windows Program Is Running](#)
[INF: How to Determine the View in Which a Form Is Displayed](#)
[INF: How to Determine When a Shelled Process Has Terminated](#)
[INF: How to Determine Whether a Form Is Minimized](#)
[INF: How to Determine Whether a Menu Item Is Selected](#)
[INF: How to Dial a Phone Number in Microsoft Access](#)
[INF: How to Dim \(Gray out\) Menu Items with Access Basic](#)
[INF: How to Disable the Menu Bar When Previewing a Report](#)
[INF: How to Display Immediate Window Without Module Window](#)
[INF: How to Obtain Access Startup Directory with API Calls](#)
[INF: How to Perform a Screen Dump of Your Forms](#)
[INF: How to Play .WAV Sounds on Events in Microsoft Access](#)
[INF: How to Position Microsoft Access on the Screen](#)
[INF: How to Remove the Caption Bar from a Form](#)
[INF: How to Retrieve Information from the Clipboard](#)
[INF: How to Retrieve Windows For Workgroups User Information](#)
[INF: How to Send Information to the Clipboard](#)
[INF: How To Tell If Windows For Workgroups Is Running](#)
[INF: How to Use Get and Write PrivateProfileString](#)
[INF: How To Use The CTL3D.DLL In Your Access Database](#)
[INF: How to Use the Windows OpenFile Dialog Box](#)
[INF: Maximizing, Minimizing, and Restoring Access in Modules](#)
[INF: Obtain Name of Current User Database Using Access Basic](#)
[INF: Passing a Null Pointer to an External DLL](#)
[INF: Placing the User Login Name on a Form or in the Title Bar](#)
[INF: Sending a MAPI Mail Message from Microsoft Access](#)
[INF: Using Access Basic CopyFile\(\) Function to Copy Disk Files](#)
[INF: Using Microsoft Access Macro to Quit Windows](#)
[INF: Using SetSysModalWindow API in MS Access Basic](#)
[INF: Using the GetSystemMetrics API Call in Access Basic](#)
[INF: Using the Windows 3.1 API to Connect to Network Resources](#)

Other

[INF: '#Name?' or Control is Blank on Form Based on a Query](#)
[INF: Adding a Dynamic Counter to a Query to Count Records](#)
[INF: Calculated Fields Must Be Shown to Perform Calculations](#)
[INF: Copy/Paste Operations from Forms to Queries](#)
[INF: Count\(\) Ignores Null Values](#)
[INF: Cut/Paste for Immediate Window and SQL Window](#)
[INF: Determining the Existence of a Query in Access](#)
[INF: Equivalent of Excel's Fill Down Functionality](#)
[INF: Exporting Queries to ASCII](#)
[INF: How to Create a Parameter In\(\) Statement](#)
[INF: How to Export the Results of a Query](#)
[INF: How to Query SQL Server Tables Without Attaching](#)
[INF: How to use Aggregate Functions in a Query Without Errors](#)
[INF: Like vs. '=' with Wildcard Characters in Query Searches](#)
[INF: Limit of AND's in SQL Select Statement](#)
[INF: Online Help for Sigma Button Is Incorrect](#)
[INF: Operators as Field Names in Query Cause Error Message](#)
[INF: Queries Sort by Order of IN\(\) Parameters](#)
[INF: Reporting the Median Value of a Group of Records](#)
[INF: Sample Table Design to Support Questionnaire Applications](#)
[INF: Searching for Characters/Strings in Records](#)
[INF: Sorting String Values Based on their Numeric Values](#)
[INF: Sorting/Grouping a Query Using a Portion of Field's Value](#)
[INF: Unsupported SQL Keywords Not Listed in Access](#)
[INF: Use of Constants as Field Names in a Query](#)
[INF: Using a Subtract Query to Find Unmatched Records](#)

[PRACC316: Can't Paste Column If Focus Is on One Cell in Query](#)
[PRACC9211: Unique Values Only Select Query Shows Duplicates](#)
[PRACC9302: Syntax Error When "\(" is First Chr in Criteria Cell](#)
[PRACC9303: Can't Scroll Query Grid Using Arrow Keys or Mouse](#)
[PRACC9303: Descending Sort Query Returns Wrong Number of Rows](#)
[PRACC9303: Error: Not Enough System Resources in Access](#)
[PRACC9304: Query Progress Meter Reports Status Incorrectly](#)

[PRB: 'Circular Reference Caused by Alias' Error Message](#)
[PRB: 'Field0' Appears in a Query But Not in the Source Table](#)
[PRB: 'Show All Records' Menu Choice not Available](#)
[PRB: 'Too Many Expressions in Distinct Output' Error](#)
[PRB: Can't Scroll Query Grid with ARROW Key or Mouse](#)
[PRB: Case Sensitivity is Different with Attached Tables](#)
[PRB: Changed Field Names In Table Are Not Reflected In Query](#)
[PRB: Column in QBE Grid Keeps Focus After Clicking Field List](#)
[PRB: Criteria Sections Combined into Same Column in QBE Grid](#)
[PRB: CTRL+C Does Not Copy SQL Statement in Windows 3.0](#)
[PRB: Error Message: 'Couldn't Find Input Table or Query'](#)

[PRB: Error Message: Query Must Have at Least One Output Field](#)

[PRB: Expression in Query Grid Doesn't Show All Text](#)

[PRB: Long WHERE Clause Is Truncated After View SQL Command](#)

[PRB: Microsoft Access Cannot Paste Entire Column in Query Grid](#)

[PRB: NWIND.MDB Query with Criteria Displays Data in Odd Order](#)

[PRB: ORDER BYs Must Be Output Columns in DISTINCT Queries](#)

[PRB: Query by Form Returns No Rows When All Rows Expected](#)

[PRB: Strange Characters Appear When Pasting in the Query Grid](#)

[PRB: When Column Selected and Deleted, Query Grid Disappears](#)

Excel/Lotus

[INF: Access Table with Time Field Exports Strangely to Excel](#)

[INF: Imported Excel Date Fields Off By 4 Years](#)

[INF: Sample DDE Macro Communicates with Microsoft Excel 4.0](#)

[PRACC9211: Table Export to Excel Does Not Set Database Name](#)

[PRACC9303: Access Hangs Importing Big Lotus 1-2-3 Spreadsheets](#)

[PRACC9304: Importing Numeric Column Names in Spreadsheets](#)

[PRACC9308: Problem Exporting "Single" Numeric Data to MS Excel](#)

[PRB: "Invalid Range" Error When Importing from MS Excel](#)

[PRB: 'Invalid Range' Error When Importing Spreadsheet Range](#)

[PRB: Excel Custom Date Format Not Detected by Import](#)

[PRB: Paste Link to MS Excel Displays First Field Name Only](#)

Actions

[INF: A Macro to Check for Duplicate Values in a Field](#)
[INF: AutoExec Macro Invoked Only When Database Is Opened](#)
[INF: Checking for Last Or First Record in a Macro](#)
[INF: CopyObject Appears to Copy Wrong Object](#)
[INF: Error Msg: Can't Hide the Control that Has Focus](#)
[INF: Form Controls to Increment and Decrement a Date](#)
[INF: Form OnOpen Event May Not Set Values of Unbound Controls](#)
[INF: How to Create For Next Loops in a Macro](#)
[INF: How to Hide the Database Window at Startup](#)
[INF: How to Remove Attached Tables](#)
[INF: How to Requery Combo Box on Subform](#)
[INF: Macro to Enter Date Record Was Last Modified](#)
[INF: Minimized Popup Forms Cannot be Easily Restored](#)
[INF: Recent File List Displays Deleted .MDB Files](#)
[INF: Requery, Repaint, and F9/Records Refresh](#)
[INF: Set Focus to a Subform Control Using GoToControl Action](#)
[INF: Using SendKeys to Change Option Menus](#)
[INF: Using TransferDatabase Macro to Attach to SQL Server Data](#)

[PRACC1115: ApplyFilter Action Causes Form to be Read-Only](#)
[PRACC9303: GPF Using OpenForm Action to Hide Modal Pop-Up Form](#)
[PRACC9304: RunMacro May Not Work with Large Repeat Count](#)

[PRB: GoToRecord Macro Doesn't Work on Subform](#)
[PRB: Alert Appears When it Should not When Running a Macro](#)
[PRB: CancelEvent Action in OnClose Macro Prevents Exit](#)
[PRB: Cannot Set Focus to Any Object Except MS Access Menu](#)
[PRB: Echo Without Arguments Causes Error](#)
[PRB: GoToControl to Control with Focus in AfterUpdate Ignored](#)
[PRB: No AutoRepeat with GoToRecord/Next and DoMenuItem](#)
[PRB: Opening a Database Using a Sendkeys Macro](#)
[PRB: OpenTable Action Ignores Data Mode If Table Open](#)
[PRB: Out of Stack Memory Error Message with SendKeys Macro](#)
[PRB: Restore Macro Action Does Not Appear to Work](#)
[PRB: RunCode Macro May Not Evaluate Conditions](#)
[PRB: Sendkeys Infinite Loop Macro](#)
[PRB: SendKeys Macro Does Not Run](#)
[PRB: Unable to Close Access From Control Box](#)
[PRB: UndoCurrentRecord In BeforeUpdate Property Causes Error](#)
[PRB: Wait Argument in SendKeys May Not Stop Macro Processing](#)

Database Design

[INF: How to Create a Multiuser Custom Counter](#)

[INF: Nulls Allowed in Foreign Key with Referential Integrity](#)

[INF: Sample Table Design to Support Questionnaire Applications](#)

[PRACC9210: Null Boolean Values Default to False](#)

[PRB: Can't Change Combo Box Value If ControlSource Is Counter](#)

[PRB: Can't Check to See If Database Was Opened Exclusively](#)

[PRB: Error Message: Too Many Memo or OLE Fields](#)

[PRB: New OLE Field is Created as Text Box in a New Form](#)

[PRB: ValidationRule Parses Function Name w/o Parens to String](#)

Data Types

[INF: Access Table with Time Field Exports Strangely to Excel](#)

[INF: Field Size of 0 Treated as 255](#)

[INF: How to Create a Multiuser Custom Counter](#)

[INF: Implementing Counter Values on Attached SQL Server Tables](#)

[INF: Maximum Number of Rows in Table When Counter is the Key](#)

[INF: Use Append Query to Set Initial Value of Counter Field](#)

[INF: Using Date/Time Data Type and Calculating Elapsed Time](#)

[PRB: Append Query Results in Duplicate Counter Numbers](#)

[PRB: Cannot Use 'y' 'n' 't' or 'f' in Yes/No Field](#)

[PRB: Concatenation of Memo Fields Creates Text Field](#)

[PRB: New OLE Field is Created as Text Box in a New Form](#)

Other

[INF: How to Create a Multiuser Custom Counter](#)
[INF: Introduction to Windows Programming for MS-DOS Programmer](#)
[INF: Determining the Existence of a Query in Access](#)
[INF: Programming with the Object Analyzer Engine and Utilities](#)
[INF: Easy Way to Add Items to the Help Menu](#)
[INF: Practice Using Object Analyzer Commands](#)
[INF: Creating a Proper Function for Data Entry](#)
[INF: How to List the Related Tables in a Database](#)
[INF: The Immediate Window for More than Access Basic](#)
[INF: Generating a Random Temp File Name](#)
[INF: Deciding When to Use Macros and When to Use Access Basic](#)
[PRB: Reserved Words won't Format \(Capitalize\) in Module Window](#)
[INF: Cut/Paste for Immediate Window and SQL Window](#)
[INF: Obtain Name of Current User Database Using Access Basic](#)
[PRB: 'Compile All' is Enabled After Saving a Module](#)
[INF: Flexible Formatting of Phone Numbers](#)
[INF: How to Get SQL Pass-Through Functionality Using Q+E](#)
[PRB: DoCmd OpenReport Opens Rpt in 3 Views or Prints to File](#)
[PRB: Reason Val\(\) Function Returns Invalid Use of Null Message](#)
[PRB: Invalid Use of Null Alert Assigning Null Control](#)
[INF: How to Create For Next Loops in a Macro](#)
[INF: How To Tell If An Object Exists in the Database](#)
[PRB: Scrolling to Bottom of Module Window Goes Past Typed Info](#)
[INF: Currency is Significant to Four Digits](#)
[INF: Unsupported SQL Keywords Not Listed in Access](#)
[INF: Converting Julian Dates with Access Basic](#)
[INF: How to Find Out If a Form Exists in a Database](#)
[INF: How to 'Freeze' a Record on a Form](#)
[INF: DeleteControl, DeleteReportControl, and CreateGroupLevel](#)
[INF: CreateControl\(\) and CreateReportControl\(\) Functions](#)
[PRB: Cannot Rollback in Transaction without Closing Database](#)
[INF: How to Pass a Single Byte of Data to Windows API Calls](#)
[PRB: Hourglass May Not Change After Illegal Function Call](#)
[INF: How to Use DDE to Fill a List Box](#)
[PRB: Can't Check to See If Database Was Opened Exclusively](#)
[PRB: Full Path Causes Endless Loop in StartApp\(\) Example](#)
[INF: How to Retrieve Information from the Clipboard](#)
[INF: Creating Custom Navigation \(VCR\) Buttons on a Form](#)
[INF: How to Reference Controls or Fields on a Subform](#)
[INF: Filling a Multiple-Column List Box by Using Access Basic](#)
[INF: How to Get the Last Two Digits of the Year](#)
[INF: Using Access Basic to Derive a Statistical Median](#)
[INF: How to Implement a Timer](#)
[PRB: DateAdd 'w' Option Doesn't Work As Expected](#)
[PRB: DateDiff 'W' Option Does Not Work](#)
[INF: Criteria Parameters Require Concatenated References](#)

[INF: Changing Area Codes Based on a Telephone Number Prefix](#)
[INF: dBASE's REPORT FORM Command in Access Basic](#)
[INF: Accessing a Report Dynaset Requires a .Form Reference](#)
[PRB: Compile Error When Attempting to Use an AccessWizard](#)
[INF: Items in a Module Run Menu Are Unavailable \(Grayed Out\)](#)
[PRB: Objects Cannot Be User-Defined Data Type Elements](#)
[INF: How to Determine When a Shelled Process Has Terminated](#)
[INF: How to Convert to and from Unsigned Integers](#)
[INF: Using Microsoft Word to Edit and Spell Check Memo Field](#)
[INF: How to Use Access Basic to Derive Statistical Mode](#)
[INF: Error Messages When Concatenating Variables/Controls](#)
[INF: Conversions for C Data Types in Access Basic](#)
[PRACC9303: Setting OLE Object Property May Cause GP Fault](#)
[PRACC9210: StDevP\(\) with All NULL Records Returns #ERROR](#)
[PRACC9210: Opening Objects in Transactions Can Cause Rollback](#)
[PRACC9303: GPF Referencing Form Dynaset with Form Variables](#)
[INF: Writing Functions Called from Events or Expressions](#)
[INF: Round or Truncate Values to Desired Number of Decimals](#)
[PRB: Access Basic Does Not Support Yes and No Action Arguments](#)
[INF: How to Determine Number of Working Days Between Two Dates](#)
[INF: Explanation of Option Compare Database](#)
[INF: Sorting Titles Without Leading Articles \(The, An, etc\)](#)
[INF: Sorting Titles Without Leading Articles \(The, An, etc\)](#)
[INF: Data Type of Value Overrides Data Type of Variable](#)
[INF: How to Import a Microsoft Windows Cardfile Data File](#)
[INF: Finding the OLE ClassKey of an Embedded Object](#)
[INF: Calling RegQueryValue\(\) to get an OLE Object Class Name](#)
[INF: Part 2, DDE in Visual Basic to Request Data from Access](#)
[INF: Removing alpha characters from an alpha/numeric field](#)
[INF: How to Wait for a Shelled Process to Finish](#)
[INF: Calculating a Credit Card Expiration Date](#)
[INF: How to Create a Parameter In\(\) Statement](#)
[PRB: Abnormal Action When Debugging Runtime Error](#)
[INF: Searching for Characters/Strings in Records](#)
[INF: Two Functions to Calculate Age in Months and Years](#)
[INF: How to Find the Difference Between 2 Time Values](#)
[PRB: Error When Using OpenQueryDef Method on Parameter Query](#)
[PRB: Macro on Subform Control Executed by Main Form](#)
[PRACC9305: GP Fault with Button Caption Text Longer Than 2048](#)
[INF: How to Specify a Custom Starting Page Number for a Report](#)
[INF: How to Call Functions Using a String Variable](#)
[INF: How to Pass Information to Microsoft Access Using DDE](#)
[PRACC9305: GP Fault When Using Erase Statement on String Array](#)
[PRACC9305: GP Fault When Allocating Huge Arrays](#)
[INF: How to Disable the Close Option on the System Menu](#)
[INF: How to Perform a Screen Dump of Your Forms](#)

[INF: Using SendKeys to Change Option Menus](#)
[PRB: Domain Parameter of Dom Funct Cannot Be SQL Expression](#)
[PRB: Cannot open multiple instances of the same form](#)
[PRB: Cannot Set Focus to Any Object Except MS Access Menu](#)
[INF: How to Make Forms Always Stay on Top of Other Windows](#)
[INF: How to Create a Custom Startup "Splash" Screen](#)
[INF: How to Get a List of Object Names in Microsoft Access](#)
[INF: Fill New Record with Data from Prev Record Automatically](#)
[INF: Debugging & Troubleshooting Tips for Attached SQL Tables](#)
[INF: Using Access Basic CopyFile\(\) Function to Copy Disk Files](#)
[INF: Function to Get Date of Monday Prior to Current Day](#)
[INF: How to Find the Last Day of the Month](#)
[INF: Reading, Storing, Writing Binary Large Objects \(BLOBS\)](#)
[INF: How to Get Record Position Number from Record Selector](#)
[INF: Custom Domain Function Similar to DCount](#)
[INF: Custom Domain Functions Similar to DFirst\(\), DLast](#)
[INF: How to Use the Status Bar Progress Meter](#)
[INF: Another Method of Quitting Windows](#)
[PRB: Invalid Use of Null or #Error in User Defined Function](#)

Modify

[INF: Deleting Duplicate Records from Two Tables](#)

[INF: Equivalent of Excel's Fill Down Functionality](#)

[INF: How to Programmatically Create a Table](#)

[PRACC9210: Transaction Limit Can Cause 'Out of Memory' Error](#)

[PRACC9307: "Table Does Not Exist" Error when Deleting Table](#)

[PRB: All Data 'Lost' When Doing Save As from Design View](#)

[PRB: Changed Field Names In Table Are Not Reflected In Query](#)

Sorting and Grouping

[INF: 'Continued...' Header for Groups on More than One Page](#)

[INF: Expressions to Count Yes/No and Other Responses](#)

[INF: How to Avoid Abandoned Group Headers in Reports](#)

[INF: Repeat Group Name at the Top of New Column or Page](#)

[PRACC9303: Screen Redraw Problems When Reordering Groups](#)

[PRACC9306: Run Time Partially Displays Sorting and Grouping](#)

[PRACC9308: Cannot Use First\(\) and Last\(\) Functions in a Report](#)

[PRB: Negative GroupOn Interval Reverses Sort Order](#)

NWIND/Sample Apps

[FastTips: Interoperability Questions & Answers](#)

[INF: "Form is read only" error](#)

[INF: "Form is read only" error](#)

[INF: 'Continued...' Header for Groups on More than One Page](#)

[INF: A Macro to Check for Duplicate Values in a Field](#)

[INF: Adding a New Record to a Combo Box with a Double-Click](#)

[INF: Attached Table Must Be Opened as Dynaset from Modules](#)

[INF: Base Combo Box on Parameter Query to Filter Values](#)

[INF: Changing the Relationship Between Two Existing Tables](#)

[INF: Changing the RowSource of a ListBox/ComboBox at Runtime](#)

[INF: ComboBox On Toolbar Do Not Use Window Background Color](#)

[INF: Commonly Asked Questions About Access Version 1.0](#)

[INF: Communicating with Microsoft Access using NetDDE](#)

[INF: Concatenating Fields in a Text box to Remove Blank Lines](#)

[INF: Contents: MS Access 1.0 Disks 1-4 \(1/14/93, 3.5-Inch\)](#)

[INF: Contents: MS Access 1.0 Disks 4-6 \(1/14/93, 3.5-Inch\)](#)

[INF: Controlling Design Mode at Runtime to Change Properties](#)

[INF: Correct Syntax for TransferDatabase to dBASE and Paradox](#)

[INF: Creating, Debugging, and Using an Access Library](#)

[INF: Creating Queries to Execute Cascading Updates](#)

[INF: Custom Setup May Show Zero or Negative Disk Space Needed](#)

[INF: Default Value Of A Combo Box Equal To The First Row](#)

[INF: Deleting Duplicate Records from Two Tables](#)

[INF: Determining the Existence of a Query in Access](#)

[INF: Displaying Totals for Each Row in Crosstab Queries](#)

[INF: Equivalent of Excel's Fill Down Functionality](#)

[INF: Example Has Wrong Number for Dlookup Ship Postal Code](#)

[INF: Expanded Size of the Companion files](#)

[INF: Export Btrieve Table Without Existing Xtrieve FILE.DDF](#)

[INF: Expressions to Count Yes/No and Other Responses](#)

[INF: Fill New Record with Data from Prev Record Automatically](#)

[INF: Finding the OLE ClassKey of an Embedded Object](#)

[INF: Formatting a Label for Each Axis of a Graph](#)

[INF: Four Ways to Move to a Record from a Combo Box Selection](#)

[INF: Getting the Data Types of Fields in a Table through DDE](#)

[INF: Graphing Current Sales Compared To Accumulated Sales](#)

[INF: How NWIND.MDB Employee Field Photographs Were Created](#)

[INF: How to Buffer SQL Strings Using DDE](#)

[INF: How to Check for Duplicate Values in Primary Key Fields](#)

[INF: How to Create a "Please Wait" Message](#)

[INF: How to Create a Dynamic Crosstab Report](#)

[INF: How to Create a Parameter In\(\) Statement](#)

[INF: How to Create a Query By Form](#)

[INF: How to Create a Running Sum on a Form](#)

[INF: How to Create Synchronized Combo Boxes](#)

[INF: How to Determine Whether a Menu Item Is Selected](#)

[INF: How to Disable the Close Option on the System Menu](#)
[INF: How to Execute Macro Actions Using DDE](#)
[INF: How to Export the Results of a Query](#)
[INF: How to Force a Cascading Delete](#)
[INF: How to Pass Information to Microsoft Access Using DDE](#)
[INF: How to Prevent Users from Adding New Records to a Form](#)
[INF: How to Print a Single Record from a Form in a Report](#)
[INF: How to Print Duplicate Mailing Labels for Each Record](#)
[INF: How to Print Odd or Even Pages of a Report](#)
[INF: How to Query SQL Server Tables Without Attaching](#)
[INF: How to Remove Relationships](#)
[INF: How to Remove the Menu Bar from a Form](#)
[INF: How to Simulate Changing the Record Source of a Form](#)
[INF: How to Specify a Custom Starting Page Number for a Report](#)
[INF: How to Synchronize Two Combo Boxes on a Form](#)
[INF: How to Update a Table Based on Values in a Second Table](#)
[INF: How to Use a Toggle Button to Hide and Display a Subform](#)
[INF: How to use Aggregate Functions in a Query Without Errors](#)
[INF: How to Use DDE to Fill a List Box](#)
[INF: How to Use Graphic Images As Wallpaper For Forms](#)
[INF: How to Use the Status Bar Progress Meter](#)
[INF: Install Process of the Companion Disk](#)
[INF: Join Type Mismatch Errors Caught at Runtime](#)
[INF: Macro to Enter Date Record Was Last Modified](#)
[INF: Microsoft Access 1.0 Disk Contents \(3.5 inch, 1.44 MB\)](#)
[INF: Microsoft Access 1.0 Disks 1-4 Contents \(5.25 Inch 1.2MB\)](#)
[INF: Microsoft Access and Scanners](#)
[INF: MS Access Does Not Support Parameter Queries as DDE Topic](#)
[INF: Northwind 'Sales Totals by Amount' Report Is Incorrect](#)
[INF: Nulls Allowed in Foreign Key with Referential Integrity](#)
[INF: Number of Times a Custom Function Executes in a Query](#)
[INF: OLE: New Graph Object Frames are Blank](#)
[INF: OpenDatabase and CloseDatabase Pseudo Actions](#)
[INF: Ordering Data in a Graph](#)
[INF: Part 1: DDE in Visual Basic to Request Data from Access](#)
[INF: Part 2, DDE in Visual Basic to Request Data from Access](#)
[INF: Passing Values from a Form to a Parameter Query](#)
[INF: Practice Using Object Analyzer Commands](#)
[INF: Print a Sequential Number for Each Record/Group on Report](#)
[INF: Programming with the Object Analyzer Engine and Utilities](#)
[INF: Query with Parameters to Evaluate Complex Criteria](#)
[INF: Reporting the Median Value of a Group of Records](#)
[INF: Requery Control Err Msg: There Is No Control Called...](#)
[INF: Returning Case Sensitive Match in Queries](#)
[INF: Round or Truncate Values to Desired Number of Decimals](#)
[INF: Row Fix-up in the NWIND.MDB Order Form](#)

[INF: Sending the Current Record to Word for Windows with DDE](#)
[INF: Set Focus to a Subform Control Using GoToControl Action](#)
[INF: Setting Form's Dynaset Property Puts Bookmark at EOF](#)
[INF: Showing All Records \(Including Null\) in a Parameter Query](#)
[INF: Shrinking Empty or Null OLE Object on Report](#)
[INF: Snapshot and Underlying Table Updated Until Snapshot Full](#)
[INF: Sorting/Grouping a Query Using a Portion of Field's Value](#)
[INF: Topics Supported by Access as a DDE Server](#)
[INF: Two Functions to Calculate Age in Months and Years](#)
[INF: Two Methods to Refer to a Field in a Previous Record](#)
[INF: Updating a Separate Table When a Value Changes on a Form](#)
[INF: Use Row Fix-up Technique to Look Up Info Automatically](#)
[INF: Using a Conditional Macro to Confirm Changes to a Field](#)
[INF: Using a Subquery to Simulate a SQL Sub-SELECT Statement](#)
[INF: Using a Subtract Query to Find Unmatched Records](#)
[INF: Using DDE to Communicate with a Runtime Application](#)
[INF: Using Microsoft Access as a DDE Server](#)
[INF: Using Microsoft Word to Edit and Spell Check Memo Field](#)
[INF: Using the .Parent Property of a Subform](#)
[INF: Using the DDE Item SQLText to Request the Text of a Query](#)
[INF: Where Condition Refers to Fields not Controls](#)

[PRACC1115: ApplyFilter Action Causes Form to be Read-Only](#)
[PRACC9210: Cannot Use ReportWizard Without Default Win Printer](#)
[PRACC9301: Wrong RowSourceType for Combo Box Causes GP Fault](#)
[PRACC9302: Syntax Error When "\(" is First Chr in Criteria Cell](#)
[PRACC9303: Error When Use SQL in Report RecordSource Property](#)
[PRACC9303: Error: Not Enough System Resources in Access](#)
[PRACC9303: Orders Form Doesn't Print/Preview SubTotal & Total](#)
[PRACC9303: Screen Redraw Problems When Reordering Groups](#)
[PRACC9304: Calculated Control on Subreport Displays Wrong Data](#)
[PRACC9305: GP Fault when Creating a Dynaset from Form Dynaset](#)
[PRACC9305: Hard-Coded and Passed Wildcards Different in Query](#)
[PRACC9305: List Box Controls on Forms Painted Incorrectly](#)
[PRACC9306: ERR: Undefined Function Referencing Column Property](#)
[PRACC9306: Exporting Nulls to Paradox Actually Exports a Space](#)
[PRACC9306: Hidden Subform Maximized If Main Form Is Maximized](#)
[PRACC9306: Import/Append Actions Return Errors for Each Record](#)
[PRACC9306: Run Time Partially Displays Sorting and Grouping](#)
[PRACC9306: Run-Time Icon Flashes Application Workspace Color](#)
[PRACC9307: "Table Does Not Exist" Error when Deleting Table](#)
[PRACC9307: GP Fault: Printing Large Report with Low Disk Space](#)
[PRACC9307: Invalid Characters Added to IN Clause Arguments](#)
[PRACC9308: Cannot Use First\(\) and Last\(\) Functions in a Report](#)
[PRACC9308: Error Previewing Subreports that Call Filter Macro](#)
[PRACC9308: GP Fault Referencing Column Property of ComboBox](#)

[PRACC9308: Open OLE Objects Not Saved When Record Changes](#)
[PRACC9308: Printer Fonts Cause Text to Shift Down and Right](#)
[PRACC980: DISTINCTROW Keyword Ignored in a Cartesian Product](#)

[PRB: GoToRecord Macro Doesn't Work on Subform](#)
[PRB: "#ERROR" Referencing Null Subreport Values in Main Report](#)
[PRB: 'Circular Reference Caused by Alias' Error Message](#)
[PRB: 'Fieldname' Is not an Index in This Table](#)
[PRB: <Expression> Not Part of Aggregate Function or Grouping](#)
[PRB: ADK Custom Setup Hangs During Installation](#)
[PRB: Alt+ Any Key Combination Does Not Work With Active Popup](#)
[PRB: Bitmap May Not Display Correctly](#)
[PRB: Can't Change Combo Box Value If ControlSource Is Counter](#)
[PRB: Can't Scroll Query Grid with ARROW Key or Mouse](#)
[PRB: Cannot open multiple instances of the same form](#)
[PRB: Changes in Subform Design do not Update in Main Form](#)
[PRB: Clicking Mouse in AccessWizard Background Changes Focus](#)
[PRB: Close Dynaset Based on Form Multiple Times](#)
[PRB: Combo Boxes Open When Window Sized to Vertical Minimum](#)
[PRB: Companion Disk install and Out of Disk Space](#)
[PRB: Control Menu Move Command Requires Keyboard Input](#)
[PRB: Current Record Not Immediately Updated with Replace All](#)
[PRB: Cursor Position Unchanged in Undo Paste](#)
[PRB: Database Window Loses its Focus](#)
[PRB: DDE App Name for Access Is Incorrect in Manual](#)
[PRB: Don't have Permission to Modify Secured Report](#)
[PRB: Echo Without Arguments Causes Error](#)
[PRB: Err Msg: Macro " Not Found](#)
[PRB: Error in Column Property Help Menu Topic](#)
[PRB: Error Message If Path Info for Attached Databases Changes](#)
[PRB: Error Message: 'Couldn't Find Input Table or Query'](#)
[PRB: Error Message: Couldn't Open File](#)
[PRB: Expression in Query Grid Doesn't Show All Text](#)
[PRB: Extra Unwanted Information Appears in List Box](#)
[PRB: Focus Doesn't Shift When Find Accelerators Are Depressed](#)
[PRB: Form Displays Data from Deleted Field in Query](#)
[PRB: Form Handle Not Released When Closed with Basic Code](#)
[PRB: Form May Lose the Focus When SHIFT+F8 Is Pressed](#)
[PRB: Form Objects Not Placed on Grid by Wizard](#)
[PRB: Form Opened in Form View Is Completely Blank](#)
[PRB: Form Opens Blank Except for Header](#)
[PRB: Full Path Causes Endless Loop in StartApp\(\) Example](#)
[PRB: GoToControl to Control with Focus in AfterUpdate Ignored](#)
[PRB: Incorrect Results Received with DSUM Function](#)
[PRB: Insert Object Dialog Box Has No 'All Files \(*.*\)' Option](#)
[PRB: Installing Companion Files into nested subdirectories](#)

[PRB: Join Is Broken by Value\(s\) in Fields '\[Field Name\]'](#)
[PRB: Label Placed in, Not to Left of, Command Button on Form](#)
[PRB: Layout Changes to Subform in Datasheet View Not Saved](#)
[PRB: Leading Space in Property Sheet Entry May Cause Problems](#)
[PRB: List Box Not Dimmed When Disabled](#)
[PRB: Macro on Subform Control Executed by Main Form](#)
[PRB: Minimizing Popup/Modal Form Loses Application's Focus](#)
[PRB: Negative GroupOn Interval Reverses Sort Order](#)
[PRB: No AutoRepeat with GoToRecord/Next and DoMenuItem](#)
[PRB: No Field List When Form Based on SQL Statement](#)
[PRB: No Implicit Parameters on CrossTab Queries](#)
[PRB: Not Enough Stack Memory When Opening Form](#)
[PRB: NWIND Order Form Updates Order Amount Field with Zero](#)
[PRB: NWIND Order Form Updates Order Amount Field with Zero](#)
[PRB: NWIND.MDB Query with Criteria Displays Data in Odd Order](#)
[PRB: Opening a Database Read Only Forces Others to Read Only](#)
[PRB: Opening a Database Using a Sendkeys Macro](#)
[PRB: ORDER BYs Must Be Output Columns in DISTINCT Queries](#)
[PRB: Order Entry Ship To Address Contains City Info](#)
[PRB: Out of Memory Error Message when Exporting](#)
[PRB: Out of Stack Memory Error Message with SendKeys Macro](#)
[PRB: Paste Link to MS Excel Displays First Field Name Only](#)
[PRB: Properties Do not Reset Between Crosstab and Make Table](#)
[PRB: Query Window Does not Always Scroll to Record](#)
[PRB: Referential Integrity Error Message with Shared Databases](#)
[PRB: Report Wizard Defaults To 8 1/2 by 11 Inch Page](#)
[PRB: Seek Method Is Faster Than Find Method](#)
[PRB: SetValue Does not Trigger Form Events](#)
[PRB: Trailing CR/LFs Stripped From Text Box When Retrieved](#)
[PRB: Trailing CR/LFs Stripped from Text Box when Retrieved](#)
[PRB: Two Accelerator Keys in Custom Menu Show Only First Key](#)
[PRB: Unable to Add All Fields in Graph FormWizard](#)
[PRB: Unable to Paste Information into Property Sheet](#)
[PRB: Unable to Use Hexadecimal or Octal Values in Properties](#)
[PRB: Use "MSACCESS," Not .MDB Filename, in DDE Conversation](#)
[PRB: Using CTRL+SHIFT+HOME to Move to Main Form](#)
[PRB: Valid Sum Function Returns '#Name?' in Calculated Control](#)
[PRB: ValidationRule Parses Function Name w/o Parens to String](#)

Events (Calling Macro/Mods)

[INF: Adding a New Record to a Combo Box with a Double-Click](#)
[INF: Choosing Toolbar Design Button Causes Unexpected Behavior](#)
[INF: Custom Menus Cannot Be Implemented with Pop-up Forms](#)
[INF: Delete Command on Form Open if Counter has Focus](#)
[INF: Double-click Also Executes OnPush Twice](#)
[INF: Form Controls to Increment and Decrement a Date](#)
[INF: Form OnOpen Event May Not Set Values of Unbound Controls](#)
[INF: Four Ways to Move to a Record from a Combo Box Selection](#)
[INF: How Not to Display the Form Name in the Caption Bar](#)
[INF: How to Create an Undo Record Button on a Form](#)
[INF: How to Remove the Menu Bar from a Form](#)
[INF: How to Use a Toggle Button to Hide and Display a Subform](#)
[INF: Misleading Documentation for Cancel Property](#)
[INF: OnCurrent Is Not Triggered on Unbound Forms](#)
[INF: Updating a Separate Table When a Value Changes on a Form](#)

[PRACC9301: Access GP Faults in Datasheet View of a Form](#)
[PRACC9303: Form Record Selector Buttons Disabled](#)
[PRACC9303: GPF Hiding Control in Form Footer in Datasheet View](#)

[PRB: GoToRecord Macro Doesn't Work on Subform](#)
[PRB: Can't Refer to Record in Function Called by OnClose Event](#)
[PRB: Close Action from AB Does Not Trigger OnClose Errors](#)
[PRB: Focus Returns to Control After OnDbClick](#)
[PRB: Macro on Subform Control Executed by Main Form](#)
[PRB: NWIND Order Form Updates Order Amount Field with Zero](#)
[PRB: OnDbClick Macro Requires CancelEvent to Change Focus](#)
[PRB: SetValue Does not Trigger Form Events](#)

PRB: Query Results in an "Overflow" Error

Article Number: Q104488
CREATED: 19-SEP-1993
MODIFIED: 19-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
 - Microsoft Access Distribution Kit version 1.1
-

SYMPTOMS

A query that uses a criteria may return an "Overflow" error. When the criteria is removed from the QBE grid, the error does not occur and the query executes correctly.

STATUS

This is by design.

RESOLUTION

A possible solution is to change the way the data is represented.

For example, an expression resulting in a percentage represented by a decimal value such as 0.95 for 95% and containing a criteria such as >0.1 could be changed by multiplying the expression by 100 and changing the criteria to >10.

MORE INFORMATION

You will get an overflow error message if you:

1. Divide a very large number by a very small fractional number relative to each other so that the result is larger than the data type can hold.
2. Multiply two numbers that result in a number larger than the data type can hold.

An example would be as follows:

```
Dim MyVal As Integer  
MyVal = 32767 / 0.1
```

This would result in setting MyVal to 327,670 which exceeds the maximum integer size of 32,767.

Using certain data, the following expression and criteria may cause the "Overflow" error:

```
Expression:"Test: ([Quantity] * [Unit Cost])/SumOfAmount"  
Criteria:">.1"
```

If the criteria is removed, the query will run. In this case, to apply a criteria and remove the error, change the expression and criteria to the following:

```
Expression:"Test: (([Quantity] * [Unit Cost])/SumOfAmount)*100"  
Criteria:">10"
```

Both methods are looking for cases over 10% but the second [method](#) does not generate the "Overflow" error.

[References](#)

INF: MakeTable Query Does not Copy Field Descriptions

Article Number: Q104487
CREATED: 19-SEP-1993
MODIFIED: 19-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

A MakeTable query will not include in the new table any field descriptions which may be in the table on which the query is based.

Edit Copy and Paste can be used to ensure that the field descriptions are copied, either with copy "Structure Only" or with "Structure and Data". An Append query can be used to fill the table as required.

MORE INFORMATION

1. In NorthWind, create a new query and add the Categories table.

Query: Copy Category Table

Type: MakeTable

Field: Category ID
Field: Category Name
Field: Description
Field: Picture

2. For the new table name enter "Categories (MkTbl)" and run the query.
3. View the "Categories" table in design view and note the field descriptions. View the "Categories (MkTbl)" table in design view and note the absence of any descriptions.

A possible work around, depending on the application, is to use the clipboard to copy the table.

1. Highlight the table (e.g. Categories) and choose the Copy command from the Edit menu..
2. Choose the Paste command from the Edit menu and enter the new table name (e.g. Categories (copy)).
3. Copy either Structure Only or Structure and Data - both will copy the field descriptions.
4. View the "Categories (copy)" table in design view and notice that the field descriptions were copied. An Append query can now be used to move the data from the original table to the new table.

[References](#)

PRACC9309: SQORA.DLL Does Not Allow Lengthy SQL

Article Number: Q104452
CREATED: 16-SEP-1993
MODIFIED: 16-SEP-1993
VERSION(S): 1.10

The information in this article applies to:

- Microsoft Access version 1.1
-

SYMPTOMS

When executing a query against an Oracle table, or updating a record in an Oracle table, the following error message is displayed

Statement was longer then allowable maximum 2000+ chars

if the table and field names are very long or the query is very complex.

CAUSE

This is a problem with SQORA.DLL, the Oracle ODBC Driver.

RESOLUTION

Obtain and install the updated driver, or use queries to do updates, rather than updating records while the Oracle table is in Datasheet view. The query should yield only the columns to be updated.

For complex queries, reduce the number of tables or joins in the query and reduce the number of fields used or shown in the query.

These workarounds reduce the length of the SQL statements.

STATUS

Microsoft has confirmed this to be a problem in Oracle ODBC driver shipped Microsoft Access version 1.1 and Visual Basic for Windows v3.0 - Professional Edition. An updated driver that corrects this specific problem is available from Microsoft.

MORE INFORMATION

How to obtain the updated driver:

ORA110.EXE, a self-extracting zip file, can be downloaded from Library 11, ODBC Connectivity, in the MSACCESS forum on CompuServe, or from MSDL.

If the user is not able to download the file, PSS can send them the information on a disk, which will also include an updated Btrieve driver.

Note: This driver is only for use by registered owners of Microsoft Access v1.1 and Visual Basic for Windows v3.0 - Professional Edition. By installing these files, you are indicating that you own one of these two products.

Disk from PSS:

There is just one README.TXT file on the disk that documents the changes to the Btrieve and Oracle drivers and the installation procedures for each.

The following files are on the disk:

README.TXT
BTRV110.DLL
ORACLE.TXT
SQORA.DL_
SQORASTP.DL_
ODBC.INF
SETUP.EXE

Note: The SETUP.EXE is called by the ODBC control panel facility. It will not run by itself.

The following two files will actually be updated:

	Old	New
SQORA.DLL	version: 1.00.2816	version: 1.00.3112
	size: 143,600 bytes	size: 144,096 bytes
	date: 4/16/93	date: 7/12/93
SQORASTP.DLL	version: 1.00.2403	version: 1.00.3106
	size: 9,328 bytes	size: 9,632 bytes
	date: 5/7/93	date: 7/6/93

[References](#)

INF: Database Structure Questions & Answers

Article Number: Q104392
CREATED: 14-SEP-1993
MODIFIED: 14-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

=====
Microsoft(R) Product Support Services Application Note (Text File)
WX0635: DATABASE STRUCTURE QUESTIONS & ANSWERS
=====

Revision Date: 9/93
No Disk Included

The following information applies to Microsoft Access(R), versions 1.0, and 1.1.

| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |
| PURPOSE. The user assumes the entire risk as to the accuracy and |
| the use of this Application Note. This Application Note may be |
| copied and distributed subject to the following conditions: 1) All |
| text must be copied without modification and all pages must be |
| included; 2) If software is included, all files on the disk(s) |
| must be copied without modification (the MS-DOS(R) utility |
| diskcopy is appropriate for this purpose); 3) All components of |
| this Application Note must be distributed together; and 4) This |
| Application Note may not be distributed for profit. |
| |
| Copyright (C) 1992-1993 Microsoft Corporation. All Rights Reserved |
| Microsoft, FoxPro, Microsoft Access, and MS-DOS are registered |
| trademarks |
| and Windows is a trademark of Microsoft Corporation. |
| Paradox is a registered trademark of Ansa Software, a Borland |
| company. |
| dBASE III PLUS and dBASE IV are registered trademarks of |
| Borland International, Inc. |
| 1-2-3 and Lotus are registered trademarks of Lotus Development |
| Corporation. |
| Btrieve is a registered trademark of Novell, Inc. |
| Oracle is a registered trademark of Oracle Corporation. |
Sybase is a registered trademark of Sybase, Inc.

1. Q. What foreign data formats does Microsoft Access support?

- A. Microsoft Access can [link](#) to Btrieve(R), dBASE III PLUS(R), dBASE IV(R), FoxPro(R) versions 2.x, and Paradox(R) versions

3.x data, including indexes.

Microsoft Access can import data from Btrieve, dBASE III PLUS, dBASE IV, FoxPro versions 2.x, Lotus(R) 1-2-3(R) versions 2.x and 3.x, Lotus 1-2-3/W, Microsoft Excel versions 2.x and later for Windows, Paradox 3.x, and fixed-length and delimited ASCII text.

Microsoft Access can connect to Microsoft [SQL](#) Server, Sybase(R) SQL Server, and Oracle(R) RDBMS Server using the Open Database Connectivity ([ODBC](#)) specification. These ODBC drivers are the only drivers currently certified for Microsoft Access version 1.1.

2. Q. What are primary and foreign keys in a relational [database](#)?

A. Primary keys uniquely identify a specific [row](#) in a [table](#). They are usually composed of a single [field](#) or [column](#) whose data is unique to each row of information (for example, a customer identification number, employee social [security](#) number, or order identification number). Primary keys also can be composed of more than one field. For example, the fields Company Identification and Contact Name could be combined to [form](#) a [primary key](#) for a table that contains information about companies that have more than one contact.

Foreign keys identify rows in one table that match the primary key of another table. For example, if the field Employee ID is the primary key in the Employee table, in the Orders table, Employee ID is the [foreign key](#). The [relationship](#) between primary and foreign keys enables you to access related data in multiple tables.

3. Q. Which FieldSize setting should I select for the Number data type in order to use a Counter field of another table as the foreign key in the current table?

A. To use a Counter field of another table as the foreign key in the current table, select the Long Integer setting.

4. Q. How do you change the starting value of a Counter column to a number other than 1?

A. To set the starting value of a Counter column to a number other than 1, use the following four steps:

1. Create an additional single-column temporary table. Give the Number column the same name as the Counter column in the original table.

2. Create a new row in the temporary table by inserting a value in the Number column. This value should be 1 less than the desired starting value for the original table.

3. Create an [append query](#) to append the single row from the temporary table to the original table.

4. Delete the temporary table. You can then delete the dummy

row from the original table.

NOTE: Do not compact the database before you add the first row to the original table. If you do this, the counter value resets to begin at 1.

For additional information and an example of how to set the initial value of a counter, please obtain Technical Note #94821. You can order this technical note by selecting the FastTips Technical Library option from the FastTips Main Menu.

5. Q. Is data in tables stored in a sorted order? How can I view my data in sorted order?
 - A. Data in tables is not stored in sorted order. Data is stored in the order in which it was entered. To view data in sorted order, create a [query](#), or a form based on a query, that uses the Sort field on the query grid. By default, if the table includes a primary key, the [Datasheet view](#) of the table will display the data sorted by the primary key. To view the data sorted by something other than the primary key alone, use a query, as previously described.
6. Q. How can I delete the primary key from a table?
 - A. You can delete the primary key for a table by opening the Table Properties [dialog box](#) and deleting the entry for the Primary Key field. Please note that the primary key cannot be deleted from a table that is the primary table in a relationship.
7. Q. How can I create computed columns in tables?
 - A. You can create computed columns or expressions with queries. Within the query, create a column that is defined as an [expression](#). In general, it is helpful to think of queries as virtual tables; you can use a query wherever you can use a table. If you're familiar with SQL terminology, this is very similar to creating a view. Unlike most implementations of views, however, Microsoft Access views can be updated even if they involve joins from different data sources, such as Paradox, Btrieve, or separate Microsoft Access data sources.
8. Q. Two of my tables have [Counter data type](#) columns in common. Why can't I define a relationship between the two tables based on these columns?
 - A. You can define only one-to-one relationships between [Counter data type](#) columns. If the linking column on the "one" side is a counter, define the linking column on the "many" side as a long integer.

For more information on table relationships, see the "Microsoft Access User's Guide," Chapter 1, "Designing a Database," pages 13-18.

9. Q. How many databases can I open at a time in Microsoft Access?

- A. Through the user interface menus in Microsoft Access, you can open only one database at a time; therefore, you can view only one database window in each session of Microsoft Access. Through [Access Basic](#), however, you can use the `OpenDatabase()` [function](#) to open several databases at once. The number of databases you can have open simultaneously through Access Basic is limited only by your computer's memory.

Also, by choosing Attach Table from the File menu, you can attach to as many databases as your computer's memory allows.

10. Q. Can Microsoft Access version 1.0 use databases created in version 1.1?

- A. Microsoft Access 1.0 cannot read from or write to databases created in Microsoft Access 1.1. However, Microsoft Access 1.1 can read from and write to databases created in version 1.0 and still allow version 1.0 to read from and write to the same database without conversion.

If you want to upgrade your version 1.0 database to version 1.1 format, use the following four steps:

1. From the version 1.1 File menu, choose Compact Database. The Database To Compact From dialog box is displayed.
2. In the File Name box, select the database that you want to convert. In the List Files Of Type box, select Databases (*.mdb).
3. Choose OK. The Database To Compact To dialog box is displayed.
4. In the List Files Of Type box, select Access V1.1 (*.MDB).

NOTE: If you want to convert your version 1.1 database back to version 1.0 format, select Access V1.0 (*.MDB) instead.

[References](#)

KBCategory:

KBSubcategory:

INF: Setup Questions & Answers

Article Number: Q104391
CREATED: 14-SEP-1993
MODIFIED: 14-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

=====
Microsoft(R) Product Support Services Application Note (Text File)
WX0812: SETUP QUESTIONS & ANSWERS
=====

Revision Date: 9/93
No Disk Included

The following information applies to Microsoft Access(R), versions 1.0, and 1.1.

| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |
| PURPOSE. The user assumes the entire risk as to the accuracy and |
| the use of this Application Note. This Application Note may be |
| copied and distributed subject to the following conditions: 1) All |
| text must be copied without modification and all pages must be |
| included; 2) If software is included, all files on the disk(s) |
| must be copied without modification (the MS-DOS(R) utility |
| diskcopy is appropriate for this purpose); 3) All components of |
| this Application Note must be distributed together; and 4) This |
| Application Note may not be distributed for profit. |
| |
| Copyright (C) 1993 Microsoft Corporation. All Rights Reserved. |
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |
| and Windows is a trademark of Microsoft Corporation. |
Quicken is a registered trademark of Intuit.

1. Q. Why does Microsoft recommend that I remove SHARE.EXE from my AUTOEXEC.BAT file if I run Microsoft Windows for Workgroups?

A. We recommend that you remove SHARE.EXE from your AUTOEXEC.BAT file only if you run Microsoft Windows for Workgroups in 386 enhanced mode. When Windows for Workgroups is run in 386 enhanced mode loads and uses a file-sharing program called VSHARE.386, which replaces SHARE.EXE. There is an entry for VSHARE.386 in the SYSTEM.INI file.

If you plan to run Windows for Workgroups exclusively in 386 enhanced mode and not run other applications that require SHARE.EXE, you can save approximately 5 kilobytes (K) of

conventional memory by not loading VSHARE.386 at all. If you have already loaded SHARE.EXE, delete it from your AUTOEXEC.BAT file and restart your computer.

You are likely to find SHARE.EXE in your AUTOEXEC.BAT file because SETUP.EXE automatically inserts the following MS-DOS command in your AUTOEXEC.BAT file when you install Microsoft Access

```
<directory>\share.exe /l:500
```

where <directory> is the name of your MS-DOS directory.

If you load VSHARE.386 and SHARE.EXE is already present, VSHARE.386 temporarily disables it. VSHARE.386 takes over file-sharing tasks until you exit Windows for Workgroups. VSHARE.386 then transfers the tasks back to SHARE.EXE, so file sharing is available to your MS-DOS-based applications.

Microsoft Access runs correctly with either SHARE.EXE or VSHARE.386, but SHARE.EXE limits the number of available locks to the number you specified when you loaded the file originally. The /L parameter specifies the number of locks; the default is 20. VSHARE.386, on the other hand, dynamically allocates the number of locks available, based on demand. The number of locks available is especially important if you are running Windows for Workgroups in a client-server environment.

For additional information on the SHARE.EXE file, please obtain Technical Note #95047. You can order this technical note by selecting the FastTips Technical Library option from the FastTips Main Menu.

2. Q. Why do I get the error message "Outdated 'COMMDLG.DLL' found. Please reinstall Microsoft Access" when I install Microsoft Access?

A. This error can occur for a number of reasons. To isolate the cause, check the following:

NOTE: This example uses the COMMDLG.DLL file. The same problem can occur with other .DLL files. These steps can be used to resolve any problems with outdated .DLL files.

- Make sure that you have no more than one COMMDLG.DLL file on your hard disk. If you have several earlier versions of this file, either delete or rename the earlier COMMDLG.DLL files and keep

only

the latest one.

If you are running MS-DOS version 5.0 or later, use the following two steps to search for all occurrences of the COMMDLG.DLL file:

1. Quit Windows.

2. At the MS-DOS prompt, type the following command for all

logical drives

```
dir <drive>:\commdlg.dll /s
```

where <drive> is the drive on which your Windows directory is located, the drive on which you installed Microsoft Access, and each drive located in the PATH statement in your AUTOEXEC.BAT file.

For versions of MS-DOS earlier than 5.0, you must either go into each directory and execute the above DIR command without the /S parameter, or use a utility to search for all occurrences of the COMMDLG.DLL file.

- Make sure that the COMMDLG.DLL file is located in your Windows SYSTEM subdirectory only.
- Make sure that the COMMDLG.DLL file is up to date. The file should have one of the three dates below, or later:

03/10/92	(for Microsoft Windows 3.1)
10/01/92	(for Microsoft Windows for Workgroups)
10/25/92	(for Microsoft Access)

- Make sure that you are not using a third-party, incompatible COMMDLG.DLL file.
- Make sure that no other applications are running with Windows before you run Microsoft Access Setup.

You cannot update the COMMDLG.DLL file during installation if another application is currently using the file. To ensure that all other Windows-based applications are closed, use the following five steps:

1. Create a temporary Program Manager group.
2. Move all the icons from the Startup group into the temporary group.
3. Back up your WIN.INI file. In the original copy, delete all the items from both the run= and the load= lines to read:

```
[windows]
run=
load=
```

4. Restart Windows.
5. After installing Microsoft Access, move the icons from the temporary group back into the Startup group and restore the WIN.INI file from your backup. Quit and restart Windows.

If you follow all the previous instructions and you still get

an "Outdated DLL" error message, your copy of the COMMDLG.DLL file may be corrupt or damaged. To solve this problem, reinstall a new copy of this file from your Microsoft Access package using the following three steps:

1. Copy the COMMDLG.DL_ file from Disk 1 of your Microsoft Access package to a directory on your hard disk.

NOTE: Be sure to copy the COMMDLG.DL_ file, not the COMMDLG.DL\$ file.

2. Copy the EXPAND.EXE file from the appropriate Windows setup disk to the same directory you used in step 1.
3. Use EXPAND.EXE to unpack the COMMDLG.DL_ file and place it in your Windows SYSTEM subdirectory. The [syntax](#) to complete this action is as follows

```
<drive>:\<directory>\expand commdl_g.dl_  
<destination>\commdl_g.dll
```

where <drive> and <directory> correspond to the location of EXPAND.EXE and <destination> is the drive and location of your Windows SYSTEM subdirectory.

For additional information about outdated DLLs, please obtain Technical Notes #93694 and #90863. You can order these technical notes by selecting the FastTips Technical Library option from the Fast Tips Main Menu.

3. Q. When I try to install Microsoft Access, Setup fails. What should I do?

- A. Setup problems are often caused by terminate-and-stay-resident (TSR) programs that are loaded at the MS-DOS level and/or at the Microsoft Windows level.

To determine if TSRs are causing Setup to fail, remove all unnecessary TSRs and device drivers from the CONFIG.SYS and the AUTOEXEC.BAT files, remove all icons from the Startup group in Program Manager, and delete all entries under the load= and run= lines in your WIN.INI file.

NOTE: Be sure to back up your CONFIG.SYS, AUTOEXEC.BAT, and WIN.INI files before modifying them.

The [table](#) below describes some TSRs that are known to cause problems with the Microsoft Access Setup program:

Filename	Description
APPEND.EXE	An MS-DOS TSR loaded in the AUTOEXEC.BAT file
SUBST.EXE	An MS-DOS TSR loaded in the AUTOEXEC.BAT file
BILLMNRD.EXE	An automated bill reminder loaded with Quicken(R) that can reside on the load= line of the

WIN.INI file

For additional troubleshooting steps to resolve Setup problems, please obtain Technical Note #96109. You can order this technical note by selecting the FastTips Technical Library option from the FastTips Main Menu.

4. Q. A directory called MS-SETUP.T is left on my hard disk after I install Microsoft Access. Can I safely delete this directory?
- A. Yes. After Microsoft Access installation is complete, it is okay to delete the MS-SETUP.T directory and all the files in it.

This directory is not deleted automatically because of a known problem that occurs in Microsoft Access Setup when you choose the Reboot Your System After Completing Setup option. If you choose the Return To Windows After Setup option instead, Microsoft Access automatically deletes the MS-SETUP.T directory and its contents.

5. Q. Why do I get a general protection (GP) fault when I run Setup?
- A. Running the Microsoft Access version 1.1 Setup program in Windows standard mode results in a GP fault. To run Setup correctly, run it in Windows 386 enhanced mode instead. To start Windows in 386 enhanced mode, type the following at the MS-DOS prompt:

```
win /3
```

6. Q. The AccessWizards are unavailable after I upgrade to Microsoft Access version 1.1. How do I get the AccessWizards to work?
- A. If the AccessWizards button is unavailable when you create a new [form](#) or [report](#), it is likely that the MSACCESS.INI file was not properly updated. To solve this problem, use the following three steps:
1. Rename your current MSACCESS.INI file to a different name, such as MSACCESS.BAK. The MSACCESS.INI file is located in your Windows program directory.
 2. Back up any [database](#) (.MDB) files that you have created.
 3. Run the Microsoft Access Setup program again.

The MSACCESS.INI file should now be correctly updated.

[References](#)

KBCategory:

KBSubcategory:

INF: Querying Questions & Answers

Article Number: Q104390
CREATED: 14-SEP-1993
MODIFIED: 14-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

=====
Microsoft(R) Product Support Services Application Note (Text File)
WX0639: QUERYING QUESTIONS & ANSWERS
=====

9/93
No Disk Included

The following information applies to Microsoft Access(R), versions 1.0, and 1.1.

| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |
| PURPOSE. The user assumes the entire risk as to the accuracy and |
| the use of this Application Note. This Application Note may be |
| copied and distributed subject to the following conditions: 1) All |
| text must be copied without modification and all pages must be |
| included; 2) If software is included, all files on the disk(s) |
| must be copied without modification (the MS-DOS(R) utility |
| diskcopy is appropriate for this purpose); 3) All components of |
| this Application Note must be distributed together; and 4) This |
| Application Note may not be distributed for profit. |
| |
| Copyright (C) 1992-1993 Microsoft Corporation. All Rights Reserved |
| Microsoft, FoxPro, Microsoft Access, and MS-DOS are registered |
| trademarks |
| and Windows is a trademark of Microsoft Corporation. |
| Paradox is a registered trademark of Ansa Software, a Borland |
| company |
| dBASE III PLUS and dBASE IV are registered trademarks of Borland |
| International, Inc. |
| Btrieve is a registered trademark of Novell, Inc. |
Sybase is a registered trademark of Sybase, Inc.

1. Q. How can I export the results of a [query](#)?

- A. To export the results of a query, you can either create a [table](#) that contains the results or create a [macro](#) to export the query itself.

METHOD 1: To create a table that contains the results of the

query, change the [select query](#) to a [make-table query](#) by opening it in [Design view](#) and choosing Make Table from the Query menu. The resulting table will contain the data to be exported. Then choose Export from the File menu to export the data in the format you want. For more information, please refer to Chapter 4 of the "Microsoft Access User's Guide."

METHOD 2: To create a macro to export the query, use the TransferText macro action to export the data in text file format or use the TransferSpreadsheet macro action to export the data in spreadsheet format. Using the name of a query in the TableName parameter of these macro actions allows you to export the query in either file format.

NOTE: This feature is undocumented. To export a parameter query, you must use Method 1. For specific instructions on using the TransferText or TransferSpreadsheet macro actions, search for "TransferText" or "TransferSpreadsheet" using the Help menu.

2. Q. Can I [join](#) tables from different databases into one query?
 - A. Yes, you can join tables from different databases into one query by choosing Attach Table from the File menu. Using these remotely-attached tables, you can generate queries from any supported data source (for example, Btrieve(R), dBASE III PLUS(R), dBASE IV(R), FoxPro(R), or Paradox(R)).
3. Q. Is data in tables stored in a sorted order? How can I view my data in sorted order?
 - A. Data in tables is not stored in sorted order. Data is stored in the order in which it was entered. To view data in sorted order, create either a query or a [form](#) based on a query that uses the Sort [field](#) on the query grid. By default, if the table includes a [primary key](#), the [Datasheet view](#) of the table will display the data sorted by the primary key. To view the data sorted by something other than the primary key alone, use a query, as previously described.
4. Q. How can I create computed columns in tables?
 - A. You can create computed columns or expressions with queries. Within the query, create a [column](#) that is defined as an [expression](#). In general, it is helpful to think of queries as virtual tables; you can use a query wherever you can use a table. If you're familiar with [SQL](#) terminology, this is very similar to creating a view. Unlike most implementations of views, however, Microsoft Access views can be updated even if they involve joins from different data sources, such as Paradox, Btrieve, or separate Microsoft Access data sources.
5. Q. What is the difference between the keywords DISTINCT and DISTINCTROW?
 - A. DISTINCT is part of the SQL standard and causes a query to return unique data, rather than unique records. For example,

even if there are 10 customers named Jones, the query "SELECT DISTINCT Name FROM Customer" returns only one [row](#) containing Jones. With Microsoft Access queries, you specify DISTINCT by choosing Query Properties from the View menu and then selecting the Unique Values Only [check box](#).

DISTINCTROW is unique to Microsoft Access and is not part of the SQL standard. It causes a query to return unique records, rather than unique data values. For example, if there are 10 customers named Jones, the query "SELECT DISTINCTROW Name FROM Customer" returns all 10 rows containing Jones.

The main reason for adding the DISTINCTROW keyword to Microsoft Access SQL is to support semi-joins that can be updated (for example, one-to-many joins in which the output columns all come from the one-sided table). DISTINCTROW is specified by default in Microsoft Access queries and is ignored in queries for which it has no effect.

6. Q. Why do queries change the order of my columns?
- A. When you close a query, Microsoft Access moves the sorted fields to the leftmost columns in the Query-By-Example (QBE) grid. For example, if you open an existing query in Design view, revise it, and save your changes, Microsoft Access displays the sorted fields to the left of the [datasheet](#). You can rearrange the fields if necessary.
7. Q. I am sending a query to Microsoft SQL Server or Sybase(R) SQL Server through Microsoft Access, but the query runs very slowly. Why does this occur, and how can I increase the query speed?
- A. The query speed most often decreases when you send implicit, instead of explicit, parameters to the remote server.

Query parameters can be either implicit or explicit. A parameter entry made only in the Query-By-Example (QBE) grid is called an implicit parameter. A parameter entry made in both the QBE grid and the Query Parameters [dialog box](#) is called an explicit parameter.

When you use implicit parameters in your query, Microsoft Access does not send the query to the remote server, because Microsoft Access cannot verify what implicit conversions the remote server provides. As a result, Microsoft Access processes the query locally, which can noticeably reduce the query speed. However, when you use explicit parameters in your query, Microsoft Access knows the data types of the parameters. Subsequently, Microsoft Access sends the query to the remote server for processing, which speeds up the query.

Note also that Microsoft Access automatically treats unrecognized or misspelled names and expressions as implicit parameters, rather than as errors. Microsoft Access then makes a "best guess" about the implicit parameter's [data type](#). If Microsoft Access guesses the wrong data type, a query can

return unexpected results or values, either when you enter a parameter value or when you execute queries on a remote server.

For additional optimization tips for attached SQL Server tables, please obtain Technical Note #99321. You can order this technical note by selecting the FastTips Technical Library option from the Fast Tips Main Menu.

[References](#)

KBCategory:

KBSubcategory:

INF: Access Basic and Macros Questions & Answers

Article Number: Q104389
CREATED: 14-SEP-1993
MODIFIED: 14-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

=====
Microsoft(R) Product Support Services Application Note (Text File)
WX0636: ACCESS(R) BASIC AND MACROS QUESTIONS & ANSWERS
=====

Revision Date: 9/93
No Disk Included

The following information applies to Microsoft Access, versions 1.0,
and 1.1.

| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |
| PURPOSE. The user assumes the entire risk as to the accuracy and |
| the use of this Application Note. This Application Note may be |
| copied and distributed subject to the following conditions: 1) All |
| text must be copied without modification and all pages must be |
| included; 2) If software is included, all files on the disk(s) |
| must be copied without modification (the MS-DOS(R) utility |
| diskcopy is appropriate for this purpose); 3) All components of |
| this Application Note must be distributed together; and 4) This |
| Application Note may not be distributed for profit. |
| |
| Copyright (C) 1992-1993 Microsoft Corporation. All Rights Reserved |
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |
and Windows is a trademark of Microsoft Corporation.

1. Q. How can I run macros from [Access Basic](#)?

A. The following are three suggested ways to run macros from
Access Basic:

- Create the [macro](#) using the macro editor and then call it from
your Access Basic procedure. For example, if you create a macro
called Macro1, you can call the macro from a procedure with the
following command:

```
DoCmd RunMacro "macro1"
```

- Call the macro from the [Immediate window](#) using the same [syntax](#)
as you would in Access Basic. Again, the syntax is as follows:

DoCmd RunMacro "macro1"

- Use DoCmd to run a macro action, rather than creating a macro specifically for that action. The basic format of DoCmd is as follows:

DoCmd <Action> <Argument1>, <Argument2>

For more information on DoCmd, RunMacro, and macro actions, please refer to the "User's Guide" and "An Introduction to Programming;" or, search Help on the keywords DOCMD and Expressions in Macros and Actions.

2. Q. When should I use the [Variant data type](#)?

- A. Use the Variant data type if you are unsure what type of data you are working with. When you use the Variant data type, Microsoft Access evaluates the data and then handles it appropriately. Using variants avoids the step of assigning data types to variables and manually converting data types in your programs. Further, when you store [table](#) data in variables, using variants eliminates complications with NULL data values.

3. Q. How do I comment out a macro action?

- A. With the macro in [Design view](#), choose Conditions from the View menu. This adds the Condition [column](#) to the macro design. Type "false" (without the quotation marks) on the condition line for the macro action you want to comment out.

4. Q. How do I bypass the AutoExec startup macro when I open a [database](#)?

- A. Hold down the SHIFT key when you open the database to prevent Microsoft Access from executing that database's AutoExec macro.

5. Q. How do I include a [variable](#) or [form](#) reference in the WHERE clause of my [SQL](#) statement?

- A. The syntax for including a variable or form reference in the [WHERE clause](#) of an SQL statement depends on the data type of the variable. Numeric variables do not require delimiters, [string](#) variables should be enclosed in single quotation marks, and date variables require the number sign (#). You must concatenate the variable and use the appropriate delimiter, if required, as shown in the examples below:

For a numeric variable, use the following syntax:

```
myq.sql = "select * from orders where [employee id]=" &  
mynum & ";"
```

For a string variable, use the following syntax:

```
myq.sql = "select * from orders where [customer id]='" &  
mytext & "';"
```

For a date variable, use the following syntax:

```
myq.sql = "select * from orders where [order date]=#" &  
mydate & "#;"
```

For additional information on concatenating variables, please obtain Technical Note #96576. You can order this technical note by selecting the FastTips Technical Library option from the FastTips Main Menu.

[References](#)

KBCategory:

KBSubcategory:

INF: Reports Questions & Answers

Article Number: Q104388
CREATED: 14-SEP-1993
MODIFIED: 14-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

=====
Microsoft(R) Product Support Services Application Note (Text File)
WX0640: REPORTS QUESTIONS & ANSWERS
=====

Revision Date: 9/93
No Disk Included

The following information applies to Microsoft Access(R), versions 1.0, and 1.1.

| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |
| PURPOSE. The user assumes the entire risk as to the accuracy and |
| the use of this Application Note. This Application Note may be |
| copied and distributed subject to the following conditions: 1) All |
| text must be copied without modification and all pages must be |
| included; 2) If software is included, all files on the disk(s) |
| must be copied without modification (the MS-DOS(R) utility |
| diskcopy is appropriate for this purpose); 3) All components of |
| this Application Note must be distributed together; and 4) This |
| Application Note may not be distributed for profit. |
| |
| Copyright (C) 1992-1993 Microsoft Corporation. All Rights Reserved |
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |
and Windows is a trademark of Microsoft Corporation.

1. Q. Why is every other page of my report blank? How can I correct this problem?

- A. This problem commonly occurs when the width of your report combined with the left and right margins is greater than the width of your paper. Adjust the width of your report and the margins so that your report fits on the page.

For additional information on preventing blank pages, please obtain Technical Note #95920. You can order this technical note by selecting the FastTips Technical Library option from the FastTips Main Menu.

2. Q. How can I convert a form to a report?

- A. To convert a form to a report, open the form in [Design view](#) and choose Save As Report from the File menu.
3. Q. How can I sort the data in a report by a [field](#) not displayed on the report?
- A. To sort report data by a field in a separate [table](#), create a [query](#) that includes the field from the other table, then base the report on the query. If you have already created the report, verify that the query includes all the fields used in the report and change the RecordSource property of the report to refer to the new query. Now you can adjust the Sorting and Grouping properties of the report to sort on the new field.
4. Q. How can I prevent a [page break](#) from occurring in the middle of a [group](#)?
- A. The KeepTogether property keeps all the controls for one [record](#) together within a particular section of a report. However, it does not keep multiple records together, nor does it keep a [group header](#) and its detail records together.

The following methods can prevent page breaks from occurring in the middle of groups:

- Set the ForceNewPage property of the report's Group Header section to Before Section. This ensures that each group will start on a new page and that as many detail records as possible will fit on one page.
- Repeat the group name, or place a "Continued" [label](#) at the top of each consecutive page.

For additional information on how to create these labels for consecutive pages of a group, please obtain Technical Notes #88156 and #93927. You can order these technical notes by selecting the FastTips Technical Library option from the FastTips Main Menu.

- Before placing the group [header](#) on the current page, create an [Access Basic function](#) that calculates how much space is left on the page.

[References](#)

KBCategory:

KBSubcategory:

INF: Forms Questions & Answers

Article Number: Q104387
CREATED: 14-SEP-1993
MODIFIED: 14-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

=====
Microsoft(R) Product Support Services Application Note (Text File)
WX0637: FORMS QUESTIONS & ANSWERS
=====

Revision Date: 9/93
No Disk Included

The following information applies to Microsoft Access(R), versions 1.0, and 1.1.

| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |
| PURPOSE. The user assumes the entire risk as to the accuracy and |
| the use of this Application Note. This Application Note may be |
| copied and distributed subject to the following conditions: 1) All |
| text must be copied without modification and all pages must be |
| included; 2) If software is included, all files on the disk(s) |
| must be copied without modification (the MS-DOS(R) utility |
| diskcopy is appropriate for this purpose); 3) All components of |
| this Application Note must be distributed together; and 4) This |
| Application Note may not be distributed for profit. |
| |
| Copyright (C) 1992-1993 Microsoft Corporation. All Rights Reserved |
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |
and Windows is a trademark of Microsoft Corporation.

1. Q. How do I invoke my Access Basic code from within a form?

A. To call a function from a property, type the following:

```
=<functionname>()
```

The equal sign and parentheses are required. You can call an Access Basic function from form or control properties, such as AfterUpdate or DefaultValue. Search for the name of a specific property in Help for examples. Subroutines cannot be called from properties. For more information, please refer to "An Introduction to Programming," Chapter 7, page 1.

2. Q. Do form rules override table rules?

- A. Yes, form rules do override table rules. When you position fields on a form by dragging them from the Field list, the [validation](#) property for the [field](#) inherits the [validation rule](#) that was defined for the field in the table design.
3. Q. When is it appropriate to use the exclamation point versus the period when identifying objects and properties in an [expression](#)?
- A. A good rule of thumb is to use the exclamation point before anything you specifically name (such as the name of your form or a control on the form) and to use a period before anything Microsoft Access names (such as a property).
4. Q. Why doesn't the [header](#) I created show in [Form view](#)?
- A. The header you created was most likely a [page header](#). There are two different types of headers available on forms: form headers and page headers. A [form header](#) is displayed on-screen and when printed; a page header is displayed only when printed.
5. Q. How do I reference a control on a [subform](#)?
- A. To reference the subform control itself, you must use the form property of the subform control as follows:
- ```
forms!<masterformname>!<subformcontrolname>.form!<controlname>
```
6. Q. When are validation rules on a form evaluated?
- A. Microsoft Access evaluates a validation rule only when data is entered or edited in a field and the cursor is moved to a different field or [record](#). If you leave the field unchanged, the validation rule is not evaluated. Microsoft Access also validates a field on a form when you leave the form, when you switch views, or when you close the form. To check for nulls, you must use a [macro](#). For more information, please refer to Chapter 22 of the "User's Guide."
7. Q. How do I create my own record-navigation system on a form without using the [navigation buttons](#) (called "VCR buttons")?
- A. The Customers form in ORDENTRY.MDB, a sample [database](#) supplied with Microsoft Access, is an excellent example you can follow to create your own record-navigation controls. The navigation buttons on this form use generic Access Basic code, which you can import to your application.
8. Q. Why do I get the message "#Error" in some controls on my forms or reports?
- A. Microsoft Access places an error value in a field or [text box](#) on a [datasheet](#) or form when it can't find information, execute an expression, or store a value within the field's prescribed limits. The following list describes the possible error values:

- #Error: Microsoft Access cannot execute the expression. You may have supplied either too few or incorrect arguments for an [aggregate function](#). This error also occurs in a table or form in which the DefaultValue property for a field or control is not appropriate for the DataType or FieldSize property setting, or in a [query](#) in which the value of a calculated field is larger than the FieldSize property setting allows. For example, this error message appears if you add or multiply two integer values and the resulting integer is larger than is permitted in an integer field.

This error also occurs when you substitute an expression for a text box and then use the text box's control name within the expression, as in the following example:

```
ControlName: [Test]
ControlSource: =[Test]
```

Since Microsoft Access automatically sets the ControlName property to match the field name, this error commonly occurs if you create a text box by dragging a field from the [Field list box](#) to a form or [report](#).

- #Num!: The numeric value is too large (either positively or negatively) for Microsoft Access to store in the field, based on the current DataType or FieldSize property settings.
  - #Name?: The ControlSource property you entered for the field's value is invalid. This error can result from any of the following conditions:
    - You have misspelled the ControlSource property.
    - You have deleted the ControlSource property itself.
    - You have omitted the equal sign (=) when you entered an expression for the ControlSource property.
  - #Div/0!: You tried to divide a number by zero. You can neither do this directly in an expression (for example, 8/0) nor indirectly by using a value from a field whose numeric value is zero.
  - #Deleted: The record to which you referred has been deleted. For example, if you or another user deletes a record from an underlying table's datasheet, this error value may replace a record in a form's datasheet.
  - #Locked: Another user has locked this record; thus, Microsoft Access cannot read the data.
9. Q. How can I check for duplicate records immediately after I enter a value in a [primary key](#) field?
- A. For a detailed answer to this question, please order Technical Note #102527 by selecting the FastTips Technical Library option from the FastTips Main Menu.

## [References](#)

KBCategory:  
KBSubcategory:

## INF: Interoperability Questions & Answers

Article Number: Q104386  
CREATED: 14-SEP-1993  
MODIFIED: 14-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0811: INTEROPERABILITY WITH OTHER APPLICATIONS QUESTIONS & ANSWERS  
=====

Revision Date: 9/93  
No Disk Included

The following information applies to Microsoft Access, version 1.0,  
1.1.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |  
| PURPOSE. The user assumes the entire risk as to the accuracy and |  
| the use of this Application Note. This Application Note may be |  
| copied and distributed subject to the following conditions: 1) All |  
| text must be copied without modification and all pages must be |  
| included; 2) If software is included, all files on the disk(s) |  
| must be copied without modification (the MS-DOS(R) utility |  
| diskcopy is appropriate for this purpose); 3) All components of |  
| this Application Note must be distributed together; and 4) This |  
| Application Note may not be distributed for profit. |

| Copyright (C) 1993 Microsoft Corporation. All Rights Reserved. |  
| Microsoft, FoxPro, Microsoft Access, and MS-DOS are registered |  
| trademarks and Visual Basic and Windows are trademarks of Microsoft |  
| Corporation. |  
| Paradox is a registered trademark of Ansa Software, a Borland |  
| company. |  
| dBASE is a registered trademark of Borland International, Inc. |  
| Sybase is a registered trademark of Sybase, Inc. |  
| ----- |

1. Q. When I export a Microsoft Access table to Microsoft SQL Server or Sybase SQL Server and then reattach it to Microsoft Access, I can't edit the data. Why does this problem occur?
  - A. Microsoft Access can update an attached SQL Server table only if the table has a unique index. However, Microsoft Access does not build SQL Server table indexes, unique or otherwise, when you export data; Microsoft Access allows you only to export data to SQL Server back ends. For example, if you attach the

table immediately after exporting data to it from Microsoft Access, the data is displayed in Datasheet or [Form view](#) as read-only; therefore, updates to the exported table's data are not allowed.

To work around this problem, you must create the SQL Server table indexes manually after you export the data from Microsoft Access to SQL Server. After the SQL Server indexes are created, you can create forms based on the attached SQL Server tables and update the data.

For information on creating indexes based on SQL Server [database](#) tables, see the "Microsoft SQL Server System Administrator's Guide," or pages 67-72 in the "Microsoft SQL Server Language Reference."

2. Q. How do I [link](#) a Microsoft Access table to a Microsoft Excel spreadsheet using dynamic data exchange ([DDE](#))?
  - A. The easiest way to use DDE is to create a paste link between Microsoft Access and Microsoft Excel. To do this, use the following steps:
    1. In the Microsoft Access [Database window](#), highlight the table you want to link to Microsoft Excel.
    2. From the Microsoft Access Edit menu, choose Copy.
    3. From the Microsoft Excel Edit menu, choose Paste Link.

The formula resulting from the Paste Link command should look like the following example

```
{=MSAccess|'NWIND.MDB;Table <Customers'>!All}
```

where <Customers> is the name of the table you copied.

For more information on linking Microsoft Access data to data in other applications, see the "Microsoft Access User's Guide," version 1.0, Chapter 13, "Using Pictures, Graphs, and Other Objects," pages 337-339.

3. Q. Are any new DDE commands included with Microsoft Access version 1.1?
  - A. Microsoft Access 1.1 supports the entire DDE functionality of version 1.0. In addition, it supports several new topics and items that allow DDE clients to do the following:
    - Open and close databases
    - Obtain the [data type](#) of a [field](#) in a table
    - Execute individual [macro](#) actions
    - Buffer and execute long SQL strings

- Quit Microsoft Access

For more information about the new DDE topics and items in the Microsoft Access 1.1 upgrade package, use Microsoft Access Help. To do this, use the following six steps:

1. Press F1 to start Help.
2. From the File menu, choose Open.
3. In your Microsoft Access program directory, select the MSA110.HLP file.
4. Choose OK. The "What's New in Microsoft Access Version 1.1" Help file is loaded.
5. At the top of the selection list, choose the jump topic "Common Questions and Answers."
6. Go to question #22, "How can I use Microsoft Access as a dynamic data exchange (DDE) server?," and select "Answer."

You are moved to DDE Server Topics.

To locate the DDE topics in the full Microsoft Access package, search for the keywords "DDE" and "Using Microsoft Access As a DDE Server" using the Help menu.

For additional information about and examples of using Microsoft Access as a DDE Server, please obtain Technical Note #89586. You can order this technical note by selecting the FastTips Technical Library option from the FastTips Main Menu.

4. Q. Why do the records I delete from an attached dBASE(R) table continue to reappear?
  - A. When you delete a record in a dBASE database, the record is actually only marked for deletion. It is not permanently removed until the database is packed in dBASE.

In dBASE, records that are marked for deletion still appear in any output form (such as a table in Datasheet view or a report in Print Preview), unless SET DELETED is ON. Because of this, Microsoft Access continues to display records in an attached dBASE table, even if the records are marked for deletion.

To prevent confusion, Microsoft Access provides functionality identical to the dBASE SET DELETED command. To take advantage of this functionality, use the following four steps:

1. Open the MSACCESS.INI file using NOTEPAD.EXE.
2. Find the section called [dBASE ISAM].
3. Change the Deleted=Off setting to:

Deleted=On

4. Save the MSACCESS.INI file and restart Microsoft Access.

When you view the dBASE data, records marked for deletion no longer appear. To permanently remove records marked for deletion, you must open the dBASE table in dBASE and run the PACK command.

5. Q. When I attempt to import or attach Btrieve files, why do I receive the error message "Couldn't find object <table name>"?

A. Microsoft Access uses the FILE.DDF file to determine the location of the Btrieve data file. The path statements in FILE.DDF may not be correct for the current environment. You should verify that the path is correct in the FILE.DDF file.

For additional information on how to resolve this problem, please obtain Technical Note #93685. You can order this technical note by selecting the FastTips Technical Library option from the FastTips Main Menu.

6. Q. Should I use Microsoft Access or Microsoft FoxPro(R) as my database solution?

A. Microsoft FoxPro is the best database solution for users and developers of Xbase products, for developers who need a cross-platform solution, and for all customers who can benefit from FoxPro's incredible speed and unequalled development environment.

Microsoft Access is the best database solution for users who need seamless access to data in multiple formats, for users who are not professional programmers but need to develop powerful database applications, and for users and developers who need the most productive development environment to design decision-support systems.

7. Q. What are the differences between Microsoft Access and Microsoft Visual Basic(TM)? Does Microsoft Access support custom controls designed for Visual Basic? Do these two applications use the same or different programming languages?

A. While Microsoft Access and Visual Basic share the same basic programming concepts (an [event](#)-driven programming model, visual interface creation, and so forth), each has different specialties.

Visual Basic version 3.0 is a general-purpose programming environment, which uses visual tools that allow software developers to design both database applications and other applications. Visual Basic has a more granular event model than Microsoft Access and supports custom controls.

Microsoft Access is a programmable database designed specifically for creating and running database applications. Its rich set of tools allows you to create many database applications without programming. Microsoft Access supports

database-oriented events, such as FindRecord and OnUpdate, rather than Visual Basic-like properties and events, such as MouseMove and MinButton. Microsoft Access currently does not support custom controls.

The nature and [syntax](#) of [Access Basic](#) and Visual Basic are very similar, but not identical. Access Basic supports database commands and methods, such as CreateDynaset and Clone. Visual Basic supports properties and events, such as PathChange and MouseUp, that Access Basic does not have. However, the many similarities between Access Basic and Visual Basic allow you to move between the two products easily.

## [References](#)

KBCategory:

KBSubcategory:

## INF: Access Distribution Kit Questions & Answers [B\_WAccADK]

Article Number: Q104385  
CREATED: 14-SEP-1993  
MODIFIED: 14-SEP-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0867: ACCESS(R) DISTRIBUTION KIT QUESTIONS & ANSWERS  
=====

Revision Date: 9/93  
No Disk Included

The following information applies to Microsoft Access, version 1.1.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |  
| PURPOSE. The user assumes the entire risk as to the accuracy and |  
| the use of this Application Note. This Application Note may be |  
| copied and distributed subject to the following conditions: 1) All |  
| text must be copied without modification and all pages must be |  
| included; 2) If software is included, all files on the disk(s) |  
| must be copied without modification (the MS-DOS(R) utility |  
| diskcopy is appropriate for this purpose); 3) All components of |  
| this Application Note must be distributed together; and 4) This |  
| Application Note may not be distributed for profit. |  
| |  
| Copyright (C) 1993 Microsoft Corporation. All Rights Reserved. |  
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |  
| and Windows is a trademark of Microsoft Corporation. |  
| ----- |

1. Q. My machine appears to hang when I first run the SUFILES utility (SUFILES.EXE), whether I start the SUFILES utility by double-clicking the SUFILES icon or by selecting SUFILES from the first dialog box after starting the Microsoft Access SetupWizard. My mouse pointer does not switch from pointer to hourglass to indicate that the program is still processing. The progress meter remains at 0% and I do not see any hard disk activity. Why does this occur?
  - A. The program is not hung; it just does not update the screen regularly. The screen eventually updates, allowing you to see the program's progress. This behavior is by design so that you can choose the Cancel button at any time.

The amount of time the SUFILES.EXE file takes to run varies,

depending on the machine type and the amount of memory. Average completion time is approximately 20 minutes. For example, using a 486/50 machine with 16 megabytes (MB) of memory, the process takes approximately 10 minutes to complete; using a 386/16 machine with 4 MB of memory, the process takes approximately 45 minutes.

You need to run SUFILE.EXE only once. SUFILE.EXE compresses files from the various directories, including the Windows and Access program directories, into a subdirectory called SUFILES. When you build your distribution disks, the SetupWizard needs only to copy the compressed files to the disk directories and does not have to recompress all the files each time distribution disks are built.

2. Q. At the end of the Microsoft Access installation process, why does Custom Setup hang immediately after the SETUP.EXE file creates the program [group](#)?

A. This problem occurs only when you choose to have the Change Workgroups icon installed but have not specified an icon for the application.

To correct this problem, run the SetupWizard again. When the "Where are your application's initialization, icon, and Help files located?" dialog box appears, select an icon file from the .ICO File (Optional) [list box](#).

You can also manually modify the STFSETUP.IN\_ file. For information on modifying the STFSETUP.IN\_ file, please refer to the "Microsoft Access Distribution Kit Guide," Chapter 3, "Creating Your Custom Setup Program."

3. Q. How can I prevent other users from seeing my code?

A. You can [control](#) what users of your application see by taking the following precautions:

- Using forms and macros to control a user's movements within your [database](#)
- Securing the objects in your database
- Encrypting your database

4. Q. What is the best way to distribute updates of my application to other users?

A. The best way to distribute updates of your application is by creating two databases: one database (DATA.MDB) to store the application's data, the other database (PROGRAM.MDB) to store the application's forms and other objects. This [method](#) simplifies your application updates, since you can redistribute an updated PROGRAM.MDB file (containing updated queries, forms, reports, macros, and modules) without destroying your user's data.

All objects in the PROGRAM.MDB file should be based on attached tables from the DATA.MDB file. When you create a database with attached tables, the specific path to these tables is stored in the database. If the path to these tables changes, you receive the "Couldn't find table <table name>" error message. To avoid this problem, have your AutoExec macro, which the Access Distribution Kit (ADK) requires, call an Access Basic function to verify that the attached tables are indeed attached. If the tables for some reason are not attached, your function attaches them automatically.

When you start your application, make sure that the application's working directory property is set to be the same directory in which the .MDB file is located. The SetupWizard automatically sets this property in Program Manager. Use the CurDir() function to determine the directory from which the application is being executed, then attach the files from that directory.

5. Q. Why do I get a general protection (GP) fault when I run my application's Custom Setup program?
- A. Running Setup in Windows standard mode causes a GP fault. You should run Setup in 386 enhanced mode. To start Windows in 386 enhanced mode, type the following command at the MS-DOS prompt:

```
win /3
```

Once Setup is complete, your application will run correctly in Windows standard mode.

6. Q. When I start the run-time version of my application, the following error message is displayed:

A run-time error occurred in module: <module name>. You don't have permission to view this module.

Why do I get this error message?

- A. This error message appears when a run-time error occurs in one of your modules. When this problem occurs in the retail package of Microsoft Access, the module is opened and the line of code that caused the error is selected. Because the run-time version of the application cannot open the module nor display the line causing the error, the previous error message is displayed.

To correct this problem, open your application in the full retail package of Microsoft Access and correct the error in your code from there.

7. Q. I split my database into two files using the COMPRESS.EXE utility. After running Custom Setup, I am unable to open my database file. How do I put the two .MDB files back together?
- A. Once a database file has been split into two files, you must use entries in the Custom Setup Information file, STFSETUP.IN\_, to rejoin the files. The following steps give an example of how

to combine two fictitious files, called MYAPP.MD\$ and MYAPP2.MD\$, into one file, called MYAPP.MDB:

1. Open the STFSETUP.IN\_ file in NOTEPAD.EXE or other text editor.
2. In the [Standard Setup Files] section of the STFSETUP.IN\_ file, change the following line

```
3 , MYAPP2.MD$, RENAME=MYAPP2.MDB, DEST=$(AppPath)
to
3 , MYAPP2.MD$, APPEND=MYAPP.MDB, DEST=$(AppPath)
```

NOTE: If your database file was split into more than two files, make similar changes for each entry in the STFSETUP.IN\_ file.

3. Make similar changes to the [Network Setup Files] and/or the [Multiuser Files] sections for any server installations of your application.

### [References](#)

KBCategory:

KBSubcategory:

## PRACC9309: Incorrect Error Message Using OpenTable Method

Article Number: Q104066  
CREATED: 05-SEP-1993  
MODIFIED: 20-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

When you use the OpenTable [method](#) in [Access Basic](#) on tables with certain [table](#) names that do not exist, you get the following error message

Syntax error in <xxxxx> statement

instead of the expected error message

Object does not exist

For example, if you try open a nonexistent table called "Delete This Table," you would receive a "Syntax error in DELETE statement" error message, rather than an "Object does not exist" error message.

### CAUSE

---

If the table referenced in the OpenTable method does not exist, Microsoft Access tries to interpret the table name as a [SQL](#) statement. Since the statement is not valid, you receive the SQL parsing error "Syntax error in DELETE statement."

Because of this, other SQL statements used as valid but nonexistent table names can cause other SQL parsing error messages. Some of the other possible error messages are listed in the table below:

| If Table Name Is...    | Microsoft Access Error Message Is...                                                         |
|------------------------|----------------------------------------------------------------------------------------------|
| -----                  | -----                                                                                        |
| Select                 | Syntax error in Select statement                                                             |
| Select * from authors  | Missing semicolon at end of SQL statement (when Authors table does not exist)                |
| Select * from authors; | Couldn't find input table or <a href="#">query</a> "Authors" (when Authors table does exist) |
| Select * from authors; | Couldn't find object "Select * from authors;"                                                |

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access

versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

#### [References](#)

## INF: Visual Basic 3.0 and Microsoft Access 1.x

Article Number: Q104065  
CREATED: 05-SEP-1993  
MODIFIED: 15-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Visual Basic Professional 3.0
- 

### SUMMARY

---

Microsoft is committed to enhancing all of our desktop [database](#) tools over time. With the release of Visual Basic Professional 3.0 and the ability to manipulate Microsoft Access databases directly through the language, there is now confusion as to which product the developer should purchase. You can expect to see even greater overlap between their capabilities in the future, until eventually your choice of tools will boil down to your choice of language: Basic, C, or Xbase. The two main principles that guide Microsoft's development plans are responding to your needs through innovation and solid execution, and sharing innovations between products.

This article outlines some important features of Microsoft Access 1.x and Visual Basic Professional so that the developer can make the decision as to which of the current products to use as a development tool.

### MORE INFORMATION

---

Microsoft Access  
-----

Microsoft Access is an interactive end-user tool that is optimized for customizability and database-centric solutions. The user will find that it is designed exclusively for databases, with simple yet powerful integrated database tools. These tools, such as data aware forms and reports, are designed for manipulating data within the Microsoft Access interface.

Manipulating the [form](#) and [report](#) objects is an easy task with the use of macros and the [macro](#) builder. There is also the full feature language that is very similar to that of Visual Basic.

Since there is no concept of an executable database file, an enduser that receives the developers application must have one of the following:

1. The retail version of Microsoft Access.
2. The necessary runtime version of Microsoft Access, which is shipped with the Access Distribution Kit files. These files can be distributed with the application royalty free by the

developer of the application.

Since the user is required to have Microsoft Access, the enduser would have complete ability to change or modify the MDB file unless security is implemented on the actual objects.

Some other features as compared numerically below with Visual Basic features are:

1. Microsoft Access is an interactive end-user tool that is designed for databases.
2. Full-featured report writing tool that allows flexible report writing for a custom database solution.
3. Ships code that supports all of the Microsoft Access functionality for manipulating an MDB file.
4. Drive Access application through Access macros or Access Basic.
5. Supports one .MDB with access to other formats via user interface-set linked tables. You can attach to multiple .MDBs, through the user interface or programmatically.
6. Data-aware controls bound to data controls on a form or report.
7. Multiple subforms supported on form. The data within the subform can be linked to fields on the parent form so that updating the parent form, will cause the subform to reflect the changes. This can be done completely by setting properties of the subforms without any code or macros.
8. End-user usability - Cue Cards walk inexperienced users through data-related tasks such as creating a crosstab query.
9. All project code is stored in the database and is managed through the Microsoft Access interface. Code in other databases are accessible through the use of Microsoft Access libraries.
10. Additional tools for interactive users:
  - Report, Form, and Graph Wizards
  - Macros (fill in the blank programming)
  - Graphical Query by Example
  - Cue Cards
11. Full featured security on all objects.
12. There is no implementations of custom controls available.
13. User friendly objects that are easy to implement/modify for database manipulation such as macros, forms, reports and queries.

14. Microsoft Access efficiently uses system resource even when you display vast amounts of data in a grid like presentation. In many instances, Microsoft Access will use the same control to display several pieces of data in the same window/form. Because of this, developers may find that they do not have the fine control of the form objects by using standard windows API functions.

#### Visual Basic Professional 3.0

-----

Visual Basic is optimized for programmer versatility and extensibility. It is the code-centric tool for multi-function applications with complete custom control over the user interface. Visual Basic creates .EXEs: pre-compiled code tuned to load and execute which is processed through a dynamic link library (VBRUN300.DLL). The Visual Basic compiler is not needed to run the developers application. Starting with Visual Basic Professional 3.0, the Access Data Engine was also included as part of the product giving complete accessibility to the Microsoft Access data which can include any foreign data type attached to the MDB file such as Paradox and Btrieve. It does not give access to other objects within the MDB file such as the reports and forms. There is no graphically means of defining the projects queries. You must construct them within code.

Visual Basic does not provide access to modifying the security of an MDB file. The developer must implement security through the Microsoft Access security interface.

To distribute an application, the end user would only require the .EXE that the developer developed, VBRUN300.DLL and the .MDB file. They would not require Microsoft Access or Visual Basic.

The only way that the end-user could modify the application's objects would be for the developer to implement the logic within the code, or supply the source code.

Some other features as compared to Microsoft Access above in numeric equivalent order:

1. Full programmatic object layer for "run-time" database solution.
2. Full featured report writer with the ability to execute predefined reports.
3. Ship only the code for the objects being used.
4. Complete Visual Basic language set plus a subset of the Microsoft Access Basic to fully manipulate data. Microsoft Access macro actions are not available to the Visual Basic EXE's.
5. Visual Basic attaches identically to any database format that Microsoft Access can through attached tables within the MDB file.
6. Has data-aware controls bound to fields within the tables.
7. Multiple instances of MDI child forms as specified by the user. There is no built in way to link the data on the subform with

that of the parent form. This must be handled programmatically.

8. Developer usability - full [debugging](#) tools including watch statements and call threads.
9. ASCII code in distinct modules and works with CASE and SLM tools.
10. Additional developer tools for commercial development:
  - Setup toolkit w/DLL version check
  - Win API reference
  - Help compiler
11. Visual Basic cannot modify the security implemented in the .MDB's but must oblige to the security established on the objects. All security must be establish through the Microsoft Access interface.
12. Supports all .VBX files for additional add on custom controls.
13. VB can not read any of the macros, modules, forms, or reports within an MDB file. In VB there also is no real object-creation user interface for constructing queries. The developer must either hard code a [SQL string](#) or construct the string on the fly with code.
14. Visual Basic has true window controls that you create on your form. Through incorrect use by the developer, this can lead to resource inefficiency. Because the controls are true windows, you do have more control over the actual controls with standard Windows API's.

#### Microsoft Access Distribution Kit

-----

The Microsoft Access Distribution Kit (ADK) is the run-time for Microsoft Access. It is designed for the developer who wants to distribute their application to other users who do not have Microsoft Access in their machines. Many of the design feature and tools of Microsoft Access are not included in the ADK. This gives more control to the developer as to how the end-user will use the application.

The ADK also comes with advanced developer tools such as:

- Setup toolkit with DLL version check
- Help compiler
- Documentation on AccessWizards

#### [References](#)

## PRACC9308: GP Fault Referencing Column Property of ComboBox

Article Number: Q103999  
CREATED: 02-SEP-1993  
MODIFIED: 02-SEP-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SYMPTOMS

---

The following message will appear when trying to reference the Column property of a combo box in Datasheet view:

An error has occurred in you application. If you choose Ignore, you should save your work in a new file. If you choose Close, your application will terminate".

Followed by the message:

Application Error: Microsoft Access caused a General Protection Fault in module MSAIN110.DLL at 0005:1E18

### CAUSE

---

The General Protection Fault will happen under the following conditions:

1. The combo box has a Query or Table as the RowSource.
2. The form must be opened directly into Datasheet view.
3. A selection has been made in the combo box.
4. The Column property of the combo box is referenced in an expression.

### RESOLUTION

---

The following work arounds can be used:

1. If the form will only be used in Datasheet view, use a list box instead of a combo box. (A list box is functionally equivalent to a combo box in Datasheet view).
2. If the form will be used in both Datasheet view and Form view, make sure the form is opened first in Form view before going to Datasheet view. The following macro can be used to open the form into Form view, then into Datasheet view:

| Action   | Described Below |
|----------|-----------------|
| =====    |                 |
| Echo     | 1.              |
| OpenForm | 2.              |

OpenForm            3.  
Echo                4.

#### Macro Actions

=====

1. Echo  
    Echo On: No
2. OpenForm  
    Form Name: MyForm  
    View: Form
3. OpenForm  
    Form Name: <your form name here>  
    View: Datasheet
4. Echo  
    Echo On: Yes

## STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.1. Microsoft is researching this problem and will post new information here as it becomes available.

## MORE INFORMATION

---

### Steps to Reproduce Problem

-----

1. Open the Northwind Traders [database](#) (NWIND.MDB) in Microsoft Access version 1.1.
2. Create a blank, unbound, form with the following [form properties](#):  
  
    DefaultView: Datasheet
3. Create a combo box on the form with the following properties:  
  
    ControlName: MyCombo  
    RowSourceType: Table/Query  
    RowSource: Employees  
    ColumnCount: 2
4. Close and save the form with the name "MyForm".
5. Open the form from the [Database window](#) and make a selection in the combo box.
6. With the form left opened, create a new module.
7. Choose [Immediate window](#) from the View menu.
8. Type the following and press Enter:  
  
    ? Forms!MyForm!MyCombo.Column(1)

## [References](#)

## **PRACC9309: Formatted Memo Truncates Data At 255 Characters**

**Article Number:** Q103998  
**CREATED:** 02-SEP-1993  
**MODIFIED:** 20-SEP-1993  
**VERSION(S):** 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

When you apply formatting, such as a closing or opening angle bracket (> or <), to a field with the Memo data type, the data in the field appears truncated after the 255th character.

### **CAUSE**

---

When you apply formatting to a Memo field, it is treated as a Text field, which has a 255-character limit. The data is not physically truncated; rather, only 255 characters are viewable.

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### [References](#)

## PRB: "Error Writing to File 'MSAIN100.DLL'" During Install

Article Number: Q103997  
CREATED: 02-SEP-1993  
MODIFIED: 02-SEP-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SYMPTOMS

---

If you get the message

Error writing to file 'MSAIN100.DLL'

when upgrading a network installation from Microsoft Access 1.0 to Microsoft Access 1.1.

### CAUSE

---

Make sure no one is using Microsoft Access on the network. During the installation process of Microsoft Access 1.1, the file MSAIN100.DLL is deleted and replaced with MSAIN110.DLL. This file provides the international support for Microsoft Access.

If someone is using Microsoft Access when you start the installation process, the message,

Error writing to file 'MSAIN100.DLL'

is received when the setup tries to delete that file. Once everyone has closed out of Microsoft Access, restart Microsoft Access setup.

### MORE INFORMATION

---

The following files are deleted during the installation process:

BTRVISAM.DLL  
DBSIASM.DLL  
MSABC100.DLL  
MSAES100.DLL  
MSAIN100.DLL  
MSAJU100.DLL  
MSAJT100.DLL  
PDXISAM.DLL

References: README.TXT

[References](#)

## INF: Imported Excel Date Fields Off By 4 Years

Article Number: Q103996  
CREATED: 02-SEP-1993  
MODIFIED: 02-SEP-1993  
VERSION(S): 1.00 1.10

---

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Excel versions 2.0, 3.0 and 4.0
- 

### SUMMARY

---

If date fields imported from an Excel spreadsheet are off by four years, then, on the original excel spreadsheet, the 1904 date system was used. This setting can be verified in Excel by selecting Calculation under the Options menu. Then in the Sheet Options group you will find the check box labeled "1904 Date System". If the box is checked, the spreadsheet is based on the 1904 Date System. If the box is not checked, the spread sheet is based on the 1900 Date System.

### MORE INFORMATION

---

The date systems used by Microsoft Excel can be based on one of two different dates. By default, Excel for Windows starts its dates at January 1, 1900. A serial number of 1 in Excel for Windows represents January 1,1900. Excel can be set to start on January 1, 1904. The reason this option was included in Excel for Windows was to make it compatible with Excel for the Macintosh which starts at January 1, 1904. The starting serial numbers also are different. A serial number of 0 in Excel for the Macintosh represents January 1, 1904.

If the spreadsheet you need to import into Access was based on the 1904 date system, complete the following steps before importing the spreadsheet into Microsoft Access:

1. Open the spreadsheet in Excel.
2. From the Options menu choose Calculation.
3. In the Sheet Options Group, uncheck the box labeled "1904 Date System".
4. Save and close the spreadsheet.

The spreadsheet is now ready to be imported into Microsoft Access.

If you do not have Excel available to switch the date system before importing the spreadsheet, you can do an Update Query to correct the dates after the spreadsheet has been imported into Microsoft Access:

1. Go into the Query container and click on "New".
2. Add the table that was created when importing the

spreadsheet.

3. Drag the DateField name to the "field:" in the query grid for this example the date field's name is "DateField").
4. From the Query menu select Update
5. In the "Update To:" field enter: [DateField] + 1462
6. From the Query menu select Run.

A message box will appear and tell you how many records were updated. Your dates have been successfully corrected.

### [References](#)

**PRACC9308: Setup Ignores \$(AppPath); Puts Files in MS-SETUP.T [B\_WAccADK]**  
**Article Number: Q103995**  
**CREATED: 02-SEP-1993**  
**VERSION(S): MODIFIED: 02-SEP-1993**  
ENDUSER |

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

## **SYMPTOMS**

---

When you run Setup and specify that you want to install the files into a particular directory, all files except SYSTEM.MDA and MSACCESS.REG are instead copied to the MS-SETUP.T subdirectory.

## **CAUSE**

---

This problem occurs because Setup is case-sensitive. If the \$(AppPath) information file token is either misspelled or capitalized incorrectly (for example, \$(APPPATH) instead of \$(AppPath),) Setup does not recognize the misspelled or miscapitalized token, so by default it copies all files to the current directory, which is MS-SETUP.T.

This limitation is noted on page 42 of the "Microsoft Access Distribution Kit Guide".

## **RESOLUTION**

---

Be sure that the \$(AppPath) information file token is spelled correctly and capitalized as shown.

## **STATUS**

---

Microsoft has confirmed this to be a problem in the Microsoft Access Distribution Kit version 1.1. Microsoft is researching this problem and will post new information here in the knowledge base as it becomes available.

## REFERENCES

=====

"Microsoft Access Distribution Kit Guide," version 1.1, Chapter 3,  
"Creating Your Custom Setup Program," "Using Tokens," pages 42-43.

[References](#)

## **PRACC9309: Custom Setup Gives Wrong Low Disk Space [B\_WAccADK]**

Article Number: Q103994  
CREATED: 02-SEP-1993  
MODIFIED: 02-SEP-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

When you run a custom setup with little hard disk space available, Setup alerts you that you need additional disk space. However, the amount of space it tells you to free up is not enough to complete the entire setup, and subsequently you get a Setup error message.

### **CAUSE**

---

This error occurs because Setup only calculates the remaining space required for successful setup based on the files that it provides. It does not take into account the space that your application's files also require.

For example, say that you have only 10K of hard disk space remaining, you attempt to run a custom setup, and you get a message telling you that you need an additional 90K of disk space. In reality, you would need more space than this, since this number only reflects how much more room you need to install the ADK Setup support files and does not take into consideration how much room your application's files require.

### **RESOLUTION**

---

Based on your estimates of the size of your application, free at least that amount, or more, of disk space to allow room for all required files.

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access Distribution Kit version 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### **References**

## **PRB: DDEPoke To WinWord Contain Incorrect Character Format**

**Article Number:** Q103993  
**CREATED:** 02-SEP-1993  
**MODIFIED:** 02-SEP-1993  
**VERSION(S):** 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit versions 1.1
  - Microsoft Word for Windows
- 

### **SYMPTOMS**

---

When poking, or sending, data to a bookmarked location in a Microsoft Word for Windows document during a [DDE](#) conversation, the information may arrive formatted for the incorrect font and size.

### **CAUSE**

---

Microsoft has confirmed this to be a problem with Word for Windows. We are researching this problem and will post new information here as it becomes available.

### **RESOLUTION**

---

The Word for Windows product support [group](#) has described a work around for any program trying to Poke data into that program. To review that article [query](#) in the Word for Windows knowledgebase on:

ddepoke bookmark format character

### [References](#)

## PRB: Cannot Add Security Items to Custom Menu

Article Number: Q103992  
CREATED: 02-SEP-1993  
MODIFIED: 02-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

When you add a [security](#) command to a custom menu by using the DoMenuItem action, the item will be grayed out.

### CAUSE

---

You can only perform DoMenuItem actions with commands available from the current context. If you have a [form](#) opened, you can only perform menu items available to a form. Since the security items are only available from the [Database window](#), they will not work when invoked from a form.

### RESOLUTION

---

This behavior is by design. See the following work around.

### MORE INFORMATION

---

Steps to Reproduce Behavior

-----

1. Create the following [macro](#) and save it with the name "Security Menu":

| Action | Defined Below |
|--------|---------------|
|--------|---------------|

-----

|         |    |
|---------|----|
| AddMenu | 1. |
|---------|----|

Security Menu Actions

-----

1. AddMenu

Menu Name: Security Menu

Menu Macro Name: Security Menu Password

Status Bar Text: Change Password

2. Create the following macro and save it with the name "Security Menu Password":

| Macro Name | Action | Defined Below |
|------------|--------|---------------|
|------------|--------|---------------|

-----

|                |            |    |
|----------------|------------|----|
| Change Passord | DoMenuItem | 1. |
|----------------|------------|----|

Security Menu Password Actions

-----

1. DoMenuItem
  - Menu Bar: Database
  - Menu Name: Security
  - Command: Change Password

3. Create/Open a form and set the OnMenu property to "Security Menu"
4. View the form and note that Change Password is grayed out on the Security menu.

#### Work Around

-----

The following macro can be used to add a Change Password item on a custom form menu.

| Macro Name      | Action       | Defined Below |
|-----------------|--------------|---------------|
| Change Password | Echo         | 1.            |
|                 | SelectObject | 2.            |
|                 | DoMenuItem   | 3.            |
|                 | DoMenuItem   | 4.            |
|                 | SelectObject | 5.            |

#### Change Password Actions

1. Echo
  - Echo On: No
2. SelectObject
  - Object Type: Form
  - Object Name: <name of the form that the menu is on>
  - In Database window: Yes
3. DoMenuItem
  - Menu Bar: Database
  - Menu Name: Security
  - Command: Change Password
4. DoMenuItem
  - Menu Bar: Database
  - Menu Name: Window
  - Command: Hide
5. SelectObject
  - Object Type: Form
  - Object Name: <name of the form that the menu is on>
  - In Database window: No

Note: Actions 1 and 5 above are used in the [event](#) you need to have the Database window hidden from view while using the retail version of Microsoft Access. The Echo action ensures that when the Database window is activated in step 2, that it is not visible. Step 4 re-hides the [database](#) window.

If you are using the Microsoft Access Distribution Kit (ADK) steps 1 and 5 do not need to be included since the database window is always invisible.

The only side affect to this workaround is that when the change password dialog appears, you will see the Database window menu items

in the background. However, the user will not be able to activate them when the dialog is up. Once the user chooses OK, or Cancel, the macro re-hides the Database windows and switches focus back to the form.

References:

[References](#)

## **PRB: Microsoft Access Issues with MS-DOS 6.0**

**Article Number:** Q103991  
**CREATED:** 02-SEP-1993  
**MODIFIED:** 02-SEP-1993  
**VERSION(S):** 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

You receive an error message reporting that you are out of memory. The message may indicate that you have insufficient space in your TEMP directory, insufficient disk space, or simply that you are 'out of memory'.

### **CAUSE**

---

The cause may be because of the way the memory on your machine is configured or because of disk compression on your hard drive.

### **MORE INFORMATION**

---

#### DOUBLESPACE

-----

DOS 6.0 includes its own disk compression software. You may increase your compression ratio with DoubleSpace. In fact what this is doing is making your available free disk space look larger than it actually is and larger than it will most likely be.

For example you could set a compression ratio of 16 to 1. If your hard drive has 2 megs of free disk space, it will look like you now have 32 megs of free space. A DIR command will also tell you that there is 32 megs of available disk space. This is where there is a problem with Microsoft Access. You have 32 megs of free disk space, Microsoft Access recommends only 14 megs of available disk space. You now attempt to install Microsoft Access and you get an error message reporting that you are out of disk space. Even though you thought you started with 32 megs free.

The average compression ratio of a file is about 1.5 to 1. That is the actual amount that an average file can be compressed. So what you can actually use on that 2 megs of free space is about 3 megs. For example a \*.BMP file will compress to 16 to 1, so you could put 32 megs of bitmaps on that 2 megs. An executable (\*.EXE) file will only compress to 1.1 to 1 or not at all. You will only be able to put 2 megs of executable files on that 2 megs. This shows the extremes, but for a specific example with Microsoft Access lets use the MSACCESS.HLP, which is Microsoft Access' largest file. It is 3166132 bytes in actual size. It has an actual compression ratio of 1.1 to 1. It will compress, but only to about 3 megs and if the compression ratio was set to 16 to 1, you just lost 45 megs that you thought you were going to be able to use.

## RESOLUTION

---

Exit Windows and type DBLSPACE at the C:\ prompt. From the Drive menu choose Info... If the Compression ratio: is set larger than the Est. compression ratio:, choose Ratio and set the New estimated compression ratio to the same as the Estimated compression ratio for drive X:. This will give you the most accurate information about the amount of actual usable free disk space.

### MEMMAKER

-----  
DOS 6.0 has a memory manager that will free up conventional memory. One of the changes that it will make on occasion is to change your FILES= statement. This is recommended to be at least 50 for Access to run reliably. MemMaker has in some cases changed this to below this setting and even as low as 8. Secondly, it will in most instances add the parameter HIGHSCAN to the EMM386.EXE line. This will include the E000-EFFF range. This will work fine on some machines, but others will have conflicts with this region. This will be especially evident in machines that are using RAM Shadowing, since this is the normal range for this feature.

### FILES

-----  
For the FILES= set it to at least 50. For the EMM386.EXE line with the HIGHSCAN option remove that parameter, all i=xxxx-xxxx parameters and make sure the NOEMS switch is present.

### BOOT

-----  
DOS 6.0 has a very welcomed feature called Clean Boot. When you see Starting MS-DOS... if you hit the F5 key it will automatically be booted clean. Only the following will be loaded Path C:\DOS, Prompt \$P\$G, and COMSPEC=C:\DOS\COMMAND.COM. Note that Windows will not be able to run at this point. To easily enable Windows to run, place the first Windows 3.1 diskette in the drive and type XMSMMGR. You will be have a clean AUTOEXEC.BAT and CONFIG.SYS. You may also press the F8 key at the same time and it will give you the option to load each line of your CONFIG.SYS, 1 line at a time. It will however load either all of the lines in your AUTOEXEC.BAT or none of them at all.

### UNDELETE

-----  
If you are using the delete option Sentry then you may have incorrect information about available disk space. Using the Sentry option simply takes the deleted file and make a hidden backup of this file in a hidden directory. By default the files in this directory will be purged after 7 days. You may run a DIR command and get a report of the amount of free disk space, but may run out of disk space before using that amount of space. The DIR command will not consider all of these hidden files when reporting free disk space. If you are showing 20 megs of free disk space and get an out of disk space error message when installing Access you will want to do a DIR /A. This will give you an idea of the amount of space being used by these hidden files. You may delete these files and free up the additional space, but these files will no longer be guaranteed recoverable.

References:

"Microsoft MS-DOS 6 User's Guide"

[References](#)

## PRB: Update Query on More than One Table Gives Inaccurate Data

Article Number: Q103990  
CREATED: 02-SEP-1993  
MODIFIED: 16-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

An update query that attempts to update more than one table may yield inaccurate results.

### RESOLUTION

---

This behavior is by design. You cannot update more than one table in an update query.

### MORE INFORMATION

---

If an update query in Microsoft Access is used to update two (or more) tables on the "many" side of a relationship, the resulting updated fields will be multiples of the updated value. Each value will be updated one time for each updated record in the corresponding table.

Steps to Reproduce Behavior

-----

1. Create the following three tables:

Table: Vendors

-----  
FieldName: Name  
    DataType: Text

Table: New Parts

-----  
FieldName: Item  
    DataType: Text  
FieldName: Vendor  
    DataType: Text  
FieldName: Cost  
    DataType: Number

Table: Rebuilt Parts

-----  
FieldName: Item  
    DataType: Text  
FieldName: Vendor  
    DataType: Text  
FieldName: Cost  
    DataType: Number

2. Add the following two records to the Vendors table:

```
Name

Bob
Jill
```

3. Enter the following three records to the New Parts table:

| Item   | Vendor | Cost   |
|--------|--------|--------|
| -----  | -----  | -----  |
| fender | Bob    | 100.00 |
| hood   | Jill   | 50.00  |
| trunk  | Bob    | 100.00 |

4. Enter the following four records in the Rebuilt Parts table:

| Item   | Vendor | Cost  |
|--------|--------|-------|
| -----  | -----  | ----- |
| Pump   | Bob    | 10.00 |
| Fan    | Jill   | 45.00 |
| Pulley | Bob    | 20.00 |
| Piston | Bob    | 30.00 |

5. Create the following query based on the three tables that you created above:

```
Query: Example Query

Type: Update Query
Join: [New Parts].[Vendor] <-> [Vendors].[Name]
Join: [Rebuilt Parts].[Vendor] <-> [Vendors].[Name]

Field: Name
 Table: Vendors
 Criteria: "Bob"
Field: Cost
 Table: New Parts
 Update To: [New Parts].[Cost] * 2.0
Field: Cost
 Table: Rebuilt Parts
 Update To: [Rebuilt Parts].[Cost] * 2.0
Field: Name
 Table: Vendors
 Criteria: "Bob"
```

6. Run the query, then save and close it.

7. Open the Rebuilt Parts table.

Note that the costs for items listing Bob as the vendor have been updated to four times their original values, rather than to two times, as specified in the update query. Each updated record in the Rebuilt Parts table has been updated once for each updated record in the New Parts table.

8. Open the New Parts table.

Note that the costs for items listing Bob as the vendor have been updated to eight times their original values, rather than to two times, as specified in the update query. Each updated record in the New Parts table has been updated once for each updated record in the Rebuilt Parts table.

This effect is called a cross product, in which the values in Table B are updated one time for each item meeting the same criteria in Table A.

#### REFERENCES

=====

"User's Guide," version 1.0, pages 174-178

[References](#)

## INF: Online Help and Text Files Installed By 1.1

Article Number: Q103989  
CREATED: 02-SEP-1993  
MODIFIED: 02-SEP-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1  
-----

### SUMMARY

---

This article lists the Online Help and text files included with Microsoft Access version 1.1 which contain information that is not in the printed documentation.

This article includes the files included with the 1.1 upgrade package as well as those installed by the full 1.1 package.

### MORE INFORMATION

---

The [table](#) belows lists the file name; whether the package is installed by the Upgrade; the full 1.1 package, or both; whether the file is placed in the directory Microsoft Access is installed in or in the System subdirectory of the Windows directory; and a brief description.

| FileName     | Package | Directory | Description:                                     |
|--------------|---------|-----------|--------------------------------------------------|
| MSA110.HLP   | Upgrade | Access    | What's New in 1.1?                               |
| README.TXT*  | Upgrade | Access    | Suppliment to Documentation                      |
| ANALYZER.TXT | Full    | Access    | Database Analyzer                                |
| ERRATA.TXT   | Full    | Access    | Errors in Documentation                          |
| README.TXT*  | Full    | Access    | Suppliment to Documentation                      |
| ORDENTRY.TXT | Full    | Access    | Order Entry Information                          |
| ORDENTRY.HLP | Full    | Access    | Order Entry Information                          |
| PIM.HLP      | Full    | Access    | PIM help Information                             |
| MSACCESS.HLP | Full    | Access    | Includes 1.0 and 1.1 Help                        |
| BTRIEVE.TXT  | Both    | Access    | Btrieve Information                              |
| PERFORM.TXT  | Both    | Access    | Enhancing Performance                            |
| DRVSSRVR.HLP | Both**  | System    | <a href="#">SQL</a> Server Access Information    |
| DRVORACL.HLP | Both**  | System    | Oracle Access Information                        |
| ORACLE.TXT   | Both**  | System    | Oracle Access Information                        |
| ODBCINST.HLP | Both**  | System    | <a href="#">ODBC</a> Driver Manager Install Help |

\* Note: the README.TXT installed by the 1.1 upgrade is different than that supplied with the Retail package.

\*\* Note: these files are only installed if you run the ODBC setup that comes on its own disk with both the upgrade and retail packages.

\*\*\* If the Microsoft Access setup detects that you do not have the

most current version of the Windows for Workgroups driver, setup will stop and you will be referred to the file WFWDRV.TXT. This file is on Disk 1 (retail or upgrade) and is not copied to your hard drive.

## [References](#)

## PRB: Minimizing Popup/Modal Form Loses Application's Focus

Article Number: Q103988  
CREATED: 02-SEP-1993  
MODIFIED: 02-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

The Microsoft Access window may lose focus and another application may get focus, if you run the minimize [macro](#) action on a [form](#) that has both the Popup and Modal properties set to Yes.

### CAUSE

---

When a popup form is minimized, Windows tries to activate another popup form in the active window. Since the form is also [modal](#), Microsoft Access' main window is disabled and cannot be activated. Thus Windows has no choice, but to activate the next available application.

### RESOLUTION

---

Change the Modal property of the form to No, then when you minimize the form Microsoft Access' main window will get activated.

### MORE INFORMATION

---

Steps to Reproduce Behavior

- 
1. Open the sample [database](#) Northwind Traders (NWIND.MDB)
  2. Open the "Forms Switchboard" form in design mode.
  3. Change the Modal Form Property to Yes.
  4. Change the Popup Form Property to Yes.
  5. Close the "Forms Switchboard" form and save changes.
  6. Open the "Forms Switchboard Buttons" macro in design mode.
  7. Add the Minimize Action as the second command in the "Categories" Macro Name, after the OpenForm Action.
  8. Open "Forms Switchboard" form and click the Categories button.
  9. The "Categories" form is minimized and the next application on the Microsoft Windows desktop now has the focus.

### [References](#)



## INF: Microsoft Access and Scanners

Article Number: Q103631  
CREATED: 26-AUG-1993  
MODIFIED: 26-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

This is a general article about Microsoft Access, Scanners and Scanning Software.

### MORE INFORMATION

---

To scan an image into Microsoft Access, a scanner and scanning software are required. Below is a list of scanners that are currently supported by Windows.

| Scanner      | Manufacturer    | Phone number   |
|--------------|-----------------|----------------|
| HP ScanJet   | Hewlett Packard | (208) 323-2551 |
| ScanMan      | Logitech        | (510) 795-0427 |
| Hand Scanner | Marstek         | (714) 833-7740 |

The HP ScanJet provides TWAIN compliance (the industry standard protocol for scanning and acquiring graphics from software applications).

As long as the software that comes with the scanner can save in a format that a Windows graphics program recognizes, Microsoft Access can display that image.

Below is a list of Graphics software packages and the formats that they currently support.

| Software   | Graphic Formats     | Phone number   |
|------------|---------------------|----------------|
| Image-In   | .BMP .EPS .TIF      | (800) 345-3540 |
| CorelDRAW  | .BMP .EPS .TIF .PCX | (613) 728-8200 |
| Paintbrush | .BMP .PCX .MSP .DIB | (206) 426-9400 |

Image-In is a CAD graphics software package that ships with the Marstek Hand Scanner.

#### Steps to Reproduce Behavior

-----

The following is a [method](#) to bring an image into Microsoft Access from a scanner by using Paintbrush:

1. Scan the image and save it to a .BMP format through the scanner software.

2. Open the Employees form in the sample database Northwind Traders (NWIND.MDB).
3. Select Data Entry form the Records menu.
4. Select the Photo field.
5. Select Insert Object from the Edit menu.
6. Select the File button in the Insert Object dialog box.
7. Select your file from the list.
8. The image will display in the photo field

References:

"Microsoft Access User's Guide," version 1.0, Chapter 13, "Using Pictures, Graphs, and Other Objects," pages 316-339

[References](#)

## INF: Using Data From Access In A Word For Windows Print Merge

Article Number: Q103629  
CREATED: 26-AUG-1993  
MODIFIED: 26-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Word for Windows Versions 1.1 1.1a 2.0 2.0a 2.0b 2.0c
- 

### Summary:

---

There is currently no Microsoft Word for Windows file converter capable of reading Microsoft Access [database](#) files. In order to use data from a Microsoft Access database in a Microsoft Word for Windows print merge, the data must first be exported from Microsoft Access in a format that Microsoft Word for Windows recognizes.

Microsoft Access version 1.1: Export the data using the new "Word for Windows Merge" format.

Microsoft Access version 1.0: The preferable [method](#) is to export the data using the "dBASE IV" format.

Note: It is NOT recommended that the user use the "Text (Delimited)" format for exporting data to Word for Windows. See information provided below.

The following information discusses, in detail, how to use both methods described above to export data for using in a Microsoft Word for Windows print merge.

The following topics are discussed below (in order):

- Exporting Data From Microsoft Access for use in a Print Merge
- Using the Exported Data in Microsoft Word for Windows
- Problems with Exporting Data Using the "Text (Delimited)" Format
- Microsoft Access 1.1 FieldName Conversion

### More Information:

---

Exporting Data From Microsoft Access for use in a Print Merge  
-----

1. Open the database in Microsoft Access containing the [table](#) to be used in a Microsoft Word for Windows print merge.
2. Choose Export from the File menu.

In Microsoft Access 1.1: Choose "Word for Windows Merge" from the Data Destination list.

In Microsoft Access 1.0: Choose "dBASE IV" from the Data

Destination list.

3. Choose the OK button.
4. Select the table to be exported from the table list, and choose OK.

Tip: Access only shows you a list of available tables to export. If you want to export a subset of the data, for example, just the records for a particular range of dates or just certain fields, you can create a [query](#) to get the results and export the results to a print merge data file. For information on how to export the results of a query to a file, query on the following keywords:

access and export and query and transfer

5. Enter a unique file name in the File Name box. Choose the appropriate destination directory in the Directory box. Choose the OK button.

Note: Microsoft Access will propose to name the same as your table name with a TXT extension in the Access directory. You may want to change the directory to your Microsoft Word for Windows directory and give the file a DOC extension to make it easier to find in Word.

Microsoft Access 1.0 will propose to name the same as your table name with a DBF extension in the Access directory. When you choose the OK button, Microsoft Access will immediately create the data file containing the data in dBASE IV format.

6. In the Export Options dialog (available only with the "Word for Windows Merge" format) you can choose any custom options. These options determine how dates, times, and numbers are formatted in the data file. The defaults are usually appropriate.

When you choose the OK button, Microsoft Access creates the data file containing all the data from your table.

#### Using the Exported Data in Microsoft Word for Windows

---

1. Open the main document (letter), or create one, in Microsoft Word for Windows, to be used to print merge data into.
2. Choose Print Merge from the File menu.
3. Choose the Attach Data File button.
4. From the file list, select the exported file and press OK.

Note: You may have to type "\*.TXT" or "\*.DBF" in the File Name box to see text or dBASE files in the file name list.

Now you are free to use the merge features of Microsoft Word for Windows. You can begin inserting merge fields into the document using the Insert Merge Fields button on the merge tool bar. Consult your Microsoft Word for Windows documentation for information on

how to use these features.

#### Problems with Exporting Data Using the "Text (Delimited)" Format

---

One popular method of exporting data to Microsoft Word for Windows is using the "Text (Delimited)" format. However, exporting the data in this format may not work correctly without some extra editing work required in Microsoft Word for Windows.

The problem is that there are a number of restrictions placed on [field](#) names used in a Microsoft Word for Windows print merge, which include the following: field names cannot contain spaces; field names cannot begin with a number; field names cannot be longer than 20 characters in length; field names cannot contain special characters.

Microsoft Access field names do not have these restrictions. Consequently, it is possible to export data from Microsoft Access that contains field names incompatible with Microsoft Word for Windows.

Using the "dBASE IV" or "Word for Windows Merge" formats, Microsoft Access will perform field name conversion to ensure that the data files will work correctly with Microsoft Word for Windows.

#### Microsoft Access 1.1 FieldName Conversion

---

As discussed previously in "Problems with Export Data Using the "Text (Delimited)" Format", Microsoft Word for Windows places a number of restrictions on the naming conventions of fields used in a print merge. Since Microsoft Access is not limited by these restrictions with its own field names, it must perform field name conversions when exporting the data to a print merge data file.

Microsoft Access performs field name conversions using the following rules:

1. Any spaces or illegal characters are converted to underscores.
2. If the first character in the field name is a number or an underscore, Microsoft Access adds the prefix "m\_." For example, 1Field becomes m\_1Field, and \_Field2 becomes m\_\_Field2.
3. Microsoft Access truncates any character after the twentieth character.
4. If the conversion results in duplicate field names, Microsoft Access adds a number to the end of each duplicate name (replacing the final character, if the field name contains twenty characters). For example, the field names Order#, Order\$, and Order% become Order\_, Order\_1, and Order\_2.

References:

[References](#)

## INF: Creating mailing labels for Dot Matrix printers

Article Number: Q103433  
CREATED: 24-AUG-1993  
MODIFIED: 25-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

Creating labels to print on continuous feed paper or dot matrix printers most often requires a "User-Defined" page size. The reason for this is, like all Windows applications, Microsoft Access prints in whole pages and the default page length of 11 inches may not be evenly divided into the various label sizes.

### More Information:

---

Below is a guideline when creating a "User-Defined" page size from Control Panel/Printers/Setup plus the settings for the report layout and File/Print Setup properties in Microsoft Access.

The User-Defined PageSize is specified by 100ths of an inch and has a default setting of 1 inch by 1 inch or 100 by 100.

Note: Toshiba is one known print driver where the "User-Defined" settings are not read, otherwise all dot matrix print drivers allow this type of customized page setting.

```

 The width setting for the User-Defined
 PageSize in Control Panel is the width of all
 of the labels to be printed across the page.

The top of label1 to { |
the top of label2 is { |
the Height setting { |
for the User-Defined { |
PageSize under Control { |
Panel/Printers/Setup. { |
 { -----
 { |
 { |
 { -----
 |
 | [The controls should be located in the
 | [middle of the detail section to allow
 | [room for the margins around each label
 | [
 |
 |
The space between { -----
labels is the Row- { |
Spacing setting under { |
```



## PRB: Invalid Use of Null or #Error in User Defined Function

Article Number: Q103432  
CREATED: 24-AUG-1993  
MODIFIED: 30-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

A user defined [function](#) may return "Invalid Use of [Null!](#)" or #Error.

### CAUSE

---

If the datatype is explicitly defined as any type other than a [Variant](#), Microsoft Access is unable to compare the [variable](#) to Null.

### RESOLUTION

---

Define the variable as a Variant which can be Date/Time, floating-point number, or [string](#).

More Information  
-----

If the following function is used to evaluate a [field](#) that may return a Null value, the function will return the error "Invalid Use of Null!" or the #Error if the field is Null.

```
Function IsFieldNull (MyValue As String)
 If IsNull(MyValue) Then
 IsFieldNull = "The field is null!"
 Else
 IsFieldNull = "The field is not null!"
 End If
End Function
```

Defining the variable MyValue as a Variant by using the [method](#) in the following function will prevent the error "Invalid Use of Null!" and #Error even if the field is Null.

```
Function IsFieldNull (MyValue As Variant)
 If IsNull(MyValue) Then
 IsFieldNull = "The field is null!"
 Else
 IsFieldNull = "The field is not null!"
 End If
End Function
```

References:

"Introduction to Programming", Chapter 5

## [References](#)

## PRACC9308: Printer Fonts Cause Text to Shift Down and Right

Article Number: Q103431  
CREATED: 24-AUG-1993  
MODIFIED: 26-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

The first page of a **report** prints fine, but the text shifted down and to the right on the other pages. This shift occurs when you use printer fonts.

### CAUSE

---

The problem is related to the GENDRV.DLL file. Any printer driver that uses this .DLL file creates the same distortion when using any printer fonts.

The following printers use the GENDRV.DLL file:

Generic Printer Driver-GENDRV.DLL (GENDRV.EXE) 06/24/93 version (1.68)

-----

|                            |              |
|----------------------------|--------------|
| Canon LBP 8-III            | (CANON.EXE)  |
| Digital DECLaser 1100      | (DEC1.EXE)   |
| Digital DECLaser 2100/Plus | (DEC1.EXE)   |
| Digital DECLaser 2200/Plus | (DEC1.EXE)   |
| Digital DECLaser 3200      | (DEC1.EXE)   |
| Digital LA70               | (DEC1.EXE)   |
| Digital LA75               | (DEC1.EXE)   |
| Digital LA75 Plus          | (DEC1.EXE)   |
| Digital LA324              | (DEC1.EXE)   |
| Digital LJ250/252          | (DEC1.EXE)   |
| Digital LN03/Plus          | (DEC1.EXE)   |
| IBM Laser Printer 4029     | (IB4029.EXE) |
| Olivetti DM 124C           | (OLIVE.EXE)  |
| Royal CJP 450              | (OLIVE.EXE)  |
| Seiko CH 4104              | (SEIKO.EXE)  |
| Seiko CH 5504              | (SEIKO.EXE)  |
| Seiko CH 5514              | (SEIKO.EXE)  |

NOTE: The IBM4029 is the only printer driver certified by Microsoft Compatibility Labs (MCL). All other printers using the GENDRV.DLL file have been neither tested nor certified by MCL.

### RESOLUTION

---

Use TrueType fonts or, for the IBM4029 printer, use the the most current Microsoft Windows drivers for this printer, which are available by calling Lexmark at (606) 232-3000. These drivers are also available from the Lexmark bulletin board at (606) 232-5238.

The printer driver for Windows version 3.0 is called 29W30\_PS.EXE; the printer driver for both Windows 3.0 and 3.1 is called 29W3X\_PP.EXE. A README.DOC file with installation instructions is included with each driver.

## STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### Steps to Reproduce Problem

-----

To duplicate this problem using the sample [database](#) NWIND.MDB, do the following:

1. Open the List of Products by Category report in [Design view](#).
2. From the File menu, choose Print Setup. Change the print driver to Canon LBP 8-III or to one of the print drivers listed above.
3. Select all controls in the Page Header section.
4. From the Edit menu, choose either Dutch Scalable or any printer font.
5. Print two pages of the report.

On the first page, the page [header](#) controls will be within the lines; on the second page, the controls will be below and to the right of the lines.

The IBM4029 printer driver is manufactured by Lexmark and the other printer drivers by other vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

### [References](#)

## PRB: Slow reports or large temporary print files

Article Number: Q103430  
CREATED: 24-AUG-1993  
MODIFIED: 24-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

Customers report that they are running out of temporary disk space when printing forms and report, or they are often frustrated by reports printing slowly and are looking for reasons to explain why.

### CAUSE

---

Forms and Reports that contain lines, bordered fields, or opaque fields (basically any graphic objects) will print much more slowly than the same Form or Report w/o the graphic objects. (This is true in general with all Windows applications.)

Because lines, bordered fields, or opaque fields (basically any graphic objects) are treated as bitmaps, a lot of data is being sent to the printer. This results in large temporary files needed to print a report.

If the under Control Panel/Printers an incorrect or low setting for printer memory is specified then Windows will "band" complex print jobs. That means it will send a small chunk of graphics and clear it after the printer has processed it. Also, since it can band graphics, but not text, it will send the TrueType fonts as graphics when it does this, the files will be large

The printing speed decreases or the temporary files are larger, when printing Landscape.

This is mainly due to the problem that 1.0 & 1.1 have with printing lines as graphics. It turns out that printing a line as a graphic isn't too bad if it's a HORIZONTAL line, because of the fact that graphics are sent to the printer a row of pixels at a time.

But a VERTICAL line is very expensive, because a fairly big graphics record gets sent down for each pixel in the line. When you print it landscape, these HORIZONTAL lines become VERTICAL lines as far as the printer is concerned.

### References

## PRB: Troubleshooting Tips, Reports with "Out of Memory" Errors

Article Number: Q103429  
CREATED: 24-AUG-1993  
MODIFIED: 20-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

You receive an "Out of Memory" error message immediately after you attempt to run your report.

### CAUSE

---

Reports create queries, combined with the underlying record sources, for each section of the report: report header, page header, group header, detail section, group footer, page footer, and report footer.

All queries are combined into a segmented virtual table (SVT). The final output, or SQL string, must be compiled within a 64K limit. When this limit is reached, the above error may occur.

### RESOLUTION

---

Consider the following as potential solutions to the problem:

- Shorter table names, column names, and control names. Reducing a 30-character name to the minimum length may help.
- Fewer expressions in underlying queries. Reducing space used for expressions in the select list helps to avoid this error. If at all possible, place the expressions directly in the report.

The following expression

```
Query: MyData: IIf(IsNull([Field1]), 0, [Field1])
Report: Sum([MyData])
```

could easily be moved to the report, as follows:

```
Query: MyData: Field1
Report: Sum(IIf(IsNull([Field1]), 0, [Field1])
```

- Fewer stacked query objects. Avoid situations in which Query1 is used to pull data from Table2, where Query2 filters the data. Pulling information together into one query is preferable to having multiple queries each doing portions of the tasks.
- No fields in the query that are not used in the final report output.

- Not basing main reports and their subreports on the same queries.  
Look for stacked query objects and for tables that are unnecessary to the subset.

## [References](#)

## INF: How to Use Graphic Images As Wallpaper For Forms

Article Number: Q103428  
CREATED: 24-AUG-1993  
MODIFIED: 24-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

You can use graphic images as "wallpaper" for your forms to make them more attractive. "Wallpaper", in this case, refers to a background graphic image underlying controls on a [form](#). The steps below give you a [method](#) to accomplish this.

### MORE INFORMATION

---

The basic method of creating wallpaper for a form is:

- Create or open the graphic image.
- Copy it to the clipboard.
- Paste it onto your form.
- Force the image to the the background using the Send to Back command from the Layout menu.

Below is a detailed example of how to do this:

1. Start the Windows Paintbrush program (or other graphics program of your choice) and open the WINLOGO.BMP [bitmap](#) file, or use a different bitmap of your choosing.
2. If using Paintbrush, select the Pick tool (a pair of open scizzors below a dotted rectangle) from the [toolbox](#). Use it to select the bitmap image by using the mouse to place the crosshairs in the upper left corner of the bitmap, then click and hold the left mouse button and drag the rectangle to the lower right corner to encompass the entire image.
3. Choose Copy from the Edit menu. This places a copy of the bitmap image into the Windows clipboard.
4. Start Microsoft Access, and open the Northwind Traders sample [database](#) (NWIND.MDB).
5. Open the Customers form in [Design view](#). Select the Form by clicking the small, white box in the upper left corner of the form.
6. Choose Paste from the Edit menu. This pastes the bitmap image into an object frame [control](#) on the form.

Notice that the graphic image does not cover the entire form, but only the upper left corner of it. This is because this particular graphic image is too small for the form. To cover the entire form, you must either use a larger graphic image, or use a different graphics program to alter the graphic's size.

7. Since the object frame overlaps existing controls, you want to move it into the background of the form. To do this, choose Send to Back from the Layout menu.
8. On this particular form, the BackStyle property for all the label controls on the form are set to Clear. In order to make them more readable against the new background, change their BackStyle properties to Normal. You can also change their BackColor properties, if you choose, using the Color Palette.

#### REFERENCES

=====

"Inside Microsoft Access," April 1993, Vol. 1, No. 2, pages 1-5.

[References](#)

## INF: Another Method of Quitting Windows

Article Number: Q103405  
CREATED: 23-AUG-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

While it is possible to quit Microsoft Windows from within a Microsoft Access program by making use of the Windows application programming interface (API) [function](#) `ExitWindows()`, Microsoft Access will not do a complete cleanup and can leave .TMP files in your TEMP subdirectory.

### More Information:

---

The function below gives you another way to accomplish the same task as above and still allow Microsoft Access to save changes prior to exiting Windows, thus avoiding the generation of .TMP files.

Enter the function below and call it whenever you want to quit Microsoft Windows:

```
Function QuitWindows ()
 AppActivate ("Program Manager")
 SendKeys "%{F4}"
End Function
```

The `QuitWindows()` function activates the Windows Program Manager, then sends the keystrokes to its menu necessary to quit Windows.

### [References](#)

## INF: How to Use the Status Bar Progress Meter

Article Number: Q103404  
CREATED: 23-AUG-1993  
MODIFIED: 24-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

The following information documents the use of the SysCmd() [function](#) in Microsoft Access to display a status meter. When you perform an operation that has a known duration, or number of steps, you can use SysCmd() to let the user know how far along a process is.

This article assumes that you are familiar with [Access Basic](#). For more information on Access Basic, please refer to the "Introduction to Programming" manual.

### More Information:

---

This function allows the Microsoft Access developer to create a progress meter in the Microsoft Access status bar to give a visual representation of the progress of an operation.

The SysCmd() function [syntax](#) is as follows:

```
SysCmd(action [, text][, value])
```

Where:

action - Numeric [expression](#) identifying the type of action to take.

It can be one of the following:

- 1 - Initialize the progress meter
- 2 - Update the progress meter with the specified value
- 3 - Remove the progress meter

text - String expression identifying the text that will appear left-aligned in the status bar to the left of the progress meter.

value - Numeric expression that controls the display of the meter. This is required when the action is 1 or 2.

When the action is 1, it indicates the maximum number the meter should display, the upper bound value for the meter indicating 100 percent.

When the action is 2, it is used to calculate and [update](#) the percentage complete in progress meter.

Returns: NULL (not used)

#### Initializing the Progress Meter

-----

When the action argument is 1, the value argument is the maximum value of the meter, or 100 percent. By typing the following into the Immediate window and pressing Enter, a progress meter will display with a maximum value of 1000 initialized:

```
? SysCmd(1, "This is my meter!", 1000)
```

#### Updating the Progress Meter

-----

When the action argument is 2, the value argument is used by SysCmd() to calculate the percentage displayed by the meter. Typing the following into the Immediate window will update the progress meter to 25% complete:

```
? SysCmd(2, 250)
```

#### Removing the Progress Meter

-----

When the action argument is 3, the progress meter is removed from the status bar:

```
? SysCmd(3)
```

#### Example: Using SysCmd() in an Access Basic Function

-----

The following function will open the Customers table supplied with the Northwind Traders database (NWIND.MDB) supplied with Microsoft Access. In the Immediate windows, the function will display a list of the Contact Names in the table. As the names are displayed, a meter will progress along the status bar indicating the relative position in the table.

1. Open the sample database NWIND.MDB.
2. Create a new module called "Meter Test".
3. Place the following statement in the (declarations) section:

```
Option Explicit
```

4. Create a new function called Meter ().

```
Function Meter ()
```

```
 Dim MyDB As Database, MyTable As Table
```

```
 Dim Count As Long
```

```
 Dim Progress_Amount As Integer, RetVal As Variant
```

```
 Set MyDB = CurrentDB()
```

```
 Set MyTable = MyDB.OpenTable("Customers")
```

```
 ' Move to last record of the table to get the total
```

```
 ' .. number of records.
```

```

MyTable.MoveLast
Count = MyTable.RecordCount

' Move back to first record
MyTable.MoveFirst

' Initialize the progress meter
RetVal = SysCmd(1, "Reading Data...", Count)

' Enumerate through all the records
For Progress_Amount = 1 To Count
 ' Update the progress meter
 RetVal = SysCmd(2, Progress_Amount)

 ' Print the contact name in the Immediate window
 Debug.Print MyTable![contact name]

 ' Goto the next record
 MyTable.MoveNext
Next Progress_Amount

' Remove the progress meter
RetVal = SysCmd(3)
End Function

```

To run the function, type the following from the Immediate window:

```
? Meter()
```

References:

Access Distribution Kit "The Secrets of Access Wizards" pg 30

[References](#)

## INF: Custom Domain Functions Similar to DFirst(), DLast

Article Number: Q103403  
CREATED: 23-AUG-1993  
MODIFIED: 10-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

By design, DFirst() and DLast() will always return the first and last record that you entered into the underlying table, not the first and last record listed in the sorted Query. This article provides examples of how to write custom domain functions that can be used similar to DFirst() and DLast().

The DStart() and DEnd() custom domain examples in this article will return the first and last records listed in a sorted Query. Also included in the article is the DFix() function, which can be used to overcome a limitation of concatenating variables into criteria strings. Information about the limitations of custom domain functions and how to use DFix() appears at the end of the article.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications with the programming tools provided with Microsoft Access.

For information about other custom domain functions, query on the following words in the Microsoft Knowledge Base:

custom and domain and function

### MORE INFORMATION

---

Create a new module within Microsoft Access and add the following functions with the appropriate declaration section. Each command should be entered on one line.

```
'-----
' GLOBAL DECLARATION
'-----
Option Compare Database
Option Explicit

'-----
' Use DStart() instead of DFirst() to return
' the first sorted record in a domain.
'-----
Function DStart (FieldName As String, DomainName As String, _
 Criteria As String)

 Dim MyDB As Database, Myset As Dynaset
```

```

' ERROR OUT IF THERE IS NO FIELDNAME SENT.
If Len(FieldName) = 0 Then
 MsgBox "You Must Specify a Field name", , "DStart"
 End
End If

' ERROR OUT IF THERE IS NO DOMAIN SENT.
If Len(DomainName) = 0 Then
 MsgBox "You Must Specify a Domain name", , "DStart"
 Exit Function
End If

' NOT SURE WHY YOU ARE LOOKING AT ISNULL HERE, AND NOT LEN
If Not IsNull(Criteria) Then
 MsgBox "Invalid Criteria", , "DStart"
 Exit Function
End If

Set MyDB = CurrentDB()
Set Myset = MyDB.CreateDynaset(DomainName)

'
If Len(Criteria) > 0 Then
 Myset.Filter = Criteria
 Set Myset = Myset.CreateDynaset()
End If

' IF THERE ARE NO RECORDS, RETURN THE NULL, ELSE RETURN THE VALUE
' OF THE FIRST RECORD.
If Myset.EOF Then
 DStart = Null
Else
 Myset.MoveFirst
 DStart = Myset(FieldName)
End If

Myset.Close
MyDB.Close
End Function

'-----
'Use DEnd()instead of DLast() to return
' the last sorted record in a domain.
'-----
Function DEnd (FieldName As String, DomainName As String, _
 Criteria As String)
 Dim MyDB As Database, Myset As Dynaset

 ' ERROR OUT IF THERE IS NO FIELDNAME SENT.
 If Len(FieldName) = 0 Then
 MsgBox "You Must Specify a Field name", , "DEnd"
 Exit Function
 End If

 ' ERROR OUT IF THERE IS NO DOMAINNAME SENT.
 If Len(DomainName) = 0 Then

```

```

 MsgBox "You Must Specify a Domain name", , "DEnd"
 Exit Function
End If

If Not IsNull(Criteria) Then
 MsgBox "Invalid Criteria", , "DEnd"
 Exit Function
End If

Set MyDB = CurrentDB()
Set Myset = MyDB.CreateDynaset(DomainName)

If Len(Criteria) > 0 Then
 Myset.Filter = Criteria
 Set Myset = Myset.CreateDynaset()
End If
If Myset.EOF Then
 DEnd = Null
Else
 Myset.MoveLast
 DEnd = Myset(Fieldname)
End If
Myset.Close
MyDB.Close
End Function

Function DFix (ByVal T, DQuote As Integer)
'-----
' Fixes string arguments that are passed
' to Criteria in domain functions
'
' DQuote should be TRUE or -1 if Double Quotes (") delimit Criteria
' DQuote should be FALSE or 0 if Single Quotes (') delimit Criteria
'
' e.g. this gives an error
' (note the quote (') in the data)
' X="Mike's Diner"
' A=DStart("ID","Clients","Name='" & X & "'")
'
' Use either:
' X=DFix("Mike's Diner",False)
' Or:
' A=DStart("ID","Clients","Name='" & DFix(X,False) & "'")
'-----
Dim P As Integer, OldP As Integer, Q As String * 1

If VarType(T) = 8 Then
 If DQuote = 0 Then
 Q = "'"
 Else
 Q = """"
 End If
 P = InStr(T, Q)
 Do While P > 0
 OldP = P + 2
 T = Left$(T, P) & Q & Mid$(T, P + 1)
 P = InStr(OldP, T, Q)
 End Do

```

```
 Loop
 End If
 DFix = T
End Function
```

#### General Limitations to Custom Domain Functions:

---

1. You can't use Forms!FormName!ControlName or Form.ID in quotes:

DStart("ID","LOG","Name=Form.Name") is not allowed.

Instead, use:

DStart("ID","LOG","Name = '" & DFix(Name,False) & "'")

The DFix function is another custom function, and is included in this article.

2. You must always specify a Criteria, even if it is "" or Null.
3. Do not place Field names or Domain names in square brackets [], i.e. use "First Name" instead of "[First Name]".

NOTE: DFix() can be used anywhere you concatenate criteria strings, such as the FindFirst [method](#) or when building custom [SQL](#) criteria.

#### [References](#)

## INF: Snapshot and Underlying Table Updated Until Snapshot Full

Article Number: Q103402  
CREATED: 23-AUG-1993  
MODIFIED: 26-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

A snapshot, along with its underlying [table](#), is updated until it is completely filled with data. The [link](#) to the underlying table is not severed until you move to the end of the snapshot recordset.

### MORE INFORMATION

---

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

#### Steps To Reproduce Behavior

-----

The [function](#) below adds a [record](#) to a table called "Cat," creates a snapshot from it, then prints the snapshot.

1. In the sample [database](#) NWIND.MDB, make a copy of the Categories table and save it as Cat.
2. Open the Cat table in [Datasheet view](#).
3. Create a new Access Basic [module](#) and add the following function:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function Udate ()
 Dim MyDB As Database, MyTable As Table, MySnap As Snapshot
 Set MyDB = CurrentDB()
 Set MyTable = MyDB.OpenTable("Cat")
 Set MySnap = MyTable.CreateSnapshot()
 MyTable.AddNew
 MyTable.[Category ID] = "TEST"
 MyTable.[Description] = "Still being updated"
 MyTable.Update
 Debug.Print "Category ID Description"
 MySnap.MoveFirst
 Do While Not MySnap.EOF
 Debug.Print MySnap.[Category ID] & "/" & MySnap.[Description]
 MySnap.MoveNext
 End While
End Function
```

```
Loop
End Function
```

4. In the Immediate window, type the following:

```
?Udate()
```

The snapshot now contains the new record.

#### How to Create a Snapshot Without Updated Records

---

It is possible to fill a snapshot with data that does not include updated records. The function below does the following:

- Creates a snapshot from the Cat table.
- Moves to the end of the recordset.
- Adds a record to the table.
- Prints the snapshot.

To illustrate this function, do the following:

1. Open the Cat table in Datasheet view.
2. Delete the record with the Category ID "TEST."
3. Add the following function to the module you created in step 3 of the previous procedure:

```
Function NoUdate ()
 Dim MyDB As Database, MyTable As Table, MySnap As Snapshot
 Set MyDB = CurrentDB()
 Set MyTable = MyDB.OpenTable("CAT")
 Set MySnap = MyTable.CreateSnapshot()
 MySnap.MoveLast
 MyTable.AddNew
 MyTable.[Category ID] = "TEST"
 MyTable.[Description] = "Still being updated"
 MyTable.Update
 Debug.Print "Category ID Description"
 MySnap.MoveFirst
 Do While Not MySnap.EOF
 Debug.Print MySnap.[Category ID] & "/" & MySnap.[Description]
 MySnap.MoveNext
 Loop
End Function
```

4. In the Immediate window, type the following:

```
?NoUdate()
```

Note that the snapshot does not contain the previously added record.

## REFERENCES

=====

"Language Reference," version 1.0, pages 91-92

[References](#)

## INF: Custom Domain Function Similar to DCount

Article Number: Q103401  
CREATED: 23-AUG-1993  
MODIFIED: 10-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

This article provides an example of how to write a custom [domain function](#) that is used similar to DCount. In addition, it provides a function DFix() to overcome limitations involved with concatenating variables in [criteria](#) strings. Information about the limitations of custom domain functions and use of DFix() appears at the end of the article.

Custom domain functions can be used to provide functionality other than what is provided by the standard domain functions DLookup, DMin, DLast, etc.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications with the programming tools provided with Microsoft Access.

For information about other custom domain functions, [query](#) on the following words in the Microsoft Knowledge Base:

custom and domain and function

### MORE INFORMATION

---

The DRecCount custom [domain function](#) is useful when the domain is a totals or aggregate query based on an attached [SQL table](#). The DRecCount custom domain example in this article will accurately return the count of such a [dynaset](#).

Create a new [module](#) within Microsoft Access and add the following functions with the appropriate declaration section. Each command should be entered on one line.

```
'-----
' GLOBAL DECLARATION
'-----
Option Compare Database
Option Explicit

Function DRecCount (FieldName, DomainName, Criteria)
'-----
' Use DRecCount to return a count
' of records when the domain is a query based on a
' totals/aggregate query on an attached SQL table.
```

```

'-----
Dim MyDB As Database, Myset As Dynaset

If VarType(Fieldname) <> 8 Or Len(Fieldname) = 0 Then
 MsgBox "You Must Specify a Field name", , "DRecCount"
 Exit Function
End If
If VarType(DomainName) <> 8 Or Len(DomainName) = 0 Then
 MsgBox "You Must Specify a Domain name", , "DRecCount"
 Exit Function
End If
If VarType(Criteria) <> 8 And Not IsNull(Criteria) Then
 MsgBox "Invalid Criteria", , "DRecCount"
 Exit Function
End If

Set MyDB = CurrentDB()
Set Myset = MyDB.CreateDynaset(DomainName)

If FieldName <> "*" Then
 If Len(Criteria) > 0 Then
 Criteria = Criteria & " AND "
 End If
 Criteria = Criteria & "[" & FieldName & "] Is Not Null"
 Myset.Filter = Criteria
 Set Myset = Myset.CreateDynaset()
End If
If Myset.EOF Then
 DRecCount = 0
Else
 Myset.MoveLast
 DRecCount = Myset.recordcount
End If
Myset.Close
MyDB.Close
End Function

Function DFix (ByVal T, DQuote As Integer)
'-----
' Fixes string arguments that are passed
' to Criteria in domain functions
'
' DQuote should be TRUE or -1 if Double Quotes (") delimit Criteria
' DQuote should be FALSE or 0 if Single Quotes (') delimit Criteria
'
' e.g. this gives an error
' (note the quote (') in the data)
' X="Mike's Diner"
' A=DRecCount("*","Clients","Name='" & X & "'")
'
' Use either:
' X=DFix("Mike's Diner",False)
' Or:
' A=DRecCount("*","Clients","Name='" & DFix(X,False) & "'")
'-----
Dim P As Integer, OldP As Integer, Q As String * 1

```

```

If VarType(T) = 8 Then
 If DQuote = 0 Then
 Q = "'"
 Else
 Q = """"
 End If
 P = InStr(T, Q)
 Do While P > 0
 OldP = P + 2
 T = Left$(T, P) & Q & Mid$(T, P + 1)
 P = InStr(OldP, T, Q)
 Loop
End If
DFix = T
End Function

```

General Limitations to Custom Domain Functions:

- 
1. You cannot use Forms!FormName!ControlName or Form.ID in quotes:

DRecCount("ID","LOG","Name=Form.Name") is not allowed.

Instead, use:

DRecCount("ID","LOG","Name = '" & DFix(Name,False) & "'")

The DFix function is another custom function, and is included in this article.

2. You must always specify a Criteria, even if it is "" or Null.
3. Do not place Field names or Domain names in square brackets [], i.e. use "First Name" instead of "[First Name]".

NOTE: DFix() can be used with other criteria strings, such as with the FindRecord [method](#) or when building custom SQL criteria.

## [References](#)

## INF: Marking Records Having Unmatched or Invalid Values

Article Number: Q103400  
CREATED: 23-AUG-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

A table created from an imported source may contain invalid or erroneous primary data. For instance, the customer name may be misspelled. This article explains how to use an update query which compares an imported table to a reference table and marks the imported records which do not match the reference table field values.

### More Information:

---

Suppose you import a customer order table from an application and find that the records have valid but misspelled customer names. You need to compare names in the imported file with a table of valid names and mark records which do not match the reference table customer names.

In the following example, you have two tables: Reference Table and Imported Table. Both of these tables have a field called Customer Name.

Your tables may look like this :

| Reference Table.Customer Name | Imported Table.Customer Name |
|-------------------------------|------------------------------|
| ACE                           | ACE                          |
| BAT                           | ACE                          |
| CAD                           | ACA                          |
| CAT                           | BAT                          |
|                               | BAR                          |
|                               | CAD                          |
|                               | CAD                          |
|                               | CAT                          |

You can do the following to mark unmatched records in the imported table:

1. Add a field named Matched to the Imported Table. Assign the DataType of Yes/No. The default value is 'No' when this field is created. We will set this field to true if the Customer Name matches a Customer Name in the Reference Table.
2. Create a query based on both tables. Join the tables by dragging the Customer Name field from the reference table to the Customer Name field on the Imported Table. Then, from the Query menu, choose Update.

Query: Mark Unmatched

---

Field name: Customer Name

Table: Imported Table

Criteria: [Reference Table].[Customer Name]

Field name: Matched

Table: Imported Table

Update to: "Yes"

3. Run the query. The matched field should be set to Yes for all records where in the Imported Table which have a Customer Name that matches the Customer Name field of the Reference Table.
4. Close the query and open the Imported Table. You should see the following results:

| Customer Name | Matched |
|---------------|---------|
| -----         | -----   |
| ACE           | Yes     |
| ACE           | Yes     |
| ACA           | No      |
| BAT           | Yes     |
| BAR           | No      |
| CAD           | Yes     |
| CAD           | Yes     |
| CAT           | Yes     |

References:

"Microsoft Access User's Guide," version 1.0, Chapter 7, "Designing Action Queries and Parameter Queries," page 176

[References](#)

## INF: How to Create Tasks in MS Project for Windows Using DDE

Article Number: Q103273  
CREATED: 19-AUG-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

Before data can be sent to Microsoft Project version 3.0 for Windows, a task must exist in the active project. This article contains sample code to start Project, create 10 tasks, and fill the task fields with data.

### MORE INFORMATION

---

This article assumes that the user has some knowledge of [Access Basic](#) and is intended as a demonstration of the technique required to create a task in Project. The following [function](#) is not intended to be replicated exactly.

1. Create a new [module](#) in a Microsoft Access [database](#). Add the following statement to the [Declarations section](#) of the module:

```
'-----
'Declarations Section
'-----
Option Explicit
```

2. Add the following function to the module:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----
'Start of Function
'-----
```

Function dde\_createtasks (ProjectPath)

```
Const DDE_ERROR = 282 'Define DDE return error.
Dim Temp, Mynum, Chan1, Chan2 'Define variables.
```

```
On Error GoTo StartUp 'If Project is not open, start it.
```

```
Chan1 = DDEInitiate("winproj", "system")
'Initiate a channel to Project.
```

```
On Error GoTo QuitNow:
```

```

Chan2 = DDEInitiate("winproj", DDERequest(Chan1, "ActiveProject"))
'Initiate a channel to the active Project.

For Mynum = 1 To 10
 DDEExecute Chan2, "EditGoto .ID=[" & Mynum & "]"
 'Go to the task.
 DDEExecute Chan2, "SetField .Field=[Name] .Value=MyNewTask" &_
 Mynum & " .Create=[Yes]"
 'Create a new task.
 DDEPoke Chan2, "T((" & Mynum & "), (Name,Duration))", "Task_
 Number " & Mynum & Chr(9) & Mynum & "d"
 'Send information to Project.
Next

DDETerminateAll 'Terminate the DDE channel.

Exit Function

QuitNow:

 DDETerminateAll 'Terminate the DDE channel.
 Exit Function

StartUp:

 If Err = DDE_ERROR Then
 Temp = Shell(ProjectPath, 6) 'Start Project.
 Resume ' Re-execute DDEInitiate() function.
 End If

End Function

```

```

'-----
'End of Function
'-----

```

3. To test this function, type the following in the [Immediate window](#)

```
?dde_createtasks ("")
```

where "

Project installs two files, MACRO.WRI and DDEINFO.WRI, in the program directory that contain more information about [macro](#) commands and dynamic data exchange (DDE) functionality. These files can be opened in Microsoft Windows Write.

REFERENCES  
=====

"Introduction to Programming," version 1.0, Chapter 9, "Dynamic Data Exchange"

[References](#)

## PRACC9308: Setup Cannot Recognize Forward Slash in Path

Article Number: Q103272  
CREATED: 19-AUG-1993  
MODIFIED: 20-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

When you are running the Microsoft Access Setup program, the following error message may appear if your path statement contains forward slashes (/):

```
Badly formed Setup Script Command
"CreateIniKeyValue C:/WIN\WINHELP.INI Files CUECARDS.DLL
C:\ACCESS\, Can't find CUECARDS.DLL O"
```

### CAUSE

---

The Setup routine used by Microsoft Access cannot interpret the forward slash (/) as used in the path statement. When Microsoft Access calls for the path [variable](#), the result is C:/WINDOWS. Microsoft Access is unable to append the addenda necessary for the various .INI file entries.

### RESOLUTION

---

Change forward slashes to back slashes (\) in all paths statements and path variables in your AUTOEXEC.BAT and CONFIG.SYS files. For example, change

```
PATH C:/WINDOWS;C:/DOS;
```

to read

```
PATH C:\WINDOWS;C:\DOS;
```

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

Steps to Reproduce Problem

-----

1. Type the following at the [command prompt](#):

```
PATH C:/WINDOWS;C:/DOS;
```

2. Start Microsoft Windows.
3. Install Microsoft Access and start the Setup program.

The error will occur at the end of Setup, when Microsoft Access tries to update the .INI files.

### [References](#)

## INF: Reporting the Median Value of a Group of Records

Article Number: Q103271  
CREATED: 19-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

The median value of a group of records is defined as:

"A value in an ordered set of values below or above which there is an equal number of values or which is the arithmetic mean of the two middle values if there is no one middle number."

Example 1) The median value of the following set of records is 6.

| Name | Age |
|------|-----|
| Jan  | 2   |
| Mark | 3   |
| Eli  | 6   |
| Sue  | 15  |
| Curt | 25  |

Example 2) The median value of the following set of records is 21.

| Pet    | Number |
|--------|--------|
| Snake  | 1      |
| Turtle | 2      |
| Dog    | 40     |
| Cat    | 45     |

### More Information:

---

=====  
This article described the steps to create a report which shows the median Order Amount for each customer in a report. The report is based on the Orders table in the example database NWIND.MDB. The report groups by the field Customer ID and sorts records within each group Ascending by the field Order Amount.

- 1) Create a new report based on the Orders table.
- 2) In the Sorting and Grouping dialog choose Customer ID. Set Group Header to Yes and Group Footer to Yes.
- 3) Select Order Amount as the second field in the Sorting and Grouping dialog. Select Ascending for the Sort Order.

- 4) Drag the field [Customer ID] to the Customer ID Header.
- 5) Drag the field [Order Amount] to the detail section.
- 6) Add the following fields to the Customer ID Header:

| ControlName | Argument            |
|-------------|---------------------|
| -----       | -----               |
| IDCount     | =Count(*)           |
| Half        | =Int([IDCount]/2)   |
| Odd         | =( [IDCount] Mod 2) |

- 7) Add the following controls to the Detail Section:

| ControlName | Argument                                                               |
|-------------|------------------------------------------------------------------------|
| -----       | -----                                                                  |
| Position    | =1                                                                     |
| Arg1        | =IIf([Position]=([Half]+[Odd]), [Order Amount], 0)                     |
| Arg2:       | =IIf([Odd]=0, IIf([Position]=([Half]+<br>[Odd]+1), [Order Amount], 0)) |

Note: The expression for SecondArg takes two lines in this article but should be entered as one.

- 8) Set the RunningSum property to Yes for each of the three controls created in step 7; Position, Arg1, and Arg2.
- 9) Add the following control to the Customer ID Footer:

| ControlName | Argument                                 |
|-------------|------------------------------------------|
| -----       | -----                                    |
| Median      | =IIf([Odd]=0, ([Arg1]+[Arg2])/2, [Arg1]) |

- 10) Save and run the report.

References: Webster's Ninth New Collegiate Dictionary; Merriam Webster, Inc.; Springfield, Massachusetts, USA

[References](#)

## INF: Graphing Current Sales Compared To Accumulated Sales

Article Number: Q103270  
CREATED: 19-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This article described the steps to create a [graph](#) which displays each customer's total sales for all years compared to their sales for just 1992. The data in the graph is also broken out according to the company that the order was shipped via.

This example uses tables in the example [database](#), NWIND.MDB.

### More Information:

---

The goal of this article is to create a graph which displays the following information about each customer:

| Ship Via:    | 1        | 2     | 3     |
|--------------|----------|-------|-------|
| Total Sales: | \$10,000 | 8,500 | 7,800 |
| 1992 Sales:  | \$ 1,500 | 2,500 | 3,400 |

Step 1: Create The Queries.  
-----

- 1) Create a new [query](#) based on the Orders [table](#):

|        |             |          |              |
|--------|-------------|----------|--------------|
| Field: | Customer ID | Ship Via | Order Amount |
| Total: | Group By    | Group By | Sum          |

- 2) Save this query as Total Sales.

- 3) Create a second query based on the Orders table.

|           |             |          |              |            |
|-----------|-------------|----------|--------------|------------|
| Field:    | Customer ID | Ship Via | Order Amount | Order Date |
| Total:    | Group By    | Group By | Sum          | Where      |
| Criteria: |             |          |              | >=#1/1/92# |

- 4) Save this query as 1992 Sales.

- 5) Create a third query based on the queries Total Sales and 1992 Sales which you just created.

- 6) Join on the [field](#) [Customer ID]. Double-click on the [join](#) line and choose the option: "Include ALL records from Total Sales ...".

7) Join on the field [Ship Via]. Double-click on the join line and choose the option: "Include ALL records from Total Sales ...".

8) Drag [Customer ID], [Ship Via], and [SumofOrder Amount] from Total Sales to the query grid. Add the label "Total" to [Sum of Order Amount]:

Total: [SumofOrder Amount]

9) Drag [SumofOrder Amount] from 1992 Sales and add the table "1992":

1992: [SumofOrder Amount]

10) Run the query to verify that it work correctly and then save it as Sales Summary.

#### Step 2: Create The Graph

1) Create a new report based on the table Customers and drag the field Customer ID to the detail section.

2) Create a new graph in the detail section by selecting the Graphing tool from the Toolbox and drawing an unbound object frame.

3) Make the following selections in the Graphing Wizard:

a) Data source: Sales Summary  
Totals: Sum

b) Which field(s) for data and labels: Ship Via, Total, & 1992

c) What labels do you want: Ship Via

d) What labels for the legend: Total & 1992

e) Link the Graph to the Report: Yes

f) Which fields will link: Customer ID = Customer ID

4) Save the report and then preview it. For each customer, the report should display a graph of that customer total sales and 1992 sales per shipping company.

References:

[References](#)

## INF: Updating Summary Information With Data From Second Table

Article Number: Q103269  
CREATED: 19-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This article describes the steps to create a query which will update a record in one table with data from a second table. An example of where this query might be useful is a table which contains the number of inches of rain that have fallen in each region for the the year. The number of inches that fall each day must be added to the accumulated total.

### More Information:

---

Note: These steps will not work if the information being added is the result of a totalling or grouping query. For information on how to update a record with the result of a summation query on:

"Access Update Dlookup"

Step 1: Create the tables  
-----

1) Create a new table called Yearly Rainfall:

| City     | Inches | Last Updated |
|----------|--------|--------------|
| Baroda   | 0.5    | 4/2/93       |
| Basildon | 22.0   | 4/2/93       |
| Beaver   | 18.0   | 4/2/93       |

2) Create a table called Current Week:

| City     | Inches | Week Of |
|----------|--------|---------|
| Baroda   | 0.1    | 4/9/93  |
| Basildon | 2.0    | 4/9/93  |
| Beaver   | 2.5    | 4/9/93  |

Step 2 - Create the Query  
-----

- 1) Create a new query based on the tables Yearly Rainfall and Current Week and join them on the field [City].
- 2) From the Query menu, select Update.

3) Drag the fields [Last Updated] and [Inches] from the Yearly Rainfall table to the query grid.

4) For the field [Last Updated] enter:

Field: [Last Updated]  
Update To: [Current Week].[Week Of]

5) For the field [Inches] enter:

Field: [Inches]  
Update To: [Yearly Rainfall].[Inches]+[Current Week].[Inches]

6) Save and run the query.

7) The data in the table Yearly Rainfall should now be:

| City     | Inches | Last Updated |
|----------|--------|--------------|
| -----    | -----  | -----        |
| Baroda   | 0.6    | 4/9/93       |
| Basildon | 20.0   | 4/9/93       |
| Beaver   | 24.5   | 4/9/93       |

## [References](#)

## INF: How to Create a Dynamic Crosstab Report

Article Number: Q103262  
CREATED: 19-AUG-1993  
MODIFIED: 17-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

You may want to create dynamic reports based on parameter crosstab queries, or have reports to match a [dynaset](#) returned by a such a [query](#). Dynamic reports allow your customized reports to show only the most recently modified data and eliminate the need for fixed [column](#) headings and empty columns.

The example below uses starting and ending dates entered on a [form](#) as the parameters in a [crosstab query](#). When a button on the form is chosen, [Access Basic](#) functions run the crosstab query, which creates a dynaset whose contents are displayed in a [report](#).

In the following example, the report shows which employees had sales for the current week, based on the dates entered on the form.

### MORE INFORMATION

---

The steps below show how to create a dynamic [crosstab report](#) based on tables in the sample [database](#) NWIND.MDB.

The following new objects must be added to the database: one [table](#), one query, one form, one report and two functions. Each item is explained in a separate section below.

TABLE: XTabResult  
-----

This table must contain enough columns to hold the maximum number of fields that the crosstab query can possibly generate. The table is later filled with data by Access Basic functions.

For this example, the maximum number of fields generated by the crosstab query is 10, with no [primary key](#) necessary. Create a new table with 10 columns (columns 0-9) with the Text [data type](#) (for example, Column0, Column1, and so forth).

QUERY: CrossQry  
-----

Create a new crosstab query based on the Employees, Orders, Order Details, and Products tables. (These tables are already joined, based on previously created relationships in the NWIND database.) To do this, use the following steps:

1. Choose the Query button in the Database window, then choose the New button.
2. Add the Employees, Orders, Order Details, and Products tables.
3. Drag the following fields to the query grid and add the values shown below:

NOTE: To display the crosstab field, choose Crosstab from the Query menu.

Field: Order Amount  
 Table: Orders  
 Total: Sum  
 Crosstab: Value

Field: Product Name  
 Table: Products  
 Total: Group By  
 Crosstab: Row Heading

Field: RowTotal:Order Amount  
 Table: Orders  
 Total: Sum  
 Crosstab: Row Heading

Field: Last Name  
 Table: Employees  
 Total: Group By  
 Crosstab: Column Heading

Field: Order Date  
 Table: Orders  
 Total: Where  
 Crosstab:  
 Criteria: Between [Start Date] and [End Date]

If you choose SQL from the View menu, the SQL statement should look like the following:

```
PARAMETERS [Start Date] DateTime, [End Date] DateTime;
TRANSFORM Sum(Orders.[Order Amount]) AS [SumOfOrder Amount]
SELECT Products.[Product Name], Sum(Orders.[Order Amount])
AS RowTotal
FROM Employees, Orders, [Order Details], Products, Orders
INNER JOIN [Order Details]
ON Orders.[Order ID] = [Order Details].[Order ID], Employees
INNER JOIN Orders
ON Employees.[Employee ID] = Orders.[Employee ID], Products
INNER JOIN [Order Details]
ON Products.[Product ID] = [Order Details].[Product ID]
WHERE ((Orders.[Order Date] Between [Start Date] And [End Date]))
GROUP BY Products.[Product Name]
PIVOT Employees.[Last Name]
WITH OWNERACCESS OPTION;
```

4. In the Query Parameters box, add two parameters with the same data type, as follows:

| Parameter    | DataType  |
|--------------|-----------|
| [Start Date] | Date/Time |
| [End Date]   | Date/Time |

FORM: XTabSample

1. Create an unbound form. Add two unbound text box controls with the following properties:

(Text\_Box\_1) ControlName: Start Date  
(Text\_Box\_2) ControlName: End Date

2. Add a button to the form with the following properties:

Caption: "Print Report"  
OnPush: =XTabPrint()

REPORT: CrossReport

1. Create a bound report with Page Header and Detail sections and the following RecordSource property:

RecordSource: XTabResult

2. In the page header, create 10 tabular text box controls, without labels, and change the ControlSource property of each to one of the matching functions below:

=GetPageHdr(0)   =GetPageHdr(1)   =GetPageHdr(2)   =GetPageHdr(3)  
=GetPageHdr(4)   =GetPageHdr(5)   =GetPageHdr(6)   =GetPageHdr(7)  
=GetPageHdr(8)   =GetPageHdr(9)

For example, the ControlSource property of the first text box should be set to =GetPageHdr(0), the second to =GetPageHdr(1), and so forth.

3. In the Detail section, create 10 tabular text box controls without labels. Bind each of these controls to one of each of the fields in the XTabResult table:

[Column0] [Column1] [Column2] [Column3] [Column4] [Column5]  
[Column6] [Column7] [Column8] [Column9]

MODULE: <Any Name>

Create a new module with the following Access Basic code and call it any name you choose:

Option Compare Database   'Use database order for string comparisons.  
Option Explicit

```
'=====
' Global variables
'=====
```

```
Dim MyFields() ' For holding field names.
Dim nColumns As Integer ' For holding the number of columns.
```

```
'=====
' GetPageHdr Function
'=====
```

```
Function GetPageHdr (col)
 If (col < nColumns) Then
 GetPageHdr = MyFields(col)
 Else
 GetPageHdr = ""
 End If
End Function
```

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'=====
' XTabPrint Function
'=====
```

```
Function XTabPrint ()
 Dim MyDB As Database, MyTable As Table
 Dim MyDyna As Dynaset, MyQueryDef As QueryDef
 Dim MySnap As Snapshot, i As Integer

 ' Create a dynaset from the query.
 Set MyDB = CurrentDB()
 Set MyQueryDef = MyDB.OpenQueryDef("CrossQry")
 MyQueryDef![Start Date] = Forms![XTabSample]![Start Date]
 MyQueryDef![End Date] = Forms![XTabSample]![End Date]
 Set MyDyna = MyQueryDef.CreateDynaset()
 MyQueryDef.Close

 ' Get field information and store the field names.
 Set MySnap = MyDyna.ListFields()
 MySnap.MoveLast
 MySnap.MoveFirst
 nColumns = MySnap.RecordCount
 ReDim MyFields(nColumns)
 i = 0
 While Not MySnap.EOF
 MyFields(i) = MySnap!Name
 i = i + 1
 MySnap.MoveNext
 Wend
 MySnap.Close

 ' Delete the contents of the XTabResult table.
 Set MyTable = MyDB.OpenTable("XTabResult")
```

```

While Not MyTable.EOF
 MyTable.Delete
 MyTable.MoveNext
Wend

' Dump the dynaset into the XTabResult table.
' Table should have a column called COLUMN# for each column in
' the crosstab dynaset.
While Not MyDyna.EOF
 MyTable.AddNew
 For i = 0 To nColumns - 1
 MyTable("Column" & i) = MyDyna(MyFields(i))
 Next
 MyTable.Update
 MyDyna.MoveNext
Wend
MyTable.Close
MyDyna.Close

' Print the report.
DoCmd OpenReport "CrossReport", A_PREVIEW
End Function

```

#### Choosing a Date Range

-----

After you create the new [database objects](#) specified above, you will be able to open the XTabSample form and enter starting and ending dates on the form. A recommended data range is 1/1/91 through 12/21/92. However, if you alternate short date ranges with long date ranges, you will see how the report dynamically changes to fit the data.

After entering the date range, choose the Print Report button on the form to display your dynamic report.

#### REFERENCES

=====

"User's Guide," version 1.0, pages 154-159

#### [References](#)

## INF: How to Remove the Caption Bar from a Form

Article Number: Q103261  
CREATED: 19-AUG-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

Microsoft Access provides no built-in mechanism for removing the caption bar from a [form](#). The caption bar runs across the top of the [Database window](#) and includes the system menu and the minimize and maximize buttons.

This article includes an [Access Basic function](#) called `RemoveCaptionBar()`, which allows you to remove the caption bar from a form. Procedures for creating a form with no caption bar also follow.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

### MORE INFORMATION

---

Creating a New Module to Delete Caption Bar  
-----

NOTE: In the following sample code, an underscore (`_`) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

NOTE: You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive the duplicate procedure name error message when you compile or run your code, remove or comment out the appropriate declarations statement.

```
!*****
' MODULE DECLARATION SECTION
!*****
```

Option Explicit

```
' Windows API Rect structure
Type RECT
 x1 As Integer
 y1 As Integer
 x2 As Integer
 y2 As Integer
End Type
```

```

' Windows API Declarations
Declare Function SetWindowLong& Lib "User" (ByVal hwnd%, _
 ByVal index%, ByVal style&)
Declare Function GetWindowLong& Lib "User" (ByVal hwnd%, _
 ByVal index%)
Declare Function GetSystemMetrics% Lib "User" (ByVal iIndex%)
Declare Function GetWindowRect% Lib "User" Alias "GetWindowRect" _
 (ByVal hwin%, rectangle As RECT)
Declare Function MoveWindow% Lib "User" Alias "MoveWindow" _
 (ByVal hwin%, ByVal x%, ByVal y%, ByVal dx%, _
 ByVal dy%, ByVal fRepaint%)

' GetWindowLong Constants
Const GWL_STYLE = -16

' Window Styles Constants
Global Const WS_CAPTION = &HC00000

' System Metrics Constants
Global Const SM_CYCAPTION = 4

'*****
' FUNCTION: RemoveCaptionBar
'
' PURPOSE:
' Removes the caption bar, which includes the system menu
' and minimize and maximize buttons, from a form.
'
' ARGUMENTS:
' F - The form object from which to remove the caption bar.
'
' USAGE:
' Enter the following for the OnOpen property of a form called
' MyForm:
'
' =RemoveCaptionBar(Forms![MyForm])
'
' SIDE AFFECTS:
' With a normal form, the form opens slightly lower than
' expected. With a dialog form (PopUp=Yes, Modal=Yes), the form
' opens slightly higher than expected. Also, you cannot close the
' form using the Microsoft Access menu; you must close it
' programmatically.
'*****
Function RemoveCaptionBar (F As Form)
 Dim OldStyle As Long, NewStyle As Long
 Dim R As RECT
 Dim RetVal As Integer
 Dim dx As Integer, dy As Integer

 ' Get the current window style of the form.
 OldStyle = GetWindowLong(F.hwnd, GWL_STYLE)

 ' Set the window style to be the same, minus the caption bar.
 NewStyle = OldStyle And Not WS_CAPTION

```

```

'Change the window style of the form.
OldStyle = SetWindowLong(F.hwnd, GWL_STYLE, NewStyle)

' The caption is removed, but the border has not snapped down
' .. onto the smaller window. We will call the Windows API to
' .. move and size the window to the same location and size,
' .. with the window shortened by the length of the caption bar.
RetVal = GetWindowRect(F.hwnd, R)
dx = R.x2 - R.x1
dy = R.y2 - R.y1 - GetSystemMetrics(SM_CYCAPTION)
RetVal = MoveWindow(F.hwnd, R.x1, R.y1, dx%, dy%, True)
End Function

```

#### Creating a Form with No Caption Bar

-----

The following instructions show how to create a form that does not have a caption bar:

1. Create or open a [database](#) in Microsoft Access.
2. Create a new, blank, unbound form and save it with the name NoCaption.
3. Change the OnOpen property of the form to read:

```
=RemoveCaptionBar(Forms![NoCaption])
```

4. From the View menu, choose Form.

#### [References](#)

## INF: How to Get Record Position Number from Record Selector

Article Number: Q103260  
CREATED: 19-AUG-1993  
MODIFIED: 14-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

Microsoft Access does not provide a way to retrieve the position number found in the [record selector](#) at the bottom of the [form](#) or [datasheet](#).

This article includes an [Access Basic function](#) called GetRecordPosition(), which can be used to retrieve the [current record](#) position. This article also explains how to use the GetRecordPosition() function to create your own custom record position box on a form.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

### MORE INFORMATION

---

The number that appears in the record selector is actually not a record number. It is a number used to indicate the record's relative position in the current record set, not the record's natural position in the [table](#).

The following Access Basic function, GetRecordPosition(), can be used to get the record's relative position in the record set. Add the this function to a new or existing [module](#).

```
'Global Declaration Section
Option Explicit
```

```

' FUNCTION: GetRecordPosition
'
' PURPOSE:
' Gets the number representing the relative position of the
' record in the currently active form. This is the number that
' appears in the record selector found at the bottom of a form
' or datasheet.
'
' RETURN:
' The relative record position. If there are no records in
' the active form's dynaset, or if there is no active form,
' Null is returned.
```

```

'
'*****
Function GetRecordPositions ()
 Const ERR_NO_CURRENT_RECORD = 3021
 Dim MyDynaset As Dynaset
 Dim RecordPos As Integer

 On Error GoTo Err_Handler

 Set MyDynaset = Screen.ActiveForm.Dynaset
 MyDynaset.Bookmark = Screen.ActiveForm.Bookmark
 RecordPos = 0

 While Not MyDynaset.BOF
 MyDynaset.MovePrevious
 RecordPos = RecordPos + 1
 Wend

 GetRecordPositions = RecordPos
 Exit Function

Err_Handler:
 Select Case Err
 Case ERR_NO_CURRENT_RECORD
 GetRecordPositions = RecordPos
 Exit Function
 End Select
End Function

```

#### Creating a Record Position Box on a Form -----

Use the following steps to create your own custom record position box on a form using the GetRecordPosition() function:

1. Create a new [macro](#) called Refresh Record Position with the following macro action:

```

Requery
 ControlName: Record Position

```

2. Open your form in [Design view](#). Make sure that the Form Hdr/Ftr option is selected on the Layout menu.

3. Create a [text box](#) in the [footer](#) with the following properties:

```

ControlName: Record Position
ControlSource: =GetRecordPosition()

```

4. Set the following properties on the form (not on the [control](#)):

```

OnCurrent: Refresh Record Position
AfterUpdate: Refresh Record Position

```

The current record position will display in the text box in the footer of the form either when you navigate to a different record or when a new record is created.

NOTE: The AfterUpdate property of the form is set in the event a new record is created and written to disk. The current record position may need to account for this new record.

#### REFERENCES

=====

For more information on record numbers, query on the following words here in the Microsoft Knowledge Base:

access and record and numbers and position

[References](#)

## PRACC9308: Problem Exporting "Single" Numeric Data to MS Excel

Article Number: Q103259  
CREATED: 19-AUG-1993  
MODIFIED: 27-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

You may be unable to export numeric data to Microsoft Excel if the Microsoft Access data has a Number data type, a Single field size, and a fixed format. When you export the table, the numbers lose their formatting and are altered slightly.

### RESOLUTION

---

Format the data stored as Number-Single in Microsoft Access as "fixed - 2 decimal places" in Microsoft Excel. All the data stored as Number-Single in Microsoft Access is also stored as Number-Double.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

Steps to Reproduce Behavior  
-----

1. Create a new table with two fields:

```
FieldName: NumSingle
Data Type: Number
Field Size: Single
Format: Fixed
Decimal Places: 2
```

```
FieldName: NumDouble
Data Type: Number
Field Size: Double
Format: Fixed
Decimal Places: 2
```

2. Add the following data:

| NumSingle | NumDouble |
|-----------|-----------|
| -----     | -----     |
| 0.00      | 0.00      |

|      |      |
|------|------|
| 0.01 | 0.01 |
| 0.02 | 0.02 |
| 0.03 | 0.03 |
| 0.04 | 0.04 |
| 0.05 | 0.05 |
| 0.06 | 0.06 |
| 0.07 | 0.07 |
| 0.08 | 0.08 |
| 0.09 | 0.09 |
| 0.10 | 0.10 |
| 0.10 | 0.10 |
| 0.11 | 0.11 |
| 0.12 | 0.12 |
| 0.13 | 0.13 |
| 0.14 | 0.14 |
| 0.15 | 0.15 |
| 0.16 | 0.16 |
| 0.17 | 0.17 |
| 0.18 | 0.18 |
| 0.19 | 0.19 |
| 0.20 | 0.20 |

3. Export the file to Microsoft Excel and open it.

The data appears as follows:

| NumSingle   | NumDouble |
|-------------|-----------|
| -----       | -----     |
| 0           | 0.00      |
| 0.01        | 0.01      |
| 0.02        | 0.02      |
| 0.029999999 | 0.03      |
| 0.039999999 | 0.04      |
| 0.050000001 | 0.05      |
| 0.059999999 | 0.06      |
| 0.07        | 0.07      |
| 0.079999998 | 0.08      |
| 0.090000004 | 0.09      |
| 0.100000001 | 0.10      |
| 0.100000001 | 0.10      |
| 0.109999999 | 0.11      |
| 0.119999997 | 0.12      |
| 0.129999995 | 0.13      |
| 0.140000001 | 0.14      |
| 0.150000006 | 0.15      |
| 0.159999996 | 0.16      |
| 0.170000002 | 0.17      |
| 0.180000007 | 0.18      |
| 0.189999998 | 0.19      |
| 0.200000003 | 0.20      |

Numbers stored as "singles" are imported as numbers with nine decimal places. Numbers stored as "doubles" are imported with no errors.

[References](#)

## INF: Err "The other Application Quit" Closing Excel 4.0

Article Number: Q103258  
CREATED: 19-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

Microsoft Excel 4.0 automatically closes down [DDE](#) channels when quitting rather than waiting for a channel to be closed properly. This generates errors in other applications. In Microsoft Access it will generate the following error:

The other Application Quit

### More Information:

---

Note: This article assumes an Excel [macro](#) named "Message" exists.

Steps to reproduce:  
-----

The Following [Access Basic function](#) is designed to open Excel and run the macro named "Message". When this function is executed, The error message "The other Application Quit", will be generated in Access.

```
Function CallExcel ()
 Dim chan
 x = Shell("c:\excel\excel.exe c:\excel\macro1.xlm", 1)
 chan = DDEInitiate("Excel", "System")
 DDEExecute chan, "[Run(""macro1.xlm!Message"")] "
 AppActivate "Microsoft excel"
 DDEExecute chan, "[quit]"
 DDETerminate chan
End Function
```

The two following workarounds will alleviate the problem:

1. Add the statement "On Error Resume Next" to the function, before the Quit command. The corrected function should look as follows:

```
Function CallExcel ()
 Dim chan
 x = Shell("c:\excel\excel.exe c:\excel\macro1.xlm", 1)
 chan = DDEInitiate("Excel", "System")
 DDEExecute chan, "[Run(""macro1.xlm!Message"")] "
 AppActivate "Microsoft excel"
ADD-> On Error Resume Next
 DDEExecute chan, "[quit]"
 DDETerminate chan
```

End Function

2. Have Excel Quit itself rather than Microsoft Access do it. In the Excel macro, have the second to last command call another small Excel macro which will quit Excel. The following is an example:

```
=ON.TIME(NOW()+"0:0:3","Leave")
```

3 seconds after the above statement is executed, it will call the Excel Macro "Leave" which would contain:

```
=QUIT()
=RETURN()
```

Remove the DDE command to Quit from the original Access "CallExcel" macro. The modified macro will look as follows:

```
Function Callexcel2 ()
 Dim chan
 x = Shell("c:\excel\excel.exe c:\excel\macro2.xlm", 1)
 chan = DDEInitiate("Excel", "System")
 DDEExecute chan, "[Run(""macro2.xlm!Message"")] "
 AppActivate "Microsoft excel"
 DDETerminate chan
End Function
```

References:

Excel 4.0 Function Reference page 305

[References](#)

## INF: Reading, Storing, Writing Binary Large Objects (BLOBS)

Article Number: Q103257  
CREATED: 19-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

The [OLE data type](#) is often used to store large data objects in a Microsoft Access [table](#), such as sound, video, and graphic data. However, some large binary data objects cannot be represented if they do not have an OLE server that understands the data being stored. A user might want to store copies of executable program files, or other large, non-OLE, data in a Microsoft Access table.

This type of data is often referred to as a Binary Large Object, or BLOB, for short.

The following information discusses how to use a Microsoft Access memo [field](#) to store and work with large binary objects.

### More Information:

---

Provided below are two [Access Basic](#) functions that can be used to manage large amounts of binary data in Microsoft Access memo fields.

ReadBLOB() facilitates reading a binary file from disk and storing it's contents in a Microsoft Access memo field.

WriteBLOB() facilitates writing binary data stored in a Microsoft Access memo field to a file on disk.

Further, information is provided that demonstrates how to uses these functions.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

Create a new [module](#) with the following Declaration section and functions:

NOTE: In the following sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****
' DECLARATION SECTION
'*****
```

Option Explicit

Const BlockSize = 32768

```
'*****
' FUNCTION: ReadBLOB
'
' PURPOSE:
' Reads a large binary file (BLOB) from disk and stores the contents
' it in the specified table/memo field.
'
' PREREQUISITES:
' The specified table with the memo field to contain the binary
' data must be opened in Access basic code and the correct
' record navigated to prior to calling ReadBLOB.
'
' ARGUMENTS:
' Source - The path\filename to be read and stored.
' T - The table object to store the data into.
' Field - The memo field name in table T to store data into.
'
' RETURN:
' The number of bytes read from the Source file.
'
```

```
'*****
Function ReadBLOB (ByVal Source As String, T As Table, _
Field As String)
```

```
Dim NumBlocks As Integer
Dim FileLength As Long, LeftOver As Long
Dim FileData As String
Dim SourceFile As Integer
Dim i As Integer, RetVal As Variant
```

```
On Error GoTo Err_ReadBLOB
```

```
' Open the source file to read from
SourceFile = FreeFile
Open Source For Binary Access Read As SourceFile
```

```
' Get the length of the file
FileLength = LOF(SourceFile)
```

```
If FileLength = 0 Then
ReadBLOB = 0
GoTo Bye_ReadBLOB
End If
```

```
' Calculate number of blocks to read and left over bytes
NumBlocks = FileLength \ BlockSize
LeftOver = FileLength Mod BlockSize
```

```
' Initialize the status bar meter
RetVal = SysCmd(1, "Reading BLOB", FileLength \ 1000)
```

```
' Put the table in edit mode to read data into
T.Edit
```

```
' Read the 'left over' amount of data, writing it to the table
```

```

FileData = String$(LeftOver, 32)
Get SourceFile, , FileData
T(Field).AppendChunk (FileData)

' Update status bar meter
RetVal = SysCmd(2, LeftOver / 1000)

' Read the remaining blocks of data, writing them to the table
FileData = String$(BlockSize, 32)
For i = 1 To NumBlocks
 Get SourceFile, , FileData
 T(Field).AppendChunk (FileData)

 ' Update status bar meter
 RetVal = SysCmd(2, BlockSize * i / 1000)
Next i

' Update the table record
T.Update

' Remove status bar meter
RetVal = SysCmd(3)

Close SourceFile

ReadBLOB = FileLength

Bye_ReadBLOB:

Exit Function

Err_ReadBLOB:

ReadBLOB = -1 * Err
Resume Bye_ReadBLOB

End Function

'*****
' FUNCTION: WriteBLOB
'
' PURPOSE:
' Writes large binary file (BLOB) information stored in the
' specified table/memo field to the specified path\file on disk.
'
' PREREQUISITES:
' The specified table with the memo field containing the binary
' data must be opened in Access basic code, and the correct record
' navigated to prior to calling WriteBLOB.
'
' ARGUMENTS:
' T - The table object containing the binary information
' to write.
' Field - The memo field name in table T containing the
' binary information to write.
' Destination - The path\filename to write the binary
' information to.

```

```

'
' RETURN:
' The number of bytes written to the Destination file.
'
'*****
Function WriteBLOB (T As Table, Field As String, _
ByVal Destination As String)
 Dim NumBlocks As Integer
 Dim FileLength As Long, LeftOver As Long
 Dim FileData As String
 Dim i As Integer, RetVal
 Dim DestFile As Integer

 On Error GoTo Err_WriteBLOB

 ' Get the size of the field
 FileLength = T(Field).FieldSize()

 If FileLength = 0 Then
 WriteBLOB = 0
 GoTo Bye_WriteBLOB
 End If

 ' Calculate number of blocks to write and left over bytes
 NumBlocks = FileLength \ BlockSize
 LeftOver = FileLength Mod BlockSize

 ' Remove any existing destination file
 DestFile = FreeFile
 Open Destination For Output As DestFile
 Close DestFile

 ' Open the destination file to write to
 Open Destination For Binary As DestFile

 ' Initialize status bar meter
 RetVal = SysCmd(1, "Writing BLOB", FileLength / 1000)

 ' Write the 'left over' amount of data to the output file
 FileData = T(Field).GetChunk(0, LeftOver)
 Put DestFile, , FileData

 ' Update status bar meter
 RetVal = SysCmd(2, LeftOver / 1000)

 ' Write the remaining blocks of data to the output file
 For i = 1 To NumBlocks
 ' Clear the buffer and read a chunk
 FileData = ""
 FileData = T(Field).GetChunk((i - 1) * BlockSize _
+ LeftOver, BlockSize)

 ' Write the chunk to the output file
 Put DestFile, , FileData

 ' Update status bar meter
 RetVal = SysCmd(2, ((i - 1) * BlockSize + LeftOver) / 1000)

```

```

 Next i

 ' Remove status bar meter
 RetVal = SysCmd(3)

 Close DestFile

 WriteBLOB = FileLength
Bye_WriteBLOB:

 Exit Function

Err_WriteBLOB:

 WriteBLOB = -1 * Err
 Resume Bye_WriteBLOB

End Function

```

#### How to use the ReadBLOB() and WriteBLOB() Functions

=====

The following subroutine, CopyFile, demonstrates how to copy a binary file into a Microsoft Access memo field and then write it back out to a new disk file.

1. Create the following table:

```

Table BLOB

FieldName: Blob
DataType: Memo

```

2. Add the following [function](#) to the BLOB module created earlier:

```

'*****
' SUB: CopyFile
'
' PURPOSE:
' Demonstrates how to use CopyBLOB and WriteBLOB.
'
' PREREQUISITES:
' A table called "BLOB" must be created which contains a
' memo field with the name "Blob".
'
' ARGUMENTS:
' Source - The source path\file name to copy
' Destination - The destination path\file name to write to
'
' EXAMPLE:
' CopyFile "c:\access\msaccess.hlp", "c:\access\msacces2.hlp"
'
'*****
Sub CopyFile (ByVal Source As String, ByVal Destination As String)
 Dim BytesRead, BytesWritten
 Dim Msg As String

```

```

Dim db As Database, T As Table

' Open the BLOB table
Set db = CurrentDB()
Set T = db.OpenTable("BLOB")

' Create a new record and move to it
T.AddNew
T.Update
T.MoveLast

BytesRead = ReadBLOB(Source, T, "Blob")

Msg = "Finished reading "" & Source & """"
Msg = Msg & Chr$(13) & ".. " & BytesRead & " bytes read."
MsgBox Msg, 64, "Copy File"

BytesWritten = WriteBLOB(T, "Blob", Destination)

Msg = "Finished writing "" & Destination & """"
Msg = Msg & Chr$(13) & ".. " & BytesWritten & " bytes written."
MsgBox Msg, 64, "Copy File"

End Sub

```

3. With the BLOB module opened in Design View, choose [Immediate window](#) from the View menu.

4. Type the following and press Enter:

```
CopyFile "c:\access\msaccess.hlp", "c:\access\msacces2.hlp"
```

Result: The Microsoft Access help file, which is 3+ megabytes in size, will be copied into the Blob memo field in the BLOB table. Then, this information will be copied back out to the hard disk with a new name.

References:

"Microsoft Access Introduction to Programming," version 1.x, Chapter 2, "Writing a New Function," pages 8-14

"Microsoft Access Introduction to Programming," version 1.x, Chapter 8, "Manipulating Data," pages 98-117

"The Secrets of Access Wizards," version 1.1 Access Distribution Kit, Chapter 2, "Inclinations: AccessWizard Functions and Properties," pages 30-32

[References](#)

## INF: Prevent Adding New Records To Editable Form

Article Number: Q103256  
CREATED: 19-AUG-1993  
MODIFIED: 16-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This illustrates two methods to create a [form](#) that the user can use to edit data, but prevent adding new records. The first [method](#) is to use a [macro](#) to move the cursor to the Last Record when the user tries to go to a New Record. The second method uses a [query](#) to make it impossible to add a new [record](#). (This second method also prevents a blank "new" record.)

### More Information:

---

Method 1: Using a Macro

-----

1. Create a macro.

Macro Name: EditOnlyMacro

| Condition:                           | Action                                                                                  |
|--------------------------------------|-----------------------------------------------------------------------------------------|
| [Product ID] Is <a href="#">Null</a> | DoMenuItem<br>Menu Bar: Form<br>Menu Name: Records<br>Command: GoTo<br>Subcommand: Last |

2. Go to Design View of the Products Form. Put the macro on the OnCurrent property of the Form.

Method 2: Using a Query

-----

This example will make it possible to edit the records in the Products [table](#), but not add any new records. The Products table is the related side of a [one-to-many relationship](#). The Categories table is the Primary, or parent, table of this one-to-many [relationship](#). The linking [field](#), [Category ID], is used from the Primary table, Categories.

(If the table in which you wish to edit records is not the related side of a one-to-many relationship, a method to create a parent table is illustrated below.)

1. Create a query.

Query: EditOnlyQuery

-----

Type: Select Query  
Join: Categories.[Category ID] <-> Products.[Category ID]

Field: Category ID  
Table: Categories  
Field: Product ID  
Table: Products  
Field: Supplier ID  
Table: Products  
Field: Product Name  
Table: Products

Drag down the rest of the fields from the Products table.

Note: [Category ID] is from the Categories table, NOT the Products table.

2. Go to Design View of the Products Form. Change the RecordSource of the Products Form from Products to EditOnlyQuery.

Creating a new Parent table:

-----  
The above method depends on using the ID field of the Primary or 'parent' table in a one-to-many relationship. If the table that you wish to edit does not have a 'parent' table, you can create one. For instance, if you want to be able to edit data in the Categories table but not be able to add any new records, do the following.

1. Create a new table.

Table: NewParent Table  
-----  
FieldName: ID  
DataType: Number  
FieldSize: Byte  
Make it a Primary Key

2. Save the table and switch to [Datasheet view](#). Put a number "1" in the first record. Close the table.
3. Go to the table that you want to Edit Only in Design View. Add a field.

Table: Categories  
-----  
FieldName: EditOnlyJoin  
DataType: Number  
FieldSize: Byte

4. You now need to put a number "1" in every record. (Hint: you could use an Update Query to do this.)
5. Establish a relationship between NewParent Table and the Categories table. NewParent Table will be the Primary table and Categories table will be the Related table with a one to many relationship. (To establish the relationship, from the [Database window](#), click

Edit and choose Edit Relationships.)

6. Add the NewParent Table and the Categories table in a query (see method 2 above). Be sure that this new query has the ID field included from the NewParent Table and DOES NOT have the EditOnlyJoin field from the Categories table.

References:

[References](#)

## PRB: Error Message: Couldn't Open File

Article Number: Q103254  
CREATED: 19-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

Multi-user options are set to shared, but several instances of a database cannot be opened. You either receive the error message:

    Couldn't Open File <database name>

or the error message:

    <Database name> This file is already in use. Use a new filename or close the file in use by another application.

### CAUSE

---

The database chosen from the Quick Pick List at the bottom of the File Menu, and not from the File Open dialog window. The Quick Pick List saves the settings of the last opened version of Microsoft Access.

### RESOLUTION

---

Running a single instance of Microsoft Access, open that particular database from the Open Database option from the File menu. Then close the database, thereby making the correct settings to the database in the Quick Pick List.

### MORE INFORMATION

---

This problem occurs while Microsoft Access is being shared on a network and can be reproduced on a standalone machine by the following steps.

Steps to Reproduce Behavior  
-----

1. Open the sample database Northwind Traders (NWIND.MDB) by choosing File, Open Database.
2. From the Open Database dialog box make sure that Read Only is unchecked and the Exclusive is checked.
3. From the database menu bar choose Options from the View menu.
4. From the Options dialog box, choose Multi-user from the Category: list box.

5. From the Items list box set Default Mode for Open Databases to Shared.
6. Exit Microsoft Access.
7. Open the sample database Northwind Traders (NWIND.MDB) by choosing Open Database from the File menu.
8. Start up second instance of Microsoft Access.
9. Open the sample database Northwind Traders (NWIND.MDB) by choosing the database from the Quick Pick List.
10. The following error message occurs:

Couldn't Open File.

Note: If steps 7 & 9 are reversed the following error message occurs:

This File Already In Use.

#### References:

"Microsoft Access User's Guide," version 1.0, Chapter 16, "Using Microsoft Access in a Multiuser Environment," page 387  
For more information, [query](#) on the following words in the Microsoft Knowledge Base:

"couldn't open file" and "this file is already in use"

#### [References](#)

## PRB: Insert Object Dialog Box Has No 'All Files (\*.\*)' Option

Article Number: Q103253  
CREATED: 19-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

When you choose Insert Object from the Edit menu, the Insert Object >From File [dialog box](#) appears. However, All Files (\*.\*) is not an option listed in the List Files of Type [combo box](#).

### CAUSE

---

This behavior is by design. The combo box displays only file types that represent an [OLE](#) server application stored in your system's registration [database](#) file, REG.DAT. The "All files (\*.\*)" option is omitted to prevent you from trying to [link](#) or embed an object using a non-OLE or unregistered application.

### RESOLUTION

---

You can type "\*.\*", without quotes, into the File Name [text box](#) if you wish to display all file types. However, if your purpose in selecting all files is to link/embed from a non-OLE or unregistered application, you must use Object Packager to create and paste an object into Microsoft Access.

For instructions on using Object Packager, see Chapter 13, "Integrating Your Windows Applications" in the "Microsoft Windows User's Guide," version 3.1.

### MORE INFORMATION

---

The following example is based on the sample database Northwind Traders, NWIND.MDB:

Steps to Reproduce Behavior  
-----

1. Open the Categories [table](#) in [Datasheet view](#).
2. Move to a new [record](#) and place your cursor in the Picture [field](#).
3. From the Edit menu, choose Insert Object.
4. Choose the File button in the Insert Object dialog box.
5. Drop down the List Files of Type combo box. Note that "All field (\*.\*)" option is not included.

At this point you may type \*.\* into the File Name text box.

References:

"Microsoft Windows User's Guide," version 3.1, pages 505-514

[References](#)

## INF: Query with Parameters to Evaluate Complex Criteria

Article Number: Q103252  
CREATED: 19-AUG-1993  
MODIFIED: 17-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

Microsoft Access allows the use of [variable](#) parameters in queries. This article discusses how to construct a [query](#) requiring more than one prompt. Each parameter can be used both as [criteria](#) and as a [field](#) to allow complicated evaluation of the value entered in each parameter.

### MORE INFORMATION

---

The following [parameter query](#) is based on the Orders [table](#) in the sample [database](#) NWIND.MDB. It selects orders written between two variable dates provided by the user.

If either the Start Date or the End Date is not entered, the query returns all dates greater than or equal to the Start Date, or less than or equal to the End Date. If neither a Start Date nor an End Date is entered, the query returns all orders.

Query: FindOrdersByDate

-----

Field: Order ID  
Show: Yes

Field: Order Date  
Sort: Ascending  
Show: Yes  
First Criteria Line: Between [Start Date] and [End Date]  
Second Criteria Line: <=[End Date]  
Third Criteria Line: >=[Start Date]  
Fourth Criteria Line:

Field: [Start Date]  
Show: No  
First Criteria Line: Is Not [Null](#)  
Second Criteria Line: Is Null  
Third Criteria Line: Is Not Null  
Fourth Criteria Line: Is Null

Field: [End Date]  
Show: No  
First Criteria Line: Is Not Null  
Second Criteria Line: Is Not Null  
Third Criteria Line: Is Null

Fourth Criteria Line: Is Null

REFERENCES

=====

"User's Guide," version 1.0, pages 178-181

For more information on similar criteria, query on the following phrase here in the Microsoft Knowledge Base:

query by [form](#)

[References](#)

## PRB: Sendkeys Infinite Loop Macro

Article Number: Q103251  
CREATED: 19-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

When you press a button to execute your [macro](#) that uses two SendKeys actions, the macro goes in an infinite loop and you must reboot your system to regain [control](#).

### CAUSE

---

If only one SendKeys is used, you can stop the macro by depressing the control and break keys simultaneously (CTRL+BREAK) and then pressing the Halt button in the [dialog box](#). The second SendKeys will toggle the default of the Script Single Step dialog, so that the macro can not be terminated without rebooting.

### MORE INFORMATION

---

Steps to Reproduce Behavior

-----  
1. Create a macro as follows:

| Macro Name | Action               |
|------------|----------------------|
| Go Loop    | SendKeys<br>SendKeys |

Go Loop Actions

-----  
SendKeys  
Keystrokes: " "  
Wait: No

SendKeys  
Keystrokes: " "  
Wait: No

2. Open a new [form](#) and place a [command button](#) on it.
3. In the command button properties window select the Go Loop macro from the list next to the OnPush property.
4. Get into [Form view](#) and push the button.

References: "Microsoft Access Language Reference", version 1.0, pages 432-433

## [References](#)

## **PRACC9308: Setup Program Not Handling All Palette Messages**

Article Number: Q103250  
CREATED: 19-AUG-1993  
MODIFIED: 20-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

The colors on the Microsoft Access Setup screen may appear completely random.

### **CAUSE**

---

This problem occurs if you load a 256-color [bitmap](#) in an application such as Paintbrush before starting the Microsoft Access Setup program, then switch to this application while the Microsoft Access Setup program is running. This problem occurs only if you are using a driver other than a 16-color video driver.

### **RESOLUTION**

---

You can correctly redraw the Microsoft Access Setup screen by maximizing and then minimizing the application in which you loaded the 256-color bitmap (for example, the Paintbrush window containing the 256-color bitmap). You can also switch to a 16-color video driver.

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### **MORE INFORMATION**

---

Steps to Reproduce Problem  
-----

1. Make sure that you are using a 256-color video driver.
2. Open Paintbrush and load a 256-color bitmap.
3. Size the Paintbrush window to fill the right half of your screen.
4. Start the Microsoft Access Setup program. At the first opportunity (after entering your name and company), size the Setup window to fill the left half of the screen.
5. Activate the Paintbrush window by clicking the [title bar](#). Random colors now appear in the Microsoft Access Setup window.

## [References](#)

## INF: Form Controls to Increment and Decrement a Date

Article Number: Q103194  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This article describes how to create [form](#) controls which increment or decrement a date. After initializing a date [control](#) to the system date, you will use command buttons to increase and decrease the displayed date.

### More Information:

---

Suppose you are designing a form to be used for entering new order records. You may wish to have the order date default to the current date, but you may also want the ability to increment or decrement the default date.

Steps to Reproduce:  
-----

1. Create a new unbound blank form named "Date Selector".
2. Add an unbound [text box](#) for the date value:

Object: Text Box

-----  
ControlName: Call Date  
Format: General Date  
Default Value: =Date()  
Locked: Yes  
SpecialEffect: Sunken

3. Add command buttons to change the date value:

Object: Command Buttons

-----  
ControlName: Decrease  
Caption: <  
Auto Repeat: Yes  
OnPush: Change Date  
ControlName: Advance  
Caption: >  
Auto Repeat: Yes  
OnPush: Change Date

3. Create a [macro](#) to increase and decrease the date.

| Macro Name  | Condition                                   | Action      |
|-------------|---------------------------------------------|-------------|
| Change Date | Screen.ActiveControl.ControlName="Advance"  | SetValue    |
|             | ...                                         | GoToControl |
|             | ...                                         | GoToControl |
|             | Screen.ActiveControl.ControlName="Decrease" | SetValue    |
|             | ...                                         | GoToControl |
|             | ...                                         | GoToControl |

#### Change Date Actions

---

##### SetValue

Item: Forms![Date Selector]![Call Date]  
 Expression: Forms![Date Selector]![Call Date]+1

##### GoToControl

ControlName: [Call Date]

##### GoToControl

ControlName: [Advance]

##### SetValue

Item: Forms![Date Selector]![Call Date]  
 Expression: Forms![Date Selector]![Call Date]-1

##### GoToControl

ControlName: [Call Date]

##### GoToControl

ControlName: [Decrease]

4. Open the Date Selector form and perform the following test:

The date will default to the system date. Press the ">" button to increase the date. Use the "<" button to decrease the date. The date should continue to change while the command button is depressed. The cursor will tab into the date text box. However the date cannot be changed except by using the command buttons.

References: "Microsoft Access User's Guide," versions 1.0 and 1.1, Chapter 9, "Designing Forms," and Chapter 12, "Using Special Design Effects on Forms and Reports"

## [References](#)

## INF: Equivalent of Excel's Fill Down Functionality

Article Number: Q103192  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This article describes how to do an Excel fill down [function](#) in Microsoft Access.

### More Information:

---

In Microsoft Access, you would use an Update [query](#) to perform this functionality. The following example creates a copy of the Orders [table](#), Orders1, before performing the fill down functionality.

#### Steps to Reproduce Behavior

-----

For this example we are going to create a scenario where the NorthWind Traders decided to make every shipping charge \$10. For many people they want to do a fill down like in Excel, this is article describes how this can be done in Microsoft Access.

1. Highlight the Orders table in the [Database window](#).
2. From the Edit menu choose the Copy option.
3. From the Edit menu choose the Paste option. You will be asked for a name for your new table. Name it Orders1.
4. Choose Query and press the New button.
5. Add your new table, Orders1, and close the Add Table window.
6. Drag the Freight [field](#) into the [QBE](#) grid.
7. From the Query menu choose the Update option.
8. On the Update To: line type \$10.00 and tab out of the field.
9. From the Query menu choose the Run option. You will be told 1078 records will be updated.
10. Choose OK.
11. Open the Orders1 table.

All the records will now have a Freight charge of \$10.00.

## [References](#)

## PRACC9307: Slow Repaint When Saving Form with Many Controls

Article Number: Q103191  
CREATED: 18-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0
- 

### SYMPTOMS

---

The first time you save a form that contains both a large number of controls and a counter field, Microsoft Access repaints the controls one at a time. This process may take two to 10 seconds, depending on the speed of your processor.

### CAUSE

---

Because the form contains a counter field, it must be repainted when the counter value is determined (the first time the record is saved). All fields on the form are repainted, not only the counter.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

### MORE INFORMATION

---

Steps to Reproduce Problem  
-----

1. Create a table with a counter field and approximately 100 fields with the Yes/No data type.
2. Create a new form based on the above table and include all fields.
3. Add the Yes/No fields to the form as check boxes.
4. Open the form and select a majority of the check boxes. When you save the record, the drive light flashes momentarily as each record is saved. Microsoft Access then takes several seconds to update the display.

NOTE: This problem occurs only the first time the record is saved. Subsequent changes to the record do not cause this behavior.

### [References](#)

## PRACC9308: Cannot Use First() and Last() Functions in a Report

Article Number: Q103190  
CREATED: 18-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

Microsoft Access returns a #ERROR message in your Report field when you use First() or Last() functions.

### RESOLUTION

---

It may be possible to use DFirst() or DLast() domain aggregate functions to achieve the results you want. However, these functions ignore any sorting or grouping in your report. You may need to supply a criteria to these domain functions to return the data desired.

### STATUS

---

This behavior is by design.

### MORE INFORMATION

---

Steps to Reproduce Problem  
-----

1. Open the sample database NWIND.MDB.
2. Open the Alphabetical List of Products report in Design view.
3. Add an unbound text box to the Report Header section as follows:

Object: Text Box

-----  
ControlName: First Product  
ControlSource: =First([Product Name])

4. From the File menu, choose Print Preview.

The text box that you added, First Product, evaluates to #ERROR.

### REFERENCES

=====

For more information on using the DFirst() and DLast() functions, refer to the "Microsoft Access Language Reference," versions 1.0 and 1.1, page 139.

### [References](#)

## INF: How to use Aggregate Functions in a Query Without Errors

Article Number: Q103189  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

You receive the following error message:

Can't group on fields selected with '\*'

### More Information:

---

This error message occurs when attempting to use Aggregate functions in a query when the Restrict Available Fields query property is not selected. To successfully run your query:

1. Open the query in Design view
2. From the View menu, choose Query Properties
3. Check the Restrict Available Fields property

Note that this is not the error message for dragging the asterisk (\*) down to the QBE grid when calculating totals. That error message:

Can't show Totals on fields added to QBE grid with the asterisk cannot be avoided by using the Restrict Available Fields property.

### Steps to Reproduce Behavior

-----

1. Open the sample database Northwind Traders (NWIND.MDB)
2. Run the Daily Order Totals query to confirm that it works perfectly. (This query is using aggregate functions.) Close the query.
3. Open the Daily Order Totals query in Design view.
4. From the View menu, choose Query Properties.

The Restrict Available Fields property is selected. To duplicate the above error message, unselect this property and try to run the query again.

For more information about Aggregate functions in a query, refer to "Microsoft Access User's Guide," versions 1.0 and 1.1, Chapter 6, "Designing Select Queries".

### References



## PRB: Unable to Close Access From Control Box

Article Number: Q103188  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

You receive the following error message when attempting to close Microsoft Access from its Control-menu box:

Currently unable to quit Microsoft Access

### RESOLUTION

---

Do not attach the Quit macro action to the OnClose property of the form if the user will be using the control menu box to quit Microsoft Access.

### MORE INFORMATION

---

The Control-menu box is the upper-left corner of each application's window. To access this menu box without the mouse, press ALT+Spacebar.

Steps to Reproduce Behavior

-----  
1. Create a blank, unbound form called Form1:

Form: Form1

-----  
OnClose: Macro1

2. Create Macro1:

| Macro Name | Action |
|------------|--------|
|------------|--------|

-----  
Macro1           Quit

3. Open Form1, and double-click the Microsoft Access Control-menu box. You will receive the error message:

Currently unable to quit Microsoft Access.

### [References](#)

## PRB: Opening a Database Using a Sendkeys Macro

Article Number: Q103187  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

You have a [macro](#) with two actions, SendKeys and DoMenuItem, which should open another [database](#). This macro does not open the specified database.

### CAUSE

---

While DoMenuItem is executing, Microsoft Access cannot open another database because the macro that contains the DoMenuItem can only execute as long as its database is active. Microsoft Access does not support opening more than one database at a time (outside of Access Basic code).

### RESOLUTION

---

Use a single SendKeys macro action to open the database.

### MORE INFORMATION

---

A workaround is to use a one-line SendKeys macro. The macro should be created as follows:

| Macro Name | Action   |
|------------|----------|
| OpenMe     | SendKeys |

OpenMe Actions

-----

SendKeys  
    Keystrokes: %foYOURFILE.MDB~  
    Wait: No

This will automatically open the specified database. You would substitute YOURFILE.MDB for the name of the database you wish to open.

#### Steps to Reproduce Behavior

-----

1. Create a new database and then create a new macro named OpenMe.

| Macro Name | Action   |
|------------|----------|
| OpenMe     | SendKeys |

## DoMenuItem

OpenMe Actions

-----  
SendKeys:

NWIND.MDB~

DoMenuItem:

Menubar: Database

Menu Name: File

Command: Open database

2. Run the macro. It does not open the NWIND.MDB database specified in the SendKeys argument. If you delete the tilde and run the macro and manually select 'OK' to select the named database, the database still does not open.

References:

"Microsoft Access Language Reference," version 1.0, pages 151-154 and pages 432-435.

[References](#)

## PRB: Error Message: Too Many Memo or OLE Fields

Article Number: Q103186  
CREATED: 18-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

The following error message

Too many Memo or OLE fields

is displayed when you try to export a Microsoft Access table that contains more than one memo field to Btrieve format.

### CAUSE

---

Microsoft Access follows the Xtrieve specification for writing to Btrieve tables. This specification allows only one memo field, which must be the leftmost field in the table.

### RESOLUTION

---

Change the table so that it contains only one memo field by doing either of the following:

- Changing the data types of all but one memo field.
- or-
- Moving the memo fields to other tables so that there is only one memo field per table.

Btrieve and Xtrieve are manufactured by Novell, Inc. a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

### References

## INF: How To Tell If Windows For Workgroups Is Running

Article Number: Q103185  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

Although Microsoft Windows and Windows for Workgroups can both display the same version number, you can determine whether Windows for Workgroups is running by using a Windows API call in an [Access Basic function](#).

### More Information:

---

To determine if Windows for Workgroups is running, create a new [module](#) and enter the following lines:

Note: In the following sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****
' Declarations section of the module
'*****
Option Explicit
Const WNNC_NET_MultiNet = &H8000
Const WNNC_SUBNET_WinWorkgroups = 4
Const WNNC_NET_TYPE = 2

Declare Function WNetGetCaps% Lib "User" (ByVal nIndex%)

'=====
```

'This function returns True if Windows for Workgroups is running,  
'or False if it isn't (generic Windows). It accomplishes this by:  
' - calling WNetGetCaps and retrieving the net type flag.  
' - inspecting the low word of the flag to see if the Windows for  
' Workgroups bit is set.  
'=====

```
Function IsWFW% ()
 Dim wNetType As Integer

 wNetType = WNetGetCaps(WNNC_NET_TYPE)
 IsWFW = False
 If (wNetType And WNNC_NET_MultiNet) Then
 IsWFW = ((wNetType And &HFFFF) And _
 WNNC_SUBNET_WinWorkgroups) <> 0
 End If
```

End Function

To test this function, open the [Immediate window](#), and enter the following line:

```
MsgBox IIf(IsWFW(), "WFW", "Windows") & " is running!"
```

If Windows for Workgroups is running, you will see the message,

```
WFW is running!
```

References:

[References](#)

## INF: How to Find the Last Day of the Month

Article Number: Q103184  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This article describes an [Access Basic function](#) that accepts a date as an argument and returns the last day of the month.

### More Information:

---

This article assumes that you are familiar with Access Basic and how to create and use Access Basic Procedures.

First open a new Module or a previously created [module](#) and enter the following code:

```
'*****
'Declarations Section of Module
'*****
Option Explicit

'*****
'Findeom Function
'*****
'This function takes a date as an argument and returns the last
'day of the month.
Function FindeOM (MyDate)
 Dim NextMonth, EndOfMonth
 NextMonth = DateAdd("m", 1, MyDate)
 EndOfMonth = NextMonth - DatePart("d", NextMonth)
 FindeOM = EndOfMonth
End Function
```

The following [expression](#), gives an example of how to call this function:

```
=FindeOM(#5/5/93#)
```

To view the results of this function, in the [Immediate window](#) type:

```
?FindeOM(#5/5/93#)
```

### [References](#)

## INF: How to Create a Running Sum on a Form

Article Number: Q103183  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

While reports have a RunningSum property, forms do not. This article shows a [method](#) to create a running sum on a [form](#).

### More Information:

---

1. You need a unique, sequential ID [field](#), like a counter or a date/time field. If one does not already exist, you can create one by opening the [table](#) in design view and adding a new field.
2. Open the sample [database](#) Northwind Traders (NWIND.MDB) and create a new blank form based on the Orders table.
3. Set up the form as follows:

NOTE: The underscore character '\_' is used below as a continuation character; you must enter the entire text as one line.

Form: Form1

-----  
RecordSource: Orders

Form Controls

-----  
Text Box: Order ID  
ControlSource: Order ID  
Text Box: Order Amount  
ControlSource: Order Amount  
Text Box: Total  
ControlSource: =DSum("[Order Amount]","Orders",\_  
"[Order ID] <= Forms![Form1]![Order ID]")

The DSum calculates the sum total of all orders with an Order number less than or equal to the currently displayed order number. For this reason, it is important to have a unique and sequential ID field.

References:

[References](#)

## INF: INF: Function to Determine Runtime or Retail Version [B\_WAccADK]

Article Number: Q103182  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

The following information provides a [function](#) called IsRuntime() that can be used to determine if your Microsoft Access application is running using the retail version of Microsoft Access or the runtime version.

This is useful if you want to prevent users from using your application with the retail version.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

### More Information:

---

Add the following function, IsRuntime, to a new or existing [module](#):

```
Function IsRuntime ()
 On Error GoTo ErrIsRuntime
 IsRuntime = SysCmd(6)

ByeIsRuntime:
 Exit Function

ErrIsRuntime:
 If (Err = 5) Then
 IsRuntime = False
 Else
 Error Err
 End If
 Resume ByeIsRuntime
End Function
```

Note: It is necessary to trap for Err = 5, Illegal Function Call, in the [event](#) that a user is running your application with Access version 1.0, which does not support the SysCmd(6) call.

The following [macro](#) demonstrates how to use IsRuntime() in a macro to prevent users from starting your application with the retail version of Microsoft Access.

Create the following macro in your application and save it with the

name "AutoExec":

| Condition       | Action | Defined Below |
|-----------------|--------|---------------|
| Not IsRuntime() | MsgBox | 1.            |
| ...             | Quit   | 2.            |

AutoExec Actions

1. MsgBox  
Message: Invalid setup, run YourAppName setup and try again.  
Beep: Yes  
Type: Critical  
Title: YourAppName
2. Quit

[References](#)

## INF: Showing All Records (Including Null) in a Parameter Query

Article Number: Q103181  
CREATED: 18-AUG-1993  
MODIFIED: 09-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

When you are running a [query](#) that takes its parameters from a [form](#), no records are returned by the query if you leave the [field](#) blank. If you type an asterisk (\*) in the field, only records with non-[Null](#) values are returned.

This article describes a [method](#) for returning all records, including those with Null values, when you leave the parameter blank.

### MORE INFORMATION

---

The following example is based on the sample [database](#) NWIND.MDB and assumes that you are familiar with Microsoft Access form, [macro](#), and query design.

1. Create the following unbound form:

```
Form: Pick Employees

Control: Textbox
 ControlName: Region
Control: Command Button
 Caption: Run Query
 OnPush: Run Employee Query
```

2. Create the following macro:

```
Macro: Run Employee Query

Action: OpenQuery
 Query Name: Employee Query
 View: Datasheet
 Data Mode: Edit
```

3. Create the following query based on the Employees [table](#):

```
Query: Employee Query

Field: First Name
 Show: True
Field: Last Name
 Show: True
Field: Region
```

Show: True  
Criteria: Like Forms![Pick Employees]!Region & "\*"   
Or: <leave blank>  
Field: Forms![Pick Employees]!Region  
Show: False  
Criteria: <leave blank>  
Or: Is Null

4. Open your query in [Design view](#). From the Query menu, choose Parameters. Type "Forms![Pick Employees]!Region" (without the quotation marks) as the parameter name, with Text as the [data type](#).
5. Open the Pick Employees form, type "WA" (without the quotation marks) in the Region field, and choose the Run Query button. The [dynaset](#) will contain five employee names.
6. Open the Pick Employees form, clear the Region field, and choose the Run Query button again. The dynaset now contains nine employee names, four with blank region codes.

By adding the parameter as a field, we can test the parameter and [control](#) the other [criteria](#). The equivalent [SQL](#) Where condition is the following:

Where Region Like Forms![Pick Employees]!Region & "\*"   
Or Forms![Pick Employees]!Region Is Null

## [References](#)

## PRB: Data Dropped when Printing or White Lines on Reports

Article Number: Q103180  
CREATED: 18-AUG-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

There have been reported problems with printing lines or graphics from Microsoft Access using the Canon BJC800 or BJ10E print drivers. The described symptoms are:

1. Horizontal white lines appear on the [report](#).
2. Portions of text are not printed.
3. If the reports or the forms are printed page by page vs. printing all at once all the data prints.
4. Changing to a lower resolution prints all the data.
5. Printing monochrome vs. color prints all of the data.

Note: this problem can happen on other printers as well.

### CAUSE

---

The problem is related to sending large amounts of data to the printer and having Windows time out trying to send the data to the printer. When this happens a line of data may be lost. Microsoft Access treats lines or borders as a [bitmap](#) or large graphic and the printer needs more time to interpret the data being sent to print.

### RESOLUTION

---

Make sure the Printer Time Out (seconds) value is set as follows:

| Option               | Setting |
|----------------------|---------|
| -----                | -----   |
| Device not selected: | 15      |
| Transmission retry:  | 90      |

These can be set by choosing the Printers icon in Control Panel and then choosing Connect for the specific printer driver.

The default settings for "Device not selected" is 15 and "Transmission retry" is 45 these settings may need to be increased to 30 and 120 or higher for printing complex graphics images from some applications such as lines and pictures, or when printing over a [network](#).

### [References](#)



## PRB: "Help Topic Does not Exist" Invoking Custom Help File [B\_WAccADK]

Article Number: Q103179  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

Pressing F1 to invoke a custom help file topic for a form control produces the contents screen followed by the message

Help topic does not exist" two times

### CAUSE

---

The number entered in the HelpContextID property could not be mapped to a help file context string in the custom help file.

### RESOLUTION

---

In the help project file, make sure there exists a [MAP] section which correctly maps the HelpContextID context number with the context string in the custom help file.

### MORE INFORMATION

---

The number entered in the HelpContextID property for a Microsoft Access form control is often mistaken for the Context String used to identify a topic in the Help system. The HelpContextID is actually known as a "Context Number". A Context Number is an optional number that you can associate, using the [MAP] section in the help project file, with context strings in the Help system. This number is assigned one to each object in an application to enable context sensitivity.

In the help project file, a typical [MAP] section might look like the following:

```
[MAP]
TitlePageContextString 3
Topic1ContextString 1
Topic2ContextString 2
```

In this example, the #-footnote for the first topic would be set to the string 'TitlePageContextString'. Within Microsoft Access, you would set HelpContextID to the number 3.

The context string appears first, followed by the HelpContextID context number. The context string and context number are separated by an arbitrary amount of white space, using either space characters or tabs.

References:

For more information, see "Mapping Context-Sensitive Topics" in chapter 6 of the Help Compiler Guide.

[References](#)

## INF: Resizing Multiple Controls Simultaneously

Article Number: Q103178  
CREATED: 18-AUG-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Changing the properties (size, font, color, etc.) of several controls (text boxes, labels, etc.) on a [form](#) or [report](#), is normally accomplished by changing each individual [control](#). This article describes how to change more than one control at a time.

### MORE INFORMATION

---

The following steps detail the process:

1. Resize one of the controls to the desired size by using the mouse or by indicating the width in the height in that controls [property sheet](#).
2. After changing the desired properties, choose Change Default from the Layout menu.

NOTE: If multiple controls are selected when Change Default is chosen, the properties of the last control selected are used as the defaults.

3. Select the remaining controls you wish to resize You can do this by either dragging the mouse pointer through all of them or by holding down the shift key and clicking.
4. Choose Apply Default from the Layout menu.

This will change the size of all of the controls on the forms.

NOTE: This will apply all defaults including fonts, colors, size, etc to the selected control(s).

References:

[References](#)

## INF: Access Security and RunSQL With Owner's Permission Denied

Article Number: Q103176  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

You may get [permissions](#) denied if you try to run an [action query SQL](#) statement in a RunSQL [macro](#) statement or as the RowSource for a [control](#) in a secured [database](#).

### More Information:

---

SQL statements used in the RunSQL macro or as the RowSource for a control are classified as "temp" queries. That means that they are compiled and built when executed, and not before. They don't exist as real objects in the [Database window](#). Therefore, they don't have an owner. The 'With Owner Access' clause is ignored. If you don't have permissions to the base [table](#) you will be denied. The actual [query](#) would be able to execute the command (provided the query owner has permissions to the table).

The Workaround:

-----  
You must have the RowSource property set to an intermediate query object which has the 'Run with Owner's permission' clause created by the owner of the secured table.

Steps to Reproduce:

-----  
If you have an [update query](#) (created by MYADMIN) based on a secured table (created by MYADMIN) and the table has no permissions, MYADMIN can run the query fine (as you would expect). Now make a macro (created by MYADMIN) with a RunSQL and copy and paste the SQL statement from the query window into its argument. MYADMIN can run this macro without a problem. Now log in as MYUSER and the query works fine HOWEVER the macro will not execute. It gives you an error message stating you don't have permissions.

References:

Microsoft Access User's Guide Chapter 25  
Microsoft Access Language Reference OpenQueryDef Method

[References](#)

## PRB: Zoom Box Does not Appear

Article Number: Q103175  
CREATED: 18-AUG-1993  
MODIFIED: 26-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

Pressing the SHIFT+F2 does not bring up the zoom box. In Microsoft Access the key combination SHIFT+F2 should display the [expression](#) zoom box which allows easy viewing and editing of expressions.

### CAUSE

---

Some possible causes include:

1. Low system resources within Windows.
2. A corrupted .LDB file.
3. Syntax errors in any of the modules in the open [database](#).

### RESOLUTION

---

To free up system resources, close as many running applications as possible. Some applications and drivers (printer/display) are known to "Leak" resources. This leak results from an application/driver requesting a certain amount of resources to run (i.e. 4%) but when the application/driver is closed it returns less than it used to run (i.e. 3%). Over time this leak leaves the Windows environment in a low system resource situation. In these cases, it may be necessary to occasionally exit and restart Windows in order to [refresh](#) resources.

To check Windows system resources while running Microsoft Access, press ALT+ESC on the keyboard to invoke Program Manager. From the Help Menu chose "About Program Manager". The help dialog will display the system resources.

Microsoft Access may generate an LDB file for Microsoft Access databases. The .LDB file maintains lock information about a specific database. Occasionally these files may get corrupted and cause problems including preventing the Zoom Box from appearing. If the LDB file is suspected of being corrupted a user may close the database in question and delete the associated LDB file (DBName.LDB). Access will create a new LDB file when/if it is deemed necessary.

The Zoom box itself is an Access [form](#) and the code that makes it work is created in [Access Basic](#). Microsoft Access will not run any code if the compiler contains [syntax](#) errors. If this is the case, try compiling the code in a Microsoft Access [module](#) to find out the source of the problem.

References:

For more information on LDB files, [query](#) in the Microsoft Access help on:

"Questions and Answers about .LDB files"

[References](#)

## PRB: Exporting Date Fields to Text Includes Time Format

Article Number: Q103174  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1  
-----

### SYMPTOMS

---

If you export a [table](#) that has a date [field](#) to a text file, Microsoft Access will automatically add the time format to the date in the text file. The information will be formatted as "mm/dd/yy hh:nn:ss".

### CAUSE

---

In Microsoft Access, a date is stored with a date/time format. During the export Microsoft Access creates both the date and time formats in the text file. If there is no data for the time format it will show up as all zeros (indicating 12am).

### RESOLUTION

---

There are two methods you can use to work around this problem:

Method 1: Create a [query](#) with a calculated field to format the date field appropriately. Export the results of the query. The details of this [method](#) are provided below.

Method 2: If your export needs can use a fixed width text file, you can export the data using the "Text (Fixed Width)" format, setting the size of the date field to 8 characters in the Import/Export specification. This will effectively truncate the time portion of the field during export.

Note: Common text exports require the data to be delimited using the "Text (Delimited)" format. Method 2 will not be useful in this case and is not described any further. For more information, see the Microsoft Access User's Guide, pages 78-79

### STATUS

---

This behavior is by design. This feature is under review and will be considered for revision in a future release.

### MORE INFORMATION

---

Step to Reproduce Behavior  
-----

1. Create a table with a new field called Birthday with "Data/Time" chosen as the [data type](#).

2. Choose Save from the File menu and save the table with a unique name, same "MyTable".
3. Choose Datasheet from the View menu, type "11/15/67" in the first record, and close the table.
4. Choose Export from the File menu.
5. Choose "Text (Delimited)" from the list of available formats and choose the OK button.
6. Choose the table saved in step 2, from the list of available tables and choose the OK button.
7. Type a unique name in the File Name box and choose the OK button.
8. In the Export Text Options dialog box, choose the OK button.
9. Open the exported text file in Notepad, with the name saved in step 7, or any text editor, and note that the date will be formatted as "11/15/67 00:00:00".

#### Work Around

-----

#### Method 1: Using a Query to Export Date Fields Appropriately

1. Create a new query and add the desired table to be exported.
2. Include all desired fields in the query grid ACCEPT for the date field(s).
3. Suppose the date field is called "Birthday", in an empty column, type the following in the FieldName box:

```
Birthday2: Format([Birthday], "mm/dd/yy")
```

Note: The new calculated field name must be unique from any field(s) in the table(s) that are included in the query.

4. Repeat step 3 for all other date fields.
5. Choose Make Table from the Query menu, type in a unique table name in the Table Name box, and choose OK.
6. Choose Run from the Query menu to execute the query.

The query will run with the results being stored into the new table named in step 5.

7. Close the query (saving it with a unique name if you will need it in the future).
8. Activate the Database window by pressing the F11 key, then export the new table using the Export command from the File menu.

#### References:

"Microsoft Access User's Guide," version 1.0, pages 78-79.

"Microsoft Access User's Guide," version 1.0, pages 162-167.

## [References](#)

## PRACC9307: Options in Form and Report Wizards Do Not Display

Article Number: Q103173  
CREATED: 18-AUG-1993  
MODIFIED: 16-SEP-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SYMPTOMS

---

When you select the Form Wizard button or the Report Wizard button, an empty list box is displayed.

### CAUSE

---

The [Form Wizards] and [Report Wizards] sections of the MSACCESS.INI file either do not exist or are empty.

### RESOLUTION

---

Using Notepad or another text editor, add the following sections in your MSACCESS.INI file, which is located in your WINDOWS subdirectory:

```
[Form Wizards]
Single-Column=zwInitWizard, 1
Tabular=zwInitWizard, 2
Graph=zwInitWizard, 6
Main/Subform=zwInitWizard, 7
```

```
[Report Wizards]
Single-Column=zwInitWizard, 3
Groups/Totals=zwInitWizard, 4
Mailing Label=zwInitWizard, 5
```

```
[Libraries]
wizard.mda=ro
```

### REFERENCES

=====

"User's Guide," version 1.1, page 332

### [References](#)

## PRB: Printing on Avery 5267 labels

Article Number: Q103171  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

When printing to the Avery 5267 labels, care needs to be taken when setting up the page otherwise the text will appear to "creep" down the page.

### CAUSE

---

The Avery 5267 labels are not cut on the page in precisely 1/2 inch increments. The labels are slightly larger than 1/2 inch so if you print in 1/2 inch increments then the data will appear to "creep" down the page.

### RESOLUTION

---

When placing the controls in the report be sure to allow room for the label discrepancy.

Below are guidelines for printing three lines of data on the Avery 5267 labels. The example uses a function which concatenates the fields together for optimum use of space on the within each label.

1. From Print Setup, set the Top and Bottom margins to .50 inches and set the Left and Right margins to .25 inches. Press the More button and set Items Across to 4, Row Spacing to 0, and Column Spacing to 0.3 inches. Under Item Size, the Same as Detail option should be checked.
2. Make the Width of the report 1.75 inches
3. Set the Height property for the Detail section to .5 inches and set the CanGrow and CanShrink properties to No.
4. Locate one text box control and set the properties manually as noted:

```
Control: Text Box
 Top: .01
 Height: .48
 CanGrow: Yes
 CanShrink: Yes
 Font: Arial
 FontSize: 8
```

ControlSource:

```
=AddressBlock([First],[Last],[Address],[City],[State],[Zip])
```

5. Create a new [Access Basic module](#), and type the following code into it:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

Option Explicit

```
Function AddressBlock$ (First, Last, Address, City, State, Zip)
```

```
 Dim A1$, A2$, A3$, CR$
```

```
 CR$ = Chr(13) & Chr(10) 'Carriage return and line feed
```

```
 A1$ = IIf(ISB(First),"",First & " " & IIf(ISB(Last),"",Last _
 & CR$))
```

```
 A2$ = IIf(ISB(Address),"",Address & CR$)
```

```
 A3$ = City & ", " & State & " " & Zip
```

```
 AddressBlock = A1$ & A2$ & A3$ 'Concatenate the strings.
```

```
End Function
```

```
Function ISB (V) As Integer
```

```
 If IsNull(V) or V = "" Then ISB = True Else ISB = False
```

```
End Function
```

The spacing above will create a .01 reserve above and below the text control. Variations of this layout will also print. Note: the Mailing Label Report Wizard creates two controls each with a height of .17, leaving a reserve below and above the text for label adjustments.

For another example of this search the Knowledge Base for the following keywords:

white and space and canshrink and reports

## [References](#)

## INF: Where Condition Refers to Fields not Controls

Article Number: Q103140  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

Page 348 of the Microsoft Access Language Reference states under the Where Condition argument of the OpenForm action:

"To open a form in which the value of one of its controls equals the value of a control on another form, use this syntax: [controlname on opened form] = Forms!formname![controlname on other form]"

This is incorrect. It should state:

"To open a form and show only records where one of the fields matches the value of a control on another form, use this syntax: [field in opened form's recordsource] = Forms!formname![controlname on other form]"

### EXPLANATION

=====

The Where Condition filters the records in the form's dynaset. Access compares the specified field(s) in each record of the form being opened to the control(s) on the initial form. Records are included in the dynaset only if the field(s) satisfy the Where Condition. The presence or absence of controls on the form being opened does not affect the records that are included in the dynaset. i.e. the Where Condition operates on the fields shown in the Field List, which can be seen during form design.

### EXAMPLE

=====

These steps will clarify the confusion that can be caused by this documentation error. The example uses the NWIND.MDB sample database.

Create the objects for the demo:

-----

1. Create a new unbound form with the following controls:

Textbox

-----

ControlName: ChosenState

Command Button

-----  
OnPush: Open Emp 2

2. Save the form as "ChooseState".
3. Create a second form bound to the Employees [table](#):

Textbox  
-----  
ControlName: Last Name  
ControlSource: Last Name

Textbox  
-----  
ControlName: State  
ControlSource: State

4. Save the form as "Emp 2".
5. Create a new [macro](#) with the following Action:

Action: OpenForm  
-----  
Formname: Emp 2  
Where Condition: [State] = Forms!ChooseState![ChosenState]

6. Save the macro as "Open Emp 2".

Test the objects:

- 
1. Open the form ChooseState, enter "WA" in the [text box](#), and press the button.

Note: The form ChosenState opens and shows only those employees that live in Washington state.

Demonstrate the issue:

- 
1. Open the form, ChosenState, in design mode. Change the ControlName property of the control bound to the field State to "Test".
  2. Close and save the form.
  3. Open the form ChooseState, enter "WA" in the text box, and press the [command button](#).

Note: The form ChosenState opens and show the same records. This demonstrates that you are not filtering the controls of the form but the underlying data.

References:  
Microsoft Access Language Reference, page 348

[References](#)

## INF: Function to Get Date of Monday Prior to Current Day

Article Number: Q103138  
CREATED: 18-AUG-1993  
MODIFIED: 17-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

This article contains a [function](#) to find the date of the Monday prior to the current day.

### MORE INFORMATION

---

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

The following function determines Monday's date according to the following [criteria](#):

- If the current date is Tuesday through Sunday, it yields the prior Monday's date.
- If the current day is Monday, it returns the current date.
- If the date is [Null](#) or is not a valid date, it returns Null.

```
Function GetMonDate(CurrentDate)
 If VarType(CurrentDate)<>7 then
 GetMonDate=Null
 Else
 Select Case WeekDay(CurrentDate)
 Case 1 ' Sunday
 GetMonDate=CurrentDate-6
 Case 2 ' Monday
 GetMonDate=CurrentDate
 Case 3 to 7 ' Tuesday..Saturday
 GetMonDate=CurrentDate-WeekDay(CurrentDate)+2
 End Select
 End If
End Function
```

To test the above function, type the following in the Immediate window:

```
Print GetMonDate(#4/30/93#)
```

The result is as follows:

#4/26/93#

[References](#)

## PRB: Undo Cut Does Not Restore Previous Clipboard Contents

Article Number: Q103137  
CREATED: 18-AUG-1993  
MODIFIED: 20-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

Choosing Undo Cut from the Edit menu does not restore the [Clipboard](#) to its previous state.

### RESOLUTION

---

This behavior is by design.

### MORE INFORMATION

---

There are many cases in which restoring previous data to the Clipboard does not work (for example, if Clipboard data was imported from another application).

Steps to Reproduce Behavior

-----

1. Create a new, unbound [form](#).
2. Add a [label](#) containing the text "Hello".
3. Add another label containing the text "Good-bye".
4. Highlight the "Hello" label to select it. From the Edit menu, choose Copy. The Clipboard now contains this label.
5. Highlight the "Good-bye" label to select it. From the Edit menu, choose Cut. The previous contents of the Clipboard have been replaced with the "Good-bye" label.
6. From the Edit menu, choose Undo Cut. From the Edit menu again, choose Paste.

The current contents of the Clipboard (the "Good-bye" label) is pasted onto your form. The previous contents of the Clipboard (the "Hello" label) was restored.

### REFERENCES

=====

For more information, search for "undo changes", then "undoing changes" using the Microsoft Access Help menu.

### [References](#)



## PRACC9308: Open OLE Objects Not Saved When Record Changes

Article Number: Q103136  
CREATED: 18-AUG-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

If you are editing an OLE object that is opened from an OLE field and you then move to a different record, there is no indication that the record was changed. The OLE object in the OLE server is closed and all changes to the object are lost.

### RESOLUTION

---

Version 1.0 Workaround  
-----

One way to work around this problem in Microsoft Access version 1.0 is to call an Access Basic function in the OnExit property of the object. The function prompts the user to verify that the object has been updated. The following is an example of this function:

1. Place the function called =ProcessOLEObject() in the OnExit property of the OLE object.
2. Create the following module:

```
Function ProcessOLEObject ()
 Saved = MsgBox("Have you saved your changes for this object?", 36)
 If Saved = 7 Then
 DoCmd CancelEvent
 DoCmd DoMenuItem A_FormBar, A_EditMenu, A_Object, A_ObjectUpdate
 End If
End Function
```

Every time the user passes through the OLE field, the function is run and the dialog box is displayed. Unfortunately, there is no way to test whether or not the object has been edited.

Version 1.1 Workaround  
-----

In Microsoft Access version 1.1, a dialog box appears, indicating that the object is editable and giving the user the option to update the object before moving on to the next record.

For example, the following message appears after you open one of the pictures in the Employees table in the sample database NWIND.MDB:

"The object "Paintbrush Picture" is open for editing and may have been modified. Update before saving? Yes, No, Cancel or Help."

#### Steps to Reproduce Problem

1. Open the sample database NWIND.MDB in Microsoft Access version 1.0.
2. Open the Employees [form](#).
3. Double-click the image that you want to edit. This action will start Paintbrush.
4. Make some change the image.
5. Press ALT+TAB to return to Microsoft Access without saving the image.
6. Move to the next record. The record will now change.

If you press CTRL+ESC to view the task list, you will note that Paintbrush is no longer available. If you go back to the record you were editing, the picture will not contain the changes you made previously.

#### [References](#)

## INF: A Macro to Check for Duplicate Values in a Field

Article Number: Q103135  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This article describes a simple [macro](#) with a condition using the DLookup() [function](#) that checks for duplicates.

### More Information:

---

Microsoft Access provides automatic checking for duplicate values in a [field](#) which has been designated as the Primary Key. However, this check occurs after all fields for a [record](#) have been entered, just before the record is committed to the [database](#). This article explains how to create a macro that will perform a duplicate check as the data in a field is entered.

1. In the Nothwind database, create a new macro with the following condition:

```
DLookup("[Customer ID]", "Customers", "[Customer ID] =
Form.[Customer ID]") is Not Null
```

The action corresponding to this condition is your choice. A good alternative is a Message Box that alerts the user to the entry of a duplicate value.

#### Steps to Reproduce Behavior

-----

1. Open the NWIND database.
2. Create a new [form](#) based on the Customers [table](#).
3. Create a new macro as follows:

| Macro Name | Condition  | Action   |
|------------|------------|----------|
| MyMacro    | *see above | MsgBox   |
|            | ...        | SendKeys |

#### MyMacro Actions

-----

MsgBox  
Message: That value already exists.

SendKeys  
Keystrokes: +{TAB}

Wait: No

The ellipses (...) under the condition indicates that the condition controls both macro actions. The SendKeys action will return the user to the field just exited.

4. In Design view for the form, attach this macro to the AfterUpdate property of the Customer ID control.
5. In Form view, make a note to the the value of the current Customer ID.
6. From the menu, choose Records, DataEntry to go to a new record.
7. In the Customer ID control, enter the value of the earlier noted Customer ID. The Message Box with your message should appear and the cursor should be returned to that control.

References:

"Microsoft Access User's Guide," versions 1.0 and 1.1, Chapter 22, "Using Macros with Forms".

"Microsoft Access User's Guide," version 1.0, "Validation Expressions" on page 652.

[References](#)

## INF: How to Search the CompuServe Knowledge Base

Article Number: Q103134  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This article describes how to access the Microsoft Product Support Knowledge Base (MSKB) Forum on the CompuServe Information Service (CIS).

CompuServe Information Service is independent of Microsoft; we make no warranty, implied or otherwise, regarding performance or reliability.

### MORE INFORMATION

---

The MSKB is completely menu driven and will help the user [query](#) the Knowledge Base for specific articles that might be helpful to the user.

Articles are broken up into three different groups, PRB, INF and PRACC articles. PRB articles are perceived problems but may actually be associated with "by-design", or incorrect use of the product. INF articles are articles that fall into the category of "nice to know", or advanced tips and tricks. PRACC articles are known problems with Microsoft Access or another application that may cause problems with Microsoft Access such as video drivers.

The text below is a sample script in which a user is searching for all articles that are associated with PRACC articles. The >> symbol is only used in this article to point out user input.

=====  
Start of sample script  
=====

>>GO MSKB

One moment please...

Knowledge Base+

Welcome to The Microsoft Knowledge Base  
Copyright (c) 1993 Microsoft Corporation

- 1 What's New in the Knowledge Base
- 2 Description of Database
- 3 Online User's Guide
- 4 Search the Knowledge Base

- 5 Search using Expert Mode
- 6 Quick Search - By Document Id Number
- 7 Microsoft Software Library

>>4

Knowledge Base+

Current selection: 33378 record(s)

SELECT SEARCH CRITERIA:

- 1 Product Name [ ]
- 2 Product Version [ ]
- 3 Publication Date [ ]
- 4 Operating Environment [ ]
- 5 Document Type [ ]
- 6 Document Text [ ]
- 7 Title Text [ ]
- 8 Display Results
- 9 Begin a New Search

>>7

You may enter up to 3 words/phrases, one at each prompt.  
Only records indexed under all specified words/phrases will be located.

Words/phrases:

1: >>PRACC

and 2:

Knowledge Base+

Current selection: 79 record(s)

SELECT SEARCH CRITERIA:

- 1 Product Name [ ]
- 2 Product Version [ ]
- 3 Publication Date [ ]
- 4 Operating Environment [ ]
- 5 Document Type [ ]
- 6 Document Text [ ]
- 7 Title Text [ PRACC ]
- 8 Display Results
- 9 Begin a New Search

>>8

Knowledge Base+

Search Results: 7 records

Enter "/DOWN" to download document

- 1 PRACC9301: Wrong RowSourceType for combo box Causes GP Fault
- 2 PRACC9302: Importing Delimited Text Files Strips Off Decimals
- 3 PRACC9303: Screen Redraw Problems When Reordering Groups

- 4 PRACC9303: Error: Not Enough System Resources in Access
- 5 PRACC9303: Setting OLE Object Property May Cause GP Fault
- 6 PRACC9303: Orders Form Doesn't Print/Preview SubTotal & Total
- 7 PRACC9210: '#' Appears in Attached Table Name in Error Message

=====  
End of sample script  
=====

References:

[References](#)

## PRACC980: DISTINCTROW Keyword Ignored in a Cartesian Product

Article Number: Q103133  
CREATED: 18-AUG-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SYMPTOMS

---

The query dynaset includes duplicate records, which should have been eliminated by the presence of the keyword DISTINCTROW.

### CAUSE

---

This problem occurs if no join is specified between the tables in the query.

### RESOLUTION

---

To hide the duplicate records, verify that the query output includes one field that uniquely identifies each record, then replace the keyword DISTINCTROW with the keyword DISTINCT.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

#### STEPS TO REPRODUCE

=====

1. Open the sample database NWIND.MDB.
2. Create a new query based on the Customers and Orders tables. Join the tables on the CustomerID field.
3. Drag the Contact Name field to the query grid and run the query. Note that your dynaset contains 90 records.
4. Switch to Design view and delete the join line between the tables.
5. Add the following criteria in the Contact Name field on the query grid:  
  
Customers.[Customer ID] = Orders.[Customer ID]
6. Run the query. The dynaset now contains 1078 records. Contact names from companies that have placed multiple orders are displayed multiple times.

## Workaround

-----

You can hide the duplicate records by replacing the keyword DISTINCTROW with the keyword DISTINCT. This action is equivalent to setting Query Properties to Unique Records Only. It is very important to include a query output field that is unique for each record. In this example, Contact Name is not a good choice; two different companies may have contacts with identical names. Including the Company ID field in the query ensures that all unique companies are included.

## REFERENCES

=====

Search for "ALL," "DISTINCT," "DISTINCTROW," and "Predicates ([SQL](#))" using the Help menu.

"Microsoft Access User's Guide," version 1.1, page 187

[References](#)

## INF: OpenQuery is not a QueryDef Method

Article Number: Q103132  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### Summary:

---

Two of the examples on pages 132-133 of the Microsoft Access "Introduction To Programming" attempt to use an OpenQuery [method](#). OpenQuery is an action. OpenQueryDef is a method and should be used in these two examples.

### More Information:

---

In the example on page 132:

```
Dim db As Database, Q As QueryDef, DataRecs As Dynaset
Set db = CurrentDB()
Set Q = db.OpenQuery("Sales Totals")
Set DataRecs = Q.CreateDynaset()
```

Change the Third line to:

```
Set Q = db.OpenQueryDef("Sales Totals")
```

In the example on page 133:

```
Dim db As Database, Q As QueryDef
Set db = CurrentDB()
Set Q = db.OpenQuery("Update Sales")
Q.Execute
Q.Close
```

Change the Third line to:

```
Set Q = db.OpenQueryDef("Update Sales")
```

For more information search for "OpenQueryDef", then "OpenQueryDef Method" using the Help menu.

### [References](#)

## PRACC9308: Error Previewing Subreports that Call Filter Macro

Article Number: Q103131  
CREATED: 18-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

When you attempt to print a form or report containing a subreport/subform that calls a macro that in turn executes an ApplyFilter action, then you call that macro from the OnOpen property of the subreport/subform, the following error message is displayed:

Action isn't valid because the form or report isn't bound to a table or query.

Even if you do not receive this error message, note that the ApplyFilter action has not been executed. Currently, this functionality is not available.

### CAUSE

---

When printing, Microsoft Access erroneously attempts to apply the filter that is run from the macro to the outermost (main) report/form, rather than to the subreport/subform.

NOTE: The error does not occur if the main report/form contains only the subreport control and no other controls. However, the data from the subreport/subform still is not filtered.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

Steps to Reproduce Problem  
-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new macro that executes an ApplyFilter action. Enter criteria such as the following for the Where condition:

[Lastname]="Fuller"

3. Create a new, blank report bound to the Employees table. Place a

control, such as a [text box](#), on it.

4. Drag the Employee Sales Subreport from the [Database window](#) to create a subreport on the new report.
5. Double-click the subreport to edit it. For the OnOpen property of the subreport, select the macro you created in step 2 from the [combo box](#).
6. Preview the report. The following error message appears:

Action isn't valid because the form or report isn't bound to a table or query.

Adding a text box to the main report allows you to print the form or report, but does not filter the data from the subreport/subform.

## [References](#)

## INF: How to Export the Results of a Query

Article Number: Q103130  
CREATED: 18-AUG-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

Microsoft Access does not provide a direct way to export the results of a [query](#) directly to a file using the Export command from the File menu. After the user selects Export from the File menu and selects an output format, the "Select Microsoft Access Object" dialog is presented which only lists tables, not queries, from the current [database](#).

### More Information:

---

The following information discusses two ways to export the results of a query to a file.

Method 1:

-----  
Uses a [method](#) of the Transfer [macro](#) actions (TransferText, TransferDatabase, TransferSpreadsheet) to export the query results directly to a file.

Method 2:

-----  
Discusses how to use a Make Table query to save the results of the query to a new [table](#). Once this table exists, it can be exported by choosing Export from the File menu.

More Information:

-----  
Method 1: Exporting Query Results Using a Transfer Macro Action  
-----

The three Transfer macro actions: TransferText, TransferSpreadsheet, and TransferDatabase were designed to help automate the process of exporting (and importing) data from the file menu. These commands can also be used to facilitate exporting data from a query directly to a file.

The following examples use the "Employee Sales for 1991" query included with the Northwind Trader (NWIND.MDB) database supplied with Microsoft Access.

Example A: - Exporting the query to a comma [delimited text file](#) using  
----- TransferText

1. Open the Northwind Traders database (NWIND.MDB) that comes with Microsoft Access.
2. Choose the Macro button, then the New button, from the Database window. This will present you with a new macro grid containing Action and Comments columns.
3. In the Action column choose TransferText.
4. Below the macro grid you will be presented with the arguments that need to be filled in for the TransferText action. Fill them in with the following settings:

Action: TransferText

-----

Transfer Type: Export Delimited  
 Specification Name: <leave empty>  
 Table Name: Employee Sales for 1991  
 File Name: C:\EMPSALES.TXT  
 Has Field Names: Yes

Note: You can specify the name of a query in the Table Name argument.

5. Choose Save from the File menu, type a unique macro name, and press ENTER.
6. Choose Run from the Macro menu to execute the macro. Microsoft Access will run the query and save the results to the specified destination file name (EMPSALES.TXT).

Example B: - Exporting the query as a DBase IV file using  
 ----- TransferDatabase:

1. Follow steps 1-3 for TransferText, but choose TransferDatabase as the macro action.
2. Fill in the macro arguments with the following settings:

Action: TransferDatabase

-----

Transfer Type: Export  
 Database Type: dBASE IV  
 Database Name: c:\  
 Object Type: Query  
 Source: Employee Sales for 1991  
 Destination: EMPSALES  
 Structure Only: No

3. Follow steps 5-6 for TransferText.

Example C: - Exporting the query as an Excel spreadsheet using  
 ----- TransferSpreadsheet:

1. Follow steps 1-3 for TransferText, but choose TransferSpreadsheet as the macro action.

2. Fill in the macro arguments with the following settings:

Action: TransferSpreadSheet

-----

Transfer Type: Export  
Database Type: Microsoft Excel  
Table Name: Employee Sales for 1991  
File Name: c:\empsales.xls  
Has Field Names: No  
Range: <leave empty>

Note: You can specify the name of a query in the Table Name argument.

3. Follow steps 5-6 for TransferText.

-----  
Method 2: Exporting Query Results Using a Make Table Query  
-----

A Make Table query can be used to export the results of a query to a table in Microsoft Access. Once the data is in a new table, the Export command from the File menu can be used to export the data to a file.

The disadvantage to this approach is that creating a new table can increase the size of your database. After you export the data, the table is no longer needed. If you delete the table, Access will not reclaim the once occupied space until you compact the database file. See "Compacting a Database", page 627 in the Microsoft Access User's Guide for more information.

The following examples use the "Employee Sales for 1991" query included with the Northwind Trader (NWIND.MDB) database supplied with Microsoft Access.

1. Open the Northwind Traders database (NWIND.MDB) that comes with Microsoft Access.
2. Choose the Query button in the Database window, select "Employee Sales for 1991", and choose the Design button.
3. Choose Make Table from the Query menu.
4. In the Table Name box, type a unique table name where the data will be exported to. Choose OK.
5. Choose Run from the Query menu. The query will execute with the results being stored in the new table.

You may want to save this query and give it a new name via the Save As command on the File menu if you need to export the results regularly.

6. Use the Export command from the File menu to export the new tables contents to file.

References :

[References](#)

## PRB: #NAME? Referring to Combo Box Column in Subform

Article Number: Q103129  
CREATED: 18-AUG-1993  
MODIFIED: 20-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

Normally, you can refer to a particular column of a combo box in a text box of a form using the .column property as follows:

```
=Forms![Formname]![ControlName].Column(x).
```

If, however, the combo box is in a subform, the textbox returns:

```
#Name?
```

### RESOLUTION

---

To work around this, the textbox must refer to the column of the combo box in the subform through the main form, even though the textbox is actually on the subform.

Given a situation where a subform has a combo box and a text box, the following syntax returns the correct result:

```
=Forms![MainForm]![Subform].Form![Controlname].Column(x)
```

NOTE: Columns are numbered from 0-n, not 1-n.

### [References](#)

## PRB: Tops of Letters May Be Cut Off with Some PostScript Fonts

Article Number: Q102921  
CREATED: 12-AUG-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0
- 

### SYMPTOMS

---

If you print Microsoft Access documents to a PostScript printer using certain PostScript fonts, the tops of taller letters may be cut off, even though the letters display correctly in Print Preview.

Microsoft has reproduced this problem with the CoventryScript font, supplied in the Fluent Laser Fonts package manufactured by Casady & Greene, Inc. This problem may also occur with other PostScript Type 1 fonts that have tall ascenders or tall capital letters.

### CAUSE

---

This problem occurs because of the [method](#) Microsoft Access uses to define cutoff regions for [control](#) boxes.

### RESOLUTION

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

### MORE INFORMATION

---

Fluent Laser Fonts are manufactured by Casady & Greene, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

### [References](#)

## INF: List of Articles Relating to MSACCESS.INI File

Article Number: Q102761  
CREATED: 10-AUG-1993  
MODIFIED: 10-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

Below is a list of articles in the knowledgebase relating in some manner to the MSACCESS.INI file:

| ARTICLE# | DESCRIPTION                                                       |
|----------|-------------------------------------------------------------------|
| -----    | -----                                                             |
| Q88175   | Refers to the [Libraries] section.                                |
| Q88648   | Refers to the [Menu Add-Ins] section.                             |
| Q90105   | Refers to the [Options] section.                                  |
| Q98224   | Refers to the [Options] section.                                  |
| Q96996   | Refers to the [Options] section.                                  |
| Q90170   | Refers to the [Microsoft Access] section (Tutorial setting only). |
| Q95051   | Refers to the [SQLSERVER] section of the WIN.INI file.            |
| Q101323  | Refers to the [ISAM] section.                                     |
| Q95388   | Refers to the [dBase ISAM] section (Deleted setting only).        |
| Q98805   | Refers to the [Paradox ISAM] section (CollatingSequence only).    |
| Q100176  | Refers to the [ODBC] section.                                     |
| Q100972  | Refers to the [ODBC] section.                                     |
| Q95050   | Refers to the ODBC.INI file.                                      |

### [References](#)

## INF: List of Articles Relating to MSACCESS.INI File

Article Number: Q102759  
CREATED: 10-AUG-1993  
MODIFIED: 10-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

Below is a list of articles in the knowledgebase relating in some manner to the MSACCESS.INI file:

| ARTICLE# | DESCRIPTION                                                       |
|----------|-------------------------------------------------------------------|
| -----    | -----                                                             |
| Q88175   | Refers to the [Libraries] section.                                |
| Q88648   | Refers to the [Menu Add-Ins] section.                             |
| Q90105   | Refers to the [Options] section.                                  |
| Q98224   | Refers to the [Options] section.                                  |
| Q96996   | Refers to the [Options] section.                                  |
| Q90170   | Refers to the [Microsoft Access] section (Tutorial setting only). |
| Q95051   | Refers to the [SQLSERVER] section of the WIN.INI file.            |
| Q101323  | Refers to the [ISAM] section.                                     |
| Q95388   | Refers to the [dBase ISAM] section (Deleted setting only).        |
| Q98805   | Refers to the [Paradox ISAM] section (CollatingSequence only).    |
| Q100176  | Refers to the [ODBC] section.                                     |
| Q100972  | Refers to the [ODBC] section.                                     |
| Q95050   | Refers to the ODBC.INI file.                                      |

### [References](#)

## INF: List of Articles Relating to MSACCESS.INI File

Article Number: Q102758  
CREATED: 10-AUG-1993  
MODIFIED: 10-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

Below is a list of articles in the knowledgebase relating in some manner to the MSACCESS.INI file:

| ARTICLE# | DESCRIPTION                                                       |
|----------|-------------------------------------------------------------------|
| -----    | -----                                                             |
| Q88175   | Refers to the [Libraries] section.                                |
| Q88648   | Refers to the [Menu Add-Ins] section.                             |
| Q90105   | Refers to the [Options] section.                                  |
| Q98224   | Refers to the [Options] section.                                  |
| Q96996   | Refers to the [Options] section.                                  |
| Q90170   | Refers to the [Microsoft Access] section (Tutorial setting only). |
| Q95051   | Refers to the [SQLSERVER] section of the WIN.INI file.            |
| Q101323  | Refers to the [ISAM] section.                                     |
| Q95388   | Refers to the [dBase ISAM] section (Deleted setting only).        |
| Q98805   | Refers to the [Paradox ISAM] section (CollatingSequence only).    |
| Q100176  | Refers to the [ODBC] section.                                     |
| Q100972  | Refers to the [ODBC] section.                                     |
| Q95050   | Refers to the ODBC.INI file.                                      |

### [References](#)

## INF: List of Articles Relating to MSACCESS.INI File

Article Number: Q102745  
CREATED: 10-AUG-1993  
MODIFIED: 10-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

Below is a list of articles in the knowledgebase relating in some manner to the MSACCESS.INI file:

| ARTICLE# | DESCRIPTION                                                       |
|----------|-------------------------------------------------------------------|
| -----    | -----                                                             |
| Q88175   | Refers to the [Libraries] section.                                |
| Q88648   | Refers to the [Menu Add-Ins] section.                             |
| Q90105   | Refers to the [Options] section.                                  |
| Q98224   | Refers to the [Options] section.                                  |
| Q96996   | Refers to the [Options] section.                                  |
| Q90170   | Refers to the [Microsoft Access] section (Tutorial setting only). |
| Q95051   | Refers to the [SQLSERVER] section of the WIN.INI file.            |
| Q101323  | Refers to the [ISAM] section.                                     |
| Q95388   | Refers to the [dBase ISAM] section (Deleted setting only).        |
| Q98805   | Refers to the [Paradox ISAM] section (CollatingSequence only).    |
| Q100176  | Refers to the [ODBC] section.                                     |
| Q100972  | Refers to the [ODBC] section.                                     |
| Q95050   | Refers to the ODBC.INI file.                                      |

### [References](#)

## INF: List of Articles Relating to MSACCESS.INI File

Article Number: Q102673  
CREATED: 08-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

The following is a list of articles found in the Microsoft Knowledge Base that relate in some way to the MSACCESS.INI file:

| ARTICLE# | TITLE & DESCRIPTION                                                                                                                             |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| -----    | -----                                                                                                                                           |
| Q88175   | INF: Creating, Debugging, and Using an Access Library<br>(Refers to the [Libraries] section)                                                    |
| Q88648   | INF: Easy Way to Add Items to the Help Menu<br>(Refers to the [Menu Add-Ins] section)                                                           |
| Q90105   | INF: Couldn't Find SYSTEM.MDA or UTILITY.MDA File<br>(Refers to the [Options] section)                                                          |
| Q98224   | PRB: ErrMsg: Couldn't Open Table MSysAccounts<br>(Refers to the [Options] section)                                                              |
| Q96996   | INF: How to Change the Font for the Zoom Box<br>(Refers to the [Options] section)                                                               |
| Q90170   | INF: How to Make Access Opening Screen Display<br>(Refers to the [Microsoft Access] section, Tutorial<br>setting only)                          |
| Q95051   | PRB: Unable to Connect to Attached <a href="#">SQL</a> Server Table,<br>Error Msg<br>(Refers to the [SQLSERVER] section of the WIN.INI<br>file) |
| Q101323  | INF: Five Parameters to Set in ISAM Section of<br>MSACCESS.INI<br>(Refers to the [ISAM] section)                                                |
| Q95388   | INF: Doc Err: dBASE SET DELETED Characteristic in<br>MS Access<br>(Refers to the [dBASE ISAM] section, Deleted<br>setting only)                 |
| Q98805   | PRB: Internal Database Error (-5207) When Attaching<br>Tables<br>(Refers to the [Paradox ISAM] section,<br>CollatingSequence only)              |

- Q100176      INF: Upper Limit on QueryTimeout and LoginTimeout  
(Refers to the [\[ODBC\]](#) section)
- Q100972      INF: Forcing Access to Use [Snapshot](#) Method on Attached  
Data  
(Refers to the [\[ODBC\]](#) section)
- Q95050      PRB: Data Source Not Found, No Default Driver Specified  
Error  
(Refers to the ODBC.INI file)

[References](#)

## INF: How to Simulate Changing RecordSource Property of Report

Article Number: Q102672  
CREATED: 08-AUG-1993  
MODIFIED: 25-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

Although you cannot actually change the RecordSource property of a [report](#), you can simulate the change by using a OpenReport [macro](#) action and calling a different [query](#) source from the Filter Name argument.

### MORE INFORMATION

---

The Filter Name argument must reference a query that contains the same number of fields (with the same names) as appear in the existing report. If the fields do not exist in the query, the report may display the "#Name?" error message for controls bound to those fields.

To simulate changing the RecordSource property of a report called MyReport, do the following:

1. Create a new report called MyReport by choosing New from the File menu and following the prompts.
2. Create a new query called MyNewQuery, using the same fields that appear in MyReport.
3. Create a macro with one action as follows:

```
OpenReport

Report Name: MyReport
View: Print Preview
Filter Name: MyNewQuery
```

NOTE: A [SQL](#) statement cannot be substituted for the query name.

Using this [method](#), you can use the same report to print data from multiple data sources.

### REFERENCES

=====

"User's Guide," version 1.0, Chapter 23

### [References](#)

## INF: Using Access Basic CopyFile() Function to Copy Disk Files

Article Number: Q102671  
CREATED: 08-AUG-1993  
MODIFIED: 16-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Access Basic does not have a command, such as the MS-DOS COPY command, to copy a disk file.

This article includes two variations of an Access Basic function called CopyFile(), both of which allow you to copy disk files.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

### MORE INFORMATION

---

Two variations of the Access Basic CopyFile() function is provided below. The first version uses only Microsoft Access Basic code; the second leverages some function calls to the Windows application programming interface (API) to optimize performance, particularly with larger files.

NOTE: You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive the duplicate procedure name error message when you compile or run your code, remove or comment out the appropriate declarations statement from your code.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

Method 1: CopyFile() Function Using Standard Access Basic Code

-----

```
'*****
' DECLARATION SECTION
'*****
Option Explicit

'*****
' FUNCTION: CopyFile()
' PURPOSE:
' Facilitates copying a disk file.
' ARGUMENTS:
' Source - The path\filename of the file to copy from.
```

```

' Destination - The path\filename of the file to copy to.
' RETURN:
' The length of the file copied.
'*****
Function CopyFile (ByVal Source$ ByVal Destination$) As Long
 Dim Index As Integer, NumBlocks As Integer
 Dim FileLength As Long, LeftOver As Long, AmountCopied As Long
 Dim SourceFile As Integer, DestFile As Integer
 Dim FileData As String
 Dim RetVal As Variant
 Const BlockSize = 32768

 On Error GoTo Err_CopyFile

 ' Remove the destination file.
 DestFile = FreeFile
 Open Destination For Output As DestFile
 Close DestFile

 ' Open the source file to read from.
 SourceFile = FreeFile
 Open Source For Binary Access Read As FreeFile

 ' Open the destination file to write to.
 DestFile = FreeFile
 Open Destination For Binary As DestFile

 ' Get the length of the source file.
 FileLength = LOF(SourceFile)

 ' Calculate the number of blocks in the file and left over.
 NumBlocks = FileLength \ BlockSize
 LeftOver = FileLength Mod BlockSize

 ' Create a buffer for the leftover amount.
 FileData = String$(LeftOver, 32)

 ' Read and write the leftover amount.
 Get SourceFile, , FileData
 Put DestFile, , FileData

 ' Create a buffer for a block to be read.
 FileData = String$(BlockSize, 32)

 ' Read and write the remaining blocks of data.
 For Index = 1 To NumBlocks
 ' Read and write one block of data.
 Get SourceFile, , FileData
 Put DestFile, , FileData
 Next Index

 Close SourceFile, DestFile
 CopyFile = AmountCopied

Bye_CopyFile:
 Exit Function

```

```
Err_CopyFile:
 CopyFile = -1 * Err
 Resume Bye_CopyFile
```

```
End Function
```

```
Method 2: CopyFile() Function Using Standard Access Basic Code
```

```

'*****
' DECLARATION SECTION
'*****
Declare Function fWrite Lib "kernel" Alias "_lwrite" (ByVal hFile%,
 ByVal lpBuff&, ByVal nBuff%) As Long
Declare Function fRead Lib "kernel" Alias "_lread" (ByVal hFile%,
 ByVal lpBuff&, ByVal nBuff%) As Long
Declare Function GlobalAlloc Lib "Kernel" (ByVal wFlags%,
 ByVal dwBytes&) As Integer
Declare Function GGlobalFree Lib "kernel" (ByVal hMem%) As Long
Declare Function GlobalLock Lib "Kernel" (ByVal hMem%) As Long
Declare Function GGlobalUnlock Lib "kernel" (ByVal hMem%) As Long

'*****
' FUNCTION: CopyFile()
' PURPOSE:
' Facilitates copying a disk file.
' ARGUMENTS:
' Source - The path\filename of the file to copy from.
' Destination - The path\filename of the file to copy to.
' RETURN:
' The length of the file copied, if successful.
'*****
Function CopyFile (ByVal Source$, ByVal Destination$) As Long
 Dim FileLength As Long, AmountCopied As Long
 Dim RetVal As Variant, lpBuff As Long
 Dim DestFile As Integer, SourceFile As Integer
 Dim DestDOS As Integer, SourceDOS As Integer
 Dim ApiErr As Integer, AmtRead As Integer
 Dim hMem As Integer
 Const nBuff = 32767
 Const wFlags = &H20

 On Error GoTo Err_CopyFile

 ' Get the size of the file.
 SourceFile = FreeFile
 Open Source For Binary Access Read As SourceFile
 FileLength = LOF(SourceFile)
 Close SourceFile

 ' Allocate and lock memory to buffer file contents.
 hMem = GlobalAlloc(wFlags, nBuff)
 lpBuff = GlobalLock(hMem)

 ' Open the source file to read from.
 SourceFile = FreeFile
 Open Source For Input As SourceFile Len = 1
```

```

' Open the destination file to write to.
DestFile = FreeFile
Open Destination For Output As DestFile Len = 1

' Get the operating system file handles.
DestDOS = FileAttr(DestFile, 2)
SourceDOS = FileAttr(SourceFile, 2)

Do
 ' Read and write nBuff worth of data.
 AmtRead = fRead(SourceDOS, ByVal lpBuff, nBuff)
 ApiErr = fWrite(DestDOS, ByVal lpBuff, AmtRead)
 AmountCopied = AmountCopied + AmtRead
Loop Until AmtRead = 0

Close SourceFile, DestFile

' Unlock and free the file buffer memory.
lpBuff = GlobalUnlock(hMem)
hMem = GlobalFree(hMem)

CopyFile = AmountCopied

Bye_CopyFile:
 Exit Function

Err_CopyFile:
 CopyFile = -1 * Err
 Resume Bye_CopyFile
End Function

```

The CopyFile() function can be tested from the [Immediate window](#), as follows:

1. Create or open a [module](#) in Microsoft Access.
2. From the View menu, choose Immediate Window.
3. Type the following and press ENTER:

```
? CopyFile(C:\WINDOWS\CHESS.BMP", "C:\CHESS.BMP")
```

The CopyFile() function can be invoked from an Access Basic function

```
DimRetVal As Long
RetVal = CopyFile("C:\WINDOWS\CHESS.BMP", "C:\CHESS.BMP")
```

or by using the RunCode [macro](#) action:

```
RunCode
Function Name: =CopyFile(C:\WINDOWS\CHESS.BMP", "C:\CHESS.BMP")
```

## [References](#)

## PRB: Filtering Form with OpenForm Action Causes "#Name?" Error

Article Number: Q102670  
CREATED: 08-AUG-1993  
MODIFIED: 26-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

A [form](#) that is opened with an OpenForm action, which specifies a [query](#) for the Filter Name argument, may display a "#Name?" error message for some controls.

### CAUSE

---

When a query is specified as the [filter](#) for the OpenForm action, the form is bound to the fields in the query. If the form contains controls bound to fields that are not included in the query, those controls display the "#Name?" error message.

### RESOLUTION

---

Switch to [Design view](#) and either change the query so that it contains all the fields used in the form, or clear the Restrict Available Fields [check box](#).

### MORE INFORMATION

---

A similar problem may occur with a [report](#) that is filtered by a query that does not contain all the same fields. The report gives parameter prompts for all fields not included in the query.

Steps to Reproduce Behavior

-----

1. In the sample [database](#) NWIND.MDB, create a query called Filter Customers By Letter A based on the Customers [table](#). Set the properties as follows:

Query: Filter Customers By Letter A

-----

Type: Select Query

Field: Customer ID  
Table: Customers  
Criteria: Like "a\*"

2. From the View menu, choose Query properties to verify that the Restrict Available Fields check box is selected.
3. Create the following [macro](#) and call it Open Filtered Customers:

| Macro Name                                | Action   |
|-------------------------------------------|----------|
| Open Filtered Customers                   | OpenForm |
| -----                                     |          |
| Open Filtered Customers Actions           |          |
| -----                                     |          |
| OpenForm                                  |          |
| Form Name: Customers                      |          |
| Filter Name: Filter Customers By Letter A |          |

4. Run the macro. Note that all the controls in the Customer form, except for [Customer ID], display the "#Name?" error message.

#### Workaround

-----

1. Open the Filter Customers By Letter A query in Design view.
2. From the View menu, choose Query Properties. Clear the Restrict Available Fields check box.
3. Choose OK and save the query. The form now displays the correct values for all controls.

#### REFERENCES

=====

"User's Guide," versions 1.0 and 1.1, pages 532-534, 558

#### [References](#)

## INF: How to Check for Duplicate Values in Primary Key Fields

Article Number: Q102527  
CREATED: 03-AUG-1993  
MODIFIED: 25-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Normally, Microsoft Access does not check the values in primary key fields for duplicates until you move to the next record. If you enter an invalid or duplicate value in a control (text box) and move to the next record, you may invalidate all previous entries. However, there is a method to force an immediate check for duplicate values.

### MORE INFORMATION

---

The following procedure uses the Customers form in the sample database NWIND.MDB. The Customer form is based on the Customers table; Customer ID is the primary key.

1. Create a new macro as follows. From the View menu, choose Conditions.

NOTE: The first condition should appear on one line and the line-continuation character (\_) should be removed. The condition column for the second action includes the ellipsis (...).

| MacroName | Condition                                                                                        | Action |
|-----------|--------------------------------------------------------------------------------------------------|--------|
| IsItDup   | DCount("[customer id]","customers",<br>"[customer id]=Form.[customer id]")<>0<br>... CancelEvent | MsgBox |

IsItDup Actions

-----  
MsgBox  
    Message: Duplicate ID  
CancelEvent

2. Open the Customers form in Design view. Change the BeforeUpdate property of the Customer ID field, as shown below:

Text Box: Customer ID  
    BeforeUpdate: IsItDup

When you attempt to add a duplicate Customer ID value, you receive an error message. The indicator remains in the Customer ID field, allowing you to change the value.

### References

## **PRACC9307: GP Fault: Printing Large Report with Low Disk Space**

Article Number: Q102526  
CREATED: 03-AUG-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### **SYMPTOMS**

---

You get a general protection (GP) fault when you try to print a large [report](#) and have very little free hard disk space.

### **RESOLUTION**

---

Either free up hard disk space or reduce the size of your report.

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### **MORE INFORMATION**

---

Steps to Reproduce Problem  
-----

1. Make sure that you have 12K or less of free disk space on your computer.
2. Start Microsoft Access and open the sample [database](#) NWIND.MDB.
3. Print the Customer Mailing Labels report.

You receive a GP fault.

[References](#)

## PRACC9309: SQORA.DLL Does Not Allow Lengthy SQL Statements

Article Number: Q102525  
CREATED: 03-AUG-1993  
MODIFIED: 14-SEP-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SYMPTOMS

---

When you are executing a [query](#) against, or updating a [record](#) in, an Oracle [table](#), the following error message is displayed

Statement was longer then allowable maximum 2000+ chars

if the table and [field](#) names are long or the query is complex.

### CAUSE

---

This occurs because of a problem with SQORA.DLL, the Oracle [ODBC](#) driver.

### RESOLUTION

---

Obtain and install the updated driver, or use queries to do updates, rather than updating records with the Oracle table in [Datasheet view](#). The query should yield only the columns to be updated.

For complex queries, reduce both the number of tables or joins in the query and the number of fields used or shown in the query.

These workarounds reduce the lengths of [SQL](#) statements.

### STATUS

---

Microsoft has confirmed this to be a problem in the Oracle ODBC driver shipped with Microsoft Access version 1.1 and Microsoft Visual Basic version 3.0 (Professional Edition). An updated driver that corrects this specific problem is available from Microsoft.

### MORE INFORMATION

---

How to Obtain the Updated Driver  
-----

ORA110.EXE, a self-extracting zipped file, can be downloaded from the MSACCESS forum on CompuServe (Library 11, ODBC Connectivity), or from the Microsoft Software Data Library (MSDL).

If you are unable to download the file, Microsoft Product Support Services (PSS) will send you the file on disk by mail. This disk also contains the updated Btrieve driver.

NOTE: This driver is for use by registered owners of Microsoft Access 1.1 and Visual Basic 3.0 (Professional Edition) only. By installing these files, you are indicating that you own one of these two products.

PSS Disk Contents

The PSS disk contains the following files:

README.TXT  
BTRV110.DLL  
ORACLE.TXT  
SQORA.DL\_  
SQORASTP.DL\_  
ODBC.INF  
SETUP.EXE

NOTE: The SETUP.EXE file is called by the ODBC control panel facility and will not run as a stand-alone file.

The following two files are actually updated:

|              | Old                                                        | New                                                        |
|--------------|------------------------------------------------------------|------------------------------------------------------------|
| SQORA.DLL    | Version: 1.00.2816<br>Size: 143,600 bytes<br>Date: 4/16/93 | Version: 1.00.3112<br>Size: 144,096 bytes<br>Date: 7/12/93 |
| SQORASTP.DLL | Version: 1.00.2403<br>Size: 9,328 bytes<br>Date: 5/7/93    | Version: 1.00.3106<br>Size: 9,632 bytes<br>Date: 7/6/93    |

A README.TXT file documents the changes to the Btrieve and Oracle drivers and the installation procedures for each.

Oracle drivers are manufactured by Oracle Corporation and Btrieve drivers by Novell, Inc., vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

## INF: How to Disable the Menu Bar When Previewing a Report

Article Number: Q102524  
CREATED: 03-AUG-1993  
MODIFIED: 25-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

Although Microsoft Access allows you to display a custom menu bar when in a form, there is no similar capability when previewing a report. However, you can completely remove the menu bar when in Print Preview using Microsoft Windows API calls. This option may be a more desirable to developers than displaying menu choices that are not pertinent to a particular application.

### MORE INFORMATION

---

The technique presented in this article removes the menu bar when in Print Preview. To also remove the toolbar, choose Options from the View menu and set Show Tool Bar to No.

The code below needs to be called for each report every time you switch to Print Preview. If you display another report after calling this code, Microsoft Access will re-establish the Print Preview menu.

To remove the menu bar that displays when you are printing a report, use the following steps:

1. Create the following new module:

```
'NOTE: In the following sample code, an underscore (_) is used as a
'line-continuation character. Remove the underscore when
're-creating this code in Access Basic.
```

```
'NOTE: You may have some Windows API functions defined in
'an existing Microsoft Access library; therefore, your
'declarations may be duplicates. If you receive a duplicate
'procedure name error message, remove or comment out the
'declarations statement in your code.
```

```
'*****
'Declarations section of the module
'*****
Option Explicit
Declare Function FindWindow% Lib "user" (ByVal lpClassName As Any, _
 ByVal lpCaption As Any)
Declare Function SetMenu% Lib "User" (ByVal hWnd%, ByVal hMenu%)

'=====
```

'The following function removes the Microsoft Access menu bar by:  
' - Retrieving the Microsoft Access window handle with FindWindow.  
' - Removing the Microsoft Access menu bar with SetMenu.

```
'=====
Function RemoveAccessMenu ()
 Dim hWnd%, dummy%
 hWnd% = FindWindow("OMain", 0&)
 dummy% = SetMenu(hWnd%, 0)
End Function
```

2. In the report in which you want to disable the menu bar, set the OnFormat or OnPrint property of the first section that prints (such as the Report Header section) to call the RemoveAccessMenu() function:

```
OnFormat: =RemoveAccessMenu()
```

#### REFERENCES

=====

"Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference," Volumes 1-6, Microsoft Press, 1992

#### [References](#)

## PRACC9307: Invalid Characters Added to IN Clause Arguments

Article Number: Q102523  
CREATED: 03-AUG-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SYMPTOMS

---

When you choose OK from the [SQL dialog box](#), you receive the error message

SQL error in FROM clause.

if the [query](#) contains an IN clause.

### CAUSE

---

Microsoft Access version 1.1 incorrectly parses the arguments for the IN clause after saving the SQL statement.

### RESOLUTION

---

If you enter the arguments correctly and then save the query, the query will run without errors. However, you must correct the parsing done by Microsoft Access if you make any changes to the SQL statement that you want to commit.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

Steps to Reproduce Problem

-----

1. Open the sample [database](#) NWIND.MDB.
2. Attach to the sample dBASE IV file, NEWCUST.DBF, located in your Access program directory.
3. Create a new query base on the [attached table](#), drag several fields to the query grid, and run the query to verify that it works properly.
4. Switch to [Design view](#). From the View menu, choose SQL.
5. After the FROM clause, "FROM NEWCUST", type the following:

```
IN "<c:\access>" "dBASE IV;"
```

where <c:\access> is the directory in which you installed Microsoft Access.

6. Choose OK to close the SQL dialog box, then choose OK again.
7. Run the query.
8. Switch back to Design view. From the View menu, choose SQL.

Microsoft Access has changed your IN clause to:

```
IN 'c:\access'["dBASE IV;"]
```

If you choose OK, the following error message is displayed:

```
Syntax error in From clause
```

You must choose Cancel or correct the IN clause.

## [References](#)

## INF: "Record Lock Threshold Exceeded" with Large Action Query

Article Number: Q102522  
CREATED: 03-AUG-1993  
MODIFIED: 09-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

If you exceed the maximum number of locks set per connection for the Novell server, you will get the following error message on the server:

Record Lock threshold exceeded

This error can also cause the Novell server to crash.

### CAUSE

---

### RESOLUTION

---

To solve this problem, use the following steps:

1. Obtain the TTSFIX.NLM dynamic patch, either directly from Novell or from CompuServe. On CompuServe, look for 311PTD.ZIP in the NOVFILES forum.
2. Set the Record Lock threshold to some high value greater than the default of 500, preferably between 6,000 and 10,000, even though the server should no longer crash. However, Microsoft Access may still encounter problems when the server exceeds the lock count.

There are no reports of any problems with the Record Lock threshold set to a high value. Even with this maximum set, if the threshold is reached, Microsoft Access will stop; with the fix, the Novell [network](#) will not crash.

3. Load the TTSFIX.NLM dynamic patch on the server to ensure that it will not crash if a lock count is exceeded.

NOTE: The 311LPTD.ZIP file also contains PATCHMAN.NLM and TTSFIX.NLM. We recommend that these files be loaded in the AUTOEXEC.NCF of the server. This action requires a LOAD PATCHMAN.NLM, followed by a LOAD TTSFIX.NLM in the AUTOEXEC.NCF.

The following [table](#) details possible lock settings:

|                                                             | Range       | Defaults       |
|-------------------------------------------------------------|-------------|----------------|
| -----                                                       |             |                |
| #1 SET Maximum <a href="#">record</a> locks per connection: | 10-10,000   | default 500    |
| #2 SET Maximum file locks per connection:                   | 10-1,000    | default 250    |
| #3 SET Maximum record locks:                                | 100-200,000 | default 20,000 |
| #4 SET Maximum file locks:                                  | 100-100,000 | default 10,000 |

#1 is the setting that should be increased from the default to some value within the 6000 to 10,000 range.

[References](#)

## PRB: "#ERROR" Referencing Null Subreport Values in Main Report

Article Number: Q102521  
CREATED: 03-AUG-1993  
MODIFIED: 25-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

Referencing a subreport control from a main report when the subreport does not return any records results in a "#ERROR" message.

### CAUSE

---

Make sure that any subreport fields referenced in main reports have values associated with them.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

See "Steps to Solve Problem" in the section below for a workaround procedure.

### MORE INFORMATION

---

Steps to Reproduce Problem  
-----

1. Start Microsoft Access and open the sample database NWIND.MDB. Create a new query called Sorter.

NOTE: This query limits the dynaset to customer records beginning with the letter "P" and includes customers with no orders.

2. From the Customers table, drag the Company Name field to the query grid. In the Criteria field, type the following:

Like "P\*"

Include any other fields on the query grid that you want, such as Customer ID.

3. Create a new subreport based on the Orders table; call it SubRpt.
4. In the Detail section of the subreport, drag the Order Amount field from the Orders table. In the Report Footer section, place a text box with the following formula as the ControlSource:

```
=Sum([Order Amount])
```

Set the ControlName property of the text box to Sum Amount.

5. Create a new report called Main Report based on the Sorter query.
6. From the Layout menu, turn off the Page Hdr/Ftr command. From the View menu, turn on the Sorting And Grouping command. In the dialog box, select Company Name from the Field/Expression combo box and set the Group Header property to Yes.
7. Drag the Company Name field to the new Company Name Header section.
8. Drag the SubRpt subreport from the Database window to the Detail section of Main Report. Below the subreport, place a label control with the caption "Total Order Amount Per Customer:" and a text box control containing the following formula:

```
=([SubRpt].Report![SumAmt])
```

Set both the LinkMaster and the LinkChild properties in the subreport to [Company Name].

9. Switch to Print Preview.

Note that Parisian Specialties, which has no orders, displays a "#ERROR" message.

#### Steps to Solve Problem

1. Create the following new Access Basic function called ErrAvoid():

```
Function ErrAvoid (n As Variant)
 On Error GoTo Trap
 ErrAvoid = n
 Exit Function
Trap:
 ErrAvoid = 0
 Exit Function
End Function
```

2. Replace "=( [SubRpt].Report![SumAmt])" in step 8 of the previous procedure with the following:

```
=ErrAvoid([SubRpt].Report![SumAmount])
```

3. Switch to Print Preview.

Parisian Specialties now shows \$0.00 dollars in sales, instead of the "#ERROR" message. The ErrAvoid() function traps for errors that are caused by the subreport containing no values. When an error is encountered, a value of zero is assigned.

#### References

## INF: Getting the Data Types of Fields in a Table through DDE

Article Number: Q102520  
CREATED: 03-AUG-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1  
-----

### SUMMARY

---

Microsoft Access 1.1 expanded on the use of the [DDE](#) item 'FieldNames' by providing a way to get the [data type](#) as well as the fieldname. The item is called FieldNames;T and can be Requested on any open [database](#) in Access.

A simple example of using this feature from Excel or Word is provided below.

### MORE INFORMATION

---

Excel Example:

- 
- 1) Start Microsoft Access and load the database NWIND.MDB.
  - 2) Start Excel and open a new Worksheet.
  - 3) Highlight cells A1:D2.
  - 4) Type the formula:

```
=MSAccess|'C:\ACCESS\NWIND.MDB;Table Categories'!'Fieldnames;T'
(The formula will start typing into the active cell of the selected
range.)
```

- 5) Hold down the CONTROL-SHIFT Keys and hit Enter. (This will enter the formula as an array formula. The results will be propagated through the entire selected range.)
- 6) The range should return the following information:

| Category ID | Category Name | Description | Picture |
|-------------|---------------|-------------|---------|
| 10          | 10            | 12          | 11      |

These are the values returned and the data types they represent:

|    |                                                                          |
|----|--------------------------------------------------------------------------|
| 0  | Invalid                                                                  |
| 1  | True/False (non- <a href="#">Null</a> )                                  |
| 2  | Unsigned byte (Byte)                                                     |
| 3  | 2-byte signed integer (Integer)                                          |
| 4  | 4-byte signed integer (Long)                                             |
| 5  | 8-byte signed integer (Currency)                                         |
| 6  | 4-byte single-precision floating-point (Single)                          |
| 7  | 8-byte double-precision floating-point (Double)                          |
| 8  | Date/Time (integral date, fractional time)                               |
| 9  | Binary data, 255 bytes maximum                                           |
| 10 | ANSI text, not <a href="#">case-sensitive</a> , 255 bytes maximum (Text) |
| 11 | Long binary ( <a href="#">OLE</a> Object)                                |
| 12 | Long text (Memo)                                                         |

Word Example:

-----

- 1) Start Microsoft Access and load the database NWIND.MDB.
- 2) Start Word and open a new Document.
- 3) Choose Field... from the Insert Menu.
- 4) Type the following formula into the field code edit box:

```
DDE MSAccess "C:\\ACCESS\\NWIND.MDB;Table Categories" FieldNames;T
```

- 5) Click OK.

The following information should be displayed:

| Category ID | Category Name | Description | Picture |
|-------------|---------------|-------------|---------|
| 10          | 10            | 12          | 11      |

You may see the formula displayed rather than this data. If this happens choose Options from the Tools menu. In the Category called view there is an option called Field Codes. If you uncheck this option the values will be displayed. Also some formatting of the tabs will be necessary to get the numbers to align under the names.

References:

[References](#)

## INF: Using the DDE Item SQLText to Request the Text of a Query

Article Number: Q102519  
CREATED: 03-AUG-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1  
-----

### SUMMARY

---

Microsoft Access version 1.1 introduces the new DDE item, SQLText, which can be used to retrieve the SQL statement representing the table or query topic on which the DDE link has been established.

The SQLText item allows the DDE client to request that the SQLText be either returned as a whole or divided into substrings. The length of the substring is specified by the client. The SQLText is returned as an array of substrings.

This article describes the use of this DDE item to request the text of a query and place it in a Microsoft Excel worksheet.

### MORE INFORMATION

---

Syntax for the SQLText item name is as follows

```
SQLText;<number>
```

where <number> is the maximum number of characters for each substring. If value of <number> is not provided, the entire text is sent as one string.

The last substring of the array may be shorter than the previous substrings if there are not enough characters.

NOTE: The following example assumes that you have a working knowledge of Microsoft Excel macros. For information on using macros, please refer to the Microsoft Excel user documentation.

Re-Creating the Sample Macro  
-----

The following sample macro should be pasted or entered in a Microsoft Excel macro sheet, starting with Cell A1. (Some formulas are dependent on starting at that location in the macro sheet.) Also, formulas designated in opening and closing braces ({} ) must be entered as array formulas, using the CTRL+SHIFT+ENTER key combination, without braces.

Use the three steps below to yield the following line:

```
{=FORMULA.ARRAY (TRANSPOSE (SQL)) }
```

1. Go to the cell that contains this formula.

2. Remove the braces ({}).
3. Press CTRL+SHIFT+ENTER.

To verify that the formula is entered as an array, the braces will reappear around the formula in the Microsoft Excel formula bar. For more information on using array formulas, please refer to the Microsoft Excel user documentation.

```

SQLText
StringLength=50
chan=INITIATE("MSACCESS","NWIND.MDB;Query Sales for 1991")
=FORMULA.ARRAY(REQUEST(chan,"SqlText;255"),B1:U1)
=IF(LEN(B1)<255)
= SET.NAME("SQLLen",LEN(B1))
=ELSE()
{= SET.NAME("SQLLen",SUM(IF(ISNA(B1:U1),0,LEN(B1:U1))))}
=END.IF()
=SET.NAME("SQLPieces",INT(SQLLen/StringLength)+1)
=SET.NAME("SQL",REQUEST(chan,"SqlText;"&StringLength))
=TERMINATE(chan)
=NEW(1)
=SELECT("r1c1:r"&SQLPieces&"c1")
{=FORMULA.ARRAY(TRANSPOSE(SQL))}
=COPY()
=PASTE.SPECIAL(3)
=CANCEL.COPY()
=FORMULA.REPLACE(CHAR(13)&CHAR(10)," ",2,2,FALSE)
=COLUMN.WIDTH(,,3)
=RETURN()

```

#### How Sample Macro Works

-----

The length of result string is set to 50, then a channel is initiated to the Microsoft Access query. The SQL statement is requested in 255-character chunks. Then the total length of the query is calculated.

Using SQLLen, the number of rows needed to display the query is calculated. The SQLText is requested in chunks of 50, based on the [variable](#) StringLength. Then the DDE channel is terminated.

Finally, a new worksheet is opened and the appropriate number of cells are selected. The Formula.Array() [function](#) places the SQLText in the currently selected cells. The result originally appears as a formula, but the Copy and Paste.Special() commands change it to text. The carriage returns and line feeds are changed to spaces and the ColumnWidth property is set to Best Fit.

#### [References](#)

## INF: Displaying Totals for Each Row in Crosstab Queries

Article Number: Q102517  
CREATED: 03-AUG-1993  
MODIFIED: 14-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Crosstab queries can display totals for each row. To do this, simply repeat the field used as the crosstab value and make it a crosstab row header.

### MORE INFORMATION

---

The following steps explain how to create a crosstab query with row totals:

1. Open the sample database NWIND.MDB.
2. Create a new query based on the Order Review query, as follows:

Query: XTAB With Row Totals

-----  
Field: Company Name  
Total: Group By  
CrossTab: Row Heading

Field: Total Ordered:Order Amount  
Total: Sum  
CrossTab: Row Heading

Field: Ship Via  
Total: Group By  
CrossTab: Column Heading

Field: Order Amount  
Total: Sum  
CrossTab: Value

The Total column always appears before the other data columns, but you can display the data on a form or report in any order.

### REFERENCES

=====

Cue Cards, the Microsoft Access tutorial, walks you through the process of creating a crosstab query. On the last step, you are given the option to add row totals, using procedures similar to those shown above.

For more information, choose Cue Cards from the Help menu and click

the About Cue Cards hot spot.

[References](#)

## INF: Debugging & Troubleshooting Tips for Attached SQL Tables

Article Number: Q102437  
CREATED: 03-AUG-1993  
MODIFIED: 19-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

Listed below are some techniques that can be used to determine indirect causes of error messages when using attached [SQL](#) tables:

1. Use Microsoft Windows File Manager to search for multiple, corrupted, or outdated copies of the following files:

- [ODBC](#).DLL - dated 10-16-92 or later, with 44736 bytes
- DBNMP3.DLL - dated 9-15-92 or later, with 8241 bytes
- SQLSRVR.DLL - dated 10-16-92 or later, with 135792 bytes
- NETAPI.DLL - depends on the [network](#) being used
- COMMDLG.DLL - dated 10-25-92 or later, with 89248 bytes

NOTE: The sizes and dates referenced are correct if the files were installed by Microsoft Access version 1.0.

2. Make sure that all the files listed in step 1 are in the appropriate locations. All files except NETAPI.DLL are usually installed in the Windows SYSTEM subdirectory. NETAPI.DLL is a network-specific driver; it is usually located in the network's program directory, which should appear in the path.
3. Use another application, such as the three listed below, on the same machine and try to attach the same data source:
  - Q+E (comes with Microsoft Excel)
  - SQL Server Administration Facility (SAF)
  - PowerBuilder

If you cannot access the data source from another application, you may have a problem with the network, not with Microsoft Access or the Microsoft Open Database Connectivity (ODBC) driver.

4. Try to attach another data source, such as another SQL Server [table](#), from a different [database](#) server. If you can attach the other data source, the original server may be unavailable, or you may need to reconfigure the entry for that server using the ODBC Administration utility.
5. Try to attach the same data source from a different computer. If you still cannot gain access to the SQL Server machine, you may have a network-wide problem, or perhaps the server is down or not communicating.

6. Try increasing the ODBC time-out settings in the [ODBC] section of your MSACCESS.INI file.
7. Rename the current ODBC.DLL, DBNMP3.DLL, and SQLSRVR.DLL files, restart the server, and reinstall ODBC.

For more information on this procedure, see the "Microsoft Access User's Guide," version 1.0, Appendix D, "Setting Up Microsoft Access on a Network," pages 660-662.

PowerBuilder is manufactured by Powersoft Corporation, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

### [References](#)

## PRACC9305: Hard-Coded and Passed Wildcards Different in Query

Article Number: Q102431  
CREATED: 03-AUG-1993  
MODIFIED: 25-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

The Like operator treats hard-coded asterisk (\*) wildcard characters in queries differently from asterisks passed as [query](#) parameters.

With the asterisk as a parameter value, queries do not display records with null values for the [field](#) in which the Like [criteria](#) is set. However, with the asterisk as a hard-coded criteria value (for example, Like "\*"), null values do appear for the field with the Like condition.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

Steps to Reproduce Problem

-----

How to Create TestQuery:

1. Start Microsoft Access and open the sample [database](#) NWIND.MDB.
2. Create a new query based on the Employees [table](#).
3. Add the Last Name and Region fields to the query grid.
4. In the Criteria [cell](#) in the Region field of the query grid, type the following:

Like "\*"

5. Run the query.

The result is a [dynaset](#) containing nine records, including four records without values in the Region [column](#) (Buchanan, Suyama, King and Dodsworth). The wildcard character hard-coded with the Like operator in the query returns all records, whether or not there are null values in the field.

6. Change the Like operator in step 4 above to the following:

Like [Forms]![TestForm]![Field0]

This operator instructs the Like criteria to look in Field0 on a [form](#) called TestForm for the value it will use in the operation.

7. Save the query as TestQuery.

How to Create TestForm:

1. In the [Database window](#), create a new, unbound form.
2. Place a [text box control](#) on the form.
3. Save the form as TestForm.

How to Pass the Asterisk Wildcard Character as a Parameter:

1. Open TestForm in [Form view](#) and TestQuery in [Design view](#). Arrange the windows so that you can see both the form and the query.
2. Type "\*" (without the quotation marks) in the text box on TestForm and press ENTER. This asterisk will be passed as a parameter for the Like operator in TestQuery.
3. Run TestQuery.

Note that the query dynaset now contains only five records: those containing non-null values in the Region field. When the asterisk is passed as a parameter to the query, Microsoft Access displays only records with non-null values in the field using the Like operator (in this case, the Region field).

## [References](#)

## INF: Fill New Record with Data from Prev Record Automatically

Article Number: Q88670  
CREATED: 11-AUG-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1  
-----

### SUMMARY

---

When you are creating new records, you may want speed up the data entry process by having fields in the new record fill with values from fields in the previous record automatically.

This article contains an Access Basic function, AutoFillNewRecord(), that allows you to do this. The following procedure uses the Customer form in the sample database NWIND.MDB.

### MORE INFORMATION

---

One technique used to automate repetitive data entry is to press CTRL+apostrophe (') with your insertion point in a field. Microsoft Access then inserts the value contained in the previous record in the new field.

However, you may want certain fields in a new record to fill with data from the previous record automatically. This feature is under review and will be considered for inclusion in a future release of this product.

The following Access Basic function, AutoFillNewRecord(), can be used to fill either selected fields or all fields in a new record with values from the previous record. Use the following three steps to re-create an example of this procedure, using the Customers table in the NWIND database:

1. Create the following Access Basic module:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
' *****
' DECLARATION SECTION
' *****
Option Explicit

' *****
' FUNCTION: AutoFillNewRecord
'
' PURPOSE:
' Speeds up repetitive data entry by automatically filling
' selected fields in a new record with data contained in fields in
```

```

' the previous record.
'
' USAGE:
' Set the OnCurrent property of the form to read:
'
' =AutoFillNewRecord()
'
' If you want only certain fields filled in automatically, create
' an unbound text box on the form with the following properties:
'
' ControlName: AutoFillNewRecordCriteria
' Visible: No
' DefaultValue: ;Company Name;Contact Name;Contact Title;Address;
'
' NOTE: The DefaultValue property should be a semicolon-delimited
' list of control names on the form. You must have a semicolon
' to the left and right of each entry to avoid confusion with
' other fields with similar names.
'
' If you want all fields automatically filled in, do not create the
' AutoFillNewRecordCriteria text box described above.
'
' *****
Function AutoFillNewRecord ()
 Dim ds As Dynaset, F As Form,RetVal As Variant, i As Integer
 Dim AutoFillCriteria As String, AutoFillField As Variant
 Set F = Screen.ActiveForm

 On Error Resume Next

 ' See if we are on the new record by trying to reference
 ' .. the bookmark. An error will occur if it is a new record.
 RetVal = F.Bookmark

 ' If there was no error, we are not on new record, so exit.
 If Err = 0 Then Exit Function

 On Error Resume Next
 Set ds = F.Dynaset
 ds.MoveLast

 ' If we cannot move to, or find, last record, exit the function.
 If Err <> 0 Then Exit Function

 DoCmd Echo False

 ' Step through all the fields in the dynaset.
 For i = 0 To F.Count - 1
 ' Get the list of fields to auto fill from the form.
 AutoFillCriteria = F!AutoFillNewRecordCriteria

 ' Don't fill the current field by default.
 AutoFillField = False

 ' Fill field if there are criteria (Err=0) and the current
 ' control name being evaluated appears in the criteria.
 If (Err = 0 And InStr(1, AutoFillCriteria, ";" _

```

```

 & (F(i).ControlName) & ";")) Then AutoFillField = True
 End If

 ' Fill field if there is no criteria field (Err <> 0,
 ' indicating there was an error finding the criteria
 ' field on the form).
 ' In other words, fill all fields if there is no
 ' criteria field on form.
 If Err <> 0 Then
 Err = 0 'reset error flag
 AutoFillField = True
 End If

 ' If we are going to fill the field,
 If AutoFillField = True Then
 ' copy the value from the last record in the dynaset
 ' field to the new record field on the form.
 F(i) = ds((F(i).ControlName))

 ' Ignore error if one occurs during copy action.
 If Err <> 0 Then Err = 0
 End If
Next i

DoCmd Echo True

End Function

```

2. Open the Customers form in [Design view](#). Change the form's OnCurrent property to read:

```
=AutoFillNewRecord()
```

3. Create an unbound text box on the form with the following properties:

```

ControlName: AutoFillNewRecordCriteria
Visible: No
DefaultValue: ;Company Name;Contact Name;Contact Title;Address;

```

When you go to a new record, the Company Name, Contact Name, Contact Title, and Address fields are filled in automatically. If you want all fields to automatically fill in, skip step 3 above.

## [References](#)

## **PRACC9307: Setup (STFSETUP) GP Faults in Windows Standard Mode**

**Article Number:** Q102230  
**CREATED:** 29-JUL-1993  
**MODIFIED:** 03-AUG-1993  
**VERSION(S):** 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

When you are running Microsoft Windows in standard mode, the Setup program causes the following error message to appear near the end of the installation process

STFSETUP caused a general protection fault in STFSETUP.EXE

if you are running any one of the following:

- The retail setup of Microsoft Access version 1.1.
- A workstation setup (/n) from the version 1.1 upgrade disks.
- A custom application developed with the Microsoft Access Distribution Kit (ADK).

### **CAUSE**

---

This problem is caused by the detection and implementation of sharing mechanisms.

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.1 and the Microsoft Access Distribution Kit (ADK) version 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### **RESOLUTION**

---

Run Windows in 386 enhanced mode during Setup. You may then switch back to standard mode to run your applications.

NOTE: This problem is solely confined to the Setup process. In no way does it affect how Microsoft Access runs in either standard or 386 enhanced mode.

If for some reason you cannot run Microsoft Windows in 386 enhanced mode, modify STFSETUP.IN\_ as described below. Separate procedures follow for retail, upgrade, and ADK users.

Steps for Retail Users  
-----

1. Make a backup copy of Disk1.
2. On the backup disk, use an editor to modify the STFSETUP.IN\_ file. Change the [System Paths] section from

```
[System Paths]
 SYSTEMPATH = "" ? DETCMD.DLL GetWindowsSysDir
 WINDOWSPATH = "" ? DETCMD.DLL GetWindowsDir
 HARDDRIVELIST = "" ? DETCMD.DLL GetAllValidLocalHardDrives
 NETDRIVELIST = "" ? DETCMD.DLL GetAllValidNetworkDrives
```

to

```
[System Paths]
 SYSTEMPATH = "" ? DETCMD.DLL GetWindowsSysDir
 WINDOWSPATH = "" ? DETCMD.DLL GetWindowsDir
 HARDDRIVELIST = "" ? DETCMD.DLL GetAllValidLocalHardDrives
 NETDRIVELIST = "" ? DETCMD.DLL GetAllValidNetworkDrives
 WINDOWSMODE = "" ? DETCMD.DLL GetWindowsMode
```

3. Then change the following

```
DoShareStuff = +
 set CurrentDialog = DoShareStuff
 set MAKEBAK = "NO"
 ui start VSHARERunning
 ifstr $(ANSWER) == "NO"
```

to

```
DoShareStuff = +
 set CurrentDialog = DoShareStuff
 set MAKEBAK = "NO"
 ifstr $(WINDOWSMODE) != "ENHANCED"
 set ANSWER = "NO"
 else
 ui start VSHARERunning
 endif
 ifstr $(ANSWER) == "NO"
```

#### Steps for Version 1.1 Upgrade Users

-----

Single-user upgrades and Admin upgrades (setup /a) work fine. However, to run Setup or setup /n from the [network](#) with Windows in standard mode, you must first make the changes mentioned above to the STFSETUP.IN\_ file in the ACCESS\INSTALL subdirectory on the network.

#### Steps for ADK Users

-----

The ADK Setup works fine in Windows standard mode. However, before you create disks for your application, modify the SETUPWIZ.IN\_ file (and optionally the CUST12MB.IN\_, CUST14MB.IN\_, and CUST720K.IN\_ files) as follows:

1. Change the following

```
[System Paths]
 SystemPath = "" ? DETCMD.DLL GetWindowsSysDir
 WindowsPath = "" ? DETCMD.DLL GetWindowsDir
 HARDDRIVELIST = "" ? DETCMD.DLL GetAllValidLocalHardDrives
 NETDRIVELIST = "" ? DETCMD.DLL GetAllValidNetworkDrives
```

to

```
[System Paths]
 SystemPath = "" ? DETCMD.DLL GetWindowsSysDir
 WindowsPath = "" ? DETCMD.DLL GetWindowsDir
 HARDDRIVELIST = "" ? DETCMD.DLL GetAllValidLocalHardDrives
 NETDRIVELIST = "" ? DETCMD.DLL GetAllValidNetworkDrives
 WINDOWSMODE = "" ? DETCMD.DLL GetWindowsMode
```

NOTE: Unlike the Microsoft Access version 1.1 corrections, the first two variables in the [System Paths] section of the ADK is case-sensitive.

2. Add the test case around VSHARERunning by changing the following from

```
DoShareStuff = +
 set CurrentDialog = DoShareStuff
 set MAKEBAK = "NO"
 ui start VSHARERunning
 ifstr $(ANSWER) == "NO"
```

to

```
DoShareStuff = +
 set CurrentDialog = DoShareStuff
 set MAKEBAK = "NO"
 ifstr $(WINDOWSMODE) != "ENHANCED"
 set ANSWER = "NO"
 else
 ui start VSHARERunning
 endif
 ifstr $(ANSWER) == "NO"
```

NOTE: This is exactly the same change as the Microsoft Access version 1.1 change.

## **MORE INFORMATION**

---

### Steps to Reproduce Problem

-----

1. Start Windows in standard mode (win /s), using the Microsoft Access version 1.1 Retail disks.
2. Insert Disk 1 into drive A for 3.5-inch disks (or B for 5.25-inch disks).
3. From the File menu in Program Manager, choose Run.

4. Type `a:\setup` (or `b:\setup`) and press ENTER.

5. Follow the instructions on your screen.

When you insert the last disk, you will receive this error.

[References](#)

## INF: Trapping SQL Server RAISERROR() Function Values

Article Number: Q101678  
CREATED: 20-JUL-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SUMMARY

---

Microsoft [SQL](#) Server users often create triggers/stored procedures that perform specific functions, which are fired during certain events. Often, a custom-generated error value is desired to indicate the status of such events. This article describes a way to trap that value in Microsoft Access.

NOTE: This article assumes that the user is running Microsoft Access version 1.1, since that version corrects a problem to allow the SQL Server RAISERROR() [function](#) to return a value to Microsoft Access.

### MORE INFORMATION

---

1. Create the following stored procedure based on the sample [database](#) PUBS in Microsoft SQL Server:

```
===== Store Procedure =====
CREATE PROCEDURE TestProc
AS
 RAISERROR 25000 'This is a test error'
=====
```

2. Create a trigger on the [table](#) from which you want this stored procedure to run:

NOTE: Stored procedures cannot be executed directly from Microsoft Access unless they are triggered from a SQL Server trigger or the SQL Pass-Thru .DLL file is used. In this example, you will use a trigger that fires when a [record](#) is updated.

```
===== Trigger =====
CREATE TRIGGER TestTrig
ON Authors
FOR UPDATE
AS
 EXECUTE TestProc
=====
```

3. Create the following [module](#) in a Microsoft Access database that updates the Authors table (the [attached table](#) dbo\_authors) in some way. It then can trap the error value that is passed by the RAISERROR() function.

```
===== Access Basic Module =====
```

```

Sub TrapIt ()
 Dim db As Database, Mydyna As Dynaset
 Dim Xerr As String, Xval As Integer,
 Dim Xstart As Integer, Xlen As Integer
 On Error GoTo ErrorHandler
 Set db = CurrentDB()
 Set Mydyna = db.CreateDynaset("select * from dbo_authors;")
 Mydyna.Edit
 Mydyna!au_fname = Mydyna!au_fname
 Mydyna.Update
 Mydyna.MoveNext
 Exit Sub

ErrorHandler:

'This routine parses the error string returned from ODBC and
'extracts only the error value you assigned with the RAISERROR()
'function in SQL Server.

'traps the error message
 xerror = Error$
'finds start of error value
 Xstart = InStr(1, xerror, "#") + 1
'finds length of error value
 Xlen = InStr(Xstart, xerror, ")") - Xstart
'extracts error value from string
 Xval = Mid(xerror, Xstart, Xlen)
 MsgBox ("You have encountered error #" & CStr(Xval))
 Resume Next
End Sub

```

=====

[References](#)

## PRB: No CanShrink property for Check Boxes

Article Number: Q101677  
CREATED: 20-JUL-1993  
MODIFIED: 20-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

There is no CanShrink property for check boxes and option buttons placed on a [form](#) or [report](#).

### RESOLUTION

---

See the work around described below.

### MORE INFORMATION

---

Work Around  
-----

The following information describes a technique to simulate a check box on a form using a [text box](#). When placed on a report, if the [field](#) is not checked, the horizontal space the text box occupies will close up and not print.

1. Given a yes/no field called "Check1", create a textbox on a report with the following properties:

```
ControlName: txtCheck1
ControlSource: =IIf([Check1]=-1," X ")
CanShrink: Yes
BorderStyle: Normal
```

Note: The ControlName property must be different from any fields being referenced in the associated ControlSource property formula or a #ERROR message will be generated for the field.

2. Change the ControlName of the [label](#) to "lblCheck1".
3. Create the following [macro](#) and save it with the name "ShrinkNo":

| Condition   | Action   | Defined Below |
|-------------|----------|---------------|
| [Check1]=-1 | SetValue | 1.            |
| [Check1]=0  | SetValue | 2.            |

ShrinkNo Actions  
-----

1. SetValue  
Item: [lblCheck1].Visible  
Expression: True

2. SetValue  
Item: [lblCheck1].Visible  
Expression: False

4. Set the following properties for the Detail section:

CanShrink: True  
On Format: ShrinkNo

5. Size and position the text box and the label appropriately to look like a check box.

The text box will shrink up if and only if it is the ONLY thing sharing the same space horizontally on the report. Therefore, you can not have text boxes overlap one another or any other control to the left or right of the text box that needs to shrink. The label for the text box must be hidden for the text box to shrink, which the ShrinkNo macro accomplishes.

References:

[References](#)

## INF: How to Retrieve Windows For Workgroups User Information

Article Number: Q101676  
CREATED: 20-JUL-1993  
MODIFIED: 18-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

Using the Microsoft Windows for Workgroups API, you can retrieve information about the currently running machine, such as user name, workgroup, [domain](#), and computer name.

### MORE INFORMATION

---

Windows for Workgroups comes with a [function](#) called NetWkstaGetInfo() that returns information about a computer. This function is especially useful for multiuser applications.

The following procedure explains how to use [Access Basic](#) and several Windows API calls to access the NetWkstaGetInfo() function:

1. Create a new [module](#) containing the following declarations and functions:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

NOTE: You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive the duplicate procedure name error message when you compile or run your code, remove or comment out the appropriate declarations statement from your code. Make sure to execute Compile All by choosing Run from the [menu bar](#) and verify that you do not get any compilation errors.

```
Option Explicit
Declare Function NetWkstaGetInfo% Lib "NetAPI" (ByVal lServer&, _
 ByVal sLevel%, ByVal pbBuffer&, ByVal cbBuffer%, pcbTotalAvail%)
Declare Function GlobalAlloc% Lib "Kernel" (ByVal fFlags%, _
 ByVal nSize&)
Declare Function GlobalLock& Lib "Kernel" (ByVal hMem%)
Declare Function GlobalUnlock% Lib "Kernel" (ByVal hMem%)
Declare Function GlobalFree% Lib "Kernel" (ByVal hMem%)
Declare Sub lstrcpy Lib "Kernel" (ByVal dest As Any, _
 ByVal src As Any)
Declare Sub hmemcpy Lib "Kernel" (ByVal dest As Any, _
 ByVal src As Any, ByVal size As Long)

Function GetBinInt (sobj As String, off As Integer)
```

```

If (Asc(Mid$(sobj, off + 1, 1)) < 128) Then
 GetBinInt = Asc(Mid$(sobj, off, 1)) +
 Asc(Mid$(sobj, off + 1, 1)) * 256
Else
 GetBinInt = ((&HFF - Asc(Mid$(sobj, off + 1, 1))) * 256) -
 Asc(Mid$(sobj, off, 1))
End If
End Function

Function GetWorkstationInfo (sComputer As String, _
 sUserName As String, sWorkgroup As String, sLogonDomain As String)
Dim nRetSize As Integer, nStruct, ret As Integer, dummy As Integer
Dim hMem As Integer, lpMem As Long, sTemp As String

' Get the size of the return structure.
ret = NetWkstaGetInfo(0&, 10, 0&, 0, nRetSize)
If (ret <> 0) And (ret <> 2123) Then
 GetWorkstationInfo = False
 Exit Function
End If

' Allocate memory for structure.
hMem = GlobalAlloc(0, CLng(nRetSize))
nStruct = nRetSize
If (hMem <> 0) Then
 lpMem = GlobalLock(hMem)

' Read workstation information into structure.
ret = NetWkstaGetInfo(0&, 10, lpMem, nRetSize, nRetSize)
If (ret = 0) Then
 sTemp = Space(100)
 hmemcpy sTemp, lpMem, nStruct

' Retrieve the computer name.
sComputer = Space(100)
lpMem = CLng(GetBinInt(sTemp, 1)) +
 (CLng(GetBinInt(sTemp, 3)) * CLng(&H10000))
lstrcpy sComputer, lpMem
sComputer = Mid(sComputer, 1, InStr(sComputer, Chr(0)) - 1)

' Retrieve the user name.
sUserName = Space(100)
lpMem = CLng(GetBinInt(sTemp, 5)) +
 (CLng(GetBinInt(sTemp, 7)) * &H10000)
lstrcpy sUserName, lpMem
sUserName = Mid(sUserName, 1, InStr(sUserName, Chr(0)) - 1)

' Retrieve the workgroup name.
sWorkgroup = Space(100)
lpMem = CLng(GetBinInt(sTemp, 9)) +
 (CLng(GetBinInt(sTemp, 11)) * &H10000)
lstrcpy sWorkgroup, lpMem
sWorkgroup = Mid(sWorkgroup, 1, InStr(sWorkgroup, _
 Chr(0)) - 1)

' Retrieve the logon domain name.
sLogonDomain = Space(100)

```

```

 lpMem = CLng(GetBinInt(sTemp, 15)) +
 (CLng(GetBinInt(sTemp, 17)) * &H10000)
 lstrcpy sLogonDomain, lpMem
 sLogonDomain = Mid(sLogonDomain, 1, InStr(sLogonDomain,
 Chr(0)) - 1)
 End If
 ' Free the memory allocated.
 dummy = GlobalUnlock(hMem)
 dummy = GlobalFree(hMem)
Else
 ret = -1
End If
 GetWorkstationInfo = IIf(ret = 0, True, False)
End Function

```

2. Test this function by creating the following Access Basic function:

```

Sub ShowInfo ()
 Dim a$, b$, c$, d$

 If GetWorkstationInfo(a$, b$, c$, d$) Then
 MsgBox a$ & " " & b$ & " " & c$ & " " & d$
 Else
 MsgBox "Unable to get information."
 End If
End Sub

```

3. In the [Immediate window](#), type the following:

```
ShowInfo
```

Either a [dialog box](#) that displays the Computer Name, User Name, Workgroup Name, and Logon Domain Name will appear, or, if you are running this function on a non-Windows for Workgroups-based system, the error message "Unable to get information" will be displayed.

## [References](#)

## INF: Sample Table Design to Support Questionnaire Applications

Article Number: Q101675  
CREATED: 20-JUL-1993  
MODIFIED: 13-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

This article describes the [table](#) design suitable for an application that tallies results from questionnaires and surveys.

### MORE INFORMATION

---

The following examples outline nonrelational table design commonly used for questionnaires and surveys and suggests an improved, relational table design.

#### Nonrelational Table Design

-----

When designing tables for questionnaire applications, many users begin with a design that resembles the table below. Each [record](#) contains multiple fields, called Question1 through Question<n>, that contain responses to the questions.

```
Table: Table1 (old)

FieldName: Respondent ID [Primary Key]
FieldName: Question1
FieldName: Question2
FieldName: Question3
.
.
.
FieldName: Question<n>
```

Problems occur when you want to perform crosstab queries to summarize and/or [graph](#) the questionnaire results.

#### Relational Table Design

-----

A relational table design better supports summary queries, reports, and graphs. In the table below, the Question ID [field](#) identifies the question and the Response field contains the answer.

```
Table: Table2 (new)

Field: Respondent ID [Primary Key]
Field: Question ID [Primary Key]
Field: Response
```

## How to Switch to Relational Database Design

---

To convert data that has been entered in Table1 format, complete the following six steps:

1. Create Table2 according to the above structure, using the following data type criteria:
  - All Respondent ID fields must be the same data type.
  - The Question ID field must be a Text data type.
  - The Response field must be the same data type as the Question<x> data types.
2. Create a new query based on Table1.
3. From the Query menu, choose Append. Select Table2 as the table to which you want to append the data. Design the query as follows:

```
Query: Query1

Field: Respondent ID
 Append To: Respondent ID
Field: Question1
 Append To: Response
Field: "Question1"
 Append To: Question ID
```

4. Run Query1 to append to Table2 each participant's responses to Question1.
5. Repeat steps 2-4 above, replacing Question1 with Question2, and "Question1" with "Question2". You must re-create or modify this query for each question in Table1.
6. After running all the append queries, the result is a table (Table2) that can easily summarize your results in a crosstab query:

```
Query: CrossTabExample

Field: Question ID
 Total: Group By
 Crosstab: Row Heading
Field: Response
 Total: Group By
 Crosstab: Column Heading
Field: Response
 Total: Count
 Crosstab: Value
```

## REFERENCES

---

"User's Guide," version 1.0, Chapters 1, 6, and 7

[References](#)

## INF: How to Get a List of Object Names in Microsoft Access

Article Number: Q101674  
CREATED: 20-JUL-1993  
MODIFIED: 20-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

Microsoft Access versions 1.x provide no built in mechanism for enumerating a list of Form, Report, Macro, or Module names. The following information provides an [Access Basic function](#), `GetObjectNames()`, which can be used to list the names of these objects as well as Table and Query names.

NOTE: You can get information on [table](#) and [query](#) names using the technique described in Chapter 8 of the "Introduction to Programming Guide" titled "Listing the Tables and queries in a Database".

NOTE: The technique described below relies on the use the system tables stored with your [database](#). These tables are undocumented and are subject to change in future versions of Microsoft Access.

### More Information:

---

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

Create a new [module](#) with the code listed below.

Note: In the following sample code, an underscore (`_`) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****
' MODULE DECLARATIONS
'*****
Option Explicit

'*****
' FUNCTION: GetObjectNames
'
' PURPOSE:
' Fills the string array passed as an argument with a list of names
' of objects of type ObjectType in the currently opened user
' database.
'
' ARGUMENTS:
' ObjectType - A string representing the object whose names are to
```

```

' be retrieved. ObjectType can be one of the
' following:
'
' Tables, Queries, Forms, Reports, Macros, Modules
'
' Names - A single dimensional array of type string
'
RETURN:
' The number of names stored in the Names array. The names
' are stored beginning from 0 to GetObjectName - 1.
'
NOTES:
' This function utilizes information stored in the MySysObject table
' of the currently opened user database. The system tables are
' undocumented and are subject to change in future versions of
' Access.
'
'*****
Function GetObjectName (ByVal ObjectType, Names() As String)
 Dim db As Database
 Dim ss As Snapshot
 Dim Count
 Dim SQL
 Dim Msg As String

 SQL = "Select Name,Type from MSysObjects Where Type="

 Select Case ObjectType
 Case "Tables"
 SQL = SQL & "1 And Left(Name,1)<>'~' And Left(Name,4) <> _
 "MSys" Order By Name;"
 Case "Queries"
 SQL = SQL & "5 And Left(Name,1)<>'~' And Mid(Name, _
 Len(Name) - 3) <> ""0000"" Order By Name;"
 Case "Forms"
 SQL = SQL & "-32768 And Left(Name,1)<>'~' Order By Name;"
 Case "Reports"
 SQL = SQL & "-32764 And Left(Name,1)<>'~' Order By Name;"
 Case "Macros"
 SQL = SQL & "-32766 And Left(Name,1)<>'~' Order By Name;"
 Case "Modules"
 SQL = SQL & "-32761 And Left(Name,1)<>'~' Order By Name;"
 Case Else
 Msg = "Object Name """" & ObjectType & """" is an invalid"
 Msg = Msg & " argument to Function GetObjectName!"
 MsgBox Msg, 16, "GetObjectName"
 Exit Function
 End Select

 Set db = CurrentDB()
 Set ss = db.CreateSnapshot(SQL)

 ss.MoveLast
 If ss.RecordCount > 0 Then
 ReDim Names(0 To ss.RecordCount - 1)
 Else
 GetObjectName = 0
 End If

```

```

 Exit Function
 End If

 ss.MoveFirst
 Count = 0
 Do While Not ss.EOF
 Names(Count) = ss![name]
 Count = Count + 1
 ss.MoveNext
 Loop

 GetObjectNames = ss.RecordCount
End Function

'*****
' FUNCTION: TestGetObjectNames
'
' PURPOSE: Used to demonstrate and test the GetObjectNames function
'
' ARGUMENTS:
' ObjectType - A string representing the object whose names are to
' be retrieved. ObjectType can be one of the following:
'
' Tables, Queries, Forms, Reports, Macros, Modules
'*****
Function TestGetObjectNames (ObjectType)
 Dim Count, i
 ReDim Names(0) As String

 Count = GetObjectNames(ObjectType, Names())

 Debug.Print "Count: " & Count

 For i = 0 To Count - 1
 Debug.Print Names(i)
 Next i

End Function

```

How to use the GetObjectNames() Function

---

The function TestGetObjectNames() listed above demonstrates how to use the GetObjectNames() function.

1. With the module opened in design view, choose Immediate window from the View menu.
2. Type the following and press Enter:

```
? TestGetObjectNames("Forms")
```

Result: A list of forms in the currently opened database will print in the Immediate window.

See the text description of the GetObjectNames() function for more

details on it's usage.

[References](#)

## PRACC9306: ERR: Undefined Function Referencing Column Property

Article Number: Q101673  
CREATED: 20-JUL-1993  
MODIFIED: 26-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

When you attempt to run [macro](#) actions that contain a "Where" clause (ApplyFilter, OpenForm, OpenReport, and FindRecord), and queries that reference the [column](#) property of a [combo box](#) on a [form](#), the following error message is displayed:

Undefined Function "[Forms]![Formname]![Controlname].column" in Expression.

### CAUSE

---

Microsoft Access interprets any text followed by opening and closing parentheses as a user-defined [function](#).

### RESOLUTION

---

Create an unbound [text box](#) on the form that references the column property of the combo box. Reference this text box in your [query](#).

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

NOTE: This problem occurs only when you are attempting to reference a column other than the bound column of the combo box. If you want to reference the bound column of the combo box, just use the [control](#) name of the combo box.

Steps to Reproduce Problem  
-----

1. Using the sample [database](#) NWIND.MDB, create a new form based on the Orders [table](#).
2. Add a combo box control with the following properties:

Form: PromptForm  
Record Source: Orders  
Combo Box Control: TheCombo

ControlSource: OrderID  
RowSource: Orders  
ColumnCount: 4  
BoundColumn: 1

3. Create the following query based on the Employees table:

Query: ReferenceCombo

-----  
Type: Select Query

Field: Employee ID

Table: Employees

Criteria: =[Forms]![PromptForm]![TheCombo].column(2)

Field: Last Name

Table: Employees

Field: First Name

Table: Employees

4. Open the form PromptForm in Form view and select an Order ID record from the combo box.
5. Open the query ReferenceCombo in Datasheet view. You will receive the following error message:

Undefined Function "[Forms]![PromptForm]![TheCombo].column" in Expression.

To work around this problem, use the following four steps:

1. Open the form PromptForm in Design view.
2. Add the following unbound text box control:

Text Box Control: TheTextbox  
ControlSource: =TheCombo.column(2)  
Visible: No

3. Open the Query in Design view.
4. Change the criteria for the field [Employee ID] to the following:

=[Forms]![PromptForm]![TheTextbox]

The query will now correctly reference the text box control, which references the column property of the combo box.

## [References](#)

## PRB: "Btrieve-Out of Resources" Error Msg with Attached Table

Article Number: Q101405  
CREATED: 13-JUL-1993  
MODIFIED: 31-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

Opening an attached Btrieve [table](#) returns the following error message:

Btrieve-Out of Resources

### CAUSE

---

If you are using compressed Btrieve files, you must be sure that the compression buffer Btrieve is using is adequate for your data. The buffer size must be at least as large as the largest [record](#) in your data files.

### RESOLUTION

---

To ensure proper operation, set the compression buffer size option (/u) in the [btrieve] section of your WIN.INI file. The units for this setting are kilobytes; if your largest record is 2K, add /u:2 to the Btrieve options line.

### MORE INFORMATION

---

For more information on setting options, see the topic "Setting Btrieve Options in WIN.INI" in the BTRIEVE.TXT file, included with Microsoft Access version 1.1.

Btrieve is manufactured by Novell, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

### [References](#)

## INF: How to Query SQL Server Tables Without Attaching

Article Number: Q101375  
CREATED: 12-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

This article describes how to use the [SQL](#) clause IN to [query](#) an SQL Server [table](#) without attaching to it.

### MORE INFORMATION

---

You can use IN to query only one external [database](#) at a time.

The IN clause requires two arguments: database name and connect [string](#). When you attach to SQL Server tables, you must specify an empty string for the database name. The connect string consists of the following items:

|                          |                                                                      |
|--------------------------|----------------------------------------------------------------------|
| Data Source Name (DSN)   | The name of the SQL Server machine                                   |
| Database Name (DATABASE) | The name of the SQL Server database                                  |
| Workstation ID (WSID)    | The name of the workstation that is making the connection (optional) |
| User ID (UID)            | The login id*                                                        |
| Password (PWD)           | The user password*                                                   |

\* This item is optional within the connect string. If it is not present, the system prompts the user with a [dialog box](#) to enter this information the first time the [macro](#) is executed each session.

The following is a sample connect string to SQL Server. Note that it starts with "[ODBC](#)," which tells the system that it uses the ODBC driver (as opposed to an [installable ISAM](#) driver).

```
ODBC;DSN=ServerName;Database=Pubs;WSID=Wrk1;USID=JaneDoe;PWD=ABC123
```

The following example assumes that the ODBC driver manager and SQL Server driver are installed, that a data source called "Corp" was set up using the ODBC driver manager, and that the table "Employees" can be successfully attached to using the Attach command on the File menu.

1. Open the sample database NWIND.MDB.
2. Create a new query.

3. From the View menu, choose SQL.

4. Enter the following query:

```
Select *
FROM Employees
IN "" "ODBC;dsn=Corp;database=Payroll;uid=JSmith;pwd=KL3d";
```

5. Choose OK to exit the SQL dialog box and close and save the query.

NOTE: If you return to the View SQL window after saving the query, Access may place additional characters in the IN clause that you must remove to avoid a [syntax error](#).

#### REFERENCES

=====

Microsoft Access Help "IN clause" topic

"Language Reference," Appendix B: Microsoft Access SQL, page 513

[References](#)

## INF: How to Create a Custom Startup "Splash" Screen

Article Number: Q101374  
CREATED: 12-JUL-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

This article explains how to create a custom startup "splash" screen for your Microsoft Access application.

### MORE INFORMATION

---

A startup "splash" screen (title screen) is used by programmers to display important information about a program during startup and to give the user something colorful to look at and read while waiting for the program to complete its initial processing, such as reading and setting up user-defined preferences.

A typical splash screen contains the following information:

- The program name and version number
- The registered user's name and company
- Copyright information
- The developer's company name and logo

This article contains two [Access Basic](#) functions to implement splash functionality.

Creating Splash, the New Module

-----

1. Create a new Access Basic [module](#) with the following functions:

```
'*****
' MODULE NAME: Splash
' DECLARATION SECTION
'*****
Option Explicit

Dim gSplashStart ' The time when the splash screen opened.
Dim gSplashInterval ' The minimum time to leave the splash screen up.
Dim gSplashForm ' The name of the splash screen form.

'*****
' FUNCTION: SplashStart()
'
' PURPOSE: Used to invoke the splash screen form specified by the
' SplashForm argument.
'
```

```

' ARGUMENTS:
' SplashForm - The name of the form to use as the splash
' screen.
' SplashInterval - The minimum time, in seconds, that the splash
' screen must remain active on the screen.
'
'*****
Function SplashStart (ByVal SplashForm As String, ByVal SplashInterval_
As Integer)
' Set the starting time.
 gSplashStart = Timer

' Record the global information.
 gSplashInterval = SplashInterval
 gSplashForm = SplashForm

' Open the splash form.
 DoCmd OpenForm SplashForm
End Function

'*****
' FUNCTION: SplashEnd()
'
' PURPOSE: Used to close the splash screen form opened by the
' SplashStart() function. This function checks to ensure that
' the splash screen remains active until the user-specified
' interval has expired.
'
'*****
Function SplashEnd ()
 Dim RetVal

' Loop until the splash screen has been active for
' .. the desired interval.
 Do Until Timer - gSplashStart > gSplashInterval
 ' Yield control so other applications can process.
 RetVal = DoEvents()
 Loop

' Close the splash screen.
 DoCmd Close A_FORM, gSplashForm

End Function

```

2. Save the new module as Splash.

#### How to Use SplashStart() and SplashEnd() Functions

---

1. Create an unbound form with the text and graphics you want to display on the splash screen. For more information, see the "Setting Form Properties" section below.
2. Create a [macro](#) with the following actions and save it as AutoExec:

| Action | Arguments |
|--------|-----------|
| -----  | -----     |

```
RunCode 1.
<any startup macro actions required by your program> 2.
RunCode 3.
```

#### AutoExec Actions

-----

1. RunCode  
    Function Name: SplashStart("YourSplashFormNameHere", 5)
2. <any startup macro actions required by your program .. optional>
3. RunCode  
    Function Name: SplashEnd()

#### Setting Form Properties

-----

For best results, the splash form should have the [form properties](#) set as follows:

```
Scrollbars: Neither
Pop Up: Yes
Modal: Yes
RecordSelectors: No
```

#### Suppressing the Default Splash Screen

-----

To prevent the default Microsoft Access splash screen from appearing, you can use the run-time version of Microsoft Access, provided with the Microsoft Access Distribution Kit (ADK). You can also activate the /S switch following MSACCESS.EXE in the Command Line property of the Microsoft Access icon in Windows Program Manager.

To change the Microsoft Access icon, use the following steps:

1. Select the icon in the Program Manager by clicking it once.
2. From the File menu, choose Properties.
3. Tab once to the Command Line box, press END, type "/s" (without the quotation marks), and press ENTER.

The Command Line box should now read as follows

```
C:\<ACCESS>\MSACCESS.EXE /S
```

where <ACCESS> is your Microsoft Access program directory.

#### [References](#)

## PRACC9306: Run Time Partially Displays Sorting and Grouping

Article Number: Q101373  
CREATED: 12-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

In the run-time version of Microsoft Access, the Sorting and Grouping properties of a report are partially displayed. The user is able to see only the title bar of the dialog box, as well as the title bar of the Group Properties section.

NOTE: Using the DoMenuItem() function to display other property dialog boxes, like Field List or Palette, does not work. These dialog boxes are not displayed and do not get the focus.

### RESOLUTION

---

If you want to change the Sorting and Grouping properties of a report without allowing the user to see the dialog box, select Echo Off so that the screen is not updated while the macro is running.

You can also prevent users from switching to Design view by using macros and forms to control what actions the user can take while in the application.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access Distribution Kit version 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

Steps to Reproduce Problem

- 
1. In the sample database NWIND.MDB, create the following macro:

| Macro Name | Action                   |
|------------|--------------------------|
| Test1      | OpenReport<br>DoMenuItem |

Test1 Actions:

-----  
OpenReport  
Report Name: Catalog  
View: Design

DoMenuItem

Menu Bar: Report

Menu Name: View

Command: Sorting and Grouping

2. Save this macro as AUTOEXEC. Close the database and quit Microsoft Access.
3. Start the run-time version of Microsoft Access with NWIND.MDB. Switch to Program Manager and choose Run from the File menu. Type the following in the command line:

C:\ACCESS\MSARN110.EXE NWIND.MDB

The Sorting and Grouping dialog box is partially displayed. Note that as you tab through the cells of the dialog box, cells that contain text are visible.

## [References](#)

**PRB: Use "MSACCESS," Not .MDB Filename, in DDE Conversation [B\_WAccADK]**  
**Article Number: Q101372**  
**CREATED: 12-JUL-1993**  
**VERSION(S): MODIFIED: 24-AUG-1993**  
ENDUSER |

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

## **SYMPTOMS**

---

Applications can initiate a dynamic data exchange ([DDE](#)) conversation with a run-time application developed with Microsoft Access by specifying "MSACCESS," rather than the .MDB file, as the application name.

## **RESOLUTION**

---

Microsoft has confirmed this to be a problem with the run-time version of Microsoft Access version 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

## **MORE INFORMATION**

---

To communicate with a run-time application using DDE, you must use only the name of the .MDB file, not "MSACCESS," as the application name when you are initiating a DDE conversation.

Currently the run-time version of Microsoft Access incorrectly responds to both Microsoft Access and the name of the .MDB file as the application name.

### Steps to Reproduce Behavior

-----

1. Start Microsoft Access and open the sample [database](#) NWIND.MDB.
2. In the [Database window](#), choose the Macro button and select Sample AutoExec from the list.
3. From the File menu, choose Rename. In the Macro Name box, type "AutoExec" and press ENTER.
4. Quit Microsoft Access.
5. In Windows Program Manager, choose Run from the File menu.
6. Type the following and press ENTER:

```
C:\<ACCESS>\MSARN100.EXE NWIND.MDB
```

where <ACCESS> is your Access program directory.

7. Start Microsoft Access again and create a new database, or open a database other than NWIND.MDB.
8. Create or open a module and add the following Access Basic function:

Option Explicit

```
Function DDEWithNWIND ()
 Dim chan
 chan = DDEInitiate("MSACCESS", "NWIND;TABLE Employees")
 MsgBox DDERequest(chan, "FirstRow")
 DDETerminate chan
End Function
```

9. From the View menu, choose Immediate Window. Type the following and press ENTER:

```
? DDEWithNWIND()
```

A message containing the first record from the Employees table in NWIND.MDB is displayed.

NOTE: The run-time application responds correctly to the DDEInitiate() statement that reads:

```
chan = DDEInitiate("NWIND", "NWIND;TABLE Employees")
```

The run-time application should respond only to the NWIND.MDB file as the application name, not to "MSACCESS," as in the first example.

## [References](#)

## PRB: How the Scaling Property Affects Embedded MS Draw Object

Article Number: Q101371  
CREATED: 12-JUL-1993  
MODIFIED: 16-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

When you have an embedded Microsoft Draw object that includes both text and graphical elements, the text and graphical elements grow and shrink with different degrees of proportionality.

### RESOLUTION

---

This is by design.

### MORE INFORMATION

---

If the Scaling property is set to Scale when you adjust the size of an embedded object's frame, the object grows or shrinks to fit the size of the frame. This setting allows graphical elements, such as rectangles and ovals, to become distorted. Fonts do not become distorted by resizing. They can grow or shrink, but their aspect ratios are maintained.

Steps to Reproduce Behavior  
-----

1. Create a new unbound form.
2. Create an unbound object frame using the Object Frame tool, which is located directly beneath the Graph tool in the toolbox.
3. Select Microsoft Drawing as the Object Type.
4. Select the Text Object tool in Microsoft Draw, and type some text.
5. Select the Rectangle tool in Microsoft Draw and draw a rectangle around the text. From the Edit menu, choose Send To Back.
6. From the File menu, choose Exit And Return To Microsoft Access. When the dialog box asks you to Update Microsoft Access, choose Yes.
7. Change the Scaling property of this Object Frame control to Scale.
8. Stretch the right side of the object to the right a few inches.

The box stretches. The text stretches, but not to the extent that it becomes disproportionate.

## REFERENCES

=====

"User's Guide," version 1.0, Chapter 13, "Using Pictures, Graphs, and Other Objects," page 320

[References](#)

## PRACC9307: "Table Does Not Exist" Error when Deleting Table

Article Number: Q101370  
CREATED: 12-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

If an [Access Basic function](#) creates a [table](#) by executing a [QueryDef\(\)](#) function and then attempts to delete that table, the following error message appears:

Table does not exist.

The table has not been deleted, but the table will not be displayed again until the [Database window](#) is refreshed.

### CAUSE

---

If the Database window is displaying the table while the function is being executed, the Database window must be refreshed to display the new table.

### RESOLUTION

---

Refresh the Database window before attempting to delete the table. To do this, first select an object other than a table in the Database window and then select the table.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

The following function uses the sample [database](#) NWIND.MDB to demonstrate how to create the table and then successfully delete it. Make sure that you choose Table in the Database window before you run the function below.

NOTE: In the following sample code, an underscore ( \_ ) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function DeleteThisTable ()
 Dim mydb As Database, myq As QueryDef
 Set mydb = CurrentDB()
 Set myq = mydb.CreateQueryDef("createtablequery")
```

```
'Create the Table
myq.sql = "SELECT DISTINCTROW Customers.[Customer ID], _
 Customers.[Company Name] INTO test FROM Customers;"
myq.Execute

'Select any object other than a table to refresh Database
>window.
'NOTE: Omit the following line to reproduce the problem:

DoCmd SelectObject A_FORM, "Customers", True

>Delete the table
DoCmd SelectObject A_TABLE, "test", True
DoCmd DoMenuItem A_DATABASE, A_EDIT, A_DELETE
End Function
```

## [References](#)

## INF: How to Make Forms Always Stay on Top of Other Windows

Article Number: Q101325  
CREATED: 11-JUL-1993  
MODIFIED: 03-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

With the Microsoft Access Popup property, you can create forms that "float" (that is, they stay on top of other forms) within Access, however, if you set focus to a non-Microsoft Access application, these forms are overlapped. If you want your forms to stay on top of forms in non-Microsoft Access windows, you must call the Windows application programming interface (API) subroutine SetWindowPos().

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

### MORE INFORMATION

---

To make your [form](#) stay on top of non-Microsoft Access forms, call the following Access Basic [function](#) from an [event](#) such as your form's OnOpen event property.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive the duplicate procedure name error message, remove or comment out the declarations statement in your code.

1. Add the following to your global declarations section:

```
Declare Sub SetWindowPos Lib "User" (ByVal hwnd%, _
 ByVal hWndInsertAfter%, _
 ByVal X%, ByVal Y%, ByVal cx%, _
 ByVal cy%, ByVal wFlags%)

Global Const HWND_TOPMOST = -1
Global Const SWP_NOSIZE = &H1
Global Const SWP_NOMOVE = &H2
```

2. Create the following function:

```
Function TopMost (F As Form)
 Call SetWindowPos(F.hwnd, HWND_TOPMOST, 0, 0, 0, 0, _
 SWP_NOACTIVATE)
```

End Function

3. Create a form and set the following properties:

OnOpen: =TopMost(Form)

Popup: True

4. Close the form to save any changes; changing the Popup property in [Design view](#) has no effect unless you reopen your form.

5. Open the form. When you switch to any other application, this form should stay on top of the other application.

#### REFERENCES

=====

"Microsoft Windows 3.1 Programmer's Reference," Volume 2

[References](#)

## INF: Access Basic Error-Handling Supplement

Article Number: Q101324  
CREATED: 11-JUL-1993  
MODIFIED: 03-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

This article supplements the information available in the "Microsoft Access Language Reference" about how to handle errors in an Access Basic application.

### MORE INFORMATION

---

There are times when your code becomes susceptible to user error problems, even though the code itself is proven. The larger the program, the more often situations arise that were not anticipated at the time the code was written. For this reason, always include error-handling routines in your code to prevent your application from crashing when unexpected errors occur. Error-trapping routines offer significant help in the [debugging](#) process.

On Error Statement  
-----

[Access Basic](#) includes the On Error statement for error trapping, as well as several other commands for reporting and processing errors. The On Error statement is followed by either a GoTo clause, which causes program [control](#) to branch to a predefined [label](#) when an error occurs, or by a Resume Next clause, which tells Access Basic to ignore the error condition and continue to the next statement.

To create a label for an On Error GoTo statement, type the name of the label, followed by a colon (:). The commands that follow the label are executed when control is passed to the label by the On Error GoTo statement.

Generally, you should place the label and its accompanying error-handling routine at the bottom of the procedure, preceded by an End Sub or End Function statement, to ensure that the routine executes only when branched to and not by mistake.

NOTE: A label does not necessarily indicate a break in program flow.

Shown below is a [Sub procedure](#) that performs a [SQL](#) action based on a SQL [string](#) value. The On Error Resume Next statement is used to handle any errors generated by the first DeleteQueryDef [method](#) (for example, if "TempQuery" does not exist). On Error GoTo SQLError means that if any other errors occur, program control passes to the SQLError error-handling routine. The On Error GoTo 0 statement is used at the

end of the procedure to disable error trapping.

```
Sub PerformSQLAction (InDB As String, SQLStmt As String)
 Dim SQLDb As Database, SQLQuery As QueryDef

 On Error Resume Next
 Set SQLDb = OpenDatabase(InDB)
 SQLDb.DeleteQueryDef ("TempQuery")
 On Error GoTo SQLError:
 Set SQLQuery = SQLDb.CreateQueryDef("TempQuery", SQLStmt)
 SQLQuery.Execute
 SQLQuery.Close
 SQLDb.DeleteQueryDef ("TempQuery")
 On Error GoTo 0
 Exit Sub

SQLError:
 MsgBox "An error occurred while executing the SQL statement."
 Exit Sub
End Sub
```

The Exit Sub statement is required at the end of the error-handling routine, even though the End Sub statement follows. In place of the Exit Sub statement, you could use a Resume Next statement to return control to the line following the line of code that caused the error.

#### Err(), Erl(), and Error\$() Functions

-----

Access Basic includes the Err(), Erl(), and Error\$() functions, all of which return information about the error that occurred. The Error\$() function returns the error message as a string, the Err() function returns a number representing the error message, and the Erl() function returns the number of the line in which the error occurred. Using these functions, you can code your error-trapping routine to display meaningful error messages and trap for specific errors.

A generic error-trapping routine that uses both Error\$() and Erl() functions is shown below. This type of routine tells you what error occurred and where; it also allows you to either ignore the error and continue or cancel execution and further investigate the problem.

```
On Error GoTo ErrorHandler

ErrorHandler:
 If MsgBox("The following error has occurred at line " &
 Trim(Str(Erl)) & ":" & Chr(13) & Chr(10) & Chr(13) &
 Chr(10) & Error$, 17) = 1 Then Resume Next Else Stop
```

The following Sub procedure illustrates how the Error\$() function can create informative error messages to help you and your users debug your program:

```
Sub MyError ()
 On Error GoTo ErrorHandler
 INTEGERVAL% = 99999 'Generates Numeric Overflow error
 Debug.Print "Error was ignored"
```

```

Exit Sub

ErrorHandler:
 If MsgBox("The following error has occurred at line " &
 Trim(Str(Erl)) & ":" & Chr(13) & Chr(10) & Chr(13) &
 Chr(10) & Error$, 17) = 1 Then Resume Next Else Stop
 Exit Sub
End Sub

```

You can use the Err() function to build a SQL SELECT statement that traps for specific errors and takes a different action depending on the error number. You can use the Erl() function to pinpoint exactly which line is causing the problem in your program, to anticipate error conditions in your program, and to handle errors more smoothly.

If you use the Erl() function, you do not need to number every line of your code to find the offending line number. Instead, number only those lines of code most likely to cause problems. (The line numbers represent labels, rather than classic line numbers, as in other Basic languages.) Then, if an error occurs in one of the numbered lines, the Erl() function reports only that line number.

You can also use this method if you have two lines of code that may generate the same error and you want to handle each case separately, as in the following example:

```

Function ErlTest()
 On Error Goto ErlTest_Err
 10: Open "AUTOEXEC.BAT" For Input As #1 'causes an error.
 20: Open "CONFIG.SYS" For Input As #2
 Close
Exit Sub

ErlTest_Err:
 If Erl = 10 Then
 MsgBox "Could not open AUTOEXEC.BAT file."
 ElseIf Erl = 20 Then
 MsgBox "Could not open CONFIG.SYS file."
 End If
 Exit Sub
End Sub

```

Since there is no possibility that a file called "AUTOEXEC.BAT" exists, this line of code errors out and program control passes to the error-handling routine "ErlTest\_Err". The Erl() function detects the number of the line in which the error occurred and displays the "Could not open AUTOEXEC.BAT file" error message.

#### REFERENCES

=====

"Microsoft Access Language Reference," version 1.0, pages 331-333

For more information about error-handling routines, search for "Error handling" using the Microsoft Access Help menu.

#### [References](#)



## INF: Five Parameters to Set in ISAM Section of MSACCESS.INI

Article Number: Q101323  
CREATED: 11-JUL-1993  
MODIFIED: 19-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

This article describes the five parameters that you can set in the ISAM section of the MSACCESS.INI file, found in your WINDOWS subdirectory.

The MSACCESS.INI file can contain the following five parameters:

```
MaxBufferSize
ReadAheadPages
PageTimeout
LockRetry
CommitLockRetry
```

NOTE: These parameters are not automatically placed in the MSACCESS.INI file. If necessary, you must add them.

### MORE INFORMATION

---

ISAM parameters are read at only one time: during the launch of the first instance of Microsoft Access. While the parameters can be changed while Microsoft Access is running, these changes do not take effect until all instances of Microsoft Access are closed and Microsoft Access is restarted.

The examples in the following five sections assume the sample parameter values below:

```
MaxBufferSize = 512
ReadAheadPages = 5
PageTimeout=20 (sets PageTimeout to 2 seconds)
LockRetry = 100
CommitLockRetry = 100
```

MaxBufferSize  
-----

Microsoft Access physically stores all [database](#) data in 2K-memory pages, generally in a large storage medium, such as a hard disk, RAM drive, or CD-ROM. Once placed in the page cache, memory pages are then used by Microsoft Access to hold data for tables, reports, and so forth.

The MaxBufferSize ISAM parameter defines, in kilobytes, how large a memory page cache to set up. A memory page cache is a section of memory used to temporarily store (or, buffer) the 2K-memory pages that Microsoft Access creates. Microsoft Access uses physical memory (RAM), and, if necessary, virtual memory (for example, a hard disk) to create the page cache that you specify.

Since the page cache stores data in 2K segments, you should set the MaxBufferSize parameter only to an even number (for example, 18, 512, 1024, and so forth). If the parameter is set to an odd number, Microsoft Access uses a MaxBufferSize setting of the given number minus 1.

Since the page cache size is measured in kilobytes, and since each memory page is 2K in size, you can calculate how many memory pages will fit in your new page cache by using the following formula:

$$\text{maximum number of memory pages} = (\text{MaxBufferSize} / 2)$$

For example, if you set the MaxBufferSize to 18, you set up a memory page cache of 18K. Applying this number to the above formula yields the following result:

$$(18 / 2) = 9 \text{ possible memory pages}$$

Microsoft Access responds to data read requests by checking the page cache to verify whether or not the requested data is already present.

NOTE: The PageTimeout parameter can also be a factor in this process. Please read the section below titled "PageTimeout."

If the data you need is present in the page cache, Microsoft Access is able to immediately read the memory pages from it. Since Microsoft Access does not have to wait for the cache to be filled from the storage medium, this feature saves you time. However, if the data does not appear on a memory page already present in the page cache, Microsoft Access fills the cache from the storage medium and then retrieves the data from the cache itself.

All memory pages stay in the page cache indefinitely, either until it becomes full or until the computer is turned off. If the page cache becomes full, pages are purged as necessary to make room for the new data.

#### Specifications

---

Maximum Setting: 4096 (4096 = 4MB)

Minimum Setting: 18 (18 = 18K)

Default Setting: 512

Granularity: 2 (for example, 512 is valid, but 511 is not)

#### ReadAheadPages

---

The ReadAheadPages ISAM parameter specifies the size of another cache, known as the "read-ahead" cache. The read-ahead cache is used by Microsoft Access to cache sequential page reads from a database.

Reading data from a read-ahead cache is more efficient than reading it from a storage medium, since sequential data reads can occur more quickly. This is especially true when you are accessing data stored on a [network](#), where speed is increased by sending small numbers of large packets, rather than large numbers of small packets.

The read-ahead cache is not used for all reads, but triggered instead when Microsoft Access detects that a sequential read is taking place. Microsoft Access attempts to detect a sequential read in the following manner:

When Microsoft Access receives a read request, it checks to see if the previous read is adjacent to the current request. If it is, Microsoft Access reads the requested page, plus the next <n> pages (where <n> is the value that the ReadAheadPages parameter is set to) in that direction and places the data in the read-ahead cache. If Microsoft Access correctly detects a sequential read, it then reads the next <n> reads directly from the read-ahead cache, thus improving the speed.

When the read-ahead cache is created, Microsoft Access attempts to place it in the first 640K of memory (also called conventional memory) to benefit from the ability of Microsoft Windows to directly read from and write to conventional memory. A separate read-ahead cache is created for each database open on the host machine. Each library database has its own read-ahead cache.

#### Specifications

-----  
Maximum Setting: 31  
Minimum Setting: 0  
Default Setting: 16  
Granularity: 1 (1 = 1 page)

#### PageTimeout

-----  
NOTE: The PageTimeout parameter is enabled for non-[exclusive](#) (shared) data only.

The PageTimeout ISAM parameter defines the amount of time, in tenths of a second, that a page is held in the memory page cache before it must be refreshed to fulfill the next read of the same page. The following scenario describes how the PageTimeout parameter is used:

If page <x> is read and placed in the memory page cache at time T1, and later another read request is made for page <x> at time T2, the memory page can be read directly from the memory page cache, if T2 minus T1 is less than the PageTimeout setting. Otherwise, Microsoft Access must reread the data from the storage medium to [refresh](#) that particular memory page.

NOTE: PageTimeout applies only to situations in which a database or database object is opened with shared access. When a database is opened with exclusive access, only one user has access to the data; as a result, that user always holds the most current data and has no need for refreshed data.

The PageTimeout parameter is used only when operating through the Microsoft Access user interface. It is ignored when running an Access Basic program, unless you allow background processing to occur by periodically calling the DoEvents() statement.

#### Specifications

-----  
Maximum Setting: 2147483647 (maximum long integer)  
Minimum Setting: 0  
Default Setting: 20  
Granularity: 1 (1 = 1/10 second)

#### LockRetry

-----

The LockRetry ISAM parameter defines the number of times Microsoft Access tries to lock a page before it reports an error.

For example, if user A tries to lock page <x>, which was previously locked by another user, user A's lock attempt fails. After this failure, Microsoft Access tries <n> more times (where <n> is the value LockRetry is set to) to lock the page. If the lock cannot be secured by the <n>th retry, an error is reported either in the user interface or in [Access Basic](#) code.

#### Specifications

-----  
Maximum Setting: 2147483647 (maximum long integer)  
Minimum Setting: 0  
Default Setting: 20  
Granularity: 1

#### CommitLockRetry

-----

The CommitLockRetry ISAM parameter is used in conjunction with the LockRetry parameter to determine the number of retries performed when locking pages within transactions. The total number of retries performed is determined by multiplying the two parameters (LockRetry and CommitLockRetry).

For example, if LockRetry has a value of 5 and CommitLockRetry has a value of 6, 30 more retries (5 \* 6 = 30) are made after the initial failure to secure a lock.

#### Specifications

-----  
Maximum Setting: 2147483647 (maximum long integer)  
Minimum Setting: 0  
Default Setting: 20  
Granularity: 1

#### REFERENCES

=====

For more information on the MSACCESS.INI file, search for "MSACCESS.INI" using the Help menu. Also, see both "Setting [ODBC](#)

Options in the MSACCESS.INI File" in the README.TXT file and the PERFORM.TXT file in your ACCESS subdirectory.

## [References](#)

## PRACC9307: GP Fault with MS Packager Object in Bound OLE Field

Article Number: Q101322  
CREATED: 11-JUL-1993  
MODIFIED: 16-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

A general protection (GP) fault may occur when you attempt to insert or update Microsoft Packager objects in a bound OLE field in Microsoft Access.

### CAUSE

---

During the cleanup process when you quit Microsoft Packager, the program inappropriately frees memory blocks that it does not own. This memory block, which is still being used by OLE, is made available to other applications.

If another application begins using this memory block, OLE and the other application may write information to the same memory location, overwriting each other's information. This conflict may eventually result in a GP fault.

### RESOLUTION

---

A corrected version of Microsoft Packager has been created to work around this problem. This patch, PACKPT.EXE, is a self-extracting zipped file and is available both in the Microsoft Access section of the Microsoft Download Service (MSDL) and in the OLE/DDE section of the Microsoft Access forum on CompuServe.

You can reach the MSDL by calling (206) 936-6735. You can reach the Microsoft Access forum on CompuServe by typing "go msaccess" from any CompuServe exclamation point (!) prompt.

NOTE: To install PACKPT.EXE, copy this file to a new directory and run PACKPT.EXE at the DOS command prompt. The files README.TXT and PACKAGER.EXE are extracted automatically. PACKAGER.EXE should be copied to your Windows directory.

WARNING: MICROSOFT PROVIDES THIS PATCH "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE. The user assumes the entire risk as to the accuracy and the use of this information.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Packager. We

are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

#### [References](#)

## PRB: Cannot Set Focus to Any Object Except MS Access Menu

Article Number: Q101321  
CREATED: 11-JUL-1993  
MODIFIED: 19-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

One of the following two conditions occur:

- You can set the focus only to the Microsoft Access menu. All other objects appear normal, but you cannot set the focus to them.
- or-
- Your application has stopped on a [breakpoint](#) that you set in an [Access Basic function](#), but you cannot use the indicator to set the focus to the Module window.

### CAUSE

---

Your code has executed an Echo action and turned the echo off. You need to turn the echo back on by setting Echo On to Yes (the default).

### RESOLUTION

---

To correct this problem, you need to complete one of the following tasks:

- Correct the logic of your program.
- or-
- Add a key assignment in your Key Assignment [macro](#).

NOTE: By default, the Key Assignment macro is set to the AutoKeys macro.

### MORE INFORMATION

---

Steps to Reproduce Behavior

-----

1. Create the following function:

```
Function FunctionThatAppearsHung()
 DoCmd Echo False
 Debug.Print i
 DoCmd Echo True
End Function
```

- Using the F8 key, set a breakpoint on the following line:

```
Debug.Print i
```

- From the Immediate window, run the function. The following error occurs:

```
?FunctionThatAppearsHung()
```

- To give control back to Microsoft Access, either quit Microsoft Access or define the AutoKeys macro as follows and press F12:

| <u>MacroName</u> | <u>Action</u> | <u>Arguments</u> |
|------------------|---------------|------------------|
| {F12}            | Echo          | On               |

## [References](#)

## PRB: Cannot Auto Update an OLE Linked Object in a Table

Article Number: Q101320  
CREATED: 11-JUL-1993  
MODIFIED: 11-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### SYMPTOMS

---

The Auto Update checkbox is grayed out in the Edit Paste Special [dialog box](#) when attempting to Paste Link an [OLE object](#).

### CAUSE

---

You cannot paste [link](#) an object into a [field](#) of type OLE Object in a [table](#) in Microsoft Access and specify that the link to be "hot", or automatically updated.

You can place linked OLE objects on forms and [report](#) when they are in Design mode. If you paste a link to an OLE Object into a [form](#) or report in design mode, the Auto Update checkbox will be available.

### RESOLUTION

---

This is by design. See the work around described below.

Steps to Reproduce Behavior  
-----

1. Start Microsoft Excel.
2. In [cell](#) A1, type "123", and press ENTER.
3. Select the cell and choose Copy from the Edit menu.
4. Switch to/Start Microsoft Access and create a table that has a field with the datatype set to "OLE Object".
5. Create a new, blank form, based on the table created in step 4.
6. From the View menu, check Field List.
7. Drag the OLE field from the Field List window to the middle of the blank form.
8. Choose Form from the View menu.
9. Place focus in the OLE field, and choose Paste Special from the Edit menu.

Result: "Microsoft Excel Worksheet Object" will appear in the Data Type box, the Paste and Paste Link buttons are available, but the Auto

Update checkbox is grayed out.

## **MORE INFORMATION**

---

There is a significant amount of system resources required to facilitate hot linking, or automatically updating, an OLE object. If the user were to create 10 to 100 records, each with a auto updatable OLE linked object, the penalty on system resources would be enormous. You may experience significant out of memory errors or other system resource related problems.

However, Auto Update is made available to unbound OLE objects inserted in a form or report in design view. This is because the number of object inserted in form is generally minimal, and your system should be quite capable of managing them.

Typically, users tend to work with one, or a few records, at a time. There is little need for Windows to maintain hot links to data between Access and the other OLE servers that are currently not needed by the user. The following work around can be used to "simulate" auto updating.

### Work Around

-----

You can force Access to update the OLE linked field manually by selecting the object to update, choosing Object from the Edit menu, and selecting Update.

This process can be automated using the UpdateOLE() Access Basic function provided below, which can be assigned to a button on the form, or the OnCurrent property of the form. If it is assigned to the OnCurrent property, Access will automatically perform the macro whenever the user navigates to a different record. This will ensure that the data in the record the user is working with will always be current, effectively simulating auto updating for the user.

Create a new module with the following declaration and UpdateOLE() function:

```
'*****
' DECLARATION SECTION
'*****
Option Explicit

'*****
' FUNCTION: UpdateOLE
'
' PURPOSE: Automates updating an OLE linked field.
'
' ARGUMENTS:
' fieldName - A string with the name of the field to update.
'
' EXAMPLE USAGE:
' The OnPush property for a button could be set to:
' =UpdateOLE("OLEField")
'
```

```
' SIDE EFFECTS:
' Focus will be left on the OLE object field.
' The record will be put in Edit Mode.
'
'*****
Function UpdateOLE (ByVal FieldName As String)
 On Error GoTo ErrUpdateOLE
 DoCmd GoToControl FieldName
 DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_OBJECT, A_OBJECTUPDATE

ByeUpdateOLE:
 Exit Function

ErrUpdateOLE:
 Resume ByeUpdateOLE

End Function
```

Using the UpdateOLE() Function

---

You can use the UpdateOLE() function by assigning it to a button on the form or to the OnCurrent property of the form. If performance is a problem with the OnCurrent solution, the button approach allows the user to decide when the data should be refreshed.

Given that a linked OLE field on your form has the name "OLEField", set the OnPush property of the button or the OnCurrent property for the form to the following:

```
=UpdateOLE("OLEField")
```

If you have multiple OLE linked fields, you can create a macro to call UpdateOLE() for each OLE field on the form.

Given that you have 3 linked OLE fields on a form with the names "OLEField1", "OLEField2", and "OLEField3", create the following macro and save it with the name "UpdateAllOLEFields".

| Action  | Argument                               |
|---------|----------------------------------------|
| RunCode | Function Name: =UpdateOLE("OLEField1") |
| RunCode | Function Name: =UpdateOLE("OLEField2") |
| RunCode | Function Name: =UpdateOLE("OLEField3") |

Then set the OnPush property of the button or the OnCurrent property of the form to the following:

```
UpdateALLOLEFields
```

[References](#)

## PRB: Unexpected Error from External Database Driver [22]

Article Number: Q101319  
CREATED: 11-JUL-1993  
MODIFIED: 11-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SYMPTOMS

---

When you try to open a attached Btrieve table, you receive the following error:

Unexpected error from external database driver [22]

### CAUSE

---

This error message is actually a Novell error message that is returned when there is a mismatch of information between the FIELD.DDF and the Btrieve data table. This usually occurs when someone has changed the internal structure of the Btrieve table but did not reflect those same changes to the FIELD.DDF file. The Xtrieve manual lists this error with the following description:

- 22 Data buffer length too short  
The data buffer parameter is not long enough to accommodate the length of the data record defined when the file was created. Verify that the length of the data buffer is at least as long as the file's defined record length.

### RESOLUTION

---

Make sure that the internal structural definition of the Btrieve table is the same for the FIELD.DDF file. This can be accomplished by using Novell's Xtrieve or a third party DDF builder to update the FIELD.DDF file.

### MORE INFORMATION

---

Refer to the Novell Xtrieve manual, Appendix A for a complete list of error messages and status codes.

### References

## INF: Using DDE to Communicate with a Runtime Application

Article Number: Q101318  
CREATED: 11-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

Runtime applications developed with Microsoft Access have the same [DDE](#) server capabilities as the retail version of Microsoft Access.

However, in order to communicate with a runtime application using DDE, the name of the MDB file must be used as the application name, instead of MSACCESS, when initiating a DDE conversation.

For more information on using Microsoft Access as a DDE Server, [query](#) on the following keywords:

DDE and server and topics

### More Information:

---

The following example demonstrates how to communicate with a runtime application developed with Microsoft Access.

1. Start the retail version of Microsoft Access and open the Northwind Traders [database](#) (NWIND.MDB) supplied with Microsoft Access.
2. Choose the Macro button from the [Database window](#) and select the [macro](#) named "Sample Autoexec" in the list.
3. Choose Rename from the File menu. In the Macro Name box, type "Autoexec" and press Enter.
4. Exit Microsoft Access.
5. Choose Run from the File menu in the Windows Program Manager.
6. Type the following and press enter:  

```
c:\access\msarn110.exe nwind.mdb
```
7. Start the retail version of Microsoft Access and create or open a database other than NWIND.MDB.
8. Create or Open a [module](#) and add the following Access Basic [function](#):

Option Explicit

```
Function DDEWithNWIND ()
 Dim chan
 chan = DDEInitiate("NWIND", "NWIND;TABLE Employees")
 MsgBox DDERequest(chan, "FirstRow")
 DDETerminate chan
End Function
```

9. Activate the Immediate window from the View menu, type the following, and press Enter:

```
? DDEWithNWIND()
```

Result: A message box will display with the first record of data from the Employees table in the NWIND database.

## [References](#)

## **PRB: "Maximum Size Reached" Error Attempting to Import File**

**Article Number:** Q101316  
**CREATED:** 11-JUL-1993  
**MODIFIED:** 19-JUL-1993  
**VERSION(S):** 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

You receive the following error message when you attempt to import a file that increases your .MDB file size to larger than 128 megabytes (MB), or 131,072,000 bytes:

Maximum size reached

### **CAUSE**

---

Although Microsoft Access version 1.1 can support databases up to 1 gigabyte (GB), Microsoft Access version 1.0 supports databases only up to 128 MB. If your .MDB file is in Microsoft Access version 1.0 format and you use it in Microsoft Access version 1.1, and then try to import data that increases your database size to larger than 128 MB, the "Maximum size reached" error message is displayed.

This error occurs because your database is still subject to the 128-MB maximum size in Microsoft Access version 1.0.

This situation typically occurs when you upgrade from Microsoft Access version 1.0 to version 1.1 and continue to use databases originally created in version 1.0, without converting the files to version 1.1 format.

### **RESOLUTION**

---

To correct this problem, use the following procedure to compact your version 1.0 database file and convert it to version 1.1 format, which supports .MDB files up to 1 GB:

1. From the File menu, choose Compact Database.
2. In the List Files Of Type box, select Access V1.1 (\*.mdb).

### [References](#)

## INF: Using Microsoft Access as a DDE Server

Article Number: Q101313  
CREATED: 11-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

Microsoft Access supports dynamic data exchange (DDE) as both a destination (client) application and a source (server) application. This article documents the DDE topics that Microsoft Access supports as a DDE server and the valid DDE items for each topic.

NOTE: In this article, an asterisk (\*) indicates a new feature in Microsoft Access version 1.1.

The examples below require that you have the sample [database](#) NWIND.MDB open in Microsoft Access.

### MORE INFORMATION

---

As a DDE server, Microsoft Access supports the following topics:

- The System topic
- The name of a database, [table](#), or [query](#)
- A Microsoft Access [SQL](#) statement

This article contains descriptions of the five topics and the items that are supported in each topic.

The System Topic  
-----

The System topic is a standard topic for all Microsoft Windows-based DDE server applications and returns information about the topics supported by the application. The System topic supports the following Microsoft Access data items:

- SysItems - A list of items supported by the System topic in Microsoft Access
- Formats - A list of formats Microsoft Access can copy to the [Clipboard](#)
- Status - "Busy" or "Ready"
- Topics - A list of all open databases

Information returned from any item used with the System topic is tab delimited, Chr\$(9).

The following Microsoft Word for Windows WordBasic [macro](#) demonstrates how to use the System topic to get information on available topics from Microsoft Access:

```
Chan = DDEInitiate("MSAccess", "System")
Topics$ = DDERequest$(Chan, "Topics")
DDETerminate Chan
MsgBox Topics$, "Topics", 64
```

#### The Database Topic

---

The database topic is the filename of an existing database. After you initiate a DDE conversation with the database, you can request a list of objects in that database. This list of information is tab delimited, Chr\$(9).

NOTE: You cannot query the SYSTEM.MDB file using DDE.

The database topic supports the following items:

```
TableList
QueryList
FormList
ReportList
MacroList
ModuleList
```

The following WordBasic macro demonstrates how to retrieve a list of table names:

```
Chan = DDEInitiate("MSAccess", "NWIND")
TableList$ = DDERequest$(Chan, "TableList")
DDETerminate Chan
```

#### The TABLE TableName, QUERY QueryName, and SQL SqlString Topics

---

The TABLE, QUERY, and SQL DDE topics are used to retrieve information from Microsoft Access tables. This list is tab-delimited, Chr\$(9).

The [syntax](#) for these topics is as follows:

```
<DatabaseName>; TABLE <TableName>
<DatabaseName>; QUERY <QueryName>
<DatabaseName>; SQL <SqlString>
```

#### Description of Syntax:

<DatabaseName> The name of the database to which the table or query belongs or the SQL statement applies, followed by a semicolon (;). The database name can be either the base name only (for example, NWIND) or its full path and .MDB extension (for example, C:\ACCESS\NWIND.MDB).

<TableName> The name of an existing table.

<QueryName> The name of an existing query.

<SqlString> A valid SQL statement up to 255 characters, followed by a semicolon.

\*NOTE: You can exchange more than 255 characters by omitting this argument and using successive DDEPoke() statements to build an SQL statement.

For example, the following WordBasic code uses the DDEPoke() [function](#) to build an SQL statement and request the results of the query:

```
Chan = DDEInitiate("MSAccess", "NWIND;SQL")
DDEPoke Chan, "SQLText", "SELECT * FROM Orders "
DDEPoke Chan, "SQLText", "WHERE [Order Amount] > 1000;"
Results = DDERequest$(Chan, "Data")
DDETerminate Chan
```

The following is a list of valid DDE items for the TABLE TableName, QUERY QueryName, and SQL SqlString DDE topics:

All All data in the table, including [field](#) names.

Data All rows of data, without field names.

\*FieldNames A single-[row](#) list of field names.

FieldNames;T Two records of data, the first a list of field names and the second a list of data types.

The following are the values returned and the data type each value represents:

|    |                                                                         |
|----|-------------------------------------------------------------------------|
| 0  | Invalid                                                                 |
| 1  | True/False (non-NULL)                                                   |
| 2  | Unsigned byte (Byte)                                                    |
| 3  | 2-byte signed integer (Integer)                                         |
| 4  | 4-byte signed integer (Long)                                            |
| 5  | 8-byte signed integer (Currency)                                        |
| 6  | 4-byte single-precision floating-point (Single)                         |
| 7  | 8-byte double-precision floating-point (Double)                         |
| 8  | Date/Time (date is integer, time is fraction)                           |
| 9  | Binary data, 255-byte maximum                                           |
| 10 | ANSI text, not <a href="#">case-sensitive</a> , 255-byte maximum (Text) |
| 11 | Long binary ( <a href="#">OLE</a> Object)                               |
| 12 | Long text (Memo)                                                        |

NextRow The data in the next row in the table or query. When you first open a channel, NextRow returns the data in the first row. If the current row is the last [record](#) and you execute NextRow, the request fails.

PrevRow The data in the previous row in the table or query. If PrevRow is the first request on a new channel, the data in the last row of the table or query is returned. If

the first record is the current row, the request for PrevRow fails.

FirstRow           The data in the first row of the table or query.

LastRow            The data in the last row of the table or query.

FieldCount         The number of fields in the table or query.

\*SQLText           An SQL statement representing the table or query.  
For tables, this item returns an SQL statement in the  
format "SELECT \* FROM table;".

\*SQLText;<n>       An SQL statement in <n>-character chunks that  
represents the table or query, where <n> is an integer  
lower than 255. For example, suppose a query is  
represented by the following SQL statement:

```
SELECT * FROM Orders;
```

The item "SQLText;7" would return the following  
tab-delimited chunks:

```
"SELECT "
"* FROM "
"Orders;"
```

The following WordBasic macro demonstrates how to get information from  
the Employees table in NWIND.MDB:

```
Chan1 = DDEInitiate("MSAccess", "NWIND;TABLE Employees")
' Get a count of the number of Employee records.
SQL$ = "SELECT Count([Employee ID]) AS [CountOfEmployees] "
SQL$ = SQL$ + "FROM Employees;"
Chan2 = DDEInitiate("MSAccess", "NWIND;SQL " + SQL$)
EmployeeCount = Val(DDERequest$(Chan2, "FirstRow"))
DDETerminate Chan2

' Quit if there are no records.
If EmployeeCount <> 0 Then
 Msg$ = "NWIND Employee Information:"
 Msg$ = Msg$ + " Record Count:" + Str$(EmployeeCount)
 MsgBox Msg$

 ' Request the first row of data from the Employees table.
 Data$ = DDERequest$(Chan1, "FirstRow")

 ' Display the records.
 For i = 1 To EmployeeCount
 MsgBox Data$
 ' Get the next row of data, if not at the end.
 If i <> EmployeeCount Then
 Data$ = DDERequest$(Chan1, "NextRow")
 End If
 Next i
End If
```

## Executing Macros and Commands in Microsoft Access Using DDE

---

When you are using Microsoft Access as a DDE server, you can use the DDEExecute() function to instruct your application to execute a command. Microsoft Access recognizes any of the following valid commands:

- The name of a macro in the database that is currently open. A macro can be executed on a channel with any of the five topics described above.
- \*- Any action that you can execute in [Access Basic](#) using the DoCmd() statement. You cannot execute the following macro actions: AddMenu, MsgBox, RunApp, RunCode, SendKeys, SetValue, StopAllMacros, and StopMacro.
- \*- The OpenDatabase and CloseDatabase actions, executed only for DDE operations using the System topic.

NOTE: When you specify an action as a DDEExecute command, the action and any arguments follow the DoCmd() syntax and must be enclosed in brackets ([]). However, applications that support DDE do not recognize intrinsic constants, such as A\_NORMAL, in DDE operations. Therefore, you must use the actual number as an argument. Also, [string](#) arguments must be enclosed in quotation marks only if the string contains a comma. Otherwise, quotation marks are not required.

The following macro opens the Categories [form](#), first minimized and then restored:

```
Chan = DDEInitiate("MSACCESS", "System")
AppActivate "Microsoft Access"
DDEExecute Chan, "[OpenForm Categories,,,,,2]"
DDEExecute Chan, "[OpenForm Categories]"
DDETerminate Chan
```

### \*Using the OpenDatabase and CloseDatabase Commands

---

Microsoft Access 1.1 introduces some new commands that can be executed on a channel opened to the System topic. These commands facilitate the remote opening and closing of databases in Microsoft Access from the client application.

- OpenDatabase DatabaseName [, Exclusive[, ReadOnly]]
- CloseDatabase

DatabaseName      A string [expression](#) that is the name of an existing database. This can include the fully qualified MS-DOS path.

Exclusive          A [Boolean](#) value that is True (-1) if the database is to be opened with [exclusive](#) (nonshared) access and False (0) if the database is to be opened with shared access. The default is shared access.

ReadOnly            A Boolean value that is True if the database is to be opened with read-only access and False if it is to be opened with read/write access. The default is read/write access.

The following WordBasic macro demonstrates how to use the OpenDatabase actions to remotely open NWIND.MDB and then open the Employees form:

```
Chan = DDEInitiate("MSACCESS", "System")
DDEExecute Chan, "[OpenDatabase NWIND.MDB]"
DDEExecute Chan, "[OpenForm Employees,0,,1,0]"
DDETerminate Chan
```

#### REFERENCES

=====

"User's Guide," Chapters 9 and 13

README.TXT for Microsoft Access version 1.0

For more information on using DDE with Microsoft Access, query on the following words here in the Microsoft Knowledge Base

dde and <the name of the other application in the conversation>

or search for "DDE" using the Microsoft Access Help menu.

[References](#)

## PRB: Cannot open multiple instances of the same form

Article Number: Q101312  
CREATED: 11-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

When attempting to open a second instance of an already opened [form](#) using the OpenForm [macro](#) action, the currently opened form is activated.

### CAUSE

---

You can only have one copy of a form opened at a time.

This behavior is by design. This feature is under review and will be considered for inclusion in a future release. See the work around described below.

### MORE INFORMATION

---

The following information describes how to create and use an Access Basic [function](#), OpenFormInstance(), that allows you to open more than one copy of the same form in Microsoft Access. In order for OpenFormInstance to work, you must first make X number of copies of your original form giving them a numeric prefix greater than zero. i.e. If the form is called "Customers", make copies of the form renaming them to "Customers1", "Customers2", "Customers3" ...

When OpenFormInstance() is invoked with "Customers", it will check to see if a copy of the form is available for display, or return the following error message if all instances are currently opened:

```
Could not open form "Customers"
Please close another instance and try again.
```

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

Create a new [module](#) with the following Access Basic functions:

Note: In the following sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****
' DECLARATION SECTION
```

```

'*****
Option Explicit

'*****
' FUNCTION: OpenFormInstance
'
' PURPOSE:
' Allows you to open more than one copy of the same form
' on screen at a time.
'
' PREREQUISITES:
' You must make X number of copies of the the interested
' form, renaming them with a numeric suffix greater than zero.
' Ex:
'
' "Customers1", "Customers2", "Customers3" ...
'
' REQUIRES:
' Function IsLoaded() - Determines if form is loaded.
' Function GetFormNames() - Supplied in this module.
'
' ARGUMENTS:
' FormName - The generic name of the form to open, "Customers"
'
' RETURN:
' True - if an instance of FormName was opened.
' False - if no instance of FormName was found or all available
' instances are active.
'
' NOTES:
' You can customize the OpenForm action in the code below to add
' additional (optional) parameters, such as a Where condition or
' or alternate view.
'
'*****
Function OpenFormInstance (ByVal FormName As String)
 Dim Count, i, Msg As String, InstanceCount
 ReDim Names(0) As String

 ' Get a list of all the forms in the current database
 Count = GetFormNames(Names())

 ' Loop through the list of forms for a match with
 ' .. the requested form
 For i = 0 To Count - 1
 ' Is the name of the form, minus the suffix, the same?
 If FormName = Left(Names(i), Len(FormName)) Then

 ' .. AND is the suffix a number greater than zero?
 If Val(Mid(Names(i), Len(FormName) + 1)) > 0 Then

 ' Count the number of instances
 InstanceCount = InstanceCount + 1

 ' If the form is NOT loaded,load it
 If Not IsLoaded(Names(i)) Then
 DoCmd OpenForm Names(i)
 End If
 End If
 End If
 Next i
End Function

```

```

 OpenFormInstance = True
 Exit Function
 End If
End If
End If
Next i

' No form was found or all instances are being used
OpenFormInstance = False

' If no instance found, just return
If InstanceCount = 0 Then Exit Function

' If all instances are being used, display error message
Msg = "Couldn't open form "" & FormName & ""."
Msg = Msg & Chr$(13) & Chr$(13)
Msg = Msg & "Please close another instance and try again."
MsgBox Msg, 48

End Function

'*****
' FUNCTION: GetFormNames
'
' PURPOSE:
' Fills the string array with a list of forms names
'
' ARGUMENTS:
' Names - A single dimensional array of type string
'
' RETURN:
' The number(zero based) of names stored in the Names array.
'
' NOTES:
' This function utilizes information stored in the
' MySysObject table of the currently opened user database.
' The system tables are undocumented and are subject to
' change in future versions of Access.
'*****
Function GetFormNames (Names() As String)
 Dim db As Database, ss As Snapshot
 Dim Count, SQL
 Dim Msg As String

 SQL = "Select Name,Type from MSysObjects Where Type="
 SQL = SQL & "-32768 And Left(Name,1)<>'~' Order By Name;"
 Set db = CurrentDB()
 Set ss = db.CreateSnapshot(SQL)

 ss.MoveLast
 If ss.RecordCount > 0 Then
 ReDim Names(0 To ss.RecordCount - 1)
 Else
 GetFormNames = 0
 Exit Function
 End If

```

```

 ss.MoveFirst
 Count = 0
 Do While Not ss.EOF
 Names(Count) = ss![name]
 Count = Count + 1
 ss.MoveNext
 Loop

 GetFormNames = ss.RecordCount
End Function

```

#### How to use the OpenFormInstance() Function

---

Suppose there is a need to see more than 1 customer form on the screen at a given time in the Northwind Traders database supplied with Microsoft Access.

1. Open the Northwind Traders database (NWIND.MDB) and choose the Forms button in the [Database window](#).
2. Click, 1 time, on the Customers form, and choose Copy from the Edit menu.
3. Choose Paste from the Edit menu, and type "Customers1" in the Form Name box and press Enter.
4. Repeat step 3 for however many instances you expect your users to need... "Customers2", "Customers3"...
5. Open the module with the OpenFormInstance() function in Design and choose [Immediate window](#) from the View menu.

6. Type the following and press Enter:

```
? OpenFormInstance("Customers")
```

Note: The first instance of the Customers form opens.

7. Place your cursor on the previously typed line and press Enter.

Note: The second instance of the Customers form opens.

8. Repeat step 7 until you get an error message indicating that no more instances are available.

#### Problems to consider when using OpenFormInstance()

---

Since we are not really opening up the same form multiple times, but copies, it is important that any macros and module code that make direct references to the form using "Forms!FormName" references be reconsidered.

If you have more than one instance of the form opened, it is best to use the following techniques when referencing controls on the form.

1. Use Screen.ActiveForm form notation where ever possible.

```
MsgBox Screen.ActiveForm![ControlName]
```

2. When Screen.Active form is not appropriate, pass the name of the form to the Access Basic function to call.

```
Function MyFunction(MyForm)
 MsgBox Forms(MyForm)![ControlName]
End Function
```

3. Since you cannot pass arguments to macros, consider reworking the macro as an Access Basic function. You could call the function MyFunction() above using a RunCode macro action as follows:

```
RunCode
 Function Name: =MyFunction(Screen.ActiveForm.FormName)
```

For more information on referring to form and control objects, see Chapter 7, "Forms and Report Variables" in the Introduction to Programming Guide.

## [References](#)

## INF: Microsoft Access ADK Legal Agreement [B\_WAccADK]

Article Number: Q101311  
CREATED: 11-JUL-1993  
MODIFIED: 11-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This article provides some additional information about the legal restrictions on distributing applications developed with the Microsoft Access Distribution Kit (ADK).

### More Information:

---

Pages v and vi of the Microsoft Access Distribution Kit Guide include a list of provisions that define what developers may distribute.

1. You may distribute these files only as integrated within and as a part of your software product which operates in conjunction with Microsoft Access.

Explanation: This clause means that you may only distribute the runtime files with your Microsoft Access application. You can not purchase the ADK and then distribute individual files. This also applies to the [ODBC](#) disk as well.

Example: An employee at a large company purchases the ADK. Her manager has accidentally removed Graph and Mary gives her manager a copy of the run-time Graph application that comes with the ADK. This action is a violation of the License Agreement. Mary may ONLY distribute files as part of a [database](#) application.

2. Your software product does not substantially duplicate the capabilities of Microsoft Access or, in the reasonable opinion of Microsoft, compete with Microsoft Access.

Explanation: Developers cannot use the ADK to create a database application to compete with Access. However, it is OK for developers to create custom applications based on Microsoft Access. If developers believe they might be in competition with Microsoft Access, they should contact the Microsoft Access Product Manager.

Example 1: John Parns uses the ADK to create a Form Designer, Report Designer, and Query Grid that look almost identical and have the same functionality as Microsoft Access. He then sells this application to users as "an inexpensive alternative to Microsoft Access". This would be a direct violation of the license agreement.

Example 2: Kara Reed works at a company which uses [SQL](#) Server

extensively and purchases the ADK. She designs several forms which allow the user to connect to various SQL Servers and "query by example". She creates a custom setup which includes both the ODBC setup disk and the run-time Graph application. She runs an administrator setup to a network drive so that anyone in the company may set up her application. Because she has purchased a copy of the ADK and is only distributing files as part of a database application which she has developed this is legal.

3. You do not use Microsofts name, logo, or trademarks to market your software product.

Explanation: This simply means that you may not use the name and logo of Microsoft as though your application were a Microsoft product. They can use Microsoft Access (R) if used properly with a registered trade mark (subject to normal trade mark laws) and if it is used only as a description.

Developers are encouraged to inform their customers that the product is based on Microsoft Access. However, Microsoft cannot guarantee, recommend or provide support for products developed with the ADK.

Example: Alex Smith has developed an application that she intends to sell to other developers. The front of the box her application is packaged in reads:

AX Utilities  
Secure Screen Savers  
An Application For Use With Microsoft Access (R)

requires: 4 mb hard disk space (1.5 mb if you have already installed another application for Microsoft Access)

Because (1) the proper registered trademark appears, and (2) Microsoft Access (R) is included only in the description of her application, the use is legal under the license agreement.

4. You include a valid copyright notice on your software product.

Explanation: A copyright notice is required before you can distribute a database application that you develop with the ADK.

Example: Roger Plattan has developed a game using the ADK. He plans to post it in the bulletin board that he operates so that others can download it for free. He does tell readers that the game was developed using Microsoft Access (R) but does not include a copyright notice. This is not legal because he does not have a valid copyright notice.

5. You agree to display, if required in the software documentation, the designated patent notices on the packaging and in the READ file of your software product.

Explanation: This is part of the standard Microsoft License agreement. Any items requiring a patent notice will be indicated in the product documentation or the product readme file.

References: Microsoft Access Distribution Kit pp. v, vi

[References](#)

## INF: Controlling Design Mode at Runtime to Change Properties

Article Number: Q101308  
CREATED: 11-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

There are a number of form properties in Microsoft Access that cannot be changed while the form is active. To get around this, you can use a macro to force the form into design mode, use SetValue actions to change any property, then force the form back into browse mode.

This article how to change the form modes back and forth as cleanly as possible.

### More Information:

---

You can force the form into design mode by using a DoMenuItem macro action as shown below:

```
MacroName Action

Test1 DoMenuItem

Test1 Actions

DoMenuItem
 Menu Bar: Form
 Menu Name: View
 Command: Form Design
```

When you use this approach for changing properties on an Microsoft Access form, there are a couple of potential problems to be aware of, all of which can be addressed:

1. When changing from browse mode to design mode or vice-versa, the form is in design mode yet repainting takes place on the screen. To minimize the problem, set Echo Off before going into design mode. Turn Echo On after forcing the form back into browse mode.

```
MacroName Action

Test1 DoMenuItem
 Echo
Test2 DoMenuItem
 Echo

Test1 Actions

```

```
DoMenuItem
 Menu Bar: Form
 Menu Name: View
 Command: Form Design
```

```
Echo
 Echo On: No
```

#### Test2 Actions

```

DoMenuItem
 Menu Bar: Form
 Menu Name: View
 Command: Browse
Echo
 Echo On: Yes
```

2. After changing a property, the save form dialog appears. This can be resolved by setting SetWarnings to Off from the form's OnClose [event](#). If this is done in a macro, the SetWarnings will automatically be re-enabled after the form has closed.

To illustrate this, follow these steps:

1. Create a new form called Form1 bound to the Categories [table](#) in NWIND.MDB with the following settings:

```
Form

Form: Form1
 Caption: Form1
 ControlSource: Categories
 OnClose: Macro1.Onclose
Textbox: text box1
 ControlName: Category ID
 ControlSource: Category ID
Textbox: text box2
 ControlName: Category Name
 ControlSource: Category Name
command button: Button1
 Caption: Filter
 OnPush: =ChangeFilter()
command button: Button2
 Caption: Close
 OnPush: =Macro1.Close
```

5. Create a macro called Macro1 with the following options:

| MacroName | Action      | Defined Below |
|-----------|-------------|---------------|
| OnClose   | SetWarnings | 1.            |
| Close     | Close       | 2.            |

#### 1. OnClose Action

```

Warnings On: No
```

## 2. Close Action

-----

Object Type: Form  
Object Name: Form1

6. Add the following [function](#) to a new [module](#):

```
Function ChangeFilter ()
 DoCmd Echo False
 DoCmd DoMenuItem 0, 2, 0, 0
 Forms!Form1.AllowFilters = Not Forms!Form1.AllowFilters
 DoCmd DoMenuItem 0, 2, 1, 0
 DoCmd Echo True
End Function
```

NOTE: For development work, you may want to comment out the DoCmd Echo actions.

7. Close and save the form, macro, and module.

Note: Before closing the form, make sure that the Field List window, the Form properties window and the [toolbox](#) are closed.

8. Open the form and click the Filter button.

Notice the [filter](#) buttons in the tool bar gray out. If you choose the Filter button again, the filter buttons in the tool bar are re-enabled.

9. Choose the Close button or close the form by double-clicking on its Control Box.

Notice that, regardless of which way the form is closed, no Save Form dialog appears.

## [References](#)

## PRB: Echo Without Arguments Causes Error

Article Number: Q101126  
CREATED: 06-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### SYMPTOMS

---

Leaving the EchoOn argument out of the Echo action in a [module](#) results in an Expected Expression Error. The OnLine help misleads the user into believing that the [expression](#) will default to yes.

### RESOLUTION

---

The Echo action's EchoOn argument must be included in the line of code in order for the module to execute properly.

Steps to Reproduce Behavior

- 
1. Open the sample [database](#) Northwind Traders (NWIND.MDB).
  2. From the File menu choose New Module.
  3. Enter in the line Function Test() and hit enter.
  4. Enter the following code into the new window:

```
'=====
'The following function will produce the syntax error mentioned above
'upon typing in the Echo action.
'=====
Function Test()
 Docmd Echo 'Notice the arguments are missing
End Function
```

References:

"Microsoft Access Language Reference," version 1.0, pages 163-164

For more information search for "Echo Action" using the Help menu.

[References](#)

## PRB: SendKeys Action doesn't work on Toggle Keys

Article Number: Q101125  
CREATED: 06-JUL-1993  
MODIFIED: 06-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### SYMPTOMS

---

SendKeys [macro](#) action does not work on toggle keys such as the CAPS+LOCK and SCROLL+LOCK. You may see the CAPS+LOCK or SCROLL+LOCK lights flash on the keyboard; however, these lights will not remain on after you run the SendKeys macro. SendKeys also does not work to run the keystrokes Alt-Print Screen.

### CAUSE

---

SendKeys key strokes are sent at a high level to a given application. Windows traps toggle keys and print screen at a lower level.

### RESOLUTION

---

There is a resolution for the CAPSLOCK problem only. To capitalize user input into a [field](#), the ">" symbol can be used in the Format property of the [control](#) on a [form](#). The UCASE() [function](#) can also capitalize the text in a control.

A Windows API function SetKeyboardState() call can toggle these keys.

```
Declare Sub SetKeyboardState Lib "User" (lpKeyState As Any)
```

### MORE INFORMATION

---

Steps to Reproduce Behavior

-----  
The following sendkeys action will not toggle the CAPSLOCK key to on:

```
MacroName Action

Macrol SendKeys

Macrol Actions

SendKeys
 Keystrokes: {CAPSLOCK}
 Wait: NO
```

References: "Microsoft Access Language Reference", pg. 432, 485

[References](#)



## PRB: NWIND Order Form Updates Order Amount Field with Zero

Article Number: Q101090  
CREATED: 04-JUL-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0
- 

### SYMPTOMS

---

The Order Amount field in the Orders table is incorrectly set to zero.

### CAUSE

---

The Orders.Write Order Amount macro writes the order subtotal, calculated in the subform footer, to the Order Amount field in the Orders table before the order subtotal has finished calculating.

The order subtotal is calculated using the Sum() aggregate function, which runs asynchronously of (or, independently of) of other actions on the form. When a field is altered in the subform that affects this calculation (for example, the Quantity field), the AfterUpdate event for the subform executes the Orders.Write Order Amount macro. This macro executes before the new order subtotal can be calculated. As a result, the macro incorrectly updates the Subtotal field in the table.

### RESOLUTION

---

The Order Subtotal field in the subform footer cannot be relied on to calculate the correct order subtotal, since it is being calculated independently of other events occurring on the form. Instead, the value should be calculated using the DSum() aggregate function in the Orders.Write Order Amount macro. This result can be written to the Order Amount field in the Orders table.

### MORE INFORMATION

---

This change has already been made to the sample database NWIND.MDB that shipped with Microsoft Access version 1.1. Following the "Steps to Reproduce Behavior" section is a description of changes that must be made to version 1.0 of the sample database NWIND.MDB to correct this problem. There are additional suggestions to optimize the Orders form for both versions 1.0 and 1.1.

Steps to Reproduce Behavior  
-----

1. Open the sample database NWIND.MDB supplied with Microsoft Access version 1.0.
2. Open the Orders form in Form view.
3. Change the Quantity field for one of the products listed in the

Orders Subform. (Note the Order ID of the order that you are changing.)

4. Open the Orders table and find the record with the same Order ID.

Result: The Order Amount field is set to zero.

#### Using DSum() Function to Correctly Calculate Order Subtotal

---

The following steps describe how to change the Orders form in Microsoft Access version 1.0 to implement the DSum() function:

NOTE: The sample database ORDENTRY.MDB supplied with Microsoft Access uses the same technique to calculate the order subtotal. Please see the FrmOrders.Recalc macro in ORDENTRY.MDB for more information.

1. Open the Orders macro in Design view.
2. Find the SetValue action for the Write Order Amount macro group.
3. Change the expression argument from

Forms![Orders]![Orders Subform].Form![Order Subtotal]

to

DSum("[Extended Price]","[Order Details2]","[Order ID] = Forms![Orders]![Order ID]")

4. Save the macro.

#### Additional Optimizations for Orders Form

---

Since the Orders Subtotal field in the Order Subform footer is no longer necessary, use the following steps remove it and increase your database performance:

1. Open the Orders Subform in Design view.
2. Remove the Orders Subtotal field from the footer.
3. Open the Orders form and select the Subtotal field. Change the the ControlSource property from

=[Orders Subform].Form![Order Subtotal]

to

Order Amount

In Microsoft Access version 1.0, you can remove "Orders.Write Order Amount" from the BeforeUpdate property of the Orders form.

#### [References](#)

## PRB: Don't have Permission to Modify Secured Report

Article Number: Q101089  
CREATED: 04-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

You receive the following error message:

You don't have permissions to modify <report name> when trying to print a secured report.

### CAUSE

---

Secured reports opened in print preview cause the error. This will happen if the application is given to users that do not have the same type of default printer selected as the creator of the application.

For example:

Create a report and at the time of creation have the Generic/Text printer selected as your default printer.

Implement security on your application giving only execute permissions to the report.

Install your application to a machine that has any other printer selected as the default printer.

Open the report in print preview. After closing the report you will be prompted with the error message.

### RESOLUTION

---

Create a macro and add it to the on close of the report.

| Macro Name     | Action   | Argument |
|----------------|----------|----------|
| Secured Report | SendKeys | {ESC}    |

-----

Steps To Reproduce Behavior

=====

To properly produce this error you need to have already implemented security for a database.

1. Log into Microsoft Access as any user other than Admin
2. Create a new database

3. Import the Employees table from Nwind.mdb
4. Using the report wizard create a new report based on this table and save it as Report1
5. Create an AUTOEXEC macro with the following actions/arguments.

| Action     | Argument | View          |
|------------|----------|---------------|
| OpenReport | Report1  | Print Preview |

6. Assign execute permissions to the report for all groups and users.
7. Exit Access
8. Go to the control panel in windows and select a different printer for your default printer.
9. Start Access, login as a different user.
10. Open the database. The report will open in preview.
11. Close the report and the error will occur.

References:

[References](#)

## PRB: Domain Parameter of Dom Funct Cannot Be SQL Expression

Article Number: Q101088  
CREATED: 04-JUL-1993  
MODIFIED: 08-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

If you try to use an [SQL](#) SELECT statement as the [domain](#) parameter of a [domain function](#), error 3078 is displayed:

Couldn't find input [table](#) or [query](#) <SQL [string](#)>

where <SQL string> is the SQL string you have entered, such as the following:

Couldn't find input table or query "Select \* From Table1 Where ID = 1"

### RESOLUTION

---

"Microsoft Access Language Reference" versions 1.0 and 1.1 and Microsoft Access Help version 1.0 incorrectly suggest that the domain parameter of a domain [function](#) (for example, the DLookup(), DFirst(), DLast(), DSum(), DMax(), DMin(), and DCount() functions) can be an [SQL expression](#).

The following is the incorrect description of the domain parameter:

String expression identifying the records that constitute the domain. It can be a table name, a query name, or an SQL expression that returns data.

The correct wording is provided in Microsoft Access Help version 1.1. The following is the correct description of the domain parameter:

String expression identifying the records that constitute the domain. It can be a table name or a query name.

### [References](#)

## PRB: Cannot DDEPoke > 255 Characters of Text from Access

Article Number: Q101087  
CREATED: 04-JUL-1993  
MODIFIED: 13-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

When using the DDEPoke statement from an [Access Basic function](#) to send data to another application using [DDE](#), Microsoft Access will not send more than 255 characters of text. Any data beyond the 255th character will be truncated when sent. No error message will be generated.

### RESOLUTION

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0, 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

The DDESend() function, which can only be used from the ControlSource property of a textbox on a [form](#), is capable of sending more than 255 characters using DDE. See "Using the DDESend Function" latter in this article for more information.

Steps to Reproduce Behavior  
-----

1. Start Microsoft Word for Windows which opens with a new, default document, titled "Document1".
2. Choose Save As from the File menu and enter DOC1 as the file name.
3. Choose [Bookmark](#) from the Insert menu, type "test" in the "Bookmark Name" box, and press ENTER.
4. Start Microsoft Access, leaving Word for Windows running in the background.
5. Open/Create a [database](#) and open/create an Access [module](#).
6. Add the following function to the module:

Option Explicit

```
Function DDEPokeTest ()
 Dim chan
 chan = DDEInitiate("winword", "doc1")
```

```
DDEPoke chan, "test", String$(255, "*") & String$(100, "@")
DDETerminate chan
End Function
```

7. Choose Immediate window from the View menu.

8. Type the following and press ENTER:

```
? DDEPokeTest()
```

9. Switch back to Word for Windows and note that only 255 \* characters display, no @ characters display at all.

Using the DDESend Function  
-----

1. Start Microsoft Word for Windows which opens with a new, default document, titled "Document1".
2. Choose Save As from the File menu and enter DOC1 as the file name.
3. Choose Bookmark from the Insert menu, type "test" in the "Bookmark Name" box, and press ENTER.
4. Start Microsoft Access, leaving Word for Windows running in the background, and create a blank, unbound form with a textbox.
5. Type the following expression in the textbox ControlSource property:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
=DDESend("winword","doc1","test",String$(255,"*") &_
String$(100,"@"))
```

6. View the form. If you switch back to Word for Windows you will see 255 \* characters followed by 100 @ characters.

## [References](#)

**PRACC9307: Cannot Include Comma in App Name Using SetupWizard [B\_WAccADK]**  
**Article Number: Q101085**  
**CREATED: 04-JUL-1993**  
**VERSION(S): MODIFIED: 24-AUG-1993**  
ENDUSER |

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

## **SYMPTOMS**

---

When you run a Setup program developed using the Microsoft Access SetupWizard, the Program Manager **group** and icon text is truncated and the custom icon does not display.

## **CAUSE**

---

When prompted for the name of your application, you typed a name that included a comma. The Program Manager group and icon text displays everything up to, but not including, the comma.

## **STATUS**

---

Microsoft has confirmed this to be a problem in the Microsoft Access Distribution Kit version 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

Steps to Reproduce Problem  
-----

1. Run the Microsoft Access SetupWizard.
2. When the "What is the name of your application?" message appears, type the following:

Thomas, Rolm, Buchannan

3. Finish running the SetupWizard.
4. Run your custom Setup program.

Only "Thomas" appears in the new Program Manager group name and icon text.

## [References](#)

## INF: Storing SQL Database Login IDs and Passwords Locally

Article Number: Q101084  
CREATED: 04-JUL-1993  
MODIFIED: 20-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- Microsoft Access Distribution Kit version 1.1

-----  
with

### SUMMARY

---

Storing [SQL database](#) login IDs and passwords locally requires the database administrator create a unique [table](#) on the server.

### MORE INFORMATION

---

In Microsoft Access version 1.0, if you attach an SQL database table, you can choose whether you want Microsoft Access to store your login ID and password locally. If you don't, Microsoft Access prompts for your login ID and password each time you connect to the SQL database containing the table.

If you want Microsoft Access to store the connection information in your Microsoft Access database so you won't have to type it each time, you can select the Save Login ID and Password Locally [check box](#) in the Attach Tables [dialog box](#) when you attach the SQL database table.

This feature is also present in Microsoft Access version 1.1. For Microsoft SQL Server, Sybase SQL Server, and ORACLE Server databases, however, your SQL database administrator can now choose to disable this feature, requiring all users to enter their login IDs and passwords each time they connect to the SQL database.

To disable the ability to store login IDs and passwords locally, your SQL database administrator must create a table called MSysConf in the SQL database. When a user connects to the SQL database, Microsoft Access looks for this table in the database, and if it finds it, queries the table. If the values in the table correctly specify that local storing of login IDs and passwords should be disabled, Microsoft Access does so regardless of whether the Save Login ID and Password Locally check box is selected. If the table isn't present or doesn't specify disabling of the feature, users can store login IDs and passwords locally.

The SQL database table MSysConf should have the following structure.

| Column name | Data type                                                        | Allows <a href="#">Null</a> ? |
|-------------|------------------------------------------------------------------|-------------------------------|
| Config      | A <a href="#">data type</a> that corresponds to a 2-byte integer | No                            |
| chValue     | VARCHAR(255)                                                     | Yes                           |

|         |                                                  |     |
|---------|--------------------------------------------------|-----|
| nValue  | A data type that corresponds to a 4-byte integer | Yes |
| Comment | VARCHAR(255)                                     | Yes |

If the data source you're working with is case-sensitive, use the table and column names exactly as shown. All users must have permission to use the SELECT statement on this table, and only the system administrator may have permission to use the DELETE statement on this table.

For the purpose of disabling password and login ID storage, the table needs to contain only one row, defined as follows.

| Column name | Value  | Explanation                                                                                                                                      |
|-------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Config      | 101    | This is the only valid value for Microsoft Access version 1.1.                                                                                   |
| chValue     | NULL   | This is reserved for future use.                                                                                                                 |
| nValue      | 0 or 1 | Use 0 to prevent the password and login ID from being stored; use 1 to permit password and login ID storage as in version 1.0. The default is 1. |
| Comment     |        | Allow storage of passwords and login IDs in Microsoft Access.                                                                                    |

[References](#)

## **INF: ADK Setup installs to subdirectory of specified directory[B\_WAccADK]**

Article Number: Q101083  
CREATED: 04-JUL-1993  
MODIFIED: 04-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### **Summary:**

---

The Microsoft ADK setup installs into an ADK subdirectory beneath the subdirectory specified in the setup [dialog box](#).

### **More Information:**

---

When installing the Microsoft Access Distribution Kit (ADK), a dialog prompts for the drive and directory in which to install. If you specify C:\ADK, the ADK setup will install it to C:\ADK\ADK.

The default is C:\ACCESS and setup will install to C:\ACCESS\ADK. But, if you specify C:\ACCESS\ADK, the setup will install to C:\ACCESS\ADK\ADK.

### [References](#)

## INF: How to Requery Combo Box on Subform

Article Number: Q101082  
CREATED: 04-JUL-1993  
MODIFIED: 04-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### Summary:

---

This article describes a macro that will requery a combo box control on a subform. This macro is useful for a combo box that is based on a query that references the main form and thus should be requeryed whenever you change records on the main form.

### More Information:

---

Create the following macro:

| Macro Name   | Action      | Defined Below |
|--------------|-------------|---------------|
| RequeryCombo | GoToControl | 1.            |
|              | GotoControl | 2.            |
|              | Requery     | 3.            |

RequeryCombo Actions

- 
1. GotoControl  
ControlName: <Subform ControlName>
  2. GotoControl  
ControlName: <ComboBox ControlName>
  3. Requery  
ControlName: <ComboBox ControlName>

When this macro runs it will requery the combo box. If you want the combo box to be requeryed each time you change records in the main form, place this macro on the OnCurrent property of the main form.

For more information search for "events", then "Events for Forms" and "Events for Controls on Forms" using the Help menu.

### [References](#)

## INF: Two Methods to Refer to a Field in a Previous Record

Article Number: Q101081  
CREATED: 04-JUL-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

Sometimes a [field](#) in a previous [record](#) must be referred to in a later record, such as in a checkbook program or a mileage log. This article explains how to refer to such a record.

NOTE: The procedures provided require a unique, sequential, numeric field, such as a Counter field or a [transaction](#) number.

### MORE INFORMATION

---

This section provides two sets of examples for [database objects](#) (forms, reports, and queries): one using the DLookup() [function](#), the other using the sample database NWIND.MDB.

DLookup() Function Examples  
-----

Use the DLookup() function for the following database objects, as in the following examples:

In a Form  
-----

Type the following in the ControlSource of the [text box](#):

```
=DLookup("[Field]","Table","[ID]=Forms![Form1]![ID]-1")
```

In a Report  
-----

Type the following in the ControlSource of the text box:

```
=DLookup("[Field]","Table","[ID]=Reports![Report1]![ID]-1")
```

In a Query  
-----

Type the following in the Field [row](#) of the [query](#) grid:

```
Expr1: DLookup("[Field1]","Table","[ID]=|[ID]|"&-1)
```

The "-1" indicates the previous record. This number can be changed to reference a different record.

## Northwind Traders Examples

---

The following examples use the sample database NWIND.MDB:

### In a Form

---

1. Create a blank [form](#) based on the Suppliers [table](#).
2. Add the Supplier ID and Company Name fields.
3. Create an unbound text box.

NOTE: In the following sample code, an underscore () is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

Form: Form1

---

RecordSource: Suppliers

### Form1 Controls

---

Text Box:

ControlName: Supplier ID  
ControlSource: Supplier ID

Text Box:

ControlName: Company Name  
ControlSource: Company Name

Text Box:

ControlName: Field2  
ControlSource:  
=DLookup("[Company Name]","Suppliers","[Supplier ID] = \_  
Forms![Form1]![Supplier ID]-1")

### In a Report

---

1. Create a blank [report](#) based on the Suppliers table.
2. Add the Supplier ID and Company Name fields.
3. Create an unbound text box.

Report: Report1

---

RecordSource: Suppliers

### Report1 Controls

---

Text Box:

ControlName: Supplier ID  
ControlSource: Supplier ID

Text Box:

ControlName: Company Name

ControlSource: Company Name  
Text Box:  
ControlName: Field2  
ControlSource:  
=DLookUp("[Company Name]","Suppliers","[Supplier ID] = \_  
Reports![Report1]![Supplier ID]-1")

In a Query  
-----

1. Create a new query.
2. Add the Suppliers table.

Query: Query1  
-----

Type: Select Query  
Field: Supplier ID  
Field: Company Name  
Field: Expr1: DLookUp("[Company Name]","Suppliers","[Supplier ID] = \_  
|[Supplier ID]|"&-1)

[References](#)

## PRB: Out of Memory Error Message when Exporting

Article Number: Q101080  
CREATED: 04-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### SYMPTOMS

---

When exporting a Microsoft Access [table](#), an "Out of Memory" error message may occur.

### CAUSE

---

The table name may contain a non alphanumeric character such as a comma or colon, etc.

### RESOLUTION

---

Rename the Microsoft Access table without the special character in the table name.

### MORE INFORMATION

---

When trying to create a file name to display in the Export to File [dialog box](#), Microsoft Access uses up to the first eight letters of the table name. If this table name contains a special character, Microsoft Access tries to create a file name with the special character and causes an error. This error occurs when exporting to any data destination that creates a DOS file name for the table.

Note that changing the Data Destination in step #3 below to something other than Text (Delimited) may not result in the Out of Memory error message.

Steps to Reproduce Behavior

-----

1. Copy the Employees table in the Northwind sample [database](#) (NWIND.MDB) to Empl,Nwind (note the comma in the table name). The steps to copy a table are as follows:
  - a. Highlight the Employees table in the [Database window](#).
  - b. Select the Copy command from the Edit menu.
  - c. Select Past from the Edit menu. Type the new table name in the Paste Table As dialog box, and choose OK.
2. Select Export from the File menu.
3. In the Export dialog box, select Text (Delimited) as your Data

Destination.

4. In the Select Microsoft Access Object dialog box, select the Empl,Nwind and press OK.
5. The following error message appears:

Microsoft Access  
Out of Memory.

## [References](#)

## INF: How to Create an Undo Record Button on a Form

Article Number: Q101079  
CREATED: 04-JUL-1993  
MODIFIED: 29-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

This article describes how to create a button on a [form](#) to undo changes made to the [current record](#).

### MORE INFORMATION

---

To undo the changes to the current [record](#), you can use the DoMenuItem [macro](#) action to choose Undo Current Record from the Edit menu. However, this menu item is available only when there are changes to undo. Attempting to use the DoMenuItem action when no changes have been made to the current record causes the following error message to appear:

```
"Command not available: UndoCurrentRecord."
```

The following is a [module](#) using the [Access Basic function](#) UndoRecord(), which allows you to undo the current record, whether or not the record has been changed ("dirtied"):

```
Option Explicit
```

```
Function UndoRecord ()
 On Error GoTo ErrUndoRecord
 DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDOFIELD
 DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDO
```

```
 ByeUndoRecord:
 Exit Function
```

```
ErrUndoRecord:
 Resume ByeUndoRecord
End Function
```

To use this function on a form, create a push button and set its OnPush property to =UndoRecord().

### [References](#)

## PRB: Background Bitmap in Setup Says 'Setup' [B\_WAccADK]

Article Number: Q101078  
CREATED: 04-JUL-1993  
MODIFIED: 04-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

The Setup [bitmap](#) that appears in the background of the running setup program created with the Setup Wizard cannot be altered.

### RESOLUTION

---

This is by design. This feature is under review and will be considered for inclusion in a future release.

### MORE INFORMATION

---

The MSSetup kit that ships with the Microsoft Windows Software Development Kit, SDK, allows you to build custom setup programs with complete, programmatic [control](#) over every element of the setup environment, including background bitmaps, buttons, and dialogs. This may be used as an alternative to the setup solution provided by the Microsoft Access Distribution Kit. However, these tools CANNOT be used to alter existing setup information developed using the Microsoft Access Distribution Kit.

### [References](#)

## **PRACC9306: Windows Setup Ignores Applications Created in ADK**

Article Number: Q101077  
CREATED: 04-JUL-1993  
MODIFIED: 16-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

Microsoft Windows Setup (found in the Main [group](#) in Program Manager) ignores applications created with the Microsoft Access Distribution Kit (ADK), even when these directories are listed in the system path.

### **RESOLUTION**

---

To cause Windows Setup to recognize your ADK application, use the following steps:

1. Run Windows Setup.
2. From the Options menu, choose Set Up Applications.
3. Select the [check box](#) titled "Ask you to specify an application."
4. Choose OK, then specify your ADK application.

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### **MORE INFORMATION**

---

Steps to Reproduce Problem  
-----

1. Place all files required by your Microsoft Access ADK application in a directory in your system path.
2. Run Windows Setup.
3. From the Options menu, choose Set Up Applications.
4. Search your path.

Result: Windows Setup does not recognize your Microsoft Access ADK application.

### [References](#)

## INF: Expressions to Count Yes/No and Other Responses

Article Number: Q101076  
CREATED: 04-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1  
-----

### SUMMARY

---

This article provides expressions that you can use to count the actual number of Yes's, No's, and/or Nulls in a [field](#) whose DataType is Yes/No.

### MORE INFORMATION

---

The following expressions can be used in a [report](#)'s Group Footers or in the Report Footer to count the actual number of Yes's, No's and/or Nulls in a field whose DataType is Yes/No and whose field name is YesNoField:

| Expression                     | Sums What |
|--------------------------------|-----------|
| =Sum(IIF([YesNoField],1,0))    | Yes's     |
| =Sum(IIF([YesNoField],0,1))    | No's      |
| =Sum(IIF(Not[YesNoField],1,0)) | No's      |

You can also create a related [expression](#) to count a specific value in a field. For example, the following expression counts all the values "3" in a field called MyField:

```
=Sum(IIF([MyField]=3,1,0))
```

The following expression counts all the nulls in a field called MyField:

```
=Sum(IIF(IsNull[MyField],1,0))
```

#### Additional Steps to Demonstrate Behavior

---

1. Open the sample [database](#) Northwind Traders (NWIND.MDB).
2. Use the Report Wizard to create a Groups/Totals report based on the Products [table](#).
3. Select the Category ID as the only field.
4. Group on the Category ID.
5. In the report's [Design view](#), add two unbound text boxes with each one's ControlSource set to one of the following expressions:

```
=Sum(IIF([Discontinued],1,0))
=Sum(IIF([Discontinued],0,1))
```

6. From the File menu select Print Preview. The first expression will count the number of Products within each category whose Discontinued field is set to Yes. The second expression will count the number of Products within each category whose Discontinued field is set to No.

## [References](#)

## INF: How to Print Odd or Even Pages of a Report

Article Number: Q101075  
CREATED: 04-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### SUMMARY

---

User's commonly want to print the odd then the even pages of a report for duplex printing or to print reports with different margins for binder storage.

### MORE INFORMATION

---

The best way to print the odd then even pages of a report is to setup a form to prompt for which pages to print (odd or even) then from the form call a macro to print the report. (Page 559 of the User's Guide shows a sample form that has a user fill in values on the form then prints a report.) Our form in this example will have a text box control to hold the value "odd" or "even" plus a command button to print the report.

NOTE: These steps will produce blank pages when the report is printed for the pages that are not printed

1. Create a new form "Print Odd Even Pages"
  - a) locate an unbound text box with the ControlName = "PrintPages"
  - b) locate a command button with the Caption = "Print Report" and from the OnPush property call the following "OddEven.Print" macro group.
2. Create a new macro "OddEven" with the following macro groups.

Note: Each condition should be entered on one line. The line continuation character '\_' improves the readability of the code. Remove the continuation character when placing the line into the macro.

| Macro Group Name | Condition                                                                    | Action      |
|------------------|------------------------------------------------------------------------------|-------------|
| Print            |                                                                              | OpenReport  |
| TestOddEven      | Forms![Print Odd Even Pages] _<br>![PrintPages]="Even" And -<br>Page Mod 2=1 | CancelEvent |
|                  | Forms![Print Odd Even Pages] _<br>![PrintPages]="Odd" And -<br>Page Mod 2=0  | CancelEvent |

OpenReport Actions  
-----

Report Name: Customer Mailing Labels (Report from NWIND [database](#))  
View: Print Preview

3. From each section of the report that is printing data you will need to call the [macro group](#) OddEven.TestOddEven from the OnPrint events.

For example in the NWIND sample database report "Customer Mailing Labels" the only section that is printing is the Detail section, the Page Header and Page Footer in this report do not print any values. For the Detail Section set the OnPrint [event](#) to: OddEven.TestOddEven

4. With the form open enter a value into the text box of "Odd" or "Even" then when you push the "Print Report" command button, the "Customer Mailing Labels" report will open, (press OK for all Countries) then the report will run it's OnPrint events which call the macro group "OddEven.TestOddEven" to print only the Odd or Even pages.

References:

Page 559 of the User's Guide for 1.0 manuals

[References](#)

## INF: Using SendKeys to Change Option Menus

Article Number: Q101074  
CREATED: 04-JUL-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

### SUMMARY

---

In Microsoft Access 1.0, a SendKeys action will not execute properly in a macro following a DoMenuItem or other macro action which opens a Print Dialog Box. For example, when a DoMenuItem action is called to the File Print menu to change a printing parameter in the Print Dialog Box before printing, the keystrokes in a subsequent SendKeys Action remain in the buffer and do not execute.

### MORE INFORMATION

---

Because a dialog box suspends a macro, the SendKeys must be sent before the dialog box opens. The following example will allow you to open the print dialog box, select page range, 1-5 and print the current object.

| Action     | Comment                               |
|------------|---------------------------------------|
| -----      |                                       |
| SendKeys   | Store keystrokes for print dialog box |
| DoMenuItem | open print dialog box                 |

Set the following properties for the actions entered above:

SendKeys Action  
-----  
Keystrokes: %p1%t5{ENTER}  
Wait: No

DoMenuItem Action  
-----  
Menu Bar: Form  
Menu Name: File  
Menu Command: Print

References: "Microsoft Access Language Reference Manual",  
pages 432-433

Microsoft Access 1.0 Help Menu "SendKey Actions"

[References](#)

## INF: Connecting to Sybase SQL Server from an ODBC application

Article Number: Q101073  
CREATED: 04-JUL-1993  
MODIFIED: 25-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
  - Microsoft Visual Basic version 3.0 for Windows
  - Microsoft FoxPro version 2.5 for Windows
- 

### SUMMARY

---

You can connect to a Sybase [SQL](#) Server machine from an [ODBC](#) application (Microsoft Access, Microsoft Visual Basic, Microsoft FoxPro for Windows, and so forth) using the ODBC SQL Server driver. This article describes how to set up the ODBC data sources.

If you are already connected to Sybase SQL server from Microsoft Windows DB-Library applications, no other components are necessary.

### MORE INFORMATION

---

The key to both Microsoft SQL Server and Sybase SQL Server connectivity is a [module](#) called the Net-Library. This module consists of two interfaces: one with the [network](#) and one with the application. The network interface is customized to support a particular network; however, the application interface remains the same.

Since ODBC, DB-Library, APT-SQL, APT-Library, and so forth communicate with the Net-Library, these utilities can be written independent of the network. The ODBC SQL Server driver, in particular, can be used to connect either to a Microsoft SQL Server machine or to a Sybase SQL Server machine (on a UNIX or VMS system), long as there is a Net-Library for whatever network you are using.

The following paragraphs discuss how to connect to a Sybase SQL Server from an ODBC application. The remainder of the article assumes that you are connected to a SQL Server called "Mysqlsvr". This server should be set up as an ODBC data source, using the ODBC Control Panel utility, as follows:

1. Open ODBC Control Panel and choose Add.
2. Select SQL Server from the list of installed drivers and choose OK.
3. In the ODBC SQL Server Setup [dialog box](#), type a data source name and a description (optional).

Note that a single server can [function](#) as a multiple ODBC data sources, since each [database](#) in the server can be a data source. The data source name need not necessarily be the same as the server name.

4. Enter the true name of the server, then enter the network address.

Since a Sybase SQL Server on UNIX uses sockets and not named pipes, the network address must contain the IP address and the port identification number. This string is in the form "ip\_address,port" (for example, "11.1.14.40,3180").

For a Sybase SQL Server on a VAX system, this string should be in the form "<node address>,<process\_id>", where <node address> is the DECnet node address of the server and <process\_id> is the process identification number to use for the connection (for example, "1.997,141").

The Network Library field should contain the name of the Net-Library you need to use. This is dependent on what network you are using. For example, if you are using the FTP Software product PCTCP, the Net-Library name is "wdbftptc".

5. Choose Options to specify a database name and/or a language name.
6. Choose OK to go back to the first screen.
7. Run the INSTCAT.SQL script file.

NOTE: This is a file containing SQL statements that will create certain stored procedures needed to process ODBC calls. Without these procedures, Microsoft Access cannot attach to a SQL Server table and Visual Basic is not able to function correctly.

At the UNIX command line or the MS-DOS prompt, type the following:

```
isql -S<servername> -Usa -P<sa-password> -i<path>\instcat.sql
```

Finally, note that if the above procedure is followed correctly, the following changes appear in the ODBC.INI and WIN.INI files. The [Data Sources] section of ODBC.INI will contain the following entry:

```
<data-source-name>=SQL Server
```

where <data-source-name> is the name of the data source that you added. There will be a new section called "[data-source-name]" containing the location of the SQL Server driver and a description of the data source. The [SQLSERVER] section of the WIN.INI will contain the following entry:

```
data-source-name=<netlib-name>,<network address>
```

The following products are manufactured by vendors independent of Microsoft: DECnet, VMS, and MicroVAX by Digital Equipment Corporation; FTP Software by FTP Software, Inc.; DB-Library, Net-Library, and Sybase SQL Server by Sybase, Inc.; and UNIX by UNIX System Laboratories. We make no warranty, implied or otherwise, regarding these products' performance or reliability.

## [References](#)

## PRB: Buttons Pasted into Option Groups have Option Value

Article Number: Q101072  
CREATED: 04-JUL-1993  
MODIFIED: 04-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### SYMPTOMS

---

Stand-alone option buttons pasted into option groups have an option value of -1.

### CAUSE

---

Buttons pasted into option groups have whatever option value they had before they were copied to the [Clipboard](#). If they happened to have been created as stand-alone buttons, it's -1. Since stand-alone option buttons don't have the Option value property, this comes from it's ControlSource property, which defaults to -1 for True.

### RESOLUTION

---

Cutting and pasting does not change a button's Option Value. If you cut and paste a button into an Option Group, you must manually change its Option Value to a new value.

### MORE INFORMATION

---

The only time an arbitrary value is assigned to an [option button](#) is when a new button is created within a [group](#).

Steps to Reproduce Behavior

-----

1. Create an [option group](#) and an option button outside the group.
2. Copy the button into the clipboard.
3. Select the option group.
4. Edit Paste.

Results: The button has it's Option value set to -1.

For more information search for "Option Group", then "OptionValue" using the Help menu.

### [References](#)

## INF: Using SQL Server ODBC Driver with Sybase Servers

Article Number: Q101071  
CREATED: 04-JUL-1993  
MODIFIED: 19-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1  
-----

### SUMMARY

---

The [SQL](#) Server [ODBC](#) driver was designed to work with both Microsoft SQL Server and Sybase SQL Server machines. The ODBC organization at Microsoft has completed some limited interoperability testing with Sybase version 4.2 on the SPARC platform, using the Microsoft TCP/IP [network](#) protocol and the Sybase Net-Library for this stack, and have not uncovered any significant problems. Below two issues to consider if you use the SQL Server ODBC driver with Sybase servers.

### MORE INFORMATION

---

#### Issue 1

-----

The ODBC driver requires the catalog procedures that shipped with Sybase SQL Server 4.2a. Until the Sybase 4.9.1 release, these catalog procedures did not ship with Sybase servers. For the ODBC driver to work with Sybase servers, the INSTCAT.SQL script must be run against the Sybase server to install the catalog procedures.

Sybase changed the system catalogs in its version 4.8 release, requiring a modified script. This script is called INSTCAT.48 and is available through Microsoft Product Support Services and on CompuServe. This script is required with Microsoft Access version 1.0 and Microsoft Visual Basic version 2.0 applications.

With Microsoft Access 1.1 and Visual Basic 3.0, the INSTCAT.SQL file applies to all versions of Microsoft SQL Server, as well as to all versions of Sybase SQL Server.

#### Issue 2

-----

The SQL Server ODBC driver that shipped with Microsoft Access 1.0 and Visual Basic 2.0 came with the Sybase Net-Library. To connect to Sybase, you needed to use Microsoft SQL Bridge or install Net-Library on each client machine.

Both Microsoft Access and Visual Basic 2.0 use the network in a way that is not typical of Windows-based applications (for instance, these applications use multiple, simultaneous connections and asynchronous calls). It is probable that there are network configurations in which Net-Library will fail with these products.

Microsoft and Sybase are now discussing how best to perform interoperability testing with the various network configurations supported by Sybase (FTP Software, Wollongong, Novell, TCP/IP, DECnet, and so forth).

The following products are manufactured by vendors independent of Microsoft: DECnet is manufactured by Digital Equipment Corporation; FTP Software is manufactured by FTP Software, Inc.; Novell networking products are manufactured by Novell, Inc.; SPARC is manufactured by Sun Microsystems, Incorporated; Net-Library and Sybase SQL Server are manufactured by Sybase, Inc. We make no warranty, implied or otherwise, regarding these products' performance or reliability.

## [References](#)

## **INF: Custom Application Does Not Appear In Task List [B\_WAccADK]**

Article Number: Q101070  
CREATED: 04-JUL-1993  
MODIFIED: 04-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### **Summary:**

---

When the task list appears the Application running from the Access Distribution Kit does not appear. However it will appear by pressing Alt-Tab.

### **More Information:**

---

You need to provide a Title Bar in the application's INI file under the [Run-Time Options] Heading such as:

```
[Run-Time Options]
TitleBar=My Applications Name
```

After adding this option to the INI file you will now see the application listed in the Task List.

References:

[References](#)

## **PRB: Custom Setup Creates too many Program Groups [B\_WAccADK]**

**Article Number:** Q101069  
**CREATED:** 04-JUL-1993  
**MODIFIED:** 03-SEP-1993  
**VERSION(S):** 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

Custom setup will create a program group for each icon installed and an additional empty Program Group. For example if your custom setup needs to create one program group with two icons (your application and the Change) Setup will create three Program Groups and only two will have an Icon.

### **CAUSE**

---

This will occur if the application is being installed to a machine running Windows 3.0 and the application's name specified in the last screen of the Custom Wizard is longer than 24 characters.

### **RESOLUTION**

---

Reduce the length of the program group name.

Steps to Reproduce Behavior

-----  
Run the Setup Wizard. The last screen will prompt for the name of the application. Specify a name longer than 24 characters. Create the program diskettes and install the application to a machine running Windows 3.0.

### **[References](#)**

## **PRB: Input to Field not Capitalized [B\_WAccADK]**

Article Number: Q101068  
CREATED: 04-JUL-1993  
MODIFIED: 04-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

When specifying the Access Directory in the SUFiles setup dialog. The input is not capitalized.

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access Distribution Kit version 1.1. Microsoft is researching this problem and will post new information here as it becomes available.

### **RESOLUTION**

---

This will not effect the functionality of the SUFiles utility.

Steps to Reproduce Behavior

-----  
Start the SUFiles Utility. Enter information in the Access Directory field. Note the letters are not uppercased.

References:

[References](#)

## PRB: Out of Stack Memory Error Message with SendKeys Macro

Article Number: Q101067  
CREATED: 04-JUL-1993  
MODIFIED: 24-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

You receive the following error message when choosing a command button

Not enough stack memory left

This problem occurs when you activate a [macro](#) from the OnPush property of a [command button](#) on a [form](#). The macro includes a SendKeys action, in which the keystrokes sent include the ESC key and the Wait argument is set to Yes.

### CAUSE

---

The Cancel property of the command button is set to Yes.

### RESOLUTION

---

Change the Cancel property of the command button to No.

### MORE INFORMATION

---

You receive the above error message if you choose a command button and all the following conditions are met:

- The macro attached to the command button includes a SendKeys action.
- The command button's Cancel property is set to Yes.
- You choose the command button by pressing ESC.

If the OnPush property activates a macro that performs a SendKeys action, which in turn sends an ESC keystroke, the button that repeatedly activates the macro is chosen and an endless loop results. Note that if the Wait argument on the SendKeys action is set to No, Microsoft Access will appear to hang.

Steps to Reproduce Behavior

- 
1. Start Microsoft Access and open the sample [database](#) NWIND.MDB.
  2. Create a new macro called Test Macro:

Test Macro Actions

```

SendKeys
 Keystrokes: {esc}
 Wait: Yes
```

3. Open the Customers form in [Design view](#).

4. Add a command button to the form:

```
Object: Command Button

ControlName: Test Button
Caption: Test Button
OnPush: Test Macro
Cancel: Yes
```

5. Switch to [Form view](#).

6. Choose the Test button. The following error message appears:

```
Not enough stack memory left
```

#### REFERENCES

```
=====
```

"Language Reference," versions 1.0 and 1.1, page 59

[References](#)

## **PRACC9306: Run-Time Icon Flashes Application Workspace Color**

Article Number: Q101066  
CREATED: 04-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

A flashing Microsoft Access run-time icon is displayed using the application workspace color, rather than the desktop color.

### **CAUSE**

---

The run-time icon is not registered as a normal icon; instead, it is registered as a NULL icon. Microsoft Windows handles painting the icon background of a NULL icon and a normal icon differently.

### **RESOLUTION**

---

To work around this problem, either do not use minimized icons or specify the same color for your desktop and application workspaces. To do this, choose Color Settings from Windows Control Panel.

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access Distribution Kit version 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### **MORE INFORMATION**

---

Steps to Reproduce Problem

-----

1. Create a new icon in Program Manager by selecting New from the File menu. Choose Program Item and then choose OK.
2. In the Item Properties box, enter the following:

Description: Minimized Repair of Nwind  
Command Line: <C:\ACCESS\MSARN110.EXE NWIND.MDB> /repair  
Working Directory: <C:\ACCESS>  
Run Minimized: Yes (checked)

where <C:\ACCESS\MSARN110.EXE NWIND.MDB> and <C:\ACCESS> are the directory names and filenames on your system.

3. Choose OK to create the icon.
4. Double-click the icon.

Once NWIND.MDB has been repaired, the icon begins flashing and the application workspace color is displayed instead of the desktop color.

## [References](#)

## INF: Filenames/Version Numbers, ODBC Files for MS Access V1.1

Article Number: Q100975  
CREATED: 01-JUL-1993  
MODIFIED: 19-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SUMMARY

---

This article provides the filenames, version numbers, and other data about the [ODBC](#) files that accompany Microsoft Access version 1.1.

### MORE INFORMATION

---

The following ODBC files shipped with Microsoft Access 1.1:

| Filename                     | Version   | Size   | Date    |
|------------------------------|-----------|--------|---------|
| -----                        | -----     | -----  | -----   |
| SQLSRVR.DLL                  | 1.01.2920 | 151856 | 5-20-93 |
| SQORA.DLL                    | 1.00.2816 | 143600 | 4-16-93 |
| ODBC.DLL                     | 1.00.2709 | 44736  | 4-08-93 |
| INSTCAT. <a href="#">SQL</a> | **        | 93834  | 5-20-93 |

\*\* The version number that displays when you execute `sp_server_info 500` is 01.01.2807, though technically there is no such version of this file.

Another file that is important to Microsoft SQL Server is `DBNMP3.DLL`, which should be 9941 bytes.

### [References](#)

## INF: Using Ampersands in Shorcut Keys, Displaying in Labels

Article Number: Q100974  
CREATED: 01-JUL-1993  
MODIFIED: 23-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

The ampersand (&) is a reserved character. You can assign a shortcut key by typing an ampersand in the Caption property of a control, immediately preceding the character that you want to assign to the shortcut key. The character you have assigned will then be underlined. Press ALT+underlined character to move the focus to that control.

### MORE INFORMATION

---

To display the ampersand character in the caption, type "A&&B" (without the quotation marks). The double ampersand is displayed as "A&B". The following are examples of how "A&B" appears on a label:

| Text Typed | Text Displayed |
|------------|----------------|
| "A&B"      | AB             |
|            | -              |
| "A&&B"     | AB             |
|            | -              |
| "A&&&B"    | A&B            |
|            | -              |

### References

## INF: How to Perform a Screen Dump of Your Forms

Article Number: Q100973  
CREATED: 01-JUL-1993  
MODIFIED: 01-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### Summary:

---

Through the use of Windows API functions, you can take screen dumps of the graphic user interface of Windows. This includes screen dumps of Microsoft Access, objects of Microsoft Access, other Windows applications, or the entire screen or desktop of Windows.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

### More Information:

---

The following code will perform a screen dump to the clipboard of the currently running session of Microsoft Access. Changing the handle, referred by AccessHwnd, you can take screen dumps of other applications, including specific objects within Microsoft Access.

To see the output of the screen dump, open PaintBrush(PBRUSH.EXE) that ships with Windows, and perform the Paste command from the Edit menu. You should then see the screen shot of Microsoft Access.

To run this code, you will need to call ScreenDump() from some [event](#) , from a [macro](#), or from the [Immediate window](#). From the Immediate window, you would type:

```
?ScreenDump()
```

followed by a carriage return.

To set up this [function](#), open a new or existing [module](#) and insert the code documented below.

NOTE: You may have some Windows API functions defined in an existing Microsoft Access library, therefore your declarations may be duplicates. If you receive the duplicate procedure name error when you compile or run your code, remove or comment out the appropriate declare statement from your code.

Note: In the following sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

Option Compare Database

```
Option Explicit
```

```
Type RECT_Type
 left As Integer
 top As Integer
 right As Integer
 bottom As Integer
End Type
```

```
Declare Function GetActiveWindow% Lib "User" ()
Declare Function GetDesktopWindow% Lib "User" ()
Declare Sub GetWindowRect Lib "User" (ByVal Hwnd%, _
 lpRect As RECT_Type)
Declare Function GetDC% Lib "User" (ByVal Hwnd%)
Declare Function CreateCompatibleDC% Lib "GDI" (ByVal hdc%)
Declare Function CreateCompatibleBitmap% Lib "GDI" (ByVal hdc%, _
 ByVal nWidth%, ByVal nHeight%)
Declare Function SelectObject% Lib "GDI" (ByVal hdc%, ByVal hObject%)
Declare Function BitBlt% Lib "GDI" (ByVal hDestDC%, ByVal X%, _
 ByVal Y%, ByVal nWidth%, _
 ByVal nHeight%, ByVal hSrcDC%, _
 ByVal XSrc%, ByVal YSrc%, _
 ByVal dwRop&)
Declare Function OpenClipboard% Lib "User" (ByVal Hwnd%)
Declare Function EmptyClipboard% Lib "User" ()
Declare Function SetClipboardData% Lib "User" (ByVal wFormat%, _
 ByVal hMem%)
Declare Function CloseClipboard% Lib "User" ()
Declare Function ReleaseDC% Lib "User" (ByVal Hwnd%, ByVal hdc%)
Declare Function DeleteDC% Lib "GDI" (ByVal hdc%)
```

```
Global Const SRCCOPY = &HCC0020
Global Const CF_BITMAP = 2
```

```
Function ScreenDump ()
 Dim AccessHwnd%, DeskHwnd%
 Dim hdc%
 Dim hdcMem%
 Dim rect As RECT_Type
 Dim junk%
 Dim fwidth%, fheight%
 Dim hBitmap%

 DoCmd Hourglass True

 '-----
 ' Get window handle to Windows and Microsoft Access
 '-----
 DeskHwnd = GetDesktopWindow()
 AccessHwnd = GetActiveWindow()

 '-----
 ' Get screen coordinates of Microsoft Access
 '-----
 Call GetWindowRect(AccessHwnd, rect)
 fwidth = rect.right - rect.left
 fheight = rect.bottom - rect.top
```

```

'-----
' Get the device context of Desktop and allocate memory
'-----
hdc = GetDC(DeskHwnd)
hdcMem = CreateCompatibleDC(hdc)
hBitmap = CreateCompatibleBitmap(hdc, fwidth, fheight)

If hBitmap <> 0 Then
 junk = SelectObject(hdcMem, hBitmap)

 '-----
 ' Copy the Desktop bitmap to memory location
 ' based on Access coordinates.
 '-----
 junk = BitBlt(hdcMem, 0, 0, fwidth, fheight, hdc, rect.left, _
 rect.top, SRCCOPY)

 '-----
 ' Setup the clipboard and copy bitmap
 '-----
 junk = OpenClipboard(DeskHwnd)
 junk = EmptyClipboard()
 junk = SetClipboardData(CF_BITMAP, hBitmap)
 junk = CloseClipboard()
End If

'-----
' Cleanup handles
'-----
junk = DeleteDC(hdcMem)
junk = ReleaseDC(DeskHwnd, hdc)

DoCmd Hourglass False

```

End Function

References:

[References](#)

## INF: Force MS Access to Use "Snapshot" Mode for Attached Data

Article Number: Q100972  
CREATED: 01-JUL-1993  
MODIFIED: 10-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Microsoft Access addresses attached data differently than it does its own native data. This article discusses how Microsoft Access retrieves attached [ODBC](#) data.

### MORE INFORMATION

---

If `SQLStatistics()`, an ODBC API [function](#), reports a unique [index](#) on the [table](#), Microsoft Access goes to the [dynaset](#) to select the values from the unique key fields that adhere to the user's [WHERE clause](#) restrictions. It then selects only the data needed for display, using the unique key values (10 at a time).

If the table does not have a unique key, Microsoft Access pulls down all the data at once in [read-only](#) ("snapshot") mode. In most cases, this process will be faster once the data is pulled down, though the data cannot be updated.

If you want to enforce read-only mode all the time, add the following line to the [ODBC] section of your MSACCESS.INI file:

```
SnapshotOnly=1
```

This restriction will apply only to tables attached after you make the change and will simply short-circuit the call to the `SQLStatistics()` function during the attachment phase.

In general, a table in dynaset mode typically takes 4-5 seconds to open, as opposed to 1-2 seconds for a table in read-only mode.

### [References](#)

## PRB: ADK Custom Setup Hangs During Installation [B\_WAccADK]

Article Number: Q100971  
CREATED: 01-JUL-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### SYMPTOMS

---

If you create Setup disks using the Microsoft Access Distribution Kit (ADK) Custom Setup Wizard, the installation process hangs immediately after SETUP.EXE creates the program group.

### CAUSE

---

This problem is caused by not specifying an icon file for the application in the .ICO File (Optional) field and selecting the Change Workgroups Program Manager Item check box when the message "Would you like to install any of these optional features?" appears in the Setup Wizard.

### RESOLUTION

---

Run the Setup Wizard again and specify an icon file for the .ICO File (Optional) field, or follow the steps outlined below to modify the STFSETUP.IN\_ file.

### MORE INFORMATION

---

When the Setup Wizard creates the STFSETUP.IN\_ file, it requires that that you have an icon file if you choose to install Change Workgroups. The [Custom] section of the file contains the following line:

```
InstallChangeWorkGroup="YES"
```

If you have set this option to Yes, you must also add a line in the [Program Manager] section of the STFSETUP.IN\_ file that reads:

```
$(AppPath) "NWIND.ICO"
```

You also need to substitute NWIND.ICO with an icon name that you specify.

Updating Your STFSETUP.IN\_ File  
-----

1. Open the STFSETUP.IN\_ file in Notepad. This file is usually found in the DISK1 directory of the custom application.
2. Change the following entry

```
[Program Manager]
```

```
CreateProgManGroup $(ProgGroupDesc), "NWIND"
CreateProgManItem $(ProgGroupDesc), "NWIND",
$(AppPath)"MSARN110.EXE_ NWIND.MDB /Ini NWIND.INI"
EXIT
```

to read

```
[Program Manager]
CreateProgManGroup $(ProgGroupDesc), "NWIND"
CreateProgManItem $(ProgGroupDesc), "NWIND",
$(AppPath)"MSARN110.EXE_ NWIND.MDB /Ini NWIND.INI",
$(AppPath)"NWIND.ICO"
EXIT
```

#### Steps to Reproduce Behavior

---

1. Run the Microsoft Access Custom Setup Wizard. Do not specify an icon.
2. When you are prompted for ISAMs, select ChangeWorkGroup.
3. After the Setup Wizard creates your disks, run SETUP.EXE. The process will hang on the last disk, after the icons are added to Windows Program Manager.

#### [References](#)

## INF: How to Give Users the Ability to Change Passwords

Article Number: Q100970  
CREATED: 01-JUL-1993  
MODIFIED: 01-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 1.1  
-----

### Summary:

---

This article describes how to distribute a secure application with the Access Development Kit and still be able to give the end user the ability to change their passwords.

### More Information:

---

You must follow these two steps in order to achieve this functionality.

1. You must make the database container active to get to the Security menu. The best method is to select an object in the Database window through a macro.
2. Once the Password dialog is displayed, change the password.
2. Once the user has closed the Password dialog, set the focus back to the form you called the macro or function from.

Here is a sample macro.

```
Macro:Change Password
=====

Actions

SelectObject
DoMenuItem
SelectObject

Action arguments

SelectObject
 Object Type: Macro
 Object Name: Autoexec
 In Database window: Yes
DoMenuItem
 Menu Bar: Database
 Menu Name: Security
 Command: Change Password
 Subcommand:
SelectObject
 Object Type: Form
 Object Name: Form1
```

In Database window: No

The database container will appear as well as the dialog to change the password. You can not select anything in the Database window due to the fact the Change Password Dialog is modal.

In The Access Development Kit the functionality is identical except the Database window is NOT displayed.

References:

[References](#)

## PRB: Error Message If Path Info for Attached Databases Changes

Article Number: Q100969  
CREATED: 01-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

The following error message is displayed:

Couldn't find table <table name>

### CAUSE

---

The Access Distribution Kit (ADK) recommends splitting up databases. You can place your forms, reports, and macros in an objects file, such as PROGRAM.MDB, and place your tables in a data file called DATA.MDB.

When you create a database that has attached tables, the specific path to these attached tables is stored in the database. If the path to these attached tables has changed, you receive the above error message.

Splitting the database simplifies updating, since you can redistribute the objects file and copy over the PROGRAM.MDB file without destroying your data.

### RESOLUTION

---

The ADK requires that your database have an AUTOEXEC macro. From within the AUTOEXEC macro, call an Access Basic function to verify that the attached tables are actually attached. If they are not attached, you must attach them.

When you start your application, make sure that the program's working directory property is the same directory as the .MDB file. The Microsoft Access Setup Wizard automatically sets this property automatically in Program Manager. Use the CurDir() function to determine from what directory the program is being executed, then attach the files to that directory.

### MORE INFORMATION

---

1. Create a new directory (for example, C:\ATTACH).
2. Copy the sample database NWIND.MDB from your ACCESS directory to the C:\ATTACH directory.
3. Start ACCESS.EXE. From the File menu, choose New. Create a new database called PROGRAM.MDB in the C:\ATTACH directory.
4. From the Database window, choose Module and then choose New. Enter

the following function:

```
Function Reattach ()

 On Error GoTo attach_files
 DoCmd SelectObject a_table, "Employees", True
 Exit Function
```

attach\_files:

'NOTE: In the following sample code, an underscore (\_) is  
'used as a line continuation character. Remove the underscore  
'when re-creating this code in Access Basic.

```
 DoCmd TransferDatabase A_ATTACH, "Microsoft Access", _
 CurDir &"\nwind.mdb", A_TABLE, "Employees", _
 "Employees"

 Exit Function

End Function
```

5. Save the module as Module1.
6. Create the following new macro and save it as AUTOEXEC:

| Macro Name | Action  | Argument   |
|------------|---------|------------|
| Autoexec   | RunCode | Reattach() |

7. Close PROGRAM.MDB, then reopen it. Note that PROGRAM.MDB attaches the Employees table from NWIND.MDB.

## [References](#)

## **PRACC9307: Application Admin Setup (/A) to Empty Floppy Drive [B\_WAccADK]**

Article Number: Q100968  
CREATED: 01-JUL-1993  
MODIFIED: 24-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### **SYMPTOMS**

---

When a Microsoft Access Distribution Kit (ADK) Application Admin Setup (/A) is directed to a empty disk drive, you receive the following error message:

System Error--Cannot read from drive A.

Choosing Cancel works only for one file at a time. The error message is repeated for all files installed from Disk 1.

If you continue to choose Cancel, a Setup [dialog box](#) is displayed that shows additional space is needed on the drive. The bottom half of the last line of text in the dialog box is cut off.

If you choose Continue, several more system error messages are displayed. Then the following dialog box appears

An error has occurred during Setup. Please make sure you are not out of disk space.

and the Setup routine terminates.

### **RESOLUTION**

---

Do not install to an empty disk drive. Either install to a floppy disk drive in which you have inserted a disk or install to another hard disk drive.

### **STATUS**

---

Microsoft has confirmed this to be a problem in the Microsoft Access Distribution Kit version 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### **MORE INFORMATION**

---

Steps to Reproduce Problem  
-----

1. Insert Disk 1 of the ADK Application into your floppy disk drive.
2. Type <B>:\SETUP.EXE /A, where <B> is your floppy disk drive.

A dialog box requesting drive and directory names for program files, graphs, and databases is displayed.

3. Type <A>:\ (no subdirectory name), where <A> is the floppy disk drive that you did not type in Step 2.
4. Press ENTER. You will begin receiving system error messages.

#### [References](#)

## **PRB: "The file WFWNET.DR\_ could not be found."**

**Article Number:** Q100967  
**CREATED:** 01-JUL-1993  
**MODIFIED:** 01-JUL-1993  
**VERSION(S):** 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0
- 

### **SYMPTOMS**

---

The error message

The file WFWNET.DR\_ could not be found

appears when attempting to install the updated Microsoft Windows for Workgroups [network](#) driver.

### **CAUSE**

---

The WFWDRV setup program was executed from a drive different from the drive the disk is in.

### **RESOLUTION**

---

Change to the floppy drive containing the disk with the updated driver. This has been fixed in Microsoft Access version 1.1.

### **MORE INFORMATION**

---

Steps to Reproduce Behavior

- 
1. From the [command prompt](#), type A: or B: then Enter.
  2. Type WFWDRV and press Enter.

References:

[References](#)

## **INF: SUFILES Utility Appears to Hang [B\_WAccADK]**

**Article Number:** Q100965  
**CREATED:** 01-JUL-1993  
**MODIFIED:** 01-JUL-1993  
**VERSION(S):** 1.10

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
- 

### **Summary:**

---

After installation of the ADK you will want to run the SUFiles Utility(Setup Files Utility). This procedure compresses files from the various directories including the Windows and Access directories into the Sufiles directory. This procedure is done so that when you build your distribution diskettes, the Setup Wizard only needs to copy the compressed files to the diskette directories and will not have to recompress each time a distribution diskettes are built.

### **More Information:**

---

When this application is executed it will at first appear to be hung. Your cursor will remain a pointer and not switch to an hourglass to indicate that the process is working. The progress meter will remain at 0% and you will not see any hard disk activity. At some point, the screen will be updated.

Depending on machine type and amount of memory this process will take various amounts of time the average appears to be approximately 20 minutes.

486-50 w/16 meg of memory 10 minutes

383-16 w/4 meg of memory 45 minutes

This is designed to happen this way to give the users the ability to press the cancel button, if the pointer was changed to an hourglass you would not have that ability.

References:

[References](#)

## PRACC9306: Exporting Nulls to Paradox Actually Exports a Space

Article Number: Q100964  
CREATED: 01-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0
- 

### SYMPTOMS

---

No records are returned when a Paradox user tries to search a Paradox database created using the Microsoft Access Export command. This problem occurs only when the criteria specifies fields that are BLANK. (The Paradox BLANK criteria is equivalent to the Microsoft Access NULL criteria.)

### CAUSE

---

When Microsoft Access exports a table to a Paradox database, all fields that are NULL in Microsoft Access are replaced with a space during export to Paradox. This prevents Paradox users from correctly retrieving records based on the Paradox BLANK criteria command.

This problem may also occur if a Microsoft Access .MDB file attaches to a Paradox database and the Microsoft Access user deletes the entire contents of a field from the attached Paradox database. Microsoft Access treats the empty field as NULL, while a Paradox user treats the same field as NOT BLANK.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem has been corrected in Microsoft Access 1.1.

Steps to Reproduce Problem  
-----

1. Open the sample database NWIND.MDB.
2. From the File menu, choose Export.
3. In the Export dialog box, select the Paradox 3.X file format and the Customers table. Name the file CUSTOMER.DB.
4. Open Paradox 3.5 and create a new query, using graphical query by example (QBE) on the exported CUSTOMER.DB file.
5. Using the F6 key, check the Customer ID and Region fields in QBE.
6. Add the criteria of BLANK to the Region field.
7. Press F2 to execute the query. Zero (0) records are returned to the Paradox Answer table.

8. Change the criteria of the Region field to NOT BLANK, then press F2. Note that all the records are returned.

Paradox is manufactured by Ansa Software, a Borland company, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

### [References](#)

## PRB: Report Wizard Defaults To 8 1/2 by 11 Inch Page

Article Number: Q100963  
CREATED: 01-JUL-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0
- 

### SYMPTOMS

---

When you use the Report Wizard to create a [report](#), if the option Fit All Fields On One Page is selected, the report automatically defaults to an 8 1/2 by 11 inch portrait page size. This happens even if you have changed the printer driver in the Control Panel to be setup for 8 1/2 by 14 Landscape.

### CAUSE

---

The default page size is set by the Report Wizard and does not reflect the printer settings you may have setup in the Printer icon in the Control Panel of Windows.

### RESOLUTION

---

In design view of the report modify the layout to reflect the page size you desire. You can extend the right edge of the report and place the controls in the newly extended area.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

The following example is based on the sample [database](#) Northwind Traders (NWIND.MDB).

Steps to Reproduce Behavior  
-----

1. In the Main [group](#) select the Control Panel icon.
2. Select the Printers icon.
3. Select the Setup button to set the page size to 8 1/2 by 14, Landscape.
4. Start Microsoft Access and open NWIND.MDB.
5. Create a new report based on the Employees [table](#)

6. Chose Report Wizard and select Groups/Totals.
7. Add all of the fields to the report except the Photo field.
8. After building the report, choose the option Fit All Fields One One Page.
9. Select the option to Print Preview the report.

Result: You see the report formatted for 8 1/2 by 14 paper, but the right side of the report is blank. When you choose the Cancel button to switch to design view notice that the right edge of the report is aligned to the 8 inch mark on the horizontal ruler bar.

## [References](#)

## INF: Default Value Of A Combo Box Equal To The First Row

Article Number: Q100924  
CREATED: 30-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
  - Microsoft Access Distribution Kit version 1.1
- 

Combo boxes based on tables or queries values can change frequently. If you want the default value to reflect the first row of the results of the table or query you can do this with a domain function. With this process the default value will always reflect the first row of the list.

To demonstrate this process we can create a combo box in the sample application NorthWind.

1. Open the sample database Northwind Traders (NWIND.MDB).
2. Open the Employees form in design mode.
3. From the toolbox select a combo box and add it to the form.
4. Select View Properties and the properties for the combo box will be displayed.
5. Add the following properties.

```
Form property
=====
RowSource: Employees
ColumnCount: 2
ColumnWidth: 0 in; 1 in;
BoundColumn: 2
Default Value: =Dfirst("[Last Name]","Employees")
```

>From the menu choose View, Form

The default value is now set to the first row of the list.

### References

## INF: Sample Macro for Repair/Compact Operations

Article Number: Q100923  
CREATED: 30-JUN-1993  
MODIFIED: 10-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

This article includes a sample [macro](#) that repairs and compacts a [database](#).

### MORE INFORMATION

---

To manually repair and compact a database from the File menu, you must close the database and perform the compact and repair steps individually.

The macro provided automates this process. It repairs a database called MYDB and compacts it to a new database called NewDB. The macro prompts you for information only if the name of the database it is compacting to (for example, NewDB) already exists.

NOTE: The macro must be run from a different database than the one you are repairing.

1. From the File menu, choose New and then Macro.

-or-

Choose the Macro button in the [Database window](#) and then choose the New button.

2. Create the following macro:

| MacroName | Action      | Defined Below |
|-----------|-------------|---------------|
| -----     | -----       | -----         |
| Test      | SetWarnings | 1.            |
|           | SendKeys    | 2.            |
|           | DoMenuItem  | 3.            |
|           | SendKeys    | 4.            |
|           | DoMenuItem  | 5.            |
|           | SetWarnings | 6.            |

Test1 Actions

- 
1. SetWarnings  
    WarningsOn: No
  2. SendKeys  
    Keystrokes: MYDB.mdb{enter}  
    Wait: No
  3. DoMenuItem

- Menu Bar: Init
- Menu Name: File
- Command: Repair Database
- 4. SendKeys
  - Keystrokes: MYDB.mdb{enter}NewDB{enter}
  - Wait: No
- 5. DoMenuItem
  - Menu Bar: Init
  - Menu Name: File
  - Command: Compact Database
- 6. SetWarnings
  - Warnings On: Yes

NOTE: In most cases this macro works fine. However, remember that keystrokes sent by the SendKeys action are buffered temporarily and that results may vary. A large database may take extra time to compact or repair; in this case, the macro may not exhibit expected behavior.

## [References](#)

## PRACC9306: Import/Append Actions Return Errors for Each Record

Article Number: Q100922  
CREATED: 30-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SYMPTOMS

---

Executing import and append actions that cause every record to return an import error generates the following error message:

    Couldn't Import. No records found or all records contained errors.

After this error message is displayed, an empty Import Errors table is created. Also, the Help jump for this error message does not display a Help topic, but instead takes you to the Microsoft Access Help Table of Contents.

### RESOLUTION

---

To avoid returning an import error for all records, either modify the existing table's structure or modify the file itself. If at least one record imports correctly, you will not receive the above error message and the Import Errors table will include the correct error descriptions for each error.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. From the File menu, choose Export. Select Text [Delimited].
3. Select the Customers table. Name the file CUSTOMER.TXT.
4. In the Export Text Options box, choose OK.
5. From the File menu, choose Import. Select Text [Delimited] again.
6. Select the CUSTOMER.TXT file.
7. Choose Append to Existing Table, select the Customers table, and choose OK.

NOTE: Due to an indexing violation, trying to use a import and append actions to move the CUSTOMER.TXT file back to the Customers table causes an error for each record.

After a few seconds, the above error message is displayed. Note that the Help jump has taken you only to the Help Table of Contents. At the same time, an empty Import Errors table has been created.

## [References](#)

## PRB: Case Sensitivity is Different with Attached Tables

Article Number: Q100921  
CREATED: 30-JUN-1993  
MODIFIED: 17-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### SYMPTOMS

---

If you use an attached table in a query the criteria are case sensitive. This is not the case with a native Microsoft Access table. In a query based on a Microsoft Access table the same query would not be case sensitive.

### RESOLUTION

---

There are several methods you can use to keep this from happening. The first method is to import the attached table. Then base your query on the imported table. The second method is to include a native Microsoft Access table joined to the attached table in the query. Both of these options will allow you to enter non case-sensitive values.

Another option is to be aware that you are using an attached table and make sure that the values you use for your criteria, match the case of the values in the attached tables.

A third option is to use the UCase() function to ensure that the field in the query dynaset is returned in uppercase. Using this method, you would enter your criteria in uppercase. For example,

```
SELECT DISTINCTROW UCase([ICONTRACT_TI]) AS Title
FROM AttachedSUPPLIER
WHERE(((UCase([CONTACT_TI])) Like "*MANAGER*"));
```

### MORE INFORMATION

---

Steps to Reproduce Behavior  
-----

1. Create a new database and attach to the dBASE file named NEWCUST.DBF file. This sample dBASE file is supplied with Microsoft Access.
2. Create a new query and add the attached NEWCUST table. This query will be called QueryTest.
3. Drag all the fields to the query grid.
4. Make the following change to the CITY column:

Query: QueryTest  
-----

FieldName: CITY  
Show: True  
First Criteria Line: Like "s\*"

Notice that the "s" is lowercase.

The same is true if your criteria was "= seattle". This criteria search would return an empty set.

5. Run the query. The results will be an empty set.
6. Now import the dBASE table. While the database window has the focus select Import from the File menu. In the Data Source list box, select dBASE III. Choose the OK button.
7. Recreate the query using steps 1-4 but this time add the imported table.
8. Run the query. The results return all of the records with Seattle as the city.

## [References](#)

## INF: How to Avoid Abandoned Group Headers in Reports

Article Number: Q100920  
CREATED: 30-JUN-1993  
MODIFIED: 21-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

This article describes a routine to keep a set number of detail records together on a page with its [group header](#). The article also explains how to avoid an abandoned group [header](#) at the bottom of a page and how not to print the group header beyond a certain position on the page in a single-[column](#) or multi-column [report](#).

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

### MORE INFORMATION

---

The KeepTogether property is used to keep all of the controls for one [record](#) together in a particular section of a report. This property does not keep multiple records together, nor does it keep a group header and its detail records together.

Method 1 helps you to find the precise location on the page beyond which you should not print a group header, if you want to keep a set number of detail records with that header.

Method 2 helps you to determine the exact [twip](#) location by trial and error and provides two functions, one for a single-column report and the other for a multi-column report.

Method 1

-----

To keep a set number of detail records together on a single page with the group header, you need to know approximately how many detail records will be printed per group, or, more importantly, how many records you want to print with a particular group header. You can [preview](#) your report to find the group that prints the most detail records to determine how many records to keep with the group header.

For example, say you have determined that the largest group prints prints 24 detail records. The Detail section of your report is 1/4 inch in height; thus, 24 detail records take up 6 inches on a page. You must also consider the top and bottom margins, which default to 1 inch each, and the height of your Group Header section.

Using the Report.Top property, you can determine where on the page the

next section will print. If enough space remains to print your group header, all of its detail records, and the page footer (if any), plus the bottom margin, your group header will be printed on the same page as its detail records. If there is not enough blank space, delay printing the report ( use a combination of the MoveLayout, NextRecord, and PrintSection properties to leave the rest of the page blank) and continue the group header information on the next page.

Below are sample measurements on which this article's calculations are based:

Page Length: 11 inches  
Detail.Height: 0.25 inch

Add the following values together:

Top Margin: 1 inch  
Page Header.Height: 1 inch  
Group Header.Height: 0.5 inch  
Group Footer.Height: 0.5 inch  
Page Footer.Height: 1 inch  
Bottom Margin: 1 inch  
-----  
Total = 5 inches

To determine the blank space left on the page for detail records, subtract the total above from your Page Length property value (11 inches - 5 inches = 6 inches).

Divide the maximum space for detail records by the Detail.Height property value (6 inches / 0.25 inch). A total of 24 detail records will fit on a page. Force a new page if fewer than 8 records will print with the header (for example, if the formula returns either less than the sum of 8 \* 0.25 inches or less than 2 inches of blank space).

To determine where the top of the 2-inch limit is, or the point beyond which you should not print the group header, add the following values together:

Top margin 1 inch  
+ Page Header.Height 1 inch  
+ Remainder of detail section (6 inches - 2 inches)  
  
= 1 inch + 1 inch + (6 inches - 2 inches)

Final calculation for the function: = 6 inches or (6 \* 1440 twips)

To implement this calculation in a module, use the following steps:

1. Enter the =PrintOK() function in the OnFormat property of your group header. (This function assumes a report name of MyReport.)
2. Type the following lines in the Declarations section of a module:

Option Explicit  
  
Function PrintOK()

```

 Dim R As Report
 Set R = Reports!MyReport
 If R.Top > (6 * 1440) Then
 R.MoveLayout = TRUE
 R.PrintSection = FALSE
 R.NextRecord = FALSE
 End If
End Function

```

## Method 2

-----

Another way to avoid an abandoned group header at the bottom of a page, or not to print the group header beyond a certain position on the page, is to replace (6\*1440) in the function above with the twip location beyond which you do not want to print.

The report tests the Top property; if the condition is True, the report does not print the section. Instead, it moves down the report layout and attempts to print the section until the condition is False, which occurs at the top of the next page.

For a multi-column report, the function must be modified as follows:

```

Option Compare Database
Option Explicit

```

```

'=====
'
' Function to force to new column if at bottom of column.
' R is the name of the Report being formatted, as in
' Reports![ReportName], and
' Bottom is the value (in inches) at which a new column is forced.
'=====

```

```

Function IfBottom (R As Report, Bottom As Integer)
 Dim YPos, LastPos

 YPos = R.Top
 If YPos > Bottom * 1440 Then 'Beyond this position, go to a new column.
 R.MoveLayout = True
 R.NextRecord = False
 R.PrintSection = (YPos = LastPos) 'Set to True when at Max position.
 LastPos = YPos
 End
End Function

```

## [References](#)

## INF: Consecutively Numbering Pages for Two or More Reports

Article Number: Q100919  
CREATED: 30-JUN-1993  
MODIFIED: 30-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### Summary:

---

This article describes how to consecutively number the pages of multiple reports. For example, if the first [report](#) has 10 pages, the second report will start numbering at page 11.

### More Information:

---

This article assumes that you are familiar with [Access Basic](#) and how to create and use Access Basic Procedures.

1. First open a new Module or a previously created [module](#) and enter the following code:

```
'*****
'Declarations Section of Module
'*****
Option Explicit
Global CurrPage As Integer

'=====
```

'This [function](#) returns the page number to be used in the consecutive report's [footer](#)(s). It also increments the Global Variable CurrPage.

```
'=====
```

```
Function GetPage ()
 CurrPage = CurrPage + 1
 GetPage = CurrPage
End Function
```

'=====

'This function is called by the first report to initialize the Global [variable](#) to 0.

```
'=====
```

```
Function InitPage ()
 CurrPage = 0
End Function
```

'=====

'This function is called from the first report's [page footer](#).

```
'=====
```

```
Function SetPage (PgNumber As Integer)
 CurrPage = PgNumber
End Function
```

2. Open the first report in design view by selecting the report in the Database window and choosing the Design button.
  3. From the View menu, choose Properties to display the property sheet.
  4. This report needs to have a Report Header. If the report does not already have a Report Header, select Report Hdr/Ftr from the Layout menu.
- Note: if you do not want a Report Header or Footer, simply resize the section to a height of 0.
5. Click on any portion of the Report Header Section not covered by a control to display the Report Header properties in the property sheet.
  6. Set one of the following properties for the Report Header Section:

On format: =InitPage()  
-or-  
On print: =InitPage()

7. Click on any portion of the Page Footer Section not covered by the control to display the Page Footer properties in the property sheet.
  8. Set one of the following properties in the Page Footer Section:
- On format: =SetPage(Page)  
-or-  
On print: =SetPage(Page)
9. Open the second report in design view, by selecting the report in the Database window and choose the Design button.
  10. Create a text box control in the Page Footer for this report.

11. Set the following property for the text box control you created in the above step:

ControlSource: =GetPage()

12. Repeat steps 9-11 for any additional reports whose pages are to be consecutively numbered.
13. Print the reports in order - i.e. print the first report, then the second. The pages will be numbered consecutively.

Note: In order to ensure that the reports are numbered correctly, you should print the reports consecutively without Print Previewing each report. If you wish to use Print Preview to view the reports, be sure to then go back and print the reports consecutively.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 18, "Designing Reports," pages 422-424, 440-444

"Microsoft Access Language Reference," version 1.0, page 364

## [References](#)

## FastTips: Setup Questions & Answers

Article Number: Q100851  
CREATED: 29-JUN-1993  
MODIFIED: 27-AUG-1993  
VERSION(S): 1.00 1.10

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0812: SETUP QUESTIONS & ANSWERS  
=====

Revision Date: 5/93  
No Disk

The following information applies to Microsoft Access, version 1.0  
Copyright 1993.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR  
| PURPOSE. The user assumes the entire risk as to the accuracy and  
| the use of this Application Note. This Application Note may be  
| copied and distributed subject to the following conditions: 1) All  
| text must be copied without modification and all pages must be  
| included; 2) If software is included, all files on the disk(s)  
| must be copied without modification (the MS-DOS(R) utility  
| diskcopy is appropriate for this purpose); 3) All components of  
| this Application Note must be distributed together; and 4) This  
| Application Note may not be distributed for profit.  
|

| Copyright (C) 1993 Microsoft Corporation. All Rights Reserved.  
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks  
| and Windows is a trademark of Microsoft Corporation.

| Quicken is a registered trademark of Intuit.

| This document was created using Microsoft Word for Windows(TM).  
|

---

1. Q. Why does Microsoft recommend that I remove SHARE.EXE from my AUTOEXEC.BAT file if I run Microsoft Windows for Workgroups?

A. We recommend that you remove SHARE.EXE from your AUTOEXEC.BAT file only if you run Microsoft Windows for Workgroups in 386 enhanced mode. Windows for Workgroups in 386 enhanced mode loads and uses a file-sharing program called VSHARE.386, which replaces SHARE.EXE. There is an entry for VSHARE.386 in the SYSTEM.INI file.

If you plan to run Windows for Workgroups exclusively in 386 enhanced mode and do not run other applications that require SHARE.EXE, you can save approximately 5 kilobytes (K) of conventional memory by not loading VSHARE.386 at all. If you have already loaded SHARE.EXE, delete it from your AUTOEXEC.BAT file and restart your computer.

You are likely to find SHARE.EXE in your AUTOEXEC.BAT file because SETUP.EXE automatically inserts the following MS-DOS command in your AUTOEXEC.BAT file when you install Microsoft Access

```
<directory>\share.exe /l:500
```

where <directory> is the name of your MS-DOS directory.

If you load VSHARE.386 and SHARE.EXE is already present, VSHARE.386 temporarily disables it. VSHARE.386 takes over file-sharing tasks until you exit Windows for Workgroups. VSHARE.386 then transfers the tasks back to SHARE.EXE, so file sharing is available to your MS-DOS-based applications.

Microsoft Access runs correctly with either SHARE.EXE or VSHARE.386, but SHARE.EXE limits the number of locks available to the number you specified when you loaded the file originally. The /L parameter specifies the number of locks; the default is 20. VSHARE.386, on the other hand, dynamically allocates the number of locks available, based on demand. The number of locks available is especially important if you are running Windows for Workgroups in a client-server environment.

2. Q. Why do I get the error message "Outdated 'COMMDLG.DLL' found. Please reinstall Microsoft Access" when I install Microsoft Access?

A. This error can occur for a number of reasons. To isolate the cause, check the following:

- Make sure that you have no more than one version of the COMMDLG.DLL file on your hard disk. If you have more than one version, either delete or rename your duplicate COMMDLG.DLL files.

If you are running MS-DOS version 5.0 or later, use the following two steps to search for ALL occurrences of the

COMMDLG.DLL file:

1. Quit Windows.
2. At the MS-DOS prompt, type the following command for all logical drives

```
dir <drive>:\commdlg.dll /s
```

where <drive> is the drive on which your Windows directory is located, the drive on which you installed Microsoft Access, and each drive located in the PATH statement in your AUTOEXEC.BAT file.

For versions of MS-DOS earlier than 5.0, you must either go into each directory and execute the above DIR command without the /S parameter, or use a utility to search for all occurrences of the COMMDLG.DLL file.

- Make sure that the COMMDLG.DLL file is located in your Windows SYSTEM subdirectory only.
- Make sure that the COMMDLG.DLL file is up to date. The file should have one of the three dates below, or later:

|          |                                           |
|----------|-------------------------------------------|
| 03/10/92 | (for Microsoft Windows 3.1)               |
| 10/01/92 | (for Microsoft Windows for<br>Workgroups) |
| 10/25/92 | (for Microsoft Access)                    |

- Make sure that you are not using a third-party, incompatible COMMDLG.DLL file.
- Make sure that no other applications are running with Windows before you run Microsoft Access Setup.

You cannot update the COMMDLG.DLL file during installation if some other application is currently using the file. To ensure that all other Windows-based applications are closed, use the following five steps:

1. Create a temporary Program Manager group.
2. Move all the icons from the Startup group into the temporary group.
3. Back up your WIN.INI file. In the original copy, delete all the items from both the run= and the load= lines to read as follows:

```
[windows]
run=
load=
```

4. Restart Windows.
5. After installing Microsoft Access, move the icons from

the temporary group back into the Startup group and restore the WIN.INI file from your backup. Quit and restart Windows.

If you follow all the previous instructions and you still get an "Outdated DLL" error message, your copy of the COMMDLG.DLL file may be corrupt or damaged. To solve this problem, reinstall a new copy of this file from your Microsoft Access package using the following three steps:

1. Copy the COMMDLG.DL\_ file from Disk 1 of your Microsoft Access package to a directory on your hard disk.

NOTE: Be sure to copy the COMMDLG.DL\_ file, not the COMMDLG.DL\$ file.

2. Copy the EXPAND.EXE file from the appropriate Windows setup disk to the same directory you used in step 1.
3. Use EXPAND.EXE to unpack the COMMDLG.DL\_ file and place it in your Windows SYSTEM subdirectory. The [syntax](#) to complete this action is as follows:

```
<drive>:\<directory>\expand commdl_g.dl_
<destination>\commdl_g.dll
```

where <drive> and <directory> correspond to the location of EXPAND.EXE and <destination> is the drive and location of your Windows SYSTEM subdirectory.

3. Q. When I install Microsoft Access, why does Setup fail while copying the README.TX\$ file?  
A. Setup problems are often caused by terminate-and-stay-resident (TSR) programs that are loaded at the MS-DOS level and/or at the Microsoft Windows level.

To determine if TSRs are causing Setup to fail, remove all unnecessary TSRs and device drivers from the CONFIG.SYS and the AUTOEXEC.BAT files, remove all icons from the Startup group in Program Manager, and delete all entries on the load= and run= lines in your WIN.INI file.

NOTE: Be sure to back up your CONFIG.SYS, AUTOEXEC.BAT, and WIN.INI files before modifying them.

Some TSRs known to cause problems with the Microsoft Access Setup program include the following:

| Filename     | Description                                                                                             |
|--------------|---------------------------------------------------------------------------------------------------------|
| APPEND.EXE   | An MS-DOS TSR loaded in the AUTOEXEC.BAT file                                                           |
| SUBST.EXE    | An MS-DOS TSR loaded in the AUTOEXEC.BAT file                                                           |
| BILLMNRD.EXE | An automated bill reminder loaded with Quicken(R) that can reside on the load= line of the WIN.INI file |

4. Q. A directory called MS-SETUP.T is left on my hard disk after I install Microsoft Access. Can I safely delete this directory?
- A. Yes. After Microsoft Access installation is complete, it is okay to delete the MS-SETUP.T directory and all the files in it.

This directory is not deleted automatically because of a known problem that occurs in Microsoft Access Setup when you choose the Reboot Your System After Completing Setup option. If you choose the Return To Windows After Setup option instead, Microsoft Access automatically deletes the MS-SETUP.T directory and its contents.

#### [References](#)

KBCategory:

KBSubcategory:

## FastTips: Interoperability Questions & Answers

Article Number: Q100850  
CREATED: 29-JUN-1993  
MODIFIED: 27-AUG-1993  
VERSION(S): 1.00 1.10

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0811: INTEROPERABILITY WITH OTHER APPLICATIONS QUESTIONS & ANSWERS  
=====

Revision Date: 5/93  
No Disk

The following information applies to Microsoft Access, version 1.0  
Copyright 1993.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR  
| PURPOSE. The user assumes the entire risk as to the accuracy and  
| the use of this Application Note. This Application Note may be  
| copied and distributed subject to the following conditions: 1) All  
| text must be copied without modification and all pages must be  
| included; 2) If software is included, all files on the disk(s)  
| must be copied without modification (the MS-DOS(R) utility  
| diskcopy is appropriate for this purpose); 3) All components of  
| this Application Note must be distributed together; and 4) This  
| Application Note may not be distributed for profit.  
|  
|  
|

| Copyright (C) 1993 Microsoft Corporation. All Rights Reserved.  
| Microsoft, FoxPro, Microsoft Access, and MS-DOS are registered  
| trademarks and Windows is a trademark of Microsoft Corporation.  
| Paradox is a registered trademark of Ansa Software, a Borland  
| company.  
|

| dBASE and dBASE IV are registered trademarks of Borland

| International, Inc.

| This document was created using Microsoft Word for Windows(TM).  
|-----

1. Q. When I export a Microsoft Access [table](#) to [SQL](#) Server and then reattach it to Microsoft Access, I can't edit the data. Why does this problem occur?
  - A. Microsoft Access can [update](#) an attached SQL Server table only if the table has a unique [index](#). However, Microsoft Access does not build SQL Server table indexes, unique or otherwise, when you export data; Microsoft Access allows you only to export data to SQL Server back ends. For example, if you attach the table immediately after exporting data to it from Microsoft Access, the data is displayed in Datasheet or [Form view](#) as read-only; therefore, updates to the exported table's data are not allowed.

To work around this problem, you must create the SQL Server table indexes manually after you export the data from Microsoft Access to SQL Server. After the SQL Server indexes are created, you can create forms based on the attached SQL Server tables and update the data.

For information on creating indexes based on SQL Server [database](#) tables, see the "Microsoft SQL Server System Administrator's Guide", or pages 67-72 in the "Microsoft SQL Server Language Reference".

2. Q. How do I [link](#) a Microsoft Access table to a Microsoft Excel spreadsheet using dynamic data exchange ([DDE](#))?
  - A. The easiest way to use DDE is to create a paste link between Microsoft Access and Microsoft Excel. To do this, use the following steps:
    1. In the Microsoft Access [Database window](#), highlight the table you want to link to Microsoft Excel.
    2. From the Microsoft Access Edit menu, choose Copy.
    3. From the Microsoft Excel Edit menu, choose Paste Link.

The formula resulting from the Paste Link command should look like the following example

```
{=MSAccess|'NWIND.MDB;Table <Customers>'!All}
```

where <Customers> is the name of the table you copied.

For more information on linking Microsoft Access data to data in other applications, see the "Microsoft Access User's Guide", version 1.0, Chapter 13, "Using Pictures, Graphs, and Other

Objects," pages 337-339.

3. Q. Why do I get a "database corrupted" error message when I try to import a Paradox(R) database into Microsoft Access?

A. When you import a keyed Paradox table into Microsoft Access, you must place it in the same directory as its associated index file. Because Paradox stores information about a table's primary key in the .PX file, it is important to verify that the .PX file is present before you import the Paradox table.

If the index (.PX) file is not found, the following error message appears

```
<filename>#DB' is corrupted or isn't a Microsoft database
```

where <filename> is the name of your Paradox table.

If a Paradox table and its associated index file become separated, or if the .PX file becomes corrupted, Microsoft Access is unable to import the table. To recreate the index file, you must restructure the table in Paradox.

4. Q. When will I be able to attach Microsoft FoxPro(R) tables to Microsoft Access?

A. Although Microsoft Access can import Microsoft FoxPro tables, it currently cannot attach most FoxPro tables. However, Microsoft Access can attach a FoxPro table as a dBASE IV(R) table if BOTH of the following conditions are met:

- The FoxPro table does not have any memo fields in its structure.
- The FoxPro table is attached without any FoxPro index files.

If either of the above conditions is not met, the FoxPro table must be imported, rather than attached, to Microsoft Access. The reason for this is that although FoxPro uses a .DBF file structure similar to that of dBASE(R), the memo and index structures of FoxPro and dBASE .DBF files are different.

Microsoft is currently working on a Microsoft Access driver for FoxPro that will allow you to attach all FoxPro tables in their original formats. When the driver is complete, Microsoft will make it available free of charge.

## [References](#)

## FastTips: Reports Questions & Answers

Article Number: Q100849  
CREATED: 29-JUN-1993  
MODIFIED: 27-AUG-1993  
VERSION(S): 1.00 1.10

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0640: REPORTS QUESTIONS & ANSWERS  
=====

Revision Date: 5/93  
No Disk

The following information applies to Microsoft Access, version 1.0  
Copyright 1992-1993.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR  
| PURPOSE. The user assumes the entire risk as to the accuracy and  
| the use of this Application Note. This Application Note may be  
| copied and distributed subject to the following conditions: 1) All  
| text must be copied without modification and all pages must be  
| included; 2) If software is included, all files on the disk(s)  
| must be copied without modification (the MS-DOS(R) utility  
| diskcopy is appropriate for this purpose); 3) All components of  
| this Application Note must be distributed together; and 4) This  
| Application Note may not be distributed for profit.  
|

| Copyright (C) 1992-1993 Microsoft Corporation. All Rights  
Reserved. |

| Microsoft, Microsoft Access, and MS-DOS are registered trademarks  
| and Windows is a trademark of Microsoft Corporation.

| This document was created using Microsoft Word for Windows(TM).  
|

1. Q. Why is every other page of my [report](#) blank? How can I correct this problem?
  - A. This problem commonly occurs when the width of your report combined with the left and right margins is greater than the width of your paper. Adjust the width of your report and the margins so that your report fits on the page.
2. Q. How can I convert a [form](#) to a report?
  - A. To convert a form to a report, open the form in [Design view](#) and choose Save As Report from the File menu.
3. Q. How can I sort the data in a report by a [field](#) not displayed on the report?
  - A. To sort report data by a field in a separate [table](#), create a [query](#) that includes the field from the other table, then base the report on the query. If you have already created the report, verify that the query includes all the fields used in the report and change the RecordSource property of the report to refer to the new query. Now you can adjust the Sorting and Grouping properties of the report to sort on the new field.

## [References](#)

KBCategory:

KBSubcategory:

## FastTips: Querying Questions & Answers

Article Number: Q100848  
CREATED: 29-JUN-1993  
MODIFIED: 27-AUG-1993  
VERSION(S): 1.00 1.10

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0639: QUERYING QUESTIONS & ANSWERS  
=====

Revision Date: 5/93  
No Disk

The following information applies to Microsoft Access, version 1.0  
Copyright 1992-1993.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR  
| PURPOSE. The user assumes the entire risk as to the accuracy and  
| the use of this Application Note. This Application Note may be  
| copied and distributed subject to the following conditions: 1) All  
| text must be copied without modification and all pages must be  
| included; 2) If software is included, all files on the disk(s)  
| must be copied without modification (the MS-DOS(R) utility  
| diskcopy is appropriate for this purpose); 3) All components of  
| this Application Note must be distributed together; and 4) This  
| Application Note may not be distributed for profit.  
|

| Copyright (C) 1992-1993 Microsoft Corporation. All Rights  
Reserved. |

| Microsoft, Microsoft Access, and MS-DOS are registered trademarks  
| and Windows is a trademark of Microsoft Corporation.

| Btrieve is a registered trademark of SoftCraft, Inc.,

| a Novell Company.  
|

dBASE III PLUS and dBASE IV are registered trademarks of  
Ashton-Tate Corporation.

Paradox is a registered trademark of Ansa Software,  
a Borland company.

This document was created using Microsoft Word for Windows(TM)

---

1. Q. How can I export the result of a [query](#)?
  - A. To export the results of a query, you must create a [table](#) that contains the results. To do this, create a [make-table query](#) in [Design view](#) (choose Make Table from the Query menu). The resultant table will contain the data to be exported. After the make-table query is complete, choose Export from the File menu to export the data in the desired format. For more information, please refer to Chapter 4 of the "User's Guide."
2. Q. Can I [join](#) tables from different databases into one query?
  - A. Yes, you can join tables from different databases into one query. Using the Attach Table [method](#), you can generate queries from any supported data source (for example, Btrieve(R), dBASE III PLUS(R), dBASE IV(R), or Paradox(R)).
3. Q. Is data in tables stored in a sorted order? How can I view my data in sorted order?
  - A. Data in tables is not stored in sorted order. Data is stored in the order in which it was entered. To view data in sorted order, create either a query or a [form](#) based on a query that uses the Sort [field](#) on the query grid. By default, if the table includes a [primary key](#), the [Datasheet view](#) of the table will display the data sorted by the primary key. To view the data sorted by something other than the primary key alone, use a query, as previously described.
4. Q. How can I create computed columns in tables?
  - A. You can create computed columns or expressions with queries. Within the query, create a [column](#) that is defined as an [expression](#). In general, it is helpful to think of queries as virtual tables; you can use a query wherever you can use a table. If you're familiar with [SQL](#) terminology, this is very similar to creating a view. Unlike most implementations of views, however, Microsoft Access views can be updated even if they involve joins from different data sources, such as Paradox, Btrieve, or separate Microsoft Access data sources.
5. Q. What is the difference between the keywords DISTINCT and DISTINCTROW?
  - A. DISTINCT is part of the SQL standard and causes a query to

return unique data, rather than unique records. For example, even if there are 10 customers named Jones, the query "SELECT DISTINCT Name FROM Customer" returns only one row containing Jones. With Microsoft Access queries, you specify DISTINCT by choosing Query Properties from the View menu and then selecting the Unique Values Only check box.

DISTINCTROW is unique to Microsoft Access and is not part of the SQL standard. It causes a query to return unique records, rather than unique data values. For example, if there are 10 customers named Jones, the query "SELECT DISTINCTROW Name FROM Customer" returns all 10 rows containing Jones.

The main reason for adding the DISTINCTROW keyword to Microsoft Access SQL is to support semi-joins that can be updated (for example, one-to-many joins in which the output columns all come from the one-sided table). DISTINCTROW is specified by default in Microsoft Access queries and is ignored in queries for which it has no effect.

6. Q. Why do queries change the order of my columns?
  - A. When you close a query, Microsoft Access moves the sorted fields to the leftmost columns in the Query-By-Example (QBE) grid. For example, if you open an existing query in Design view, revise it, and save your changes, Microsoft Access displays the sorted fields to the left of the datasheet. You can rearrange the fields if necessary.
7. Q. I am sending a query to SQL Server through Microsoft Access, but the query runs very slowly. Why does this occur, and how can I increase the query speed?
  - A. The query speed most often decreases when you send implicit, instead of explicit, parameters to the remote server.

Query parameters can be either implicit or explicit. A parameter entry made only in the Query-By-Example (QBE) grid is called an implicit parameter. A parameter entry made in both the QBE grid and the Query Parameters dialog box is called an explicit parameter.

If you use implicit parameters in your query, Microsoft Access will not send the query to the remote server because Microsoft Access cannot verify what implicit conversions the remote server provides. As a result, Microsoft Access processes the query locally, which can noticeably reduce the query speed. However, if you use explicit parameters in your query, Microsoft Access knows the data types of the parameters. Subsequently, Microsoft Access sends the query to the remote server for processing, which speeds up the query.

Note also that Microsoft Access automatically treats unrecognized or misspelled names and expressions as implicit parameters, rather than as errors. Microsoft Access then makes a "best guess" about the implicit parameter's data type. If Microsoft Access guesses the wrong data type, a query can

return unexpected results or values, either when you enter a parameter value or when you execute queries on a remote server.

### References

relational

KBCategory:

KBSubcategory:

## FastTips: Nontechnical and Marketing Questions & Answers

Article Number: Q100847  
CREATED: 29-JUN-1993  
MODIFIED: 27-AUG-1993  
VERSION(S): 1.00 1.10

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0638: NONTECHNICAL AND MARKETING QUESTIONS & ANSWERS  
=====

Revision Date: 5/93  
No Disk

The following information applies to Microsoft Access, version 1.0,  
Copyright 1992-1993.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR  
| PURPOSE. The user assumes the entire risk as to the accuracy and  
| the use of this Application Note. This Application Note may be  
| copied and distributed subject to the following conditions: 1) All  
| text must be copied without modification and all pages must be  
| included; 2) If software is included, all files on the disk(s)  
| must be copied without modification (the MS-DOS(R) utility  
| diskcopy is appropriate for this purpose); 3) All components of  
| this Application Note must be distributed together; and 4) This  
| Application Note may not be distributed for profit.  
|

| Copyright (C) 1992-1993 Microsoft Corporation. All Rights  
Reserved. |

| Microsoft, FoxPro, Microsoft Access, and MS-DOS are registered  
| trademarks and VisualBasic and Windows are trademarks of  
| Microsoft Corporation.

| CompuServe is a registered trademark of CompuServe, Inc.

| GEnie is a trademark of General Electric Corporation.

|  
| This document was created using Microsoft Word for Windows(TM).  
|

---

1. Q. What sources of information about Microsoft Access, in addition to the documentation, does Microsoft provide?

A. Microsoft provides several resources with current information about Microsoft Access. If you have a modem, you can download Microsoft Knowledge Base articles about Microsoft Access from the Microsoft Download Service (MSDL). You can also download the Windows Driver Library (WDL) from this service. The WDL is also available at no charge on CompuServe(R), GENie(TM), and Microsoft OnLine. After you connect to any of these services, please read the WDL.TXT file for instructions on downloading the WDL and a complete list of files that it contains.

You can reach the MSDL at (206) 936-6735. There is no charge for using this service; however, standard connect-time fees and long-distance telephone charges do apply when you are downloading information.

On the CompuServe, GENie, and Microsoft OnLine systems, the WDL is located in the Microsoft Software Library. On CompuServe, you can reach the Microsoft Software Library by typing GO MSL at the "!" command prompt. To locate the WDL.TXT file, search on the word S13354.

If you do not have a modem, you can obtain an individual driver on disk by calling Microsoft Product Support Services at (206) 637-7098, Monday through Friday, 6:00 A.M. to 6:00 P.M., Pacific time. If you are outside the United States, contact the Microsoft subsidiary for your area. To locate your subsidiary, call Microsoft International Customer Service at (206) 936-8661.

2. Q. How many databases can I open at one time in Microsoft Access?

A. Through the user interface menus in Microsoft Access, you can open only one database at a time; therefore, you can view only one Database window in each session of Microsoft Access. In Access Basic, however, you can use the OpenDatabase() function to open several databases at once. The number of databases you can have open simultaneously through Access Basic is limited only by memory. Also, by choosing Attach Table from the File menu, you can attach to as many databases as memory allows.

3. Q. Should I use Microsoft Access or Microsoft FoxPro(R) as my database solution?

A. Microsoft FoxPro is the best database solution for users and developers of Xbase products, for developers who need a cross-platform solution, and for all customers who can benefit from FoxPro's incredible speed and unequalled development environment.

Microsoft Access is the best database solution for users who

need seamless access to data in multiple formats, for users who are not professional programmers but need to develop powerful database applications, and for users and developers who need the most productive development environment to design decision-support systems.

4. Q. What are the differences between Microsoft Access and Microsoft Visual Basic(TM)? Does Microsoft Access support custom controls designed for Visual Basic? Do these two applications use the same or different programming languages?
- A. While Microsoft Access and Microsoft Visual Basic share the same basic programming concepts (an [event](#)-driven programming model, visual interface creation, and so forth), each has different specialties.

Microsoft Visual Basic is a general-purpose programming environment that uses visual tools to make software developers more productive. It has a more granular event model and supports custom controls. However, as a general-purpose programming tool, Visual Basic lacks the [database objects](#), such as queries and reports, that Microsoft Access has.

Microsoft Access is a programmable database designed specifically for creating and running database applications. Its rich set of tools allows you to create many database applications without programming. Microsoft Access supports database-oriented events, such as FindRecord and OnUpdate, rather than Visual Basic-like properties and events, such as MouseMove and MinButton. Microsoft Access currently does not support custom controls.

The nature and [syntax](#) of Access Basic and Visual Basic are very similar, but not identical. Access Basic supports database commands and methods, such as CreateDynaset and Clone, that Visual Basic does not have. Visual Basic supports properties and events, such as PathChange and MouseUp, that Access Basic does not have. However, the many similarities between Access Basic and Visual Basic allow programmers to move between the two products easily.

5. Q. When will the run-time version of Microsoft Access be available?
- A. The run-time version of Microsoft Access will be available during the second quarter of 1993, at a suggested retail price of \$495. This run-time version will allow developers to create unlimited royalty-free applications without requiring users to purchase Microsoft Access.

## [References](#)

KBCategory:

KBSubcategory:

## FastTips: Forms Questions & Answers

Article Number: Q100846  
CREATED: 29-JUN-1993  
MODIFIED: 27-AUG-1993  
VERSION(S): 1.00 1.10

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0637: FORMS QUESTIONS & ANSWERS  
=====

Revision Date: 5/93  
No Disk

The following information applies to Microsoft Access, version 1.0.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an |  
| Application Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY |  
| KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO |  
| THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A |  
| PARTICULAR PURPOSE. The user assumes the entire risk as to the |  
| accuracy and the use of this Application Note. This Application |  
| Note may be copied and distributed subject to the following |  
| conditions: 1) All text must be copied without modification and |  
| all pages must be included; 2) If software is included, all files |  
| on the disk(s) must be copied without modification [the MS-DOS(R) |  
| utility DISKCOPY is appropriate for this purpose]; 3) All |  
| components of this Application Note must be distributed together; |  
| and 4) This Application Note may not be distributed for profit. |  
| |  
| Copyright 1992-1993 Microsoft Corporation. All Rights Reserved. |  
| Microsoft and MS-DOS are registered trademarks and Windows |  
| is a trademark of Microsoft Corporation. |  
-----

1. Q. How do I invoke my Access Basic code from within a form?

A. To call a function from a property, type the following:

```
=<functionname>()
```

The equal sign and parentheses are required. "<functionname>" is merely a placeholder for the Access Basic function. When you type in your function name, do not include the angle brackets ("<>"). You can call an Access Basic function from form or control properties, such as AfterUpdate or DefaultValue. Search for the name of a specific property in Help for examples. Subroutines cannot be called from properties. For more information, please refer to "An Introduction to Programming", Chapter 7, page 1.

2. Q. Do form rules override table rules?

A. Yes, form rules do override table rules. When you position fields on a form by dragging them from the Field list, the validation property for the field inherits the validation rule

that was defined for the field in the table design.

3. Q. When is it appropriate to use the exclamation point versus the period when identifying objects and properties in an [expression](#)?
  - A. A good rule of thumb is to use the exclamation point before anything you specifically name (such as the name of your form or a control on the form) and to use a period before anything Microsoft Access names (such as a property).
4. Q. Why doesn't the [header](#) I created show in [Form view](#)?
  - A. The header you created was most likely a [page header](#). There are two different types of headers available on forms: form headers and page headers. A [form header](#) is displayed on-screen and when printed; a page header is displayed only when printed.
5. Q. How do I reference a control on a [subform](#)?
  - A. To reference the subform control itself, you must use the form property of the subform control as follows:

forms!<masterformname>!<subformcontrolname>.form!<controlname>

6. Q. When are validation rules on a form evaluated?
  - A. Microsoft Access evaluates a validation rule only when data is entered or edited in a field and the cursor is moved to a different field or [record](#). If you leave the field unchanged, the validation rule is not evaluated. Microsoft Access also validates a field on a form when you leave the form, when you switch views, or when you close the form. To check for nulls, you must use a [macro](#). For more information, please refer to Chapter 22 of the "User's Guide."
7. Q. How do I create my own record-navigation system on a form without using the [navigation buttons](#) (called "VCR buttons")?
  - A. The Customers form in ORDENTRY.MDB, a sample [database](#) supplied with Microsoft Access, is an excellent example you can follow to create your own record-navigation controls. The navigation buttons on this form use generic Access Basic code, which you can import to your application.
8. Q. Why do I get the message "#Error" in some controls on my forms or reports?
  - A. Microsoft Access places an error value in a field or [text box](#) on a [datasheet](#) or form when it can't find information, execute an expression, or store a value within the field's prescribed limits. The following list describes the possible error values:
    - #ERROR: Microsoft Access cannot execute the expression. You may have supplied either too few or incorrect arguments for an [aggregate function](#). This error also occurs in a table or

form in which the DefaultValue property for a field or control is not appropriate for the DataType or FieldSize property setting, or in a [query](#) in which the value of a calculated field is larger than the FieldSize property setting allows. For example, this error message appears if you add or multiply two integer values and the resulting integer is larger than is permitted in an integer field.

This error also occurs when you substitute an expression for a text box and then use the text box's control name within the expression, as in the following example:

```
ControlName: [Test]
ControlSource: =[Test]
```

Since Microsoft Access automatically sets the ControlName property to match the field name, this error commonly occurs if you create a text box by dragging a field from the [Field list box](#) to a form or [report](#).

- #NUM!: The numeric value is too large (either positively or negatively) for Microsoft Access to store in the field, based on the current DataType or FieldSize property settings.
- #NAME?: The ControlSource property you entered for the field's value is invalid. This error can result from any of the following conditions:
  - You have misspelled the ControlSource property.
  - You have deleted the ControlSource property itself.
  - You have omitted the equal sign (=) when you entered an expression for the ControlSource property.
- #DIV/0!: You tried to divide a number by zero. You can neither do this directly in an expression (for example, 8/0) nor indirectly by using a value from a field whose numeric value is zero.
- #DELETED: The record to which you referred has been deleted. For example, if you or another user deletes a record from an underlying table's datasheet, this error value may replace a record in a form's datasheet.
- #LOCKED: Another user has locked this record; thus, Microsoft Access cannot read the data.

9. Q. How can I check for duplicate records immediately after I enter a value in a [primary key](#) field?

A. Normally, Microsoft Access does not check the values you enter in primary key fields for duplicates until you move to the next record. Thus, if you enter an invalid or duplicate value in a control (text box) and move to the next record, you may invalidate all your previous entries. However, there are two methods you can use to force an immediate check for duplicate

values.

Method 1: Create and call a macro that executes a DoMenuItem action of the Save Record command from the AfterUpdate property of your primary key field. If your record is a duplicate, the macro immediately causes the following error message to appear:

```
Can't have duplicate key; Index changes were unsuccessful.
```

Method 2: Write an Access Basic function that searches for a matching value in a form's [dynaset](#) and displays an error message if one is found. For example, if you have a form called "MyForm" containing a text field called "ID", you can create a function called "CheckDup" as follows:

```
Function CheckDup (CheckThis)
 On Error goto Err1:
 Dim D As Dynaset
 Set D = Forms!MyForm.Dynaset
 D.FindFirst "ID = '" & CheckThis & "'"
 ' NOTE: The symbols after the first "=" character in the
 ' line above are one single quote and one double quote.
 ' After the last "&" character there is one double
quote,
 ' one single quote, and one double quote.
 MsgBox "This is a duplicate record"
 Exit Function
Err1:
 Exit Function
 Resume Next
End Function
```

In this example, you then would type =CheckDup([ID]), where "ID" represents the value you entered in the text field, as the ID field's AfterUpdate property. When you press ENTER or TAB to move to a new control, Microsoft Access immediately searches the dynaset for duplicates of your ID value that may already appear.

## [References](#)

KBCategory:

KBSubcategory:

## FastTips: Access Basic and Macros Questions & Answers

Article Number: Q100845  
CREATED: 29-JUN-1993  
MODIFIED: 27-AUG-1993  
VERSION(S): 1.00 1.10

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0636: ACCESS BASIC AND MACROS QUESTIONS & ANSWERS  
=====

Revision Date: 5/93  
No Disk

The following information applies to Microsoft Access, version 1.0.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an |  
| Application Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY |  
| KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO |  
| THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A |  
| PARTICULAR PURPOSE. The user assumes the entire risk as to the |  
| accuracy and the use of this Application Note. This Application |  
| Note may be copied and distributed subject to the following |  
| conditions: 1) All text must be copied without modification and |  
| all pages must be included; 2) If software is included, all files |  
| on the disk(s) must be copied without modification [the MS-DOS(R) |  
| utility DISKCOPY is appropriate for this purpose]; 3) All |  
| components of this Application Note must be distributed together; |  
| and 4) This Application Note may not be distributed for profit. |  
| |  
| Copyright 1992-1993 Microsoft Corporation. All Rights Reserved. |  
| Microsoft and MS-DOS are registered trademarks and Windows |  
| is a trademark of Microsoft Corporation. |  
-----

Q. How can I run macros from [Access Basic](#)?

A. The following are three suggested ways to run macros from Access Basic:

- Create the [macro](#) using the macro editor and then call it from your Access Basic procedure. For example, if you create a macro called `Macro1`, you can call the macro from a procedure with the following command:

```
DoCmd RunMacro "macro1"
```

- Call the macro from the [Immediate window](#) using the same [syntax](#) as you would in Access Basic. Again, the syntax is as follows:

```
DoCmd RunMacro "macro1"
```

- Use `DoCmd` to run a macro action, rather than creating a macro specifically for that action. The basic format of `DoCmd` is as follows:

DoCmd <Action> <Argument1>, <Argument2>

For more information on DoCmd, RunMacro, and macro actions, please refer to the User's Guide and "An Introduction to Programming"; or, search Help on the keywords "DoCmd" and "Expressions in Macros and Actions".

2. Q. When should I use the [Variant data type](#)?
  - A. Use the Variant data type if you are unsure what type of data you are working with. When you use the Variant data type, Microsoft Access evaluates the data and then handles it appropriately. Using variants avoids the step of assigning data types to variables and manually converting data types in your programs. Further, when you store [table](#) data in variables, using variants eliminates complications with NULL data values.
3. Q. How do I comment out a macro action?
  - A. With the macro in [Design view](#), choose Conditions from the View menu. This adds the Condition [column](#) to the macro design. Type "false" (without the quotation marks) on the condition line for the macro action you want to comment out.
4. Q. How do I bypass the AutoExec startup macro when I open a [database](#)?
  - A. Hold down the SHIFT key when you open the database to prevent Microsoft Access from executing that database's AutoExec macro.

## [References](#)

KBCategory:

KBSubcategory:

## FastTips: Tables Questions & Answers

Article Number: Q100844  
CREATED: 29-JUN-1993  
MODIFIED: 27-AUG-1993  
VERSION(S): 1.00 1.10

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0635: DATABASE STRUCTURE QUESTIONS & ANSWERS  
=====

Revision Date: 5/93  
2 Pages, No Disk

The following information applies to Microsoft Access, version 1.0.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an |  
| Application Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY |  
| KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO |  
| THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A |  
| PARTICULAR PURPOSE. The user assumes the entire risk as to the |  
| accuracy and the use of this Application Note. This Application |  
| Note may be copied and distributed subject to the following |  
| conditions: 1) All text must be copied without modification and |  
| all pages must be included; 2) If software is included, all files |  
| on the disk(s) must be copied without modification [the MS-DOS(R) |  
| utility DISKCOPY is appropriate for this purpose]; 3) All |  
| components of this Application Note must be distributed together; |  
| and 4) This Application Note may not be distributed for profit. |  
| |  
| Copyright 1992-1993 Microsoft Corporation. All Rights Reserved. |  
| Microsoft and MS-DOS are registered trademarks and Windows |  
| is a trademark of Microsoft Corporation. |  
-----

Q. What foreign data formats does Microsoft Access support?

A. Microsoft Access can [link](#) to Btrieve(R), dBASE III PLUS(R), dBASE IV(R), and Paradox(R) versions 3.x data, including indexes.

Microsoft Access can import data from Btrieve, dBASE III PLUS, dBASE IV, FoxPro(R) versions 2.x, Lotus(R) 1-2-3(R) versions 2.x and 3.x, Lotus 1-2-3/W, Microsoft Excel versions 2.x and later for Windows, Paradox 3.x, and fixed-length and delimited ASCII text.

Microsoft Access also supports the Open Database Connectivity ([ODBC](#)) specification for connectivity to data on [database](#) servers. ODBC drivers for Microsoft [SQL](#) Server are included with Microsoft Access. Microsoft SQL Server ODBC drivers are the only drivers certified for Microsoft Access version 1.0.

2. Q. What are primary and foreign keys in a [relational database](#)?

A. Primary keys uniquely identify a specific [row](#) in a [table](#). They

are usually composed of a single field or column whose data is unique to each row of information (for example, a customer identification number, employee social security number, or order identification number). Primary keys also can be composed of more than one field. For example, the fields Company Identification and Contact Name could be combined to form a primary key for a table that contains information about companies that have more than one contact.

Foreign keys identify rows in one table that match the primary key of another table. For example, if the field Employee ID is the primary key in the Employee table, in the Orders table, Employee ID is the foreign key. The relationship between primary and foreign keys enables you to access related data in multiple tables.

3. Q. Which FieldSize setting should I select for the Number data type in order to use a Counter field of another table as the foreign key in the current table?
  - A. To use a Counter field of another table as the foreign key in the current table, select the Long Integer setting.
4. Q. How do you change the starting value of a Counter column to a number other than 1?
  - A. To set the starting value of a Counter column to a number other than 1, use the following four-step procedure:
    1. Create an additional single-column temporary table. Give the Number column the same name as the Counter column in the original table.
    2. Create a new row in the temporary table by inserting a value in the Number column. This value should be 1 less than the desired starting value for the original table.
    3. Create an Append query to append the single row from the temporary table to the original table.
    4. Once you append the single row, delete the temporary table. You can then delete the dummy row from the original table.

NOTE: Do not compact the database before you add the first row to the original table. If you do this, the counter value is reset to start at 1.
5. Q. Is data in tables stored in a sorted order? How can I view my data in sorted order?
  - A. Data in tables is not stored in sorted order. Data is stored in the order in which it was entered. To view data in sorted order, create a query, or a form based on a query, that uses the Sort field on the query grid. By default, if the table includes a primary key, the Datasheet view of the table will display the data sorted by the primary key. To view the data sorted by something other than the primary key alone, use a query, as previously described.

6. Q. How can I delete the primary key from a table?
- A. You can delete the primary key for a table by opening the Table Properties [dialog box](#) and deleting the entry for the Primary Key field. Please note that the primary key cannot be deleted from a table that is the primary table in a relationship.
7. Q. How can I create computed columns in tables?
- A. You can create computed columns or expressions with queries. Within the query, create a column that is defined as an [expression](#). In general, it is helpful to think of queries as virtual tables; you can use a query wherever you can use a table. If you're familiar with SQL terminology, this is very similar to creating a view. Unlike most implementations of views, however, Microsoft Access views can be updated even if they involve joins from different data sources, such as Paradox, Btrieve, or separate Microsoft Access data sources.
8. Q. Two of my tables have [Counter data type](#) columns in common. Why can't I define a relationship between the two tables based on these columns?
- A. You can define only one-to-one relationships between Counter [data type](#) columns. If the linking column on the "one" side is a counter, define the linking column on the "many" side as a long integer.

For more information on table relationships, see the "Microsoft Access User's Guide", Chapter 1, "Designing a Database," pages 13-18.

## [References](#)

KBCategory:

KBSubcategory:

## INF: Contents: MS Access 1.0 Disks 1-4 (1/14/93, 3.5-Inch)

Article Number: Q100472  
CREATED: 22-JUN-1993  
MODIFIED: 24-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0  
-----

### SUMMARY

---

This article lists the contents of Disks 1-4 shipped with Microsoft Access version 1.0 (release date 1/14/93, 5.25-inch, 1.2 MB).

The WBTRCALL.DLL file was removed from the original release of version 1.0. Because of this, the SETUP.EXE program was changed. No Microsoft Access functionality was changed for this release.

### MORE INFORMATION

---

#### DISK1 Contents

-----

|          |      |            |               |        |
|----------|------|------------|---------------|--------|
| 1        | WMF  | 4470       | 01-04-93      | 12:00a |
| 1E       | WMF  | 4022       | 01-04-93      | 12:00a |
| COMMDLG  | DL\$ | 46406      | 10-09-92      | 10:02a |
| COMMDLG  | DL_  | 89248      | 01-04-93      | 12:00a |
| CTL3D    | DL\$ | 7524       | 08-16-92      | 7:35p  |
| DDEML    | DL\$ | 20305      | 03-10-92      | 3:10a  |
| DETCMD   | DL_  | 24128      | 01-04-93      | 12:00a |
| MSACCESS | IN\$ | 519        | 01-04-93      | 12:00a |
| MSACCESS | RE\$ | 552        | 01-04-93      | 12:00a |
| MSAIN100 | DL\$ | 71625      | 01-04-93      | 12:00a |
| OLECLI   | DL\$ | 41083      | 10-09-92      | 10:02a |
| OLESVR   | DL\$ | 13129      | 10-09-92      | 10:02a |
| ORDENTRY | TX\$ | 2281       | 01-04-93      | 12:00a |
| ORDENTRY | HL\$ | 14224      | 01-04-93      | 12:00a |
| ORDENTRY | MD\$ | 200396     | 01-04-93      | 12:00a |
| PIM      | HL\$ | 8119       | 01-04-93      | 12:00a |
| SETUP    | EXE  | 21472      | 01-04-93      | 12:00a |
| SETUP    | INI  | 149        | 01-04-93      | 12:00a |
| SHARE    | EX\$ | 8112       | 10-09-92      | 10:02a |
| SHELL    | DL\$ | 23296      | 10-09-92      | 10:02a |
| SMALLB   | FO\$ | 9687       | 03-06-92      | 7:16p  |
| SMALLE   | FO\$ | 11236      | 03-06-92      | 7:16p  |
| SMALLF   | FO\$ | 9376       | 03-06-92      | 7:16p  |
| STFSETUP | EX_  | 477392     | 01-04-93      | 12:00a |
| STFSETUP | IN_  | 58041      | 01-04-93      | 12:00a |
| VER      | DL\$ | 5653       | 10-09-92      | 10:02a |
| VER      | DL_  | 9008       | 01-04-93      | 12:00a |
| WORKGRP  | IN_  | 1393       | 01-04-93      | 12:00a |
|          |      | 28 file(s) | 1183014 bytes |        |

#### DISK2 Contents

-----  
2 WMF 8022 01-04-93 12:00a  
2E WMF 6870 01-04-93 12:00a  
MSACCESS HL\$ 1189376 01-04-93 12:00a  
3 file(s) 1204268 bytes

DISK3 Contents

-----  
3 WMF 8950 01-04-93 12:00a  
3E WMF 8470 01-04-93 12:00a  
MSABC100 DL\$ 187802 01-04-93 12:00a  
MSAES100 DL\$ 16899 01-04-93 12:00a  
MSAFIN DL\$ 18553 01-04-93 12:00a  
WIZARD MD\$ 174297 01-04-93 12:00a  
MSACCESS EX\$ 790199 01-04-93 12:00a  
7 file(s) 1205170 bytes

DISK4 Contents

-----  
4 WMF 8246 01-04-93 12:00a  
4E WMF 7766 01-04-93 12:00a  
ANALYZER MD\$ 49740 01-04-93 12:00a  
NEWCUST DB\$ 1212 01-04-93 12:00a  
NWIND MD\$ 555226 01-04-93 12:00a  
PIM MD\$ 109267 01-04-93 12:00a  
SYSTEM MD\$ 98863 01-04-93 12:00a  
UTILITY MD\$ 47055 01-04-93 12:00a  
WINHELP EX\$ 147911 10-09-92 10:02a  
WINHELP HL\$ 28161 10-09-92 10:02a  
10 file(s) 1053447 bytes

[References](#)

## PRB: Order Entry Ship To Address Contains City Info

Article Number: Q100471  
CREATED: 22-JUN-1993  
MODIFIED: 22-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### SYMPTOMS

---

In the Order Entry sample application you are given the ability to print a basic invoice form. The form name is called "Orders". When printing the form the Ship To information will contain the city name twice and will not display the address line.

### CAUSE

---

This will occur only if you enter a Ship To address. There is an error in the ShipToInfo function in the Order Routines Module.

### RESOLUTION

---

Open the Order Entry Application in design mode. Interrupt the autoexec macro by holding down the shift key when starting the application.

In the database container select the module button. Highlight the Order Routines module and click on the design button. Press the 'F2' function key and the View Procedures dialog will appear. In the bottom half of this dialog is the Procedures list box. In this box scroll down to the ShipToInfo procedure and highlight it. Choose the OK button and the module window will now display the function.

Move the cursor down 12 line and you should be an the line that reads:

```
stAddress$ = ConvertNulls(DLookup("ShipCity"
```

Change this to:

```
stAddress$ = ConvertNulls(DLookup("ShipAddress"
```

leaving the remainder of the line intact.

Steps to Reproduce Behavior

-----

Open the Order Entry Application. From the menu choose View, Orders. highlight Add Orders and choose OK. Select any employee id from the combo box and choose OK. Select any Bill To customer from the combo box. In the Ship To side enter your name, address, city, state and zip. Select the printer icon on the bottom left corner of the form or choose File, Print from the menu. When the form prints you will see

the duplicate city information.

References:

[References](#)

## INF: ODBC Disk Contents (1/14/93, 5.25-Inch, MS Access 1.0)

Article Number: Q100470  
CREATED: 22-JUN-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0  
-----

### SUMMARY

---

The following lists the contents of the [ODBC](#) disk shipped with Microsoft Access version 1.0 (release date 1/14/93, 5.25-inch, 1.2 MB).

The WBTRCALL.DLL file was removed from the original release of Microsoft Access version 1.0. Because of this, the SETUP.EXE program was changed. No Microsoft Access functionality was changed for this release.

### MORE INFORMATION

---

#### DISK1 Contents

-----

|          |                     |        |                  |        |
|----------|---------------------|--------|------------------|--------|
| DBNMP3   | <a href="#">DLL</a> | 8241   | 10-22-92         | 3:06p  |
| INSTCAT  | <a href="#">SQL</a> | 93796  | 01-04-93         | 12:00a |
| MSCOMSTF | DLL                 | 74528  | 01-04-93         | 12:00a |
| MSDETSTF | DLL                 | 24544  | 01-04-93         | 12:00a |
| MSINSSTF | DLL                 | 65440  | 01-04-93         | 12:00a |
| MSSHLSTF | DLL                 | 14928  | 01-04-93         | 12:00a |
| MSUILSTF | DLL                 | 6144   | 01-04-93         | 12:00a |
| ODBC     | DLL                 | 44736  | 01-04-93         | 12:00a |
| ODBC     | INF                 | 1226   | 01-04-93         | 12:00a |
| ODBCADM  | EXE                 | 9136   | 01-04-93         | 12:00a |
| ODBCADM  | HLP                 | 4942   | 01-04-93         | 12:00a |
| ODBCINST | DLL                 | 6016   | 01-04-93         | 12:00a |
| SETUP    | EXE                 | 54464  | 01-04-93         | 12:00a |
| SETUPI   | DLL                 | 44032  | 01-04-93         | 12:00a |
| SQLSETUP | DLL                 | 11024  | 01-04-93         | 12:00a |
| SQLSRVR  | DLL                 | 135792 | 01-04-93         | 12:00a |
| VER      | DLL                 | 9008   | 10-22-92         | 3:06p  |
| 8        | WMF                 | 5846   | 01-04-93         | 12:00a |
| 8E       | WMF                 | 5366   | 01-04-93         | 12:00a |
| WFWDREV  | TXT                 | 2134   | 01-04-93         | 12:00a |
| WFWDREV  | EXE                 | 23581  | 01-04-93         | 12:00a |
| WFWNET   | DR_                 | 165600 | 01-04-93         | 12:00a |
| LANMAN21 | <DIR>               |        | 01-04-93         | 12:00a |
|          | 23 file(s)          |        | 810524 bytes     |        |
|          |                     |        | 68096 bytes free |        |

#### DISK2 Contents

-----

|              |          |        |
|--------------|----------|--------|
| NETWKSTA 500 | 01-04-93 | 12:00a |
| NETWKSTA 400 | 01-04-93 | 12:00a |
| NETWKSTA 330 | 01-04-93 | 12:00a |

[References](#)

## INF: Validation Rule to Require all Upper Case in a Field

Article Number: Q100469  
CREATED: 22-JUN-1993  
MODIFIED: 22-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1
- 

### Summary:

---

Microsoft Access does not have a setting to force all upper case to be entered in a field or control. However, a Validation Rule can be used to assure that all Capital Letters are entered in a field or control.

For example, to require that all upper case is entered in a field or control called [Last Name], use the following Validation Rule in the Table or Form.

```
StrComp(UCase([Last Name]),[Last Name],0) = 0
```

For more examples of Validation Expressions refer to page 652 in the Microsoft Access User's Guide.

References: Microsoft Access Language Reference, p.456

## **PRB: Installing Companion Files into nested subdirectories**

Article Number: Q100468  
CREATED: 22-JUN-1993  
MODIFIED: 22-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 and 1.1  
-----

### **SYMPTOMS**

---

When you try to install the Microsoft Access Companion Disk files into a nested subdirectory that does not exist (ie c:\xxx\yyy\zzz and the xxx directory or the yyy subdirectory does not exist) you will get the following messages during setup:

Unable to create directory

and four subsequent errors

Can't open output file: c:\xxx\yyy\zzz\msforms.mdb  
Can't open output file: c:\xxx\yyy\zzz\mssweets.mdb  
Can't open output file: c:\xxx\yyy\zzz\mscms.mdb  
Can't open output file: c:\xxx\yyy\zzz\msdirect.mdb

### **CAUSE**

---

This is a limitation of the install process (COMPRESS.EXE) that is shipped on the disk to expand the files.

### **RESOLUTION**

---

Create the directories first or install into a directory that the setup process only needs to create one directory.

### **MORE INFORMATION**

---

Steps to Reproduce Behavior

-----

(assuming the xxx directory does not exist)

If you type the following at the DOS prompt:

A:\install c:\xxx\yyy\zzz

You will see the following output to your screen:

```
.
creating c:\xxx\yyy\zzz
.
Unable to create directory
Decompressing Sample Applications...
.
msforms.mdb (file 1 of 4)
```

Can't open output file: c:\xxx\yyy\zzz\msforms.mdb  
mssweets.mdb (file 2 of 4)  
Can't open output file: c:\xxx\yyy\zzz\mssweets.mdb  
mscms.mdb (file 3 of 4)  
Can't open output file: c:\xxx\yyy\zzz\mscms.mdb  
msdirect.mdb (file 4 of 4)  
Can't open output file: c:\xxx\yyy\zzz\msdirect.mdb

## [References](#)

## INF: How to Determine Version of INSTCAT.SQL File on Server

Article Number: Q100467  
CREATED: 22-JUN-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Using the [SQL](#) Server [ODBC](#) driver with Microsoft Access requires that you run the INSTCAT.SQL file on the ODBC installation disks. This article describes the SQL stored procedure that you can use to [query](#) for the version number of the INSTCAT.SQL file that was run against the SQL server machine.

### MORE INFORMATION

---

The following [syntax](#) can be executed from the query window of SQL Server Administration Facility (SAF) for MS-DOS, OS/2, Microsoft Windows or any other operating system in which you can execute a stored procedure:

```
sp_server_info 500
```

The attribute\_value [field](#) of the returned [row](#) contains the version number. If an empty recordset is returned, it is likely that the INSTCAT.SQL file was not run on this server. The result will resemble the following:

| <u>attribute_id</u> | <u>attribute_name</u> | <u>attribute_value</u> |
|---------------------|-----------------------|------------------------|
| 500                 | SYS_SPROC_VERSION     | 01.00.2101             |

OS/2 is manufactured by International Business Machines Corporation, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

### [References](#)

## INF: Access 1.0 (rel. 1/14/93) Disks 4-6 Contents 5.25"

Article Number: Q100466  
CREATED: 22-JUN-1993  
MODIFIED: 22-JUN-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access Version(s) 1.0  
-----

### Summary:

---

The following lists the directory contents of diskettes 4-6 shipped with Microsoft Access version 1.0 release date: 1/14/93 (3.5 inch, 1.4 meg).

The WBTRCALL.DLL was removed from the original release of Microsoft Access 1.00. Due to this removal, the SETUP.EXE was changed. No functionality of Microsoft Access was changed for this release.

### More Information:

---

#### Contents of DISK5

-----  
5 WMF 4662 01-04-93 12:00a  
5E WMF 4150 01-04-93 12:00a  
CBTLIB4 DL\$ 11179 01-04-93 12:00a  
CUECARD EX\$ 60269 01-04-93 12:00a  
CUECARDS DL\$ 2047 01-04-93 12:00a  
CUECARDS LE\$ 887544 01-04-93 12:00a  
WFWNET DR\$ 151762 01-04-93 12:00a  
7 file(s) 1121613 bytes

#### Contents of DISK6

-----  
6 WMF 5846 01-04-93 12:00a  
6E WMF 5366 01-04-93 12:00a  
BTRVISAM DL\$ 57918 01-04-93 12:00a  
DBSISAM DL\$ 135919 01-04-93 12:00a  
GRAPH EX\$ 364877 10-09-92 10:02a  
MASJT100 DL\$....338672-01-04-93..12:00a  
MSAJU100 DL\$ 38945 01-04-93 12:00a  
MSGGRAPH HL\$ 110291 10-09-92 10:02a  
PDXISAM DL\$ 106830 01-04-93 12:00a  
REGEDIT EX\$ 18176 10-09-92 10:02a  
10 file(s) 1178552 bytes

#### Contents of DISK7

-----  
7 WMF 5846 01-04-93 12:00a  
7E WMF 5366 01-04-93 12:00a  
MSACCES2 HL\$ 1134560 01-04-93 12:00a  
PSSKB TX\$ 7951 01-04-93 12:00a  
README\_ TX\$ 10186 01-04-93 12:00a  
ERRATA TX\$ 9805 01-04-93 12:00a

6 file(s) 1173714 bytes

[References](#)

## INF: How to Prevent Users from Adding New Records to a Form

Article Number: Q100465  
CREATED: 22-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

The DefaultEditing [form](#) property, when set to AllowEdits, allows users to both modify existing records and add new records. Microsoft Access currently has no built-in mechanism to prevent users from adding new records.

This feature is under review and will be considered for inclusion in a future release of this product.

### MORE INFORMATION

---

The following information describes how to use an [Access Basic function](#) to prevent users from entering new employees in the Employees form in the sample [database](#) NWIND.MDB:

1. Add the following Access Basic function to a new or existing [module](#):

```
Function PreventNew ()
 Dim X
 On Error Resume Next
 X = Screen.ActiveForm.Bookmark
 If Err <> 0 Then DoCmd GoToRecord , , A_LAST
End Function
```

NOTE: There are two commas between GoToRecord and A\_LAST above.

2. Change the OnCurrent property of the Employees form to read as follows:

```
Form: Employees

 OnCurrent: =PreventNew()
```

RESULT: The user is placed in the last [record](#) of the Employees form when attempting to access or navigate to the new record.

How PreventNew() Works  
-----

Whenever the user navigates to a different record, the OnCurrent property executes the PreventNew() function. This function attempts to reference the bookmark property associated with the [current record](#). All records are assigned a [string](#), called a bookmark, that uniquely

identifies each record.

However, a new record that has not yet been saved has no bookmark associated with it. Any attempt to reference the bookmark of a new record generates an error message. If an error occurs, the PreventNew() function immediately performs a GoToRecord action to navigate to the last record in the form.

## [References](#)

## PRACC9304: Ordered Query on Attached Paradox Table

Article Number: Q100464  
CREATED: 22-JUN-1993  
MODIFIED: 07-JUL-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0
- 

### SYMPTOMS

---

An ordered [query](#) (a query in which one or more fields is sorted using the [SQL](#) ORDER BY clause) on an attached Paradox [table](#) with secondary indexes does not consistently select all the records that meet the designated [criteria](#).

### RESOLUTION

---

In Microsoft Access version 1.0, run the query as a [make-table query](#), without specifying a sort on secondary indexed fields. Then create and run an ordered [select query](#) on the new table.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem has been corrected in Microsoft Access version 1.1.

### MORE INFORMATION

---

This problem occurs only when the query is ordered by a secondary [index field](#).

A Paradox secondary index is an field, other than the [primary key](#), that is indexed. This is analogous to Microsoft Access tables that are indexed on fields other than the primary key.

How to Check for Secondary Indexes

-----

1. Attach a Paradox table to Microsoft Access.
2. Select the [attached table](#) from the [Database window](#) and choose Design. An error message appears, stating that some properties cannot be modified.
3. Choose Yes to open the table.
4. Select each [row](#) in the table and note its indexed property. The indexed property for secondary indexes is Yes.

How to Verify Your Query Results Are Consistent

-----  
Check for secondary indexes. If secondary indexes are found, complete the following steps:

1. Open the query in [Design view](#).
2. Remove all sort specifications based on secondary indexes.
3. Run the query and note the number of rows returned.
4. Enter the sort specifications for the secondary index fields.
5. Rerun the query.

If the number of rows returned from steps 3 and 5 are not the same, the query is not returning correct results.

Paradox is manufactured by Ansa Software, a Borland company, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

## INF: How to Disable the Close Option on the System Menu

Article Number: Q100463  
CREATED: 22-JUN-1993  
MODIFIED: 20-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

To prevent a user from manually closing a [form](#), you have two options:

1. Use the CancelEvent [macro](#) action with the OnClose [event](#) on a form.
2. Disable the Close option on the System menu, using a [function](#) on the OnOpen event of the form.

The second approach, which is described in this article, requires the use of Windows application programming interface (API) calls, but is more effective. It prevents the user from choosing the Close option and does not require special handling of the OnClose event when the Close action is invoked.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

NOTE: In this article, error trapping has been minimized.

### MORE INFORMATION

---

To disable the Close option on the System menu, create a [module](#) in your [database](#) containing the DisableCloseMenu() function, then call the function from the OnOpen event in your form.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new form called Form1.
3. Place a [command button](#) on the form.
4. Set the properties for the form and button as follows:

Object: Command Button

=====

ControlName: Button1

Caption: Close

OnPush: TestClose

Object: Form

=====

On Open: =DisableCloseMenu(Form)

5. Save and close the form.
6. Create the following macro and call it TestClose. This macro, along with the command button on the form, enables the user to close the form.

```

Macro Name Action
=====
TestClose Close

TestClose Actions

Close
 Object Type: Form
 Object Name: Form1

```

7. Create a new module called Module1. Enter the following code:

```

'*****
' Declarations section of the module
'*****
' NOTE: In the following sample code, an underscore (_) is used as a
' line-continuation character. Remove the underscore when re-creating
' this code in Access Basic.

' NOTE: You may have some Windows API functions defined in an existing
' Microsoft Access library; therefore, your declarations may be duplicates.
' If you receive the duplicate procedure name error message, remove or
' comment the declarations statement from your code.

```

```

Option Explicit
Const SC_CLOSE = &HF060
Const MF_GRAYED = 1
Declare Function GetSystemMenu Lib "user" (ByVal hWnd%, ByVal flag%)
Declare Function EnableMenuItem Lib "user" (ByVal hMenu%,ByVal nItem%,_
 ByVal nAction%)
'=====
'The following function retrieves the handle to the System
'menu and dims the Close option.
'=====
Function DisableCloseMenu (frm As Form)
 Dim hSysMenu As Integer
 Dim D As Integer
 hSysMenu = GetSystemMenu(frm.hWnd, 0)
 D = EnableMenuItem(hSysMenu, SC_CLOSE, MF_GRAYED)
End Function

```

NOTE: When you are determining the position of a menu command, separator bars count as commands. For example, to get to the Import menu command on the File menu in the Database window, use the arguments 0 (for the File menu) and 7 (for the Import command). Positions 3 and 6 correspond to the separator bars on the File menu.

8. Give Form1 the focus. Note that double-clicking the System menu does not close the form and that the Close option on the form's

System menu is unavailable (dimmed). Choose the Close button to close the form.

#### REFERENCES

=====

For more information about disabling menu items, [query](#) on the following words here in the Microsoft Knowledge Base:

menu and gray and getmenu

[References](#)

## PRB: Using CTRL+SHIFT+HOME to Move to Main Form

Article Number: Q100462  
CREATED: 22-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0
- 

### SYMPTOMS

---

Use of the CTRL+SHIFT+HOME keystrokes, to move to the first field of a main form from the subform, will be disabled if you have moved to a control on your subform with a mouse click.

### CAUSE

---

This behavior is by design. When you select the control with the mouse within the subform, you are changing into edit mode, which gives you a different keyboard definition. Using the TAB key does not change you into edit mode.

### RESOLUTION

---

Choose the record selector button in the subform to advance to the next record. After clicking on the record selector button you will again be able to use CTRL+SHIFT+HOME to move to the first field of the main form.

### MORE INFORMATION

---

The CTRL+SHIFT+HOME keyboard shortcut allows you to move to the first field of the main form from a subform.

If you tab into the subform, instead of using the mouse to select a control in the subform, CTRL+SHIFT+HOME will work properly.

The following steps demonstrate use of CTRL+SHIFT+HOME.

#### Steps to Reproduce Behavior

-----

1. Open the sample database Northwind Traders (NWIND.MDB).
2. Open the Categories form. The Category ID field has the focus.
3. Press the TAB key 4-times, thereby selecting the Product ID field in the subform.
4. Press CTRL+SHIFT+HOME to return focus the first field on the main form, the Category ID field.
5. Using the mouse, select the Product ID field on the subform.
6. Press the CTRL+SHIFT+HOME keys.

You will notice that focus did not return to the first field on the main form.

7. Use the record selector to select one of the records in the subform. Now when you press CTRL+SHIFT+HOME, focus will return to the Category ID field on the main form.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 14, "Adding and Editing Data," pages 352-353.

[References](#)

## PRACC9211: Cannot Overwrite Corrupt Paradox/dBASE Files

Article Number: Q100461  
CREATED: 22-JUN-1993  
MODIFIED: 25-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1  
-----

### SYMPTOMS

---

Microsoft Access does not allow you to export a Paradox or dBASE table and overwrite an existing Paradox or dBASE file if the destination file has been corrupted. If you try to do this, the following error message appears:

```
<path\filename> This file already exists.
Replace existing file?
```

If you choose Yes, the following error message appears:

```
Table '<tablename>' already exists.
```

This error occurs only with dBASE and Paradox files. Other database programs allow you to overwrite existing files when you export.

### CAUSE

---

Microsoft Access attempts to open the file to verify, rather than to overwrite, its contents. Microsoft Access then reports the file as corrupted and returns the above error messages.

### STATUS

---

This behavior is by design.

### MORE INFORMATION

---

Steps to Reproduce Problem  
-----

1. Copy your CONFIG.SYS file to CONFIG.DB. (CONFIG.DB simulates a corrupt Paradox database.)
2. Export any Microsoft Access table to Paradox 3.x.
3. Specify CONFIG.DB as the output file and choose the OK button.
4. When the "Replace existing file?" message appears, choose Yes.

Result: The error message "Table 'CONFIG#DB' already exists" appears.

Paradox is manufactured by Ansa Software, a Borland company, and dBASE

is manufactured by Borland International, Inc., vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

## [References](#)

## INF: Importing, Exporting, and Attaching FoxPro 2.0, 2.5 Data

Article Number: Q100177  
CREATED: 16-JUN-1993  
MODIFIED: 21-JUN-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Version 1.1
- 

### Summary:

---

In Microsoft Access version 1.1, you can import, export, and attach to data in FoxPro versions 2.0 and 2.5.

### More Information:

---

If you select either Import or Attach Table from the File menu, you can choose FoxPro 2.0 or FoxPro 2.5 from the Data Source [list box](#).

When you attach to a FoxPro file, you can also associate the appropriate FoxPro [index](#) (.IDX or .CDX file). The procedure to import, export, and attach files is the same for FoxPro files as it is for dBASE and Paradox files.

dBASE is manufactured by Borland International, Inc. and Paradox is manufactured by Ansa Software, a Borland company, both vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

### References:

"Microsoft Access User's Guide," version 1.0, chapter 4, "Importing, Exporting, and Attaching," pages 60-65.

[References](#)

## INF: Upper Limit for QueryTimeout and LoginTimeout Parameters

Article Number: Q100176  
CREATED: 16-JUN-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

This article describes the maximum limit allowed for the QueryTimeout and LoginTimeout parameters.

### MORE INFORMATION

---

As the README.TXT file for Microsoft Access 1.1 states, the two settings in the [odbc] section of the MSACCESS.INI file are as follows:

| Entry        | Value | Effect                                                                                                                                                      |
|--------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -----        | ----- | -----                                                                                                                                                       |
| QueryTimeout | S     | Wait S seconds for queries sent to <b>ODBC</b> , then stop trying to process the <b>query</b> results (for asynchronous queries only). Default: 60 seconds. |
| LoginTimeout | S     | Wait S seconds for ODBC login response, and then stop trying to connect to a server. Default: 20 seconds.                                                   |

The README.TXT fails to state the limit at which these values can be set.

Since the values for QueryTimeout and LoginTimeout are unsigned 4-byte values (long integers), the maximum value is +/- 2,147,483,648. If the value is set to zero (0), the time-out is disabled and the query/login never times out.

### [References](#)

## PRACC9305: GP Fault When Allocating Huge Arrays

Article Number: Q100175  
CREATED: 16-JUN-1993  
MODIFIED: 02-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

When you try to allocate a huge array, you may receive a general protection (GP) fault. The following error message appears:

An error has occurred in your application. If you choose Ignore, you should save your work in a new file. If you choose Close, your application will terminate.

### CAUSE

---

If you have allocated an array element that includes a [string](#) or variant [data type](#) and the size of the array exceeds 64K, you may receive the error message described above, if the following [criteria](#) are met:

- The array must be huge and span two or more data segments.
- Each array element must contain at least one dynamic string or variant.
- The number and size of the array elements must cause the last segment of data to be exactly the same size as the first segment of data.

### RESOLUTION

---

Declare the array so that the last data segment contains a different number of elements from the first data segment.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

[Access Basic](#) has the ability to define huge arrays, larger than the normal 64K data segment. Any element or [index](#) of a huge array may not extend across the 64K-segment boundary. If all elements of the array cannot be allocated within the 64K segment, only the elements of the array that fit evenly in the segment are allocated. Microsoft Access then allocates another 64K segment to the array and continues to fill

the new segment with the remaining elements.

If the size of the data type defined in the array is a power of 2, the 64K segment is completely full. If the size is not a power of 2, the segment contains wasted space, since Microsoft Access cannot place a portion of one element in one segment and the rest of that element in the next segment.

#### Example

-----

```
Type Foosball
 A(3000) As double
 d As String
End Type
```

The size of this structure is 24,012 bytes (3000 \* 8 bytes/double + 12 bytes/string). If you declare this array with three elements, you have two data segments, as follows:

| Data Segment #1  | Data Segment #2  |
|------------------|------------------|
| -----            | -----            |
| Free space 17512 | Free space 41524 |
| Array(1) 24012   | Array(3) 24012   |
| Array(2) 24012   |                  |

In Microsoft Access, you may receive a GP fault if the number of elements in the first segment is equal to the number of elements in the last segment and the data structure used to define the array is made up of either a string or variant data type. If you define the data structure above with six elements, your memory map looks like the following:

| Data Segment #1  | Data Segment #2  | Data Segment #3  |
|------------------|------------------|------------------|
| -----            | -----            | -----            |
| Free space 17512 | Free space 17512 | Free space 17512 |
| Array(1) 24012   | Array(3) 24012   | Array(5) 24012   |
| Array(2) 24012   | Array(4) 24012   | Array(6) 24012   |

The actual dimension statement looks like the following:

```
Function Sample()
 Redim Array(6) As Foosball
End Function
```

To work around this problem, declare the array so that the last data segment does not contain the same number of elements as the first data segment. In addition to the example above, you may choose to define the array with seven elements instead of six. This causes Microsoft Access to allocate four data segments, in which the first data segment contains two elements of the array and the last data segment contains only one element.

This example is very easy to duplicate with large data structures, but is more difficult to duplicate with smaller data structures. With an element size of 24,012 bytes, 4, 6, or 8 elements may cause a problem. With a data size of 1K, numbers of elements that duplicate this

problem are 128, 192, 256, and so forth.

[References](#)

## INF: How to Simulate Changing the Record Source of a Form

Article Number: Q100174  
CREATED: 16-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Although you can't change the actual RecordSource property of a [form](#), you can use the Macro Action Apply Filter to simulate the effect.

### MORE INFORMATION

---

The following steps demonstrate the [method](#) to simulate changing the [record](#) source.

1. Open NWIND.MDB and make a copy of the EmployeesTable and save the copy as Employees2. Delete 5 records from the Employees2 Table.
2. Create a Macro that has the following action and save it as Employees2 Macro.

Employees2 Macro Action

-----  
ApplyFilter

Filter Name: Select \* from Employees2;  
Where Condition:

4. Open the Employees Form in Design View and add a [command button](#) to the Employees Form and set the OnPush property of the command button to Employees2 Macro.
5. Open the form in [Form view](#) and go to the last record of the form. Note how many records are in Employees. Next, push the command button and go to the last record of the form. Notice that there are now 5 less records in the form.

If you apply this method to your own tables and the tables do not have the same fields, you will need to set the visible property of the invalid controls to no to prevent #Name? errors.

References:

Microsoft Access User's Guide, pages 375-382

[References](#)

## PRB: Unexpected Error from External Database Driver [20]

Article Number: Q100173  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access Version(s) 1.1  
-----

### SYMPTOMS

---

When you try to import, export or attach to a Btrieve [table](#) using the Network Loadable Module (NLM) version of Btrieve, you receive the following error;

Unexpected error from external [database](#) driver [20]

### CAUSE

---

This error message is actually a Novell error message that is returned when you try to access Btrieve data with Microsoft Access and the BREQUEST.EXE program was not executed prior to loading Windows. The Xtrieve manual lists this error with the following description:

20 Record Manager inactive  
You made a request before starting the Record Manager. Restart the Record Manager."

### RESOLUTION

---

Create a batch file to execute the BREQUEST.EXE program before starting Windows.

### MORE INFORMATION

---

Refer to the Novell Xtrieve manual, Appendix A for a complete list of error messages and status codes.

[References](#)

## **PRB: GPF with Diamond Viper VESA Local Bus Video Driver V1.01**

**Article Number:** Q100172  
**CREATED:** 16-JUN-1993  
**MODIFIED:** 25-JUN-1993  
**VERSION(S):** 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### **SYMPTOMS**

---

When you are using version 1.01 of the Diamond Viper VESA local bus video driver for Microsoft Windows version 3.1, you receive a general protection (GP) fault when you attempt to run Microsoft Access. The GP fault occurs immediately after you try to open a [database](#).

### **CAUSE**

---

This problem is caused by a fault in version 1.01 of the Diamond Viper VESA local bus video driver for Windows 3.1.

### **RESOLUTION**

---

According to Diamond Computer Systems, version 1.02 of the driver corrects this problem. To obtain this driver, call Diamond Computer Systems Technical Support or download the VPR102.EXE file by modem from the Diamond Computer Systems bulletin board service (BBS) at (408) 730-1100.

The Diamond Viper VESA local bus video driver is manufactured by Diamond Computer Systems, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

### **[References](#)**

## INF: Install Process of the Companion Disk

Article Number: Q100171  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access Version(s) 1.0 1.1  
-----

### Summary:

---

This article shows the entire install process (defaults only) of the Microsoft Access Companion Disk setup process.

### More Information:

---

```
B:\>install

*
* Microsoft Access Companion Disk *
* Copyright (c) Microsoft Corporation 1993 *
* May 10, 1993 - Edition 1.0 *
*

*
* This batch program installs the companion disk to your C:\ACCESS *
* directory by default. If you wish to override this default *
* installation path, enter a new installation path on the command *
* line as shown below. *
*
* REQUIREMENTS: 3.8 MB of free hard disk space *
*
* USAGE: INSTALL [drive]:\[path] *
*
* Where [drive] is the drive letter and [path] is the directory *
* into which the files will be installed. *
*
* Example: INSTALL C:\ACCESS *
*

.
Install into default directory (Yes/No/Cancel)? [ync]y
.
creating c:\access
.
Decompressing Sample Applications...
.
msforms.mdb (file 1 of 4)
mssweets.mdb (file 2 of 4)
mscms.mdb (file 3 of 4)
msdirect.mdb (file 4 of 4)
.

*
```

```
*
*
* INSTALLATION COMPLETE
*
* Please load Microsoft Access to view your sample database files.
*
* MSFORMS.MDB sample data entry forms
* MSSWEETS.MDB ... sample database for small businesses
* MSCMS.MDB sample Contact Management application
* MSDIRECT.MDB ... directory of Access products & services
*
* The sample databases contained on this 'companion disk' have been
* provided purely as an aid to help users learn more about Microsoft
* Access and are therefore provided 'as is'. Microsoft is unable to
* provide support for these samples due to the nature of their design.
*

B:\>
```

Note due to the restrictions of the KB width some spaces were removed in the psuedo dialog boxes and thus the one missing astrisk because of the last line being so long.

[References](#)

## INF: Disk image of Companion Disk

Article Number: Q100170  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access Version(s) 1.0 1.1  
-----

### Summary:

---

This is the disk image of the Companion Disk that was sent out to all registered Microsoft Access v1.0 users at the release of Microsoft Access Version 1.1.

### More Information:

---

|           |      |               |          |       |
|-----------|------|---------------|----------|-------|
| COMPRESS  | EXE  | 25563         | 03-02-90 | 1:13p |
| INSTALL   | BAT  | 4677          | 05-11-93 | 7:48a |
| MSCMS     | MD\$ | 622969        | 05-10-93 | 6:34p |
| MSDIRECT  | MD\$ | 141957        | 05-10-93 | 1:47p |
| MSFORMS   | MD\$ | 108739        | 05-11-93 | 7:22a |
| MSSWEETS  | MD\$ | 186343        | 05-10-93 | 1:51p |
| YNC       | EXE  | 13023         | 03-15-91 | 2:42p |
| 7 file(s) |      | 1103271 bytes |          |       |

This disk can also be obtain from CompuServe in the MSACCESS Access forum Library 14, file name COMPAN.ZIP.

Note this disk only went out on 1.44M 3.5in disks however it may be downloaded and fit on a 1.2M 5.25in disk.

References:

[References](#)

## INF: How NWIND.MDB Employee Field Photographs Were Created

Article Number: Q100169  
CREATED: 16-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Below is a synopsis of how the photos for the Photo field of the Employees table in the Northwind Traders sample database (NWIND.MDB) were prepared and processed.

1. Photographs of the subjects were taken with a camera and developed as prints.
2. The prints were scanned on a color scanner connected to an Apple Macintosh.
3. The resulting files from the Macintosh (in .TIFF file format) were transferred to an IBM PC/AT compatible machine (to .BMP file format).
4. The images were converted from 256-color to 16-color (dithered) so that pictures would display correctly in 16-color mode, and also because 16-color images display faster than 256-color images.
5. The images were then added as embedded objects in the Employees table of the Northwind Traders database, following the instructions given in the "User's Guide," Chapter 13, "Using Pictures, Graphs, and Other Objects," pages 314-339.

### MORE INFORMATION

---

If you plan to use this process yourself, steps can be eliminated if you use a scanner directly connected to an IBM PC/AT compatible machine since the image transfer from the Macintosh (step 3) would not be necessary and step 4 can often be handled by the scanner software.

Software helpful for processing graphic images:

Apple Macintosh  
-----

Adobe "Photoshop"

IBM PC/AT Compatibles  
-----

"PaintShop" (Shareware - can be used for image file conversions.)  
"WinGif" (Shareware - can be used for image file conversions.)  
Microsoft "Paintbrush" (Great for touching up stray pixels.)

References:

"User's Guide," version 1.0, Chapter 13, "Using Pictures, Graphs, and Other Objects," pages 314-339

[References](#)

## PRACC9305: GP Fault When Using Erase Statement on String Array

Article Number: Q100168  
CREATED: 16-JUN-1993  
MODIFIED: 02-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

When you use an Erase statement on a string array, you receive a general protection (GP) fault. The following error message appears:

An error has occurred in your application. If you choose Ignore, you should save your work in a new file. If you choose Close, your application will terminate.

### CAUSE

---

You are trying to use the Erase statement on a string array contained in a user-defined data type.

### RESOLUTION

---

Use a loop structure to initialize the string array in the data type.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

Steps to Reproduce Problem

-----

The following code generates the error message described above in the "Symptoms" section:

```
Type Foosball
 X(1) As String
End Type

Sub Main ()
 Dim Arrays As Foosball
 Erase Arrays.X
End Sub
```

To work around this problem, use the following loop structure instead of the Erase statement:

```
Sub Main ()
 Dim Arrays As Foosball
 For i = 1 To UBound(Arrays.X)
 Arrays.X(i) = ""
 Next i
End Sub
```

## [References](#)

## INF: How to Pass Information to Microsoft Access Using DDE

Article Number: Q100167  
CREATED: 16-JUN-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Microsoft Access does not allow you to poke data into a [table](#) through a dynamic data exchange ([DDE](#)) channel. However, using a [function](#) with parameters in a DDE channel to the [SQL](#) topic, you can pass information to an [Access Basic](#) function. The data can then be processed by the function and added to a table.

### MORE INFORMATION

---

The following example explains how to add a new customer to the Customers table in NWIND.MDB, using data from Microsoft Excel.

To pass data to Microsoft Access for processing, use the following steps:

1. Create a table called None, with one [field](#) of any [data type](#).
2. Open a [module](#) and create a function with the parameters that you want to pass:

```
Function AddNewCust$ (CustomerID$, CompanyName$)
 Dim MyDB As Database, MyTable As Table
 Set MyDB = CurrentDB()

 Set MyTable = MyDB.OpenTable("Customers") ' Open table.

 MyTable.Index = "PrimaryKey" ' Set index.
 MyTable.AddNew ' Prepare new record.
 MyTable("Customer ID") = CustomerID$ ' Set record key.
 MyTable("Company Name") = CompanyName$ ' Set company name.
 MyTable.Update ' Save changes.
 MyTable.Close ' Close table.
End Function
```

3. In another application, initiate to MSACCESS with a [query](#) that refers to the table called None and selects the function that you created. For example, using Microsoft Excel, the following [macro](#) is appropriate:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Microsoft Excel.

```
AddNewCustomer
```

```
chan=INITIATE("MSACCESS","NWIND.MDB;SQL SELECT_
 AddNewCust$("JOHNJ","John's Place") FROM None; ")
=TERMINATE(chan)
=RETURN()
```

When you run this command macro, the parameters are passed to Microsoft Access and the function updates the Customer table while you are initiating to the table called None.

#### REFERENCES

=====

"Microsoft Access Language Reference," version 1.0, pages 118-124

README.TXT in Microsoft Access version 1.0

[References](#)

## INF: Implementing Counter Values on Attached SQL Server Tables

Article Number: Q100166  
CREATED: 16-JUN-1993  
MODIFIED: 16-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

When you create a [table](#) in Microsoft Access and port it to [SQL](#) Server, some tasks are difficult to implement because SQL Server does not include a [Counter data type](#). This article discusses how to maintain the counter functionality of an attached SQL Server table.

### MORE INFORMATION

---

With a multiuser system, you must take special precautions. Assume that the Counter [column](#) in SQL Server is called ID and that ID is the unique [index](#) for this table. In most cases, users try to simulate counters with a SQL Server OnInsert trigger that computes the maximum value and adds 1. However, Microsoft Access uses that index (bookmark) to point to the [record](#) and is not aware that the trigger is changing the index value. Thus, Microsoft Access loses track of that record until the next time the table is queried.

Since SQL Server triggers do not work and SQL Server does not have a unique Counter [data type](#) as does Microsoft Access, you must use an [Access Basic function](#) to [control](#) the counter value, as in the procedure below:

1. In a single-column, single-[row](#) Microsoft Access table, store a value that represents the next highest counter value.

NOTE: You must store the table in Microsoft Access, since the OpenTable action does not work on an attached SQL Server table and, therefore, cannot open the table exclusively. For more information on the OpenTable action, refer to the "Microsoft Access Language Reference" manual.

2. In the data entry [form](#), apply a function to the BeforeUpdate [event](#) so that the function completes the following tasks:
  - Opens the table that contains the counter value
  - Grabs the value for the [current record](#)
  - Adds 1 to the value in the table
  - Closes the table

Each user must have [exclusive](#) access to the table to ensure that no duplicate values are recorded. You must do error checking when trying to open the table exclusively because another user may have the table open at the same time. If this occurs, the program loops and retries several times before alerting the first user.

3. Once the value has been retrieved, set the ID of the current record equal to that value and commit the record to the SQL Server [database](#).

#### REFERENCES

=====

For more information about and an example of creating a unique custom counter, [query](#) on the following words in the Microsoft Knowledge Base:

table and counter and custom and unique

#### [References](#)

## INF: How To Use The CTL3D.DLL In Your Access Database

Article Number: Q100165  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access Version(s) 1.0 1.1  
-----

### Summary:

---

This article describes how to get the 3-D look in your Access [database](#) provided by the CTL3D.DLL. Calling the CTL3D.DLL APIs will give your message boxes and other dialogs that you create in your database that 3-D look.

### More Information:

---

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

You can use CTL3D.DLL to add 3-D effects to the common dialog boxes and message boxes for an Access Basic application. CTL3D.DLL does not currently work for the controls on an Access Basic [form](#). To use CTL3D.DLL to enhance the common dialog boxes and message boxes from Access Basic, follow the steps below.

You may have some Windows API functions defined in an existing Microsoft Access library, therefore your declarations may be duplicates. If you receive the duplicate procedure name error, remove or comment the declare statement from your code.

```
' Note: Each command should be entered on one line. The line
' continuation character '_' improves the readability of the code.
'-----
```

```
'GLOBAL DECLARATIONS SECTION
'-----
```

```
Option Explicit
```

```
Declare Function GetModuleHandle% Lib "Kernel" (ByVal lpModuleName$)
Declare Function Ctl3dAutoSubClass% Lib "Ctl3D.DLL" (ByVal hInst%)
Declare Function Ctl3dRegister% Lib "Ctl3D.DLL" (ByVal hInst%)
Declare Function Ctl3dUnRegister% Lib "Ctl3D.DLL" (ByVal hInst%)
```

```
'-----
' The following function needs to be called when your application
' starts.
' This will register Access and subclass dialogs and message boxes
' with the 3D look.
'-----
```

```
Function Init ()
 Dim Inst%
```

```
Dim ret
Inst% = GetModuleHandle("MSACCESS.EXE")
ret = Ctl3dRegister(Inst%)
ret = Ctl3dAutoSubClass(Inst%)
End Function
```

```
'-----
' The following function needs to be called when your application
' quits.
'-----
```

```
Function ShutDown ()
Dim Inst%
Dim ret
Inst% = GetModuleHandle("MSACCESS.EXE")
ret = Ctl3dUnRegister(Inst%)
End Function
```

After running the Init() function from the Immediate window, an AutoExec macro or a RunCode macro action you can see the 3-D effects by running a macro or function that displays a message box.

```

Remember to run the ShutDown function before quitting Access

```

References:

[References](#)

## INF: How to Call Functions Using a String Variable

Article Number: Q100164  
CREATED: 16-JUN-1993  
MODIFIED: 14-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

This article describes how [Access Basic](#) or user-defined functions can be called when the [function](#) name is stored in a [string variable](#). This [method](#) provides a functionality similar to that of pointers to functions in other programming languages.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

### MORE INFORMATION

---

Calling a function in Access Basic when the function name is stored in a variable allows significant programming flexibility. This method is described in the following procedure:

1. Store the function name and parameters in a string variable.
2. Use the Eval() function in Access Basic to evaluate the string. The process of evaluating the variable calls the function.
3. Open a new [module](#) or a previously created module and enter the following code:

```
' NOTE: In the following sample code an underscore (_) is used as
' a line-continuation character. Remove the underscore when
' re-creating this code in Access Basic.
'
'-----
'GLOBAL DECLARATIONS SECTION
'-----
Option Explicit

'-----
' The CallMyArray() function creates an array of strings, then
' loops, using the Eval() function, to call each element of the array.
'-----
Function CallMyArray ()
 Dim MyArray$()

 For i = 0 To 2
 ReDim Preserve MyArray$(i)
 MyArray$(i) = "MyFunc" & i & "(" & i & ")"
```

```
Next i

For i = 0 To 2
 x = Eval(MyArray(i))
Next i
End Function
```

```
'-----
' The first function called by CallMyArray().
'-----
```

```
Function MyFunc0 (nParam)
 MsgBox "This is function: " & nParam
End Function
```

```
'-----
' The second function called by CallMyArray().
'-----
```

```
Function MyFunc1 (nParam)
 MsgBox "This is function: " & nParam
End Function
```

```
'-----
' The third function called by CallMyArray().
'-----
```

```
Function MyFunc2 (nParam)
 MsgBox "This is function: " & nParam
End Function
```

#### REFERENCES

=====

"Introduction to Programming," version 1.0, pages 7-14

[References](#)

## INF: Expanded Size of the Companion files

Article Number: Q100163  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access Version(s) 1.0 1.1
- 

### Summary:

---

This article discusses the files sizes of the Microsoft Access Companion disk databases after the install process.

### More Information:

---

By default (if you don't specify the directory as a parameter) the Companion disk will install the files into a directory called 'C:\ACCESS'. Note it will prompt you for an OK before proceeding. Once those file are expanded (and before they are used) the file sizes should be as follows:

|          |     |           |               |       |
|----------|-----|-----------|---------------|-------|
| MSCMS    | MDB | 2490368   | 05-10-93      | 6:34p |
| MSDIRECT | MDB | 557056    | 05-10-93      | 1:47p |
| MSFORMS  | MDB | 294912    | 05-11-93      | 7:22a |
| MSSWEETS | MDB | 458752    | 05-10-93      | 1:51p |
|          |     | 6 file(s) | 3801088 bytes |       |

### [References](#)

## **PRB: Companion Disk install and Out of Disk Space**

Article Number: Q100162  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access Version(s) 1.0 1.1
- 

### **SYMPTOMS**

---

You will get the following error message if you run out of disk space during the install of the Microsoft Access Companion Disk files.

Out of space. Insert new disk and press RETURN:

Problem is you cannot insert a new hard disk. Also even if you were installing under windows and you went and cleaned some disk space off the error message continues -- it doesn't re-check disk space on the hard drive.

### **CAUSE**

---

This is a limitation of the install process (COMPRESS.EXE) provided with the Companion disk.

### **RESOLUTION**

---

1. Press Ctl+C to break out of the batch job.
2. Delete any files that may have already been copied over (MSCMS.MDB, MSDIRECT.MDB, MSFORMS.MDB, MSSWEETS.MDB)
3. Free up more than 3.8Megs of disk space
4. Run install again

### **MORE INFORMATION**

---

Steps to Reproduce Behavior

- 
1. Create a RAM Drive or Find a drive with less than 3.0Megs of disk space free.
  2. Install into a directory on that drive
  3. When the drive fills up and the setup process hasn't completed (i.e. it can't expand all 4 files) you will get the error.

[References](#)

## INF: Contents: MS Access 1.0 Disks 4-6 (1/14/93, 3.5-Inch)

Article Number: Q100161  
CREATED: 16-JUN-1993  
MODIFIED: 24-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0  
-----

### SUMMARY

---

This article lists the contents of Disks 4-6 shipped with Microsoft Access version 1.0 (release date 1/14/93, 3.5-inch, 1.4 MB).

The WBTRCALL.DLL file was removed from the original release of version 1.0. Because of this, the SETUP.EXE program was changed. No Microsoft Access functionality was changed for this release.

### MORE INFORMATION

---

#### DISK4 Contents

-----

|          |      |        |          |        |
|----------|------|--------|----------|--------|
| 4        | WMF  | 8246   | 01-04-93 | 12:00a |
| 4E       | WMF  | 7766   | 01-04-93 | 12:00a |
| ANALYZER | MD\$ | 49740  | 01-04-93 | 12:00a |
| MSAJT100 | DL\$ | 338672 | 01-04-93 | 12:00a |
| NEWCUST  | DB\$ | 1212   | 01-04-93 | 12:00a |
| NWIND    | MD\$ | 555226 | 01-04-93 | 12:00a |
| PIM      | MD\$ | 109267 | 01-04-93 | 12:00a |
| SYSTEM   | MD\$ | 98863  | 01-04-93 | 12:00a |
| UTILITY  | MD\$ | 47055  | 01-04-93 | 12:00a |
| WINHELP  | EX\$ | 147911 | 10-09-92 | 10:02a |
| WINHELP  | HL\$ | 28161  | 10-09-92 | 10:02a |

11 file(s) 1392119 bytes

#### DISK5 Contents

-----

|          |      |        |          |        |
|----------|------|--------|----------|--------|
| 5        | WMF  | 4662   | 01-04-93 | 12:00a |
| 5E       | WMF  | 4150   | 01-04-93 | 12:00a |
| CBTLIB4  | DL\$ | 11179  | 01-04-93 | 12:00a |
| CUECARD  | EX\$ | 60269  | 01-04-93 | 12:00a |
| CUECARDS | DL\$ | 2047   | 01-04-93 | 12:00a |
| CUECARDS | LE\$ | 887544 | 01-04-93 | 12:00a |
| GRAPH    | EX\$ | 364877 | 10-09-92 | 10:02a |
| MSGGRAPH | HL\$ | 110291 | 10-09-92 | 10:02a |

8 file(s) 1445019 bytes

#### DISK6 Contents

-----

|    |     |      |          |        |
|----|-----|------|----------|--------|
| 6  | WMF | 5846 | 01-04-93 | 12:00a |
| 6E | WMF | 5366 | 01-04-93 | 12:00a |

|          |      |           |               |        |
|----------|------|-----------|---------------|--------|
| MSACCES2 | HL\$ | 1134560   | 01-04-93      | 12:00a |
| PSSKB    | TX\$ | 7951      | 01-04-93      | 12:00a |
| README_  | TX\$ | 10186     | 01-04-93      | 12:00a |
| ERRATA   | TX\$ | 9805      | 01-04-93      | 12:00a |
| WFUNET   | DR\$ | 151762    | 01-04-93      | 12:00a |
|          |      | 7 file(s) | 1325476 bytes |        |

## [References](#)

## PRB: System Error When Sharing Database on LAN Man/Workgroups

Article Number: Q100160  
CREATED: 16-JUN-1993  
MODIFIED: 12-JUL-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### **SYMPTOMS**

---

If you try to save a Microsoft Access database larger than 2000 bytes (approximately 2 MB) on a Microsoft LAN Manager server with a boot drive that uses the file allocation table (FAT) format, the following error message appears:

System error... Network error on drive <x>

This error occurs only on Microsoft Windows for Workgroups workstations if you attempt to open a database already in use by a Microsoft LAN Manager workstation or another Windows for Workgroups workstation.

### **CAUSE**

---

The Microsoft LAN Manager server responds incorrectly to a request by Windows for Workgroups to share the file. The server responds with a message that is too long for Windows for Workgroups to process correctly, resulting in an error from the NetBIOS.

### **RESOLUTION**

---

The following are two ways to work around this problem:

- Change the format of the Microsoft LAN Manager server boot drive from FAT to high-performance file system (HPFS), then place the Microsoft Access database you want to share on the boot drive.

-or-

- Move the database to a Windows for Workgroups server.

### [References](#)

## PRB: SCSI Drives can Generate "corrupt" Error Message

Article Number: Q100159  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 1.1
- 

### SYMPTOMS

---

It is possible that a Microsoft Access database file (MDB) loading from a SCSI hard drive could generate the following error message:

```
"<filename> is corrupted or is not a database file.
Attempt to Repair?"
```

Attempts to repair the Database do not solve the problem.

### CAUSE

---

Typically, the database file is OK but Microsoft Windows itself is having trouble accessing the Drive.

### RESOLUTION

---

Add the following line to the Config.Sys file.

```
Device=c:\windows\smartdrv.exe /double_buffer
```

Note: If the machine is running MS-DOS 6.0 point to the "\dos" directory for the newest version of Smartdrv.exe.

### MORE INFORMATION

---

With most SCSI (small computer system interface) drives, Windows 3.1 requires the Smartdrv double buffer statement to boot Windows 3.1 into enhanced mode and access the drive. On a machine where windows is not on the SCSI drive, it is possible that windows boots into enhanced mode but Smartdrv Double\_buffer is not loading.

References: Windows 3.1 Users Guide Chap 14

[References](#)

## INF: ODBC Disk Contents (1/14/93, 3.5-Inch, MS Access 1.0)

Article Number: Q100158  
CREATED: 16-JUN-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0  
-----

### SUMMARY

---

The following lists the contents of the [ODBC](#) disk shipped with Microsoft Access version 1.0 (release date 1/14/93, 3.5-inch, 1.44 MB).

The WBTRCALL.DLL file was removed from the original release of Microsoft Access version 1.0. Because of this, the SETUP.EXE program was changed. No Microsoft Access functionality was changed for this release.

### MORE INFORMATION

---

#### Disk Contents

-----

|          |                     |        |              |        |
|----------|---------------------|--------|--------------|--------|
| DBNMP3   | <a href="#">DLL</a> | 8241   | 10-22-92     | 3:06p  |
| INSTCAT  | <a href="#">SQL</a> | 93796  | 01-04-93     | 12:00a |
| MSCOMSTF | DLL                 | 74528  | 01-04-93     | 12:00a |
| MSDETSTF | DLL                 | 24544  | 01-04-93     | 12:00a |
| MSINSSTF | DLL                 | 65440  | 01-04-93     | 12:00a |
| MSSHLSTF | DLL                 | 14928  | 01-04-93     | 12:00a |
| MSUILSTF | DLL                 | 6144   | 01-04-93     | 12:00a |
| ODBC     | DLL                 | 44736  | 01-04-93     | 12:00a |
| ODBC     | INF                 | 1226   | 01-04-93     | 12:00a |
| ODBCADM  | EXE                 | 9136   | 01-04-93     | 12:00a |
| ODBCADM  | HLP                 | 4942   | 01-04-93     | 12:00a |
| ODBCINST | DLL                 | 6016   | 01-04-93     | 12:00a |
| SETUP    | EXE                 | 54464  | 01-04-93     | 12:00a |
| SETUPI   | DLL                 | 44032  | 01-04-93     | 12:00a |
| SQLSETUP | DLL                 | 11024  | 01-04-93     | 12:00a |
| SQLSRVR  | DLL                 | 135792 | 01-04-93     | 12:00a |
| VER      | DLL                 | 9008   | 10-22-92     | 3:06p  |
| 7        | WMF                 | 5846   | 01-04-93     | 12:00a |
| 7E       | WMF                 | 5366   | 01-04-93     | 12:00a |
| WFWDV    | TXT                 | 2134   | 01-04-93     | 12:00a |
| WFWDV    | EXE                 | 23581  | 01-04-93     | 12:00a |
| WFWNET   | DR_                 | 165600 | 01-04-93     | 12:00a |
| LANMAN21 | <DIR>               |        | 01-04-93     | 12:00a |
|          | 23 file(s)          |        | 810524 bytes |        |

#### [References](#)

## PRACC9306: Help Context ID Property Does Not Work in Subform

Article Number: Q100157  
CREATED: 16-JUN-1993  
MODIFIED: 13-JUL-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0  
-----

### SYMPTOMS

---

If you press the F1 key when your insertion point is in a subform control, Microsoft Access invokes the main form's Help file Contents window, rather than invoking the appropriate Help topic for the subform control.

### CAUSE

---

Microsoft Access ignores the Help ContextID property of the subform control.

### RESOLUTION

---

Please refer to the workaround described below in the "More Information" section.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

### MORE INFORMATION

---

The following workaround can be used in Microsoft Access version 1.0 and provides a custom Help function that you can invoke by pressing the F1 key. This function tests to determine which control is currently active on a form and calls the Microsoft Windows Help API function to invoke the appropriate Help topic for that control.

1. Create the following macro and save it as AutoKeys

```
MacroName Action
=====
{F1} RunCode

AutoKeys Actions

RunCode
 Function Name: =Help("<c:\access\yourhlp.hlp>")
```

where <c:\access\yourhlp.hlp> is the path and filename of the Help file you want to use.

2. Create the following module:

```
'*****
'Declarations section of the module.
'*****
' NOTE: In the following sample code, an underscore (_) is used as a
' line-continuation character. Remove the underscore when re-creating
' this module in Access Basic.

Option Explicit

Declare Function WinHelp% Lib "User" (ByVal hwnd%, ByVal lpHelpFile$, _
 ByVal wCmd%, ByVal dwData As
Any)

Global Const HELP_CONTEXT = &H1
Global Const HELP_CONTENTS = &H3

'*****
' FUNCTION NAME: Help
'
' PURPOSE:
' Based on the currently active control, the Windows Help API
' function is called to invoke Help on a predefined ContextID for
' the control. If the ControlName is not represented, a
' message appears, indicating no Help topic is available.
' If there is no active form available, the contents of the
' Help file are invoked.
'
' INPUT PARAMETERS:
' HelpFile: Path and filename of Help file to use
'
' RETURN
' None
'*****
Function Help (HelpFile As String)
 Dim ContextID As Long
 Dim Ret As Integer

 On Error GoTo HelpError

 ' Determine which control needs Help and set the
 ' .. appropriate ContextID for the control
 Select Case Screen.ActiveControl.ControlName

'NOTE: You must customize the Case statements that follow
' for your specific control names and the ContextID to call for
' the control name.

 Case "Category ID"
 ContextID = 1
 Case "Category Name"
 ContextID = 1
 Case "Description"
 ContextID = 1
 Case "Product ID"
 ContextID = 2
```

```
Case "Product Name"
 ContextID = 2
Case "Unit Price"
 ContextID = 2
Case "Picture"
 ContextID = 2
Case Else
 MsgBox "No help for " & Screen.ActiveControl.ControlName
 ContextID = -1
End Select
```

```
' Call the Help file with the appropriate ContextID.
If ContextID > -1 Then
 Ret = WinHelp(Screen.ActiveForm.hwnd, HelpFile, _
 HELP_CONTEXT, ContextID)
End If
```

```
GoTo ByeHelp
```

```
HelpError:
```

```
' Display the Contents window of the Help file
Ret = WinHelp(0, HelpFile, HELP_CONTENTS, 0&)
Resume ByeHelp
```

```
ByeHelp:
End Function
```

[References](#)

## INF: How to Specify a Custom Starting Page Number for a Report

Article Number: Q100156  
CREATED: 16-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.1

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

In Microsoft Access, the "Page Property" specifies the current page number on each page of a [report](#). By Default the page numbers will start at 1. This article describes two methods to allow for user defined starting page numbers. Both provide custom page numbers, however the second [method](#) uses [Access Basic](#) and can therefore include some error checking.

### MORE INFORMATION

---

#### Method 1

-----  
Change the ControlSource of the [text box](#) where the page number is being displayed from "=Page" to "=Page + [Enter a Starting Page Number] -1"

This will cause the user to be prompted to enter a starting page number each time the report is previewed or printed. For example, in NWIND.MDB, open the Alphabetical List of Products report in design view. In the right side of the page [footer](#), there is a text box with a ControlSource of "=Page". Change the ControlSource of the text box to "=Page + [Enter a Starting Page Number] -1"

#### Method 2

-----  
The second method assumes that you are familiar with Access Basic and how to create and use Access Basic Procedures. This method is more complex, but avoids the possibility that a non-numeric value be entered by the user when prompted for a starting number.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

NOTE: In the following sample code, an underscore ( `_` ) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

1. First open a new Module or a previously created [module](#) and enter the following code:

```
'*****
'Declarations Section of Module
```

```

'*****
Option Explicit
Global PageChoice As Integer

'=====
' Create the following GetPageChoice() function in the Module
'=====
'This function is called in the OnOpen property of the Report.
Function GetPageChoice ()
 Dim choice As String
 Do
 choice = InputBox("Enter a Starting Page Number:", "Number _
 Report", "1")
 If Not (IsNumeric(choice)) Then
 MsgBox "Value Entered is not a Number."
 End If
 Loop While Not (IsNumeric(choice))
 PageChoice = CInt(choice)
End Function

'=====
' Create the following ReturnPageChoice() function in the Module
'=====
'This function is called by the text box that will contain the
' pagenumber.
Function ReturnPageChoice (pgnumber As Integer)
 ReturnPageChoice = PageChoice + pgnumber - 1
End Function

```

2. Open your report in Design view.

3. From the Edit Menu choose "Select Report" and from the View menu select "Properties" to display the Property Sheet. Change the following Report Property:

```
OnOpen: =GetPageChoice()
```

4. To display the page numbers, create a text box in the Page Footer of the report with the following properties:

```
Text Box: Text1

ControlName: PageNumber
ControlSource: =ReturnPageChoice(Page)

```

When you Print Preview or Print the report, you will be prompted for a starting page number. Each page will be numbered consecutively starting with this number.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 18, "Designing Reports," pages 422-424, 440-444 and Chapter 23, "Using Macros to Print Reports or Transfer Data" pages 571-572.

"Microsoft Access Language Reference," version 1.0, page 364

## References

## INF: Access 1.0 (rel. 1/14/93) Disks 1-3 Contents 3.5" 1.4 meg

Article Number: Q100155  
CREATED: 16-JUN-1993  
MODIFIED: 06-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0  
-----

### SUMMARY

---

The following lists the directory contents of diskettes 1-3 shipped with Microsoft Access version 1.0 release date: 1/14/93 (3.5 inch, 1.44 meg).

The WBTRCALL.DLL was removed from the original release of Microsoft Access 1.00. Due to this removal, the SETUP.EXE was changed. No functionality of Microsoft Access was changed for this release.

### MORE INFORMATION

---

Contents of DISK1  
-----

|          |      |        |          |        |
|----------|------|--------|----------|--------|
| 1        | WMF  | 4470   | 01-04-93 | 12:00a |
| 1E       | WMF  | 4022   | 01-04-93 | 12:00a |
| BTRVISAM | DL\$ | 57918  | 01-04-93 | 12:00a |
| COMMDLG  | DL\$ | 46406  | 10-09-92 | 10:02a |
| COMMDLG  | DL_  | 89248  | 01-04-93 | 12:00a |
| CTL3D    | DL\$ | 7524   | 08-16-92 | 7:35p  |
| DBSISAM  | DL\$ | 135919 | 01-04-93 | 12:00a |
| DDEML    | DL\$ | 20305  | 03-10-92 | 3:10a  |
| DETCMD   | DL_  | 24128  | 01-04-93 | 12:00a |
| MSACCESS | IN\$ | 519    | 01-04-93 | 12:00a |
| MSACCESS | RE\$ | 552    | 01-04-93 | 12:00a |
| MSAIN100 | DL\$ | 71625  | 01-04-93 | 12:00a |
| MSAJU100 | DL\$ | 38945  | 01-04-93 | 12:00a |
| OLECLI   | DL\$ | 41083  | 10-09-92 | 10:02a |
| OLESVR   | DL\$ | 13129  | 10-09-92 | 10:02a |
| ORDENTRY | TX\$ | 2281   | 01-04-93 | 12:00a |
| ORDENTRY | HL\$ | 14224  | 01-04-93 | 12:00a |
| ORDENTRY | MD\$ | 200396 | 01-04-93 | 12:00a |
| PIM      | HL\$ | 8119   | 01-04-93 | 12:00a |
| SETUP    | EXE  | 21472  | 01-04-93 | 12:00a |
| SETUP    | INI  | 149    | 01-04-93 | 12:00a |
| SHARE    | EX\$ | 8112   | 10-09-92 | 10:02a |
| SHELL    | DL\$ | 23296  | 10-09-92 | 10:02a |
| SMALLB   | FO\$ | 9687   | 03-06-92 | 7:16p  |
| SMALLE   | FO\$ | 11236  | 03-06-92 | 7:16p  |
| SMALLF   | FO\$ | 9376   | 03-06-92 | 7:16p  |
| STFSETUP | EX_  | 477392 | 01-04-93 | 12:00a |
| STFSETUP | IN_  | 58041  | 01-04-93 | 12:00a |
| VER      | DL\$ | 5653   | 10-09-92 | 10:02a |
| VER      | DL_  | 9008   | 01-04-93 | 12:00a |
| WORKGRP  | IN_  | 1393   | 01-04-93 | 12:00a |

31 file(s) 1415628 bytes

Contents of DISK2

-----

|           |      |               |          |        |
|-----------|------|---------------|----------|--------|
| 2         | WMF  | 8022          | 01-04-93 | 12:00a |
| 2E        | WMF  | 6870          | 01-04-93 | 12:00a |
| MSACCESS  | HL\$ | 1189376       | 01-04-93 | 12:00a |
| PDXISAM   | DL\$ | 106830        | 01-04-93 | 12:00a |
| REGEDIT   | EX\$ | 18176         | 10-09-92 | 10:02a |
| 5 file(s) |      | 1329274 bytes |          |        |

Contents of DISK3

-----

|           |      |               |          |        |
|-----------|------|---------------|----------|--------|
| 3         | WMF  | 8950          | 01-04-93 | 12:00a |
| 3E        | WMF  | 8470          | 01-04-93 | 12:00a |
| MSABC100  | DL\$ | 187802        | 01-04-93 | 12:00a |
| MSAES100  | DL\$ | 16899         | 01-04-93 | 12:00a |
| MSAFIN    | DL\$ | 18553         | 01-04-93 | 12:00a |
| WIZARD    | MD\$ | 174297        | 01-04-93 | 12:00a |
| MSACCESS  | EX\$ | 790199        | 01-04-93 | 12:00a |
| 7 file(s) |      | 1205170 bytes |          |        |

[References](#)

## PRACC9305: GP Fault with Button Caption Text Longer Than 2048

Article Number: Q100154  
CREATED: 16-JUN-1993  
MODIFIED: 02-JUL-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SYMPTOMS

---

You receive the following error message:

Application Error - MSACCESS caused a General Protection Fault  
in module MSACCESS.EXE at 0058:40DC.

### CAUSE

---

You may have set the caption of a command button to a text string that is longer than 2048 characters.

### RESOLUTION

---

Caption length is limited to 2048 characters. Using Access Basic to set the caption to a value larger than 2048 characters can result in unexpected behaviors, such as the error message described above.

### STATUS

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

---

Steps to Reproduce Problem

-----

The following steps generate the error message described above:

1. Add the following code to an Access Basic module:

```
Function mod_CreateForm ()
 Dim F As Form
 Dim C As Control

 Set F = CreateForm("", "")
 Set C = CreateControl(F.FormName, 104, 0, "", "", 100, 100)
 C.Caption = String$(3000, "M")
End Function
```

2. Run the code by typing the following in the Immediate window:

```
?mod_CreateForm()
```

Note that an icon for the form is created.

3. Restore the form and set the focus to the command button. Click on the button as if you planned to change the text in the caption.
4. Try to switch from Design view to Form view. The error message described above is displayed.

## [References](#)

## INF: Using DataPerfect data in Microsoft Access

Article Number: Q100153  
CREATED: 16-JUN-1993  
MODIFIED: 20-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

The following information describes how to import data TO Microsoft Access from WordPerfect DataPerfect.

### MORE INFORMATION

---

To use DataPerfect data in Microsoft Access, the information must first be saved from DataPerfect in "DOS Delimited Text" format. Once this process is complete, Microsoft Access can import the data using the "Text (Delimited)" option on the Export menu.

#### Problems With Multi-Line Field Entries

-----

DataPerfect allows you to press ENTER in fields (creating a carriage return character) to create multiple-line entries. In the Customer Information panel of the CLIENT1 database supplied with DataPerfect, the Address and Comment fields in some records contain two line entries.

NOTE: In DataPerfect, a "panel" is equivalent to a Microsoft Access table.

"DOS Delimited Text" files use carriage returns to mark the end of a record and cannot be included in field data. DataPerfect does not account for this condition when it saves the data in "DOS Delimited Text" format; thus, it generates an invalid file that cannot be successfully imported to Microsoft Access. You must remove the carriage returns before importing the data to Microsoft Access.

The following sets of instructions explain how to:

1. Export the Customer Information panel supplied with DataPerfect.
2. Use Microsoft Write, the word processor supplied with Microsoft Windows, to delete the carriage returns.
3. Import the resulting data to Microsoft Access.

#### Exporting the Panel from DataPerfect

-----

1. Start DataPerfect by typing DP from the DATA directory at the DOS command prompt.

2. From the list of databases, select either your database or CLIENT1 and press ENTER.
3. From the list of panels, select your panel, or select Customer Information if you chose CLIENT1 in Step 1, and press ENTER.
4. Press SHIFT+F7 to display the report list. Select Built-In Short Reports from the list and press ENTER to access the Built-In Report/Export menu.
5. Press 2 on the keypad to select Disk File On/Off and press 1 on the keypad to select Create File. Type "CUST.TXT", or any convenient name (preferably with a .TXT extension), and press ENTER.
6. Press 8 to select Report/Export Format options. Press 6 to select Export DOS Delimited Text.
7. Press ENTER to accept the pipe symbol (|) as the Field Delimiter. Press SPACEBAR to overwrite the tilde (~) character as the Record Delimiter, leaving the <CR><LF> in place.
8. Press Shift+F7 to begin the export, then Press any key to return to the list of reports.
9. Press F1, then F7 twice, then 0 to exit DataPerfect.

#### Removing Extra Carriage Returns

-----

1. In your Windows Accessories group, start Microsoft Write.
2. From the File menu, choose Open.
3. From the list of files, select CUST.TXT and press ENTER. (You may have to navigate the directory list to your DataPerfect directory, which is usually called DATA.) Type "\*.TXT" (without the quotation marks) and press ENTER to display a list of .TXT files.
4. Choose "No Conversion" when asked to convert the text file.

The document should now appear on the screen, with each piece of data separated by a pipe symbol and each record ending with a carriage return. Lengthy records wrap to the next line.

5. Lines that begin with the following text have an extra carriage return that incorrectly continues the record to the next line:

```
1|Burnett
26|Corrales, Jr.
7|Frame
19|Rhoton
22|Tolman
```

Move to the end of each line and press DEL to make the line flush with line above it.

6. The second line of the records that begin with the following text have extra paragraph marks that incorrectly mark the end of the record.

```
10|Grayson
21|Sergeant
```

Navigate to the end of the second line of these records and remove the incorrect mark by pressing DEL.

7. Choose Save from the File menu.
8. Choose Exit from the File menu to exit Write.

Importing the exported DataPerfect data into Microsoft Access

---

1. Open the desired database in Access to import the data into.
2. Choose Import from the File menu.
3. Choose "Text (Delimited)" from the Data Source list.
4. From the file list choose "CUST.TXT" just as you did when loading the file into Write, and choose the Import button.
5. In the Import Text Options window, choose the Options button.
6. Delete the double quote mark from the Text Delimiter box.
7. Type the pipe symbol, "|", in the Field Separator box.
8. Choose the OK button to begin the import.

Access will report any errors discovered during the import. You can examine the data imported into the resulting table and check to see that the proper data was imported into the proper columns. If you find data from one record spilling into fields in another record, this suggests that this record has an extra carriage return that was not removed. Open the file with a text editor, remove the offending carriage return, and reimport the data.

References: "DataPerfect User's Guide", page 339-343

[References](#)

## **PRACC9210: Cannot Use ReportWizard Without Default Win Printer**

Article Number: Q100152  
CREATED: 16-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### **SYMPTOMS**

---

You cannot edit, [preview](#), or print a [report](#) that you created using a Microsoft Access ReportWizard without first defining a default Windows printer on your system.

If a default Windows printer is not defined, the following error message appears every time you choose Print Preview or Print from the File menu:

There is no default printer. Select or install one using Windows Control Panel.

### **RESOLUTION**

---

Be sure to define a default Windows printer in your WIN.INI file before you use ReportWizards. The following example shows one way to change your WIN.INI file:

```
[windows]
...
device=QMS-PS 810,pscript,LPT1:
```

If you do not have a default Windows printer defined, select or install one using the instructions in the Microsoft Windows "User's Guide" or in Windows Help.

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### **MORE INFORMATION**

---

Steps to Reproduce Problem  
-----

1. Open your WIN.INI file. Just before "device=", type a semicolon (;) to temporarily disable your default Windows printer.
2. Start Microsoft Access and open the sample [database](#) NWIND.MDB.
3. Using a ReportWizard, create a new report based on one of the

tables (for example, Categories). Do not create a blank report.

The following error message appears twice:

There is no default printer. Select or install one using Windows Control Panel.

## [References](#)

## **PRACC9304: TrueType Fonts Shift Vertically on HP LJ4 Printer**

Article Number: Q100151  
CREATED: 16-JUN-1993  
MODIFIED: 23-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0
- 

### **SYMPTOMS**

---

When you print a Microsoft Access form or report formatted with a TrueType font to a Hewlett-Packard (HP) LaserJet 4 printer, the text in the controls is shifted vertically.

The same symptoms occur with the IBM 4029 printer.

### **CAUSE**

---

The HP LaserJet 4 and IBM 4029 printer drivers are reporting incorrect font metrics for some TrueType fonts. This problem causes Microsoft Access to incorrectly compensate by shifting the fonts vertically when sending the form or report to the printer.

### **RESOLUTION**

---

Use a non-TrueType font when you print to either the HP LaserJet 4 or IBM 4029 printer, or use another printer driver that is compatible with these printers (for example, the HP LaserJet 3 printer driver).

### **STATUS**

---

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0, using HP LaserJet 4 printer driver versions 31.1.03 and 31.V1.08 or the IBM 4029 printer driver. Microsoft Access version 1.1 corrects this problem for the HP LaserJet 4 printer drivers.

### **MORE INFORMATION**

---

If you experience the same problem with a non-TrueType font, verify that you are using a printer font. If Microsoft Windows is remapping your font to a TrueType font, you will continue to see the same vertical shift when you try to print.

The following example assumes that you have an HP LaserJet 4 printer connection and that you are using a HP LaserJet 4 printer driver version 31.1.03 or 31.V1.08.

Steps to Reproduce Problem  
-----

1. Create a new, unbound form.
2. Add two text box controls to the form, as follows:

Object: Text Box

-----  
ControlName: Field0  
ControlSource: ="Testing TrueType font"  
FontName: Arial

Object: Text Box

-----  
ControlName: Field2  
ControlSource: ="Testing non-TrueType font"  
FontName: Microsoft Sans Serif

3. Switch to [Form view](#).
4. Verify that you are configured to print to an HP LaserJet 4 printer, then choose Print Preview from the File menu. The data looks normal and is not shifted in either [control](#).
5. From the File menu, choose Print. Note that the data in Field0 is shifted vertically in the control; the data in Field2 maintains the correct position.

The HP LaserJet 4 printer driver (HPPCL5E.DRV) is manufactured by Hewlett-Packard Company, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

## [References](#)

## PRB: Macro on Subform Control Executed by Main Form

Article Number: Q100150  
CREATED: 16-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

If a macro is attached to the OnEnter or OnExit properties of the first control in the tab order of a subform, the macro will be executed each time the main form is moved from one record to another record.

In addition, a macro attached to OnEnter of the first control in the tab order of the subform will be executed when the main form is opened and a macro attached to the OnExit of the first control in the tab order of the subform will be executed when the form is closed.

### MORE INFORMATION

---

#### Steps To Reproduce Problem

-----

1. Open the Northwind Traders database (NWIND.MDB) supplied with Microsoft Access and create the following macro called "Test":

```
MacroName Action
=====
Test MsgBox
```

#### Test1 Actions

-----

```
MsgBox
 Message: Testing!
```

2. Open the "Orders Subform" and change the OnEnter and OnExit properties of the Property ID box to "Test".

```
Form: Orders Subform
 OnEnter: Test
 OnExit: Test
```

3. Save the Orders Subform, open the Orders form and navigate between records. The message "Testing!" will appear twice for each record.

#### Workaround

-----

To cause a macro to only be executed when actually moving in and out

of the control on the subform, a test condition with a flag must be used. A macro will be used to set a flag on the mainform to indicate whether the subform was entered by navigating into the subform from the mainform. The macro attached to the OnEnter and OnExit macros can check this flag to see if the user really did navigate into the subform and continue execution if appropriate.

1. Create the following macro called "Subform Flag":

| Macro Name | Action   |
|------------|----------|
| Set        | SetValue |
| UnSet      | SetValue |

"Subform Flag" Actions

-----  
SetValue  
Item: [Subform Flag]  
Expression: True  
SetValue  
Item: [Subform Flag]  
Expression: False

2. Create an unbound textbox on the main form (Orders) with the following properties:

Form: Orders  
Text Box: Subform Flag  
ControlName: Subform Flag  
Visible: No  
Default Value: False

3. Select the subform control on the main form (Order Details) and make the following property changes:

Form: Orders Subform:  
On Enter: Subform Flag.Set  
On Exit: Subform Flag.UnSet

4. Add the following macro action to the top of the macro that is attached to the On Enter and On Exit properties of the first control in the tab order of the subform. This macro will not execute any actions if the subform flag is not set.

| Condition                     | Action    |
|-------------------------------|-----------|
| Parent![Subform Flag] = False | StopMacro |

Limitation

-----  
The macro attached to the OnEnter property of the first control in the subform will not execute when you enter the control from the main form and the control is in the selected record on the subform (denoted by a triangle in the record selector). As a result, the macro will not execute by tabbing into the subform, or by mouse clicking on the first control in the selected record on the subform. However, it will

execute if you mouse click in this control on a different record in the subform.

References:

"Microsoft Access User's Guide" Chapter 22, "Using Macros with Forms" and Chapter 9, "Designing Forms;" Appendix C

[References](#)

## INF: Quit Macro Action

Article Number: Q100149  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### Summary:

---

This article provides an example of using the Quit [macro](#) action to quit Microsoft Access over a [DDE](#) channel.

### More Information:

---

The [syntax](#) for the action Quit is:

Quit

There is no parameter for the Quit action.

The following example creates a macro in Microsoft Excel that demonstrates how the Quit action works. This macro will not work with Microsoft Access version 1.0.

Sample DDE Macro from Microsoft Excel to Microsoft Access

-----

1. Open a new macro sheet in Microsoft Excel and enter the following macro:

| Cell | Command                                 |
|------|-----------------------------------------|
| A1   | QuitDDEExample                          |
| A2   |                                         |
| A3   | =APP.MINIMIZE()                         |
| A4   | =ERROR(FALSE)                           |
| A5   | =APP.ACTIVATE("Microsoft Access",FALSE) |
| A6   | chan=INITIATE("MSACCESS","system")      |
| A7   | =APP.ACTIVATE("Microsoft Access",FALSE) |
| A8   | =EXECUTE(chan,"[QUIT]")                 |
| A9   | =TERMINATE(chan)                        |
| A10  | =APP.ACTIVATE(FALSE)                    |
| A11  | =RETURN()                               |

2. Before running this macro, size your Microsoft Access and Microsoft Excel windows so that you can view both applications at the same time.
3. To run the macro, select [cell](#) A3. Then choose Run from the Macro menu and choose OK.

For more information on using macro actions in DDE, [query](#) on the following words in the Microsoft Knowledge Base:

access and 1.1 and dde and macro and action

[References](#)

## INF: OpenDatabase and CloseDatabase Pseudo Actions

Article Number: Q100148  
CREATED: 16-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.1
- 

### SUMMARY

---

Microsoft Access version 1.1 includes two new [DDE](#) Commands, also known as pseudo actions, available through the DDE "System" Topic:

OpenDatabase and CloseDatabase.

This article describes these pseudo actions, which allow you to open and close Microsoft Access databases over a DDE channel.

### MORE INFORMATION

---

The [syntax](#) for the pseudo actions OpenDatabase and CloseDatabase is:

OpenDatabase DATABASE\_LOCATION

CloseDatabase

Where DATABASE\_LOCATION is the path and filename of the Microsoft Access [database](#) to be opened. The CloseDatabase action does not take a parameter since Microsoft Access can only have one database open at a time.

The following example creates a [macro](#) in Microsoft Excel that demonstrates how the OpenDatabase and CloseDatabase pseudo actions work. This macro will not work with Microsoft Access version 1.0.

Sample DDE Macro from Microsoft Excel to Microsoft Access

-----

1. Open Microsoft Access (This macro will only work when Microsoft Access is already running).
2. Open a new Macro Sheet in Microsoft Excel. Enter the following macro:

| Cell | Command                                   |
|------|-------------------------------------------|
| A1   | PseudoOpenCloseDDEExample                 |
| A2   |                                           |
| A3   | chan=INITIATE("MSACCESS","SYSTEM")        |
| A4   | =APP.ACTIVATE("Microsoft Access",FALSE)   |
| A5   | =EXECUTE(chan,"[opendatabase nwind.mdb]") |
| A6   | =EXECUTE(chan,"[Sample AutoExec]")        |
| A7   | =EXECUTE(chan,"[closedatabase]")          |

```
A8 =TERMINATE (chan)
A9 =APP.ACTIVATE (, FALSE)
A10 =RETURN ()
```

3. Before running this macro, size your Microsoft Access and Microsoft Excel windows so that you can view both applications at the same time.
4. To run the macro, select [cell](#) A3. Then choose Run from the Macro menu and choose OK.

The commands in cells A3 and A4 initiate a DDE channel to Microsoft Access on the SYSTEM topic, and activate the Microsoft Access application.

The command in cell A4 opens the sample database Northwind Traders (NWIND.MDB) file in the current working directory using the OpenDatabase pseudo action.

The command in cell A5 runs the Sample AutoExec macro which is in the sample database Northwind Traders (NWIND.MDB) database file.

The command in cell A6 closes the currently opened database file using the CloseDatabase pseudo action.

The commands in cells A8:A10 terminate the DDE channel, activate the Microsoft Excel application, and end the macro.

For more information on using macro actions in DDE, [query](#) on the following words in the Microsoft Knowledge Base:

dde and macro and action

## [References](#)

## PRB: Error When Using OpenQueryDef Method on Parameter Query

Article Number: Q100147  
CREATED: 16-JUN-1993  
MODIFIED: 17-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

When you are using the OpenQueryDef [method](#) on a [parameter query](#), the following error message is displayed

    Couldn't Find Object.

when you re-create the sample code on page 137 of the "Introduction to Programming" manual.

### CAUSE

---

Either the name of the [query](#) you are trying to open is misspelled, the parameter is misspelled, or the parameter is not explicitly defined. In the sample code on page 137 of the "Introduction to Programming", both the query name and the parameter are misspelled.

### RESOLUTION

---

```
'*****
' The following function will correct the documentation errors on
' page 137 of the "Introduction to Programming" and will run correctly.
' Declarations section of the module
'*****
Option Explicit
```

```
Function TestQP ()
 Dim db As Database, Q As QueryDef, DataRecs As Dynaset
 Set db = CurrentDB()
 Set Q = db.OpenQueryDef("Products by Category (Parameter)")
 Q![Enter a category id:] = "COND" 'This is an explicit parameter

 Set DataRecs = Q.CreateDynaset()
 DataRecs.Close
 Q.Close
End Function
```

REFERENCES  
=====

"Introduction to Programming," page 137

[References](#)

## INF: How to Print Duplicate Mailing Labels for Each Record

Article Number: Q100146  
CREATED: 16-JUN-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

This article describes how to print multiple labels for each customer record. The example that follows uses the Customers table in the NWIND.MDB database.

### MORE INFORMATION

---

The following example generates 10 labels for each customer record:

1. In NWIND.MDB, create a new table called Label Count with one number field. Use the default properties and do not create a primary key. Enter a 1 in the first record, 2 for the second record, and so forth:

```
Table: Label Count

FieldName: Count
 DataType: Number
```

Enter the same number of records in the table as the number of times you want to duplicate the label when you print. In this example, the number of duplicates is limited to 10.

2. Create a new query based on the Label Count and Customers tables:

```
Query: LabelQuery

Type: Select Query
Tables: Label Count and Customers
Join: None
```

3. From the View menu, choose Totals so that all fields are set to Group By, as follows:

```
Field: Company Name
 Table: Customers
 Total: Group By
Field: Contact Name
 Table: Customers
 Total: Group By
Field: Address
 Table: Customers
 Total: Group By
Field: City
```

Table: Customers  
Total: Group By  
Field: Region  
Table: Customers  
Total: Group By  
Field: Postal Code  
Table: Customers  
Total: Group By  
Field: Count  
Table: Label Count  
Total: Group By  
Criteria: <= [Enter Number of Labels to Print for Each  
Customer]

NOTE: The Count field in the Label Count table may have an optional parameter value specified in the Criteria field to enter the number of labels to print for each record in the Customers table. If this query is used without a parameter value, 10 identical labels are printed for each record in the Customers table.

4. Using the Microsoft Access ReportWizard, create a new mailing label report based on the LabelQuery query. If you specify a parameter value, you are prompted to enter the number of labels to print for each customer record.

## [References](#)

## PRB: Extra Unwanted Information Appears in List Box

Article Number: Q100145  
CREATED: 16-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00

-----  
The information in this article applies to:

- Microsoft Access version 1.0  
-----

### SYMPTOMS

---

Unwanted or excess information appears in your [list box](#), along with the desired information.

### CAUSE

---

Check for a [control](#) located behind the list box.

### RESOLUTION

---

Possible solutions involve either removing the underlying control or repositioning the list box.

### MORE INFORMATION

---

A [combo box](#) does not exhibit this same behavior.

Steps to Reproduce Behavior

- 
1. Open the sample [database](#) Northwind Traders (NWIND.MDB).
  2. Open the Employees [form](#) in design mode.
  3. Add a list box to the form, with the following properties:

Object: List Box

-----  
ControlName: List Box Test  
ControlSource: EmployeeID  
RowSourceType: Table/Query  
RowSource: Employees  
Left: 2.99 in  
Top: 0.67 in

5. From the View menu, choose Form.

Initially, the list box displays the correct information, but as you go from [record](#) to record the information in the underlying control (First Name) bleeds through. Thus, your list box appears to contain garbled or excess information.

### [References](#)

## **PRB: 'Show All Records' Menu Choice not Available**

**Article Number: Q100144**  
**CREATED: 16-JUN-1993**  
**MODIFIED: 16-JUN-1993**  
**VERSION(S): 1.00**

-----  
The information in this article applies to:

- Microsoft Access version 1.0  
-----

### **SYMPTOMS**

---

You are unable to select the Show All Records menu choice from the Records menu.

### **CAUSE**

---

The [form](#)'s AllowFilters property is set to No. This behavior is by design.

### **RESOLUTION**

---

Set the form's AllowFilters property back to Yes.

### **MORE INFORMATION**

---

When the AllowFilters property is set to No, the following commands on the Records menu are disabled: Edit Filter/Sort, Apply Filter/Sort, and Show All Records.

For more information on the AllowFilters property, search for "AllowFilters", then "AllowFilters Property" using the Help menu.

### [References](#)

## INF: How to Remove the Menu Bar from a Form

Article Number: Q100143  
CREATED: 16-JUN-1993  
MODIFIED: 17-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

To remove the menu bar from a form, enter =1 (or any Access Basic function that evaluates to a numeric result) in the OnMenu property of the form.

### MORE INFORMATION

---

Entering =1 causes Microsoft Access to evaluate the OnMenu property. Because there is nothing to place in the menu bar, the menu bar is not displayed.

#### Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Open the Categories form in Design view.
3. From the View menu, choose Properties to display the form's property sheet.
4. In the OnMenu property, type "=1" (without the quotation marks).
5. Switch to Form view.

The menu bar does not appear in the Microsoft Access Application window. The menu bar reappears either when the focus leaves the form or when the form is closed.

#### REFERENCES

=====

"Language Reference," version 1.0, pages 339-340

#### [References](#)

## **INF: How Not to Display the Form Name in the Caption Bar**

**Article Number: Q100142**  
**CREATED: 16-JUN-1993**  
**MODIFIED: 24-AUG-1993**  
**VERSION(S): 1.00 1.10**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

Microsoft Access will display the name of the form in the caption bar by default.

To alter the form so that it does not display a caption, type a single space in the Caption property of the form.

### [References](#)

## INF: How to Find the Difference Between 2 Time Values

Article Number: Q100141  
CREATED: 16-JUN-1993  
MODIFIED: 10-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 1.1  
-----

### SUMMARY

---

This article describes how to use TimeValue to find the difference between two times within a 24 hour period using values with a data type of Date/Time.

### MORE INFORMATION

---

This article uses a [form](#) with two text boxes with the following properties:

Object: Text Box

-----

ControlName: Start\_Time

Object: Text Box

-----

ControlName: End\_Time

To calculate the difference between these two times, create a third [text box](#) with the following properties:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

Object: Text Box

-----

ControlName: Time\_Difference

ControlSource: =Format(TimeValue([End\_Time])\_  
-TimeValue([Start\_Time]),"Short Time")

The format [function](#) is used to format the time difference. In this example, Short Time "hh:mm" displays the difference in time in hours and minutes.

For more information search for "Format", then "Format, Format\$ functions" using the Help menu.

### [References](#)

## INF: Database Normalization Basics

Article Number: Q100139  
CREATED: 16-JUN-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

This article explains [database](#) normalization terminology for beginners. A basic understanding of this terminology is helpful when discussing the design of a [relational database](#).

### MORE INFORMATION

---

Description of Normalization  
-----

Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables, according to rules designed both to protect the data and to make the database more flexible by eliminating two factors: redundancy and inconsistent dependency.

Redundant data wastes disk space and creates maintenance problems. If data that exists in more than one place must be changed, the data must be changed in exactly the same way in all locations. A customer address change is much easier to implement if that data is stored only in the Customers [table](#) and nowhere else in the database.

What is an "inconsistent dependency"? While it is intuitive for a user to look in the Customers table for the address of a particular customer, it may not make sense to look there for the salary of the employee who calls on that customer. The employee's salary is related to, or dependent on, the employee and thus should be moved to the Employees table. Inconsistent dependencies can make data difficult to access; the path to find the data may be missing or broken.

There are a few rules for database normalization. Each rule is called a "normal [form](#)." If the first rule is observed, the database is said to be in "first normal form." If the first three rules are observed, the database is considered to be in "third normal form." Although other levels of normalization are possible, third normal form is considered the highest level necessary for most applications.

As with many formal rules and specifications, real world scenarios do not always allow for perfect compliance. In general, normalization requires additional tables and some customers find this cumbersome. If you decide to violate one of the first three rules of normalization, make sure that your application anticipates any problems that could occur, such as redundant data and inconsistent dependencies.

NOTE: The following descriptions include examples.

#### First Normal Form

-----

- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key.

Do not use multiple fields in a single table to store similar data. For example, to track an inventory item that may come from two possible sources, an inventory record may contain fields for Vendor Code 1 and Vendor Code 2.

But what happens when you add a third vendor? Adding a field is not the answer; it requires program and table modifications and does not smoothly accommodate a dynamic number of vendors. Instead, place all vendor information in a separate table called Vendors, then link inventory to vendors with an item number key, or vendors to inventory with a vendor code key.

#### Second Normal Form

-----

- Create separate tables for sets of values that apply to multiple records.
- Relate these tables with a foreign key.

Records should not depend on anything other than a table's primary key (a compound key, if necessary). For example, consider a customer's address in an accounting system. The address is needed by the Customers table, but also by the Orders, Shipping, Invoices, Accounts Receivable, and Collections tables. Instead of storing the customer's address as a separate entry in each of these tables, store it in one place, either in the Customers table or in a separate Addresses table.

#### Third Normal Form

-----

- Eliminate fields that do not depend on the key.

Values in a record that are not part of that record's key do not belong in the table. In general, any time the contents of a group of fields may apply to more than a single record in the table, consider placing those fields in a separate table.

For example, in an Employee Recruitment table, a candidate's university name and address may be included. But you need a complete list of universities for group mailings. If university information is stored in the Candidates table, there is no way to list universities with no current candidates. Create a separate Universities table and link it to the Candidates table with a university code key.

EXCEPTION: Adhering to the third normal form, while theoretically

desirable, is not always practical. If you have a Customers table and you want to eliminate all possible interfield dependencies, you must create separate tables for cities, ZIP codes, sales representatives, customer classes, and any other factor that may be duplicated in multiple records. In theory, normalization is worth pursuing; however, many small tables may degrade performance or exceed open file and memory capacities.

It may be more feasible to apply third normal form only to data that changes frequently. If some dependent fields remain, design your application to require the user to verify all related fields when any one is changed.

#### Other Normalization Forms

-----

Fourth normal form, also called Boyce Codd Normal Form (BCNF), and fifth normal form do exist, but are rarely considered in practical design. Disregarding these rules may result in less than perfect database design, but should not affect functionality.

\*\*\*\*\*  
 \* Examples of Normalized Tables \*  
 \*\*\*\*\*

#### Normalization Examples:

Unnormalized table:

| Student# | Advisor | Adv-Room | Class1 | Class2 | Class3 |
|----------|---------|----------|--------|--------|--------|
| 1022     | Jones   | 412      | 101-07 | 211-02 | 214-01 |

#### 1. First Normal Form: NO REPEATING GROUPS

Tables should have only two dimensions. Since one student has several classes, these classes should be listed in a separate table. Fields Class1, Class2, & Class3 in the above record are indications of design trouble.

Spreadsheets often use the third dimension, but tables should not. Another way to look at this problem: with a one-to-many [relationship](#), do not put the one side and the many side in the same table. Instead, create another table in first normal form by eliminating the repeating group (Class#), as shown below:

| Student# | Advisor | Adv-Room | Class# |
|----------|---------|----------|--------|
| 1022     | Jones   | 412      | 101-07 |
| 1022     | Jones   | 412      | 143-01 |
| 1022     | Jones   | 412      | 159-02 |
| 4123     | Smith   | 216      | 201-01 |
| 4123     | Smith   | 216      | 211-02 |
| 4123     | Smith   | 216      | 214-01 |

#### 2. Second Normal Form: ELIMINATE REDUNDANT DATA

Note the multiple Class# values for each Student# value in the above table. Class# is not functionally dependent on Student# (primary key), so this relationship is not in second normal form.

The following two tables demonstrate second normal form:

| Students: | Student# | Advisor | Adv-Room |
|-----------|----------|---------|----------|
|           | 1022     | Jones   | 412      |
|           | 4123     | Smith   | 216      |

| Registration: | Student# | Class# |
|---------------|----------|--------|
|               | 1022     | 101-07 |
|               | 1022     | 143-01 |
|               | 1022     | 159-02 |
|               | 4123     | 201-01 |
|               | 4123     | 211-02 |
|               | 4123     | 214-01 |

### 3. Third Normal Form: ELIMINATE DATA NOT DEPENDENT ON KEY

In the last example, Adv-Room (the advisor's office number) is functionally dependent on the Advisor attribute. The solution is to move that attribute from the Students table to the Faculty table, as shown below:

| Students: | Student# | Advisor |
|-----------|----------|---------|
|           | 1022     | Jones   |
|           | 4123     | Smith   |

  

| Faculty: | Name  | Room | Dept |
|----------|-------|------|------|
|          | Jones | 412  | 42   |
|          | Smith | 216  | 42   |

#### References:

"Microsoft Access User's Guide," version 1.0, pages 18-19

"FoxPro 2 A Developer's Guide," Hamilton M. Ahlo Jr. et al., pages 220-225, M & T Books, 1991

"Using Access for Windows," Roger Jennings, pages 799-800, Que Corporation, 1993

#### [References](#)

## PRB: New OLE Field is Created as Text Box in a New Form

Article Number: Q100138  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 1.1  
-----

### SYMPTOMS

---

When you drag an OLE field from a form's Field List, Microsoft Access creates a text box, not a Bound Object Frame. When you switch to Form view you can type text into the text box. However, if you open the table you see 'Invalid Object' in the OLE field.

### CAUSE

---

While you were designing this form you opened the table in design view and added the OLE field to the table. The form has not been saved and reopened since adding this new OLE field to the table.

### RESOLUTION

---

Save the form before dragging the new OLE field onto the form. After the form has been saved and reopened an OLE field that is dragged from the Field List will correctly generate a Bound Object Frame.

### MORE INFORMATION

---

A new OLE field, added to a table that the form is based on, will not be properly recognized if the form has not been saved and reopened. In order for Microsoft Access to properly recognize this field's data type you must save and reopen the form.

Steps to Reproduce Behavior

-----  
1. Create and save the following table named Table1:

```
Table: Table1

FieldName: Stuff
 DataType: Text
```

2. Create a new form based on Table1 and DO NOT save it. Place the field named Stuff on the form.

```
Object: Text box

ControlName: Stuff
 ControlSource: Stuff
```

3. With the form still open in design view, open Table1 in design view

and add the following field. Save and close Table1.

```
Table: Table1

FieldName: Photo
 DataType: OLE
```

4. Go back to your form and show the Field List by choosing Field List from the View menu. Drag the new OLE field, Photo, from the Field List onto the form. Microsoft Access creates a text box instead of a Bound Object Frame.

## [References](#)

## INF: Creating Queries to Execute Cascading Updates

Article Number: Q100137  
CREATED: 16-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

This article demonstrates how to create a query that automatically updates the many side of a one-to-many relationship when you change a value on the one side.

NOTE: You can do this only if referential integrity is not enforced in the relationship.

### MORE INFORMATION

---

In the sample database NWIND.MDB, there is a one-to-many relationship between the Categories and Products tables. The following steps explain how to create a query that automatically updates information in the Products table (many side) whenever you change a value in the Categories table (one side).

1. Open NWIND.MDB.
2. Make a copy of the Categories table and name it Categories Test. Make a copy of the Products table and name it Products Test.
3. From the Edit menu, choose Relationships. Establish a one-to-many relationship between Categories Test and Products Test on the Category ID field. Do not enforce referential integrity.
4. Create a new query based on the Categories Test and the Products Test tables. From the View menu, choose Table Names.
5. Drag the Category ID field from both the Categories Test and Products Test tables to the OBE grid.
6. From the Query menu, choose Update. Design your query as follows:

Query: UpdateManySide

-----  
FieldName: Category ID  
Table: Categories Test  
Update To: [Enter New Value]  
Criteria: [Enter Old Value]

FieldName: Category ID  
Table: Products Test  
Update To: [Enter New Value]  
Criteria: [Enter Old Value]

7. From the Query menu, choose Run.
8. In the Enter New Value box, type "FISH" and choose OK. In the Enter Old Value box, type "SEAF". Note that Microsoft Access tells you how many records will be updated. Choose OK and save the query.
9. Open the Categories Test and Products Test tables. Note that "SEAF" has been changed to "FISH" on both the one side (Categories Test) and the many side (Products Test) of the one-to-many relationship.

## [References](#)

## INF: Two Functions to Calculate Age in Months and Years

Article Number: Q100136  
CREATED: 16-JUN-1993  
MODIFIED: 25-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

This article explains how to create a [function](#) to calculate the age of a person or thing based on a given date.

### MORE INFORMATION

---

Enter the following code in a [module](#):

NOTE: In the following sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
'=====
' General Declaration
'=====
Option Explicit

'*****
' FUNCTION NAME: Age()
'
' PURPOSE:
' Calculates age in years from a given date to today's date.
'
' INPUT PARAMETERS:
' StartDate: The beginning date (for example, a birth date).
'
' RETURN
' Age in years.
'*****
Function Age (varBirthDate As Variant)
 Dim varAge As Variant

 varAge = DateDiff("yyyy", varBirthDate, Now)
 If Date < DateSerial(Year(Now), Month(varBirthDate), _
 Day(varBirthDate)) Then
 varAge = varAge - 1
 End If
 Age = varAge
End Function

'*****
' FUNCTION NAME: AgeMonths()
'
' PURPOSE:
```

```

' Compliments the Age function by calculating the number of months
' that have expired since the last month supplied by the given date.
' If the given date is a birthday, the function returns the number of
' months since the last birthday.
'
' INPUT PARAMETERS:
' StartDate: The beginning date (for example, a birthday).
'
' RETURN
' Months since the last birthday.
'*****
Function AgeMonths(ByVal StartDate As String)
 Dim tAge As Double
 tAge = (DateDiff("m", StartDate, Now))
 If (DatePart("d", StartDate) > DatePart("d", Now)) Then
 tAge = tAge - 1
 End If

 If tAge < 0 Then
 tAge = tAge + 1
 End If

 AgeMonths = tAge Mod 12
End Function

```

#### Testing Age Functions in the Immediate Window

---

1. Open a module or create a new one.
2. From the View menu, choose Immediate Window.
3. Assume your friend's birth date was November 15, 1967 and today is June 3, 1993. Type the following in the Immediate Window and press ENTER:

```
? Age("11/15/67")
```

Microsoft Access responds with the value 25 (years).

4. Type the following and press ENTER:

```
? AgeMonths("11/15/67")
```

Microsoft Access responds with the value 6, indicating that six months have passed since this person's last birthday. Your friend is 25 years and six months old.

#### Using Age Functions on a Form

---

The following procedure explains how to mark old orders by placing the age value in a new [control](#):

1. In the sample [database](#) NWIND.MDB, enter the Age() and AgeMonth() functions in a new module.

2. Open the Orders form in Design view and add an unbound text box.

3. Type the following in the ControlSource property of the new control:

```
=Age([Order Date]) & " ys " & AgeMonths([Order Date]) & " mo"
```

4. Switch to Form view. The age of the order is displayed in the new control.

## [References](#)

## INF: Searching for Characters/Strings in Records

Article Number: Q100135  
CREATED: 16-JUN-1993  
MODIFIED: 16-JUN-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access version 1.0 1.1  
-----

### Summary:

---

This article shows a [method](#) for searching for and identifying records containing a particular character or [string](#). In the example below, we will search for an asterisk character.

### More Information:

---

Say, for example, you have a [table](#) of names. Some of the names have an asterisk character somewhere in them, and you would like to get a list of all the names containing asterisks.

The solution is to use the InStr() [function](#). The InStr() function returns the position of the first occurrence of a string within another string. The example below uses the InStr() function to determine whether or not the asterisk character exists in the names in the table. If it does, it becomes a member of the [query's dynaset](#). The example below illustrates how to do this:

1. Start Microsoft Access, and create a new table called Nametest.
2. Create a text [field](#) called Names and enter the following data into the table:

```
Names

*Pat
Pat*
Tom
Ri*ck
Harry
```

3. Create a new query based on the Nametest table.
4. Drag the Names field from the [field list](#) box onto the query grid.
5. In the Field [row](#) of the query grid, next to the Names [column](#), type in the [expression](#):

```
NewField: InStr(1,[Name],"*")
```

6. In the Criteria row of the same column, enter:

>0

7. Uncheck the Show [check box](#) of that column.

8. Run the query.

You should get the results:

```
*Pat
Pat*
Ri*ck
```

This method can be used to obtain a dynaset containing any sequence of characters -- just change the asterisk in the expression above to match the string or character you want to search for.

References:

"Microsoft Access Language Reference," version 1.0, page 259.

[References](#)

## PRB: SetValue Does not Trigger Form Events

Article Number: Q100134  
CREATED: 16-JUN-1993  
MODIFIED: 22-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

The [macro](#) or [function](#) attached to the BeforeUpdate or AfterUpdate [event](#) does not execute.

### CAUSE

---

When you use a macro or [Access Basic](#) code to modify the value of a [form's field](#), it will not trigger any BeforeUpdate or AfterUpdate events attached to that [control](#).

### RESOLUTION

---

This behavior is by design.

### MORE INFORMATION

---

If you modify the value of a form's field or control via a macro or Access Basic code, you must manually trigger any desired BeforeUpdate or AfterUpdate events from that macro or Access Basic code (use the RunMacro or RunCode macro command).

Steps to Reproduce Behavior

- 
1. Open the sample [database](#) Northwind Traders (NWIND.MDB).
  2. Create and save the following macro:

| Macro Name | Actions              |
|------------|----------------------|
| Test1      | OpenForm<br>SetValue |

Test1 Actions

-----

OpenForm  
Form Name: Employees  
SetValue  
Item: Forms![Employees]![Last Name]  
Expression: [Null](#)

3. Open the Employees form in design view and add the pre-existing Required Entry macro to the Last Name field:

Object: Text Box

-----  
ControlName: Last Name

BeforeUpdate: Required Entry

4. Close the Employees form. From the Database window, highlight the Test1 macro.
5. Run the Test1 macro. Note that it successfully blanked out the employee's last name (you can press Escape twice to get it back). The macro attached to the Last Name control's BeforeUpdate property does not execute.
6. Try to remove the employee's last name manually. The Required Entry macro will be triggered, displaying the following message:  
  
    You must enter a last name
7. Do not save changes when closing the Employees form. This will remove the Required Entry macro that we added to the Last Name control in step #3 above.

## [References](#)

## PRB: Abnormal Action When Debugging Runtime Error

Article Number: Q100133  
CREATED: 16-JUN-1993  
MODIFIED: 13-AUG-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SYMPTOMS

---

Abnormal things can happen when debugging a runtime error in an Access Basic function. A form will not be accessible if it calls a function that is halted due to a runtime error.

### CAUSE

---

When program execution is halted like this, the handle on the form is still being used by the system. Even though the program has stopped running the form handle has not been released.

### RESOLUTION

---

To release the form handle after the program has been halted, select Reinitialize from the Module's Run menu.

### MORE INFORMATION

---

Steps to Reproduce Behavior  
-----

1. Create and save a form with the following command button on it:

Object: Command Button  
-----

ControlName: Run Code  
Caption: Run Code  
OnPush: =Test()

2. In a new module create and save the following function called Test():

```
Function Test()
 Dim x as String
 x = left("abc", -1)
End Function
```

3. Open the form that was created in step 1.
4. Push the 'Run Code' button on the form.

When the button is pushed it calls the function Test. Program execution will be halted and an error will display on the screen (because you can't specify a negative number in the left function).

The error message that you will see is:

Illegal function call

5. Select the OK button to respond to the error message.
6. Minimize the module in which the function is being displayed. (At this point, the form should be visible on the screen.)
7. Try to push the button on the form again, or, try to move the form.

Although you can see the form, it does not seem to be an object that can be made active. The form handle is not available.

8. To release the form handle after the program has been halted, select Reinitialize from the Module's Run menu.

References:

"Microsoft Access Introduction to Programming," version 1.0, Chapter 4, "Debugging Your Access Basic Code"

[References](#)

## INF: Four Ways to Move to a Record from a Combo Box Selection

Article Number: Q100132  
CREATED: 16-JUN-1993  
MODIFIED: 03-SEP-1993  
VERSION(S): 1.00 1.10

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
- 

### SUMMARY

---

Listed below are four different methods of moving to a specific record based on a combo box selection:

1. In the AfterUpdate property of the combo box, call a macro using the FindRecord action.
2. In the AfterUpdate property of the combo box, call a macro using the ApplyFilter action.
3. Use a Form/Subform, with the combo box in the main form and the data in the subform, bound by the LinkMasterFields and LinkChildFields properties.
4. Base the form on a query joining two tables and bind the combo box to the field that controls the join, using the row fix-up or dynamic lookup techniques.

### MORE INFORMATION

---

These four methods are outlined in the sections below and are based on the sample database NWIND.MDB.

The following table compares the features (benefits and drawbacks) of the four methods:

| Method Number:              | 1 | 2 | 3 | 4 |
|-----------------------------|---|---|---|---|
| Requires no code/macros     |   |   | x | x |
| Saves on subforms           | x | x |   | x |
| Can scroll to other records | x |   | x | x |
| Does not require a query    | x | x | x |   |
| Can edit records            | x | x | x |   |

-----

NOTE: These methods can also apply to text boxes.

#### Method 1

-----

1. Create a form called ComboTest based on the Products table. You must include at least the Product ID field on the new form. (Additional fields can help to illustrate that you have changed records based

on the value selected in the combo box.) Set the DefaultValue property to Single Form.

```
Object: Text Box

ControlName: Product ID
ControlSource: Product ID
Visible: Yes
```

2. Create an unbound combo box called Comb01 with the properties described below. The combo box can be located anywhere on the form, but preferably in the form header or footer.

```
Object: Combo Box

ControlName: Comb01
ControlSource: <leave blank>
RowSourceType: Table/Query
RowSource: Products
ColumnCount: 4
ColumnWidths: 0 in;0 in;0 in;2 in
BoundColumn: 1
After Update: Locate Product
```

3. Create a new macro called Locate Product with two actions:

```
Macro: Locate Product

GoToControl
 ControlName: [Product ID]
FindRecord
 Find What: =Comb01
 Find First: Yes
```

4. Switch to Form view. Note that when you choose a product name in the combo box, you are moved to the record selected and the appropriate Product ID is displayed.

Another example of this method is illustrated in the Suppliers form in NWIND.MDB. The form calls the Find Company macro based on the value selected in the Company Pick List combo box.

#### Method 2

- ```
-----
```
1. Repeat steps 1 and 2 from Method 1 described above.
 2. Create a macro called Locate Product with one action:

```
Macro: Locate Product
-----
ApplyFilter
    Where:[Product ID]=Forms![ComboTest]![Comb01]
```

The Product ID field is the bound field in the combo box. While the combo box displays the Product Name information, it is bound to (or, holds internally) the value of the Product ID.

This method is similar to the Suppliers form in NWIND.MDB, which uses the buttons to filter records from A to Z. Also, see the Alpha Apply Filter Buttons macro.

Method 3

1. Create a new, unbound form. Add a combo box with the following properties:

```
Object: Combo box
-----
ControlName: Comb01
ControlSource: <leave blank>
RowSourceType: Table/Query
RowSource: Products
ColumnCount: 4
ColumnWidths: 0 in;0 in;0 in;2 in
BoundColumn: 1
```

2. Create a second form based on the Products table to use as a subform. Include at least the Product ID field. (Additional fields can help to illustrate that you have changed records based on the value selected in the combo box.) Set the DefaultView property to Single Form.

```
Object: Text Box
-----
ControlName: Product ID
ControlSource: Product ID
Visible: Yes
```

3. Save the form, then use it to create a subform control on the first form by dragging it from the Database window to the Detail section of the first form.
4. Set the subform control properties as follows:

```
Object: Subform
-----
LinkChildFields: [Product ID]
LinkMasterFields: Comb01
```

By changing the value in Comb01, Microsoft Access ensures that the records in the subform match the combo box.

The Orders form in NWIND.MDB illustrates this method. The Order Details subform is related by the LinkMasterFields and LinkChildFields properties.

Method 4

1. Create a table called Prd containing a single field Product ID. Set the data type to Number and the field size to Integer.

NOTE: No primary key is necessary.

2. Create a single blank record.
3. Create a query called Prd Query based on a join between the Product ID fields of the Prd and Products tables. Include the following attributes in the query:

Query: Prd Query

Field: Product ID

 TableName: Prd

 Show: X

Field: <any other fields you are interested in>

 TableName: Products

4. Create a form based on Prd Query and add all fields.

NOTE: You must add at least the Product ID field.

5. Delete the text box control for Product ID and re-create it as a combo box, as follows:

Object: Combo box

ControlName: Product ID

ControlSource: Product ID

RowSourceType: Table/Query

RowSource: Products

ColumnCount: 4

ColumnWidths: 0 in;0 in ;0 in;2 in

BoundColumn: 1

The Orders form in NWIND.MDB illustrates this method. It allows the customer name and address to be selected based on the Customer ID combo box. By changing the Customer ID in the Orders table, the related Customers record is changed and the corresponding fields are updated on the form. For another example, see the Products and Suppliers form.

REFERENCES

=====

"User's Guide," Chapter 10, pages 275-278 and 281-284, Chapter 22, pages 545-548

For more information on the dynamic lookup or row fix-up techniques, query on the following words in the Microsoft Knowledge Base:

dynamic lookup and row fix-up

For more information on how to reference controls on a form, search for "Identifiers in Expressions" using the Help menu.

[References](#)

INF: How to Create a Parameter In() Statement

Article Number: Q100131
CREATED: 16-JUN-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes a way to create a [query](#) by using the In() operator with multiple values.

The [method](#) shown below uses a query that calls a [function](#), which then is passed two parameters. The first parameter is the name of a [field](#) that exists in the [table](#) on which the query is based. The second parameter prompts the user to enter a list of values. The function processes the user's entries as the list of multiple parameters for the In() operator.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

MORE INFORMATION

The following example is based on the sample [database](#) NWIND.MDB:

1. Create a new [module](#) with the following two functions:

```
'*****  
'Declarations section of the module.  
'*****
```

Option Explicit

```
'=====
```

```
' The GetToken() function defines the delimiter character.
```

```
'=====
```

```
Function GetToken (stLn, stDelim)  
    Dim iDelim as Integer, stToken as String  
    iDelim = InStr(1, stLn, stDelim)  
    If (iDelim <> 0) Then  
        stToken = LTrim$(RTrim$(Mid$(stLn, 1, iDelim - 1)))  
        stLn = Mid$(stLn, iDelim + 1)  
    Else  
        stToken = LTrim$(RTrim$(Mid$(stLn, 1)))  
        stLn = ""  
    End If  
    GetToken = stToken  
End Function
```

```

'=====
' The InParam() function is at the heart of this article. When
' the query runs, this function causes a query parameter
' dialog box to display so that you can enter a list of values.
' The values you enter are interpreted in the same way as if you
' had entered them within the parentheses of the In() operator.
'=====
Function InParam (Fld, Param)
    Dim stToken as String
    If IsNull(Fld) Then Fld = ""
    Do While (Len(Param) > 0)
        stToken = GetToken(Param, ",")
        If stToken = LTrim$(RTrim$(Fld)) Then
            InParam = -1
            Exit Function
        Else
            InParam = 0
        End If
    Loop
End Function

```

2. Close and save the module.
3. Create a new query based on the Customers table. Drag any fields you want to the query grid.
4. Add the following field to the query grid:

```

    FieldName: InParam([Customer ID],[Enter a list of IDs separated
                    by commas and no spaces:])
    Show: False
    Criteria: True

```

NOTE: The value InParam(...) shown for the FieldName should be entered as one statement on a single line. The InParam() function works with Integer fields as well as with Text fields.

5. From the Query menu, choose Parameters. Enter the following parameter with a Text [data type](#):

Enter a list of ID's separated by commas and no spaces:

6. Choose OK and run the query.

You are prompted to enter a list of parameters. The following message is displayed in the dialog box:

Enter a list of ID's separated by commas and no spaces:

7. Type the following values as shown, without spaces:

BLUEL,CACTP,DOLLC

All records meeting the above [criteria](#) are displayed.

[References](#)

PRB: Negative GroupOn Interval Reverses Sort Order

Article Number: Q100130
CREATED: 16-JUN-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access version 1.0 1.1
-

SYMPTOMS

When a non-text field in a Microsoft Access report is selected to sort ascending and group on an interval, the sort order is actually descending when previewed if the Interval is set to -1.

RESOLUTION

This is by design, as the report is actually sorting on the negative result of the expression or interval.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new report in the sample database Northwind Traders (NWIND.MDB), and base it on a table or query with a numeric field. For this example select the Order Details table.
2. Open the Sorting and Grouping window. Select a numeric field, like Order ID, in the first row of the Field/Expression column.

Sorting and Grouping

Field/Expression:	Sort Order:
Order ID	Ascending

Group Properties:
Group Header: Yes
Group On: Interval
Group Interval: -1

3. Locate the Order ID field in a text box on the Order ID Header
4. Size the Detail section Height property to 0 in.
5. Preview the Report. Results - the sort order is descending.

[References](#)

PRB: Allow Editing Set to Unavailable Does Not Work

Article Number: Q100129
CREATED: 16-JUN-1993
MODIFIED: 25-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You can still edit, add, and delete records from a [form](#), even though you set the form's AllowEditing property to Unavailable.

RESOLUTION

This behavior is by design.

MORE INFORMATION

The AllowEditing property on a form determines whether the AllowEditing command on the Records menu is enabled when you open a form from the [Database window](#). If enabled, you can override the form's Default Editing mode from [Form view](#).

It is a form's DefaultEditing property that determines whether a form opens to allow editing, only data entry, or is [read-only](#).

To disable editing, set the DefaultEditing property to Read Only, and the AllowEditing property to Unavailable.

For more information search for "allow editing", then "AllowEditing, DefaultEditing Properties" using the Help menu.

[References](#)

PRACC9304: Importing Numeric Column Names in Spreadsheets

Article Number: Q100128
CREATED: 16-JUN-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Microsoft Access replaces the numeric [column](#) names in your imported spreadsheet with other numbers.

RESOLUTION

Before you import your spreadsheet, format the column names as text by typing a single quotation mark (') in front of each number. Or, after you import your spreadsheet to a Microsoft Access [table](#), switch to [Design view](#) and change the [field](#) names.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Numeric column names in imported spreadsheets are ignored, regardless of the spreadsheet format. Microsoft Access replaces any numeric column names with column numbers in the destination table.

Steps to Reproduce Problem

1. Create a new spreadsheet in Microsoft Excel. Enter the following records in cells A1 through C2:

```
A1: 25
B1: 1
C1: 2
A2: 123456789
B2: dog
C2: cat
```

2. Save the spreadsheet as IMPTEST.XLS, then close Microsoft Excel.
3. Open a [database](#) in Microsoft Access.
4. From the File menu, choose Import.
5. In the Import box, select Microsoft Excel as the data source.

Choose OK.

6. In the Select File box, select IMPTEST.XLS, then choose Import.
7. In the Import Spreadsheet Option box, select the First Row Contains Field Names check box. Choose OK.
8. In the Import Result box, choose OK, then close the Select File box.
9. Open the IMPTEST table. Note that Microsoft Access has replaced the numeric column names with the actual column numbers from the spreadsheet.

[References](#)

PRACC9305: List Box Controls on Forms Painted Incorrectly

Article Number: Q100127
CREATED: 16-JUN-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The gray dotted box that surrounds the top row in a list box, indicating no selection, disappears when you scroll down from the top row and minimize the form.

CAUSE

This problem occurs because list box controls on forms are painted incorrectly, but only when you minimize and then restore a form whose current control is a list box in which there is no selection and the top row is scrolled out of view.

RESOLUTION

Any event that causes the form to repaint, such as pressing the F9 key or selecting an item from the list box, causes the gray dotted box to reappear.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new, blank form.
3. Place a list box control on the form and set its RowSource property to Employee List.
4. Switch to Form view. Note that the first item in the list, "Buchanan, B.L.," is surrounded by a gray dotted box.
5. Scroll down the list until "Buchanan, B.L." no longer appears. (You should need to click the scroll bar only once.)
6. Minimize the form, then maximize it.

7. Scroll up the list until "Buchanan, B.L." is visible again. Note that the gray dotted box is no longer visible.
8. Press the F9 key once to force the form to repaint. The gray dotted box reappears around "Buchanan, B.L."

[References](#)

PRACC9304: RunMacro May Not Work with Large Repeat Count

Article Number: Q100126
CREATED: 16-JUN-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you try to execute a RunMacro action with a Repeat Count argument greater than 32,767, either Microsoft Access may use an unexpected Repeat Count argument or the following error message may appear:

Repeat Count argument in RunMacro action can't be less than 0

CAUSE

Microsoft Access does not explicitly check for Repeat Count arguments greater than 32,767. Such values may wrap around either to a negative number or to an unexpected positive number. If the resulting value is negative, Microsoft Access displays the above error message. If the value is positive, Microsoft Access runs the [macro](#) using the Repeat Count argument that it evaluated.

RESOLUTION

Make sure that your Repeat Count argument falls between 0 and 32,767. If you need to run a macro more than 32,767 times, nest your macros. For example, if you need to run a macro 60,000 times, use the following three steps:

1. Create a macro (Macro1) that has a RunMacro action with a Repeat Count argument of 30,000.
2. Create another macro (Macro2) that has a RunMacro action with a Repeat Count argument of 2.
3. Instruct Macro2 to run Macro1.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

If the Repeat Count argument falls between 0 and 32,767, Microsoft Access returns the expected results (no repeat).

If the Repeat Count argument falls between 32,767 and 65,535,

Microsoft Access returns the error message listed above in the "Symptoms" section.

If the Repeat Count argument is greater than 65,535, Microsoft Access returns either the expected results (no repeat), the error message, or unexpected results.

Steps to Reproduce Behavior

1. Open a blank, unbound [form](#) and save it as as Test Form.
2. Create two macros (Increment by One and RepeatCount Test) as follows:

```
Macro Name      Action
=====
Increment by One  SetValue
```

```
RepeatCount Test Actions
-----
SetValue
  Item: [field0]
  Expression: [field0]+1
```

```
Macro Name      Action
=====
RepeatCount Test  RunMacro
                  MsgBox
```

```
RepeatCount Test Actions
-----
RunMacro
  Macro Name: Increment by One
  Repeat Count: 32767
MsgBox
  Message: Done
```

3. Add three controls to Test Form, as follows:

```
Object: Text Box
-----
ControlName: Field0
  ControlSource: <leave blank>
```

```
Object: Command Button
-----
ControlName: Increment
  Caption: Increment
  OnPush: Increment by One
```

```
Object: Command Button
-----
ControlName: Repeat
  Caption: Repeat
  OnPush: RepeatCount Test
```

4. Open Test Form in [Form view](#). Type "1" in Field0, then choose Increment. Note that each time you choose Increment, the value in Field0 increases by 1.
5. Type "1" in Field0, then choose Repeat. When the RepeatCount Test macro has run, the message "Done" is displayed. Note that the value in Field0 is now 32,767.

To test different Repeat Count arguments, open the RepeatCount Test macro in [Design view](#), change the Repeat Count argument, and repeat step 5.

[References](#)

INF: Using the .Parent Property of a Subform

Article Number: Q100025
CREATED: 14-JUN-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article explains how to use the Parent property of a subform to obtain the name of its parent form.

MORE INFORMATION

When you refer to the Parent property of a subform, you need to use Form.Parent if using the full syntax.

The following syntax will work in macro actions called from a subform:

```
Forms![Main Form name]![Subform ControlName].Form.Parent.Formname  
Form.Parent.Formname  
Parent.Formname
```

The following is incorrect syntax and gives a syntax error:

```
Forms!Orders![Orders Subform].Parent.Formname
```

The following is incorrect and gives the name of the subform:

```
Formname
```

Steps to Reproduce Behavior

1. Open the sample database Northwind Traders (NWIND.MDB) database.
2. Create and save the following macro:

```
Macro: Get Parent Name  
-----
```

```
Action: MsgBox  
  Message: =Forms!Orders![Orders Subform].Form.Parent.Formname  
  Title: Parent Form Name
```

2. Open the Orders Subform in Design view. Add a macro to the Product ID field:

```
Object: Text box  
-----
```

```
ControlName: Product ID  
  OnDblClick: Get Parent Name
```

3. Save and close Orders Subform.

5. Open the Orders form. Double-click the Product ID in the subform.

A message box entitled 'Parent Form Name' will display 'Orders', the name of the main (or parent) form.

6. Close the Orders form. Open the Orders Subform in Design view, and remove the macro that we added in step #2.

[References](#)

PRACC9210: Null Boolean Values Default to False

Article Number: Q99945
CREATED: 10-JUN-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Null values appearing in a boolean field always default to false in tables, forms, and reports.

CAUSE

Null values are not supported in boolean fields. When a null value is inadvertently placed in a boolean field (by a query, for example), the null value is ignored. By default, "false" incorrectly appears in the field.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Start Microsoft Access and create a new table called Table1. (No primary key required.)
2. In Design view, create a field in Table1 named Test with the data type Text.
3. Create another table and name it Table2. In Design view, create two fields:

Name:	Data Type:
Test	Text
Bool	Yes/No

4. Enter the following data into Table1:

Test
A
B
C

5. Enter the following data into Table2:

Test	Bool
A	Yes
B	No

6. Create a Make Table query using Table1 and Table2. Give the new table a name, such as "Bogus."
7. Create a [join](#) between Table1 and Table2 by dragging the Test field from Table1 and dropping it onto Table2.
8. Edit the Join Properties by double-clicking the join line or by selecting View/Join Properties from the menu. Select [relationship](#) #2. This includes all records from Table1 and only those records from Table2 where the joined fields are equal.
9. Drag the Test field from Table1 and the Bool field from Table2 and drop them onto the query grid.
10. Execute the Make Table query by selecting Run from the Query menu or by clicking the exclamation point button on the [toolbar](#).

A popup [dialog box](#) states, "3 [row](#)(s) will be copied into new table."

Open the table "Bogus" and notice its contents. It should have:

Test	Bool
A	-1
B	0
C	0

Notice that a value of zero (0) has incorrectly been placed for the value of Bool for Test value C (0 = No, -1 = Yes). Change the Bool field's format in Design view to view the values as True/False, Yes/No, and so forth.

[References](#)

PRACC9211: Cannot Export Paradox/dBase Tables

Article Number: Q99944
CREATED: 10-JUN-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In Microsoft Access version 1.0, you cannot export a Paradox or dBASE table over the top of an existing corrupted Paradox or dBASE database file. If you try to do so, you receive the following error message:

```
<path\filename> This file already exists.  
Replace existing file?.
```

If you choose the Yes button, you receive the following error message:

```
Table '<tablename>' already exists.
```

This error occurs with both dBASE and Paradox formats. Other database exports correctly allow you to overwrite existing database files.

CAUSE

Microsoft Access tries to open the file to verify its contents instead of simply overwriting it.

STATUS

This is by design.

MORE INFORMATION

Steps to Reproduce the Problem

1. Copy your CONFIG.SYS file to CONFIG.DB. (This will simulate a corrupted Paradox database.)
2. In Microsoft Access, choose Export from the File menu, select Paradox 3.X and any table.
3. Specify CONFIG.DB as the output file and choose OK.
4. Choose Yes to the "Replace existing file?" message.

You receive the error message "Table 'CONFIG#DB already exists."

References

PRB: Strange Characters Appear When Pasting in the Query Grid

Article Number: Q99943
CREATED: 10-JUN-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When using the Query Grid, an attempt to cut and paste (or to copy and paste) an entire Criteria row places strange characters in the cell you paste to. These may or may not disappear when you move off of the cell.

RESOLUTION

Before you paste information from the clipboard, select an object that has the same size as the information in the clipboard. In this example, select an entire row when you paste the information into the Criteria section of the QBE grid.

STATUS

This behavior is by design.

MORE INFORMATION

Perform the following four steps to demonstrate this behavior.

Steps to Reproduce Behavior

-
1. Open the Employee Sales by Country [Parameter] query in the sample database Northwind in design view.
 2. Select the entire Criteria row in the QBE Grid (click the mouse once at the left side of the row when the mouse cursor changes to a solid right arrow).
 3. From the Edit menu, choose either Cut or Copy.
 4. Select a cell (to paste the copied row into) and choose Paste from the Edit menu.

The pasted value shows characters that may appear as rectangles at the end of the cell.

This behavior occurs pasting into any object in Microsoft Access. When you cut or copy the row to the clipboard it is transferred as tab separated data. When you paste an entire row into a single cell, you are seeing the tabs.

For more information on pasting information in the query grid in Microsoft Access, search the Knowledge Base for

queries and paste

[References](#)

PRACC9306: Timeout While Waiting for DDE Response

Article Number: Q99942
CREATED: 10-JUN-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Microsoft Access generates the following message

Timeout while waiting for [DDE](#) response

when you try to initiate a dynamic data exchange (DDE) [link](#) with another application.

CAUSE

This error occurs when the [function](#) attempting to initiate the link is invoked using a RunCode action in a [macro](#) and the RunCode action macro is called from another application using DDE.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Microsoft Access takes time-outs with all attempts to perform DDE to other applications if the other application is currently conversing with Microsoft Access as a DDE server.

Steps to Reproduce Problem

The following instructions demonstrate the problem by showing how to use Word for Windows to run a Microsoft Access macro that attempts to initiate a DDE conversation with Microsoft Excel.

1. Start Excel with the default worksheet Sheet1 and type some text in [cell](#) A1, or [row](#) 1, [column](#) 1.
2. Start Microsoft Access and create the following macro called "Test":

Test Actions

RunCode

Function Name: =TestDDE()

3. Create the following [Access Basic](#) function:

```
Option Explicit
Function TestDDE ()
    Dim chan
    chan = DDEInitiate("excel", "sheet1")
    MsgBox DDERequest(chan, "R1C1")
    DDETerminate chan
End Function
```

4. Start Word for Windows and choose Macro from the Tools menu.
5. In the Macro Name box, type "test" (without the quotation marks), and press the Edit button.
6. Type the following macro in the editing window:

```
Sub MAIN
    chan = DDEInitiate("msaccess", "system")
    DDEExecute chan, "test"
    DDETerminate chan
End Sub
```

7. Run the macro by choosing the Start button on the macro [toolbar](#) or by pressing ALT+SHIFT+S.

Approximately 45 to 60 seconds later the following message is generated by Microsoft Access:

Timeout while waiting for DDE response

[References](#)

INF: Calculating a Credit Card Expiration Date

Article Number: Q99941
CREATED: 10-JUN-1993
MODIFIED: 21-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access Version(s) 1.0 1.1

Summary:

If you enter a credit card expiration date (month/year) in a date/time [field](#) on a [form](#), Microsoft Access assumes that the card expires on the first day of the month. This article includes a [function](#) that correctly calculates the expiration date as the last day of the month.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

More Information:

Use the following function, `ExpirationDay()`, in the `AfterUpdate` property of an expiration [text box](#) on your form. The function will replace the date entered in the text box with the date computed for the last day of the month/year.

Given a date/time field bound to a text box with the `ControlName` property set to "Expiration", the `AfterUpdate` property should read:

```
=ExpirationDay([expiration])
```

Create a [module](#) with the following function:

```
Option Explicit
Function ExpirationDay (dt)
  If Not IsNull(dt) Then
    dt = DateAdd("m", 1, DateAdd("d", (-1 * DatePart("d", dt)), dt))
  End If
End Function
```

You can use the following derivative of the `ExpirationDay()` function in calculated fields in a [query](#):

NOTE: In the following sample code, an underscore (`_`) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function ExpirationDay (ByVal dt)
  If Not IsNull(dt) Then ExpirationDay = DateAdd("m", _
    1, DateAdd("d", (-1 * DatePart("d", dt)), dt))
End Function
```

References

INF: How to Wait for a Shelled Process to Finish

Article Number: Q99940
CREATED: 10-JUN-1993
MODIFIED: 17-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you are using the Shell() [function](#) or the RunApp [macro](#) action to run another program or process, Microsoft Access does not wait for the shelled process to finish before processing the next line in the macro or function. This causes problems for macros and functions whose subsequent actions depend on the results of the shelled process.

This article includes a function called WaitShell(), which, given an application name or process to run, starts the process and waits for it to terminate.

MORE INFORMATION

The following Microsoft Access function, WaitShell(), uses GetModuleUsage, a Microsoft Windows API function, to determine if the shelled process has terminated. The Shell() function, if it successfully starts the process, returns a handle to the process [module](#). If the process is no longer running, the handle is invalid and GetModuleUsage returns a value of 0. WaitShell() simply loops until the module is no longer valid.

NOTE: You may have the following GetModuleUsage Windows API function defined in an existing Microsoft Access library. If you receive a duplicate procedure name error message, delete the Declare statement from your code.

```
Option Explicit
Declare Function GetModuleUsage% Lib "Kernel" (ByVal hModule%)

Function WaitShell (AppName$)
    Dim hMod%
    hMod% = Shell(AppName$, 1)
    If (hMod% > 32) Then
        While (GetModuleUsage(hMod%))
            DoEvents
        Wend
    Else
        MsgBox "Unable to start " & AppName$
    End If
End Function
```

To test this function, create and run the following Microsoft Access macro:

```
Action
-----
RunCode
MsgBox
```

```
Macro Actions
-----
```

```
RunCode
  Function Name: =WaitShell("Command.com")
MsgBox
  Message:      Done!
```

or run the following Microsoft Access function:

```
Function TestWaitShell()
  x=WaitShell("Command.com")
  MsgBox "Done!"
End Function
```

Both examples above launch instances of the DOS prompt, which remain until you terminate them by typing "exit" at the command line and pressing ENTER.

References:

Microsoft Windows "Programmer's Reference, Volume 2: Functions,"
version 3.1, pages 403-404

[References](#)

INF: Communicating with Microsoft Access using NetDDE

Article Number: Q99939
CREATED: 10-JUN-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article demonstrates how to communicate with Microsoft Access on a remote machine using NetDDE. This article assumes the reader is well versed in communicating with [DDE](#) on a local machine and with the capabilities of Microsoft Access as a DDE server.

For more information on Microsoft Access as a DDE server, [query](#) on the following keywords:

dde and server

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

MORE INFORMATION

In order to establish a [link](#) to a remote application, a DDE share must be created on the remote machine. A DDE share is simply a name that represents an application name, a topic, and any [permissions](#) for that topic. A DDE conversation can only occur with applications and topics assigned to share names on the remote machine, and nothing more. A utility is provided with the Windows for Workgroups Resource Kit called "Network DDE Share Manager" that can be used to edit and create DDE share names.

Creating a DDE share for the Employees [table](#):

1. Run the Network DDE Share Manager and choose New from the Share menu.
2. Type "Northwind Employees" in the "Share Name:" box.
3. Type "MSACCESS.EXE" in the "Application Name:" box and "NWIND.MDB;TABLE Employees" in the "Topic Name:" box. The password information is optional and can be left empty.
4. Choose Save from the File menu.

Communicating with the remote Access DDE share

Before running the following [function](#), make sure that Access is running on the remote machine and has the NWIND.MDB file opened.

Create the following [module](#) function in Microsoft Access on your local machine:

Note: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Option Explicit
Function GetRemoteEmployeeInfo()
    Dim iChan As Integer
    Dim EmpRecord As String, FieldNames As String

    iChan = DDEInitiate("\\RemoteMachineName\NDDE$", _
        "Northwind Employees")

    FieldNames = DDERequest(iChan, "FieldNames")
    MsgBox FieldNames, 0, "Employee Table Field Names"

    On Error Resume Next
    EmpRecord = DDERequest(iChan, "FirstRow")
    Do Until EmpRecord = ""
        MsgBox EmpRecord, 0, "Employee Record"
        EmpRecord = ""
        EmpRecord = DDERequest(iChan, "NextRow")
    Loop
End Function
```

You will have to change the first argument of the DDEInitiate command (RemoteMachineName) to reference the name of your remote machine.

When this function is executed from the [Immediate window](#), a message box will appear with a list of [field](#) names in the Employees table. Next, a series of message boxes will appear displaying each [record](#) of information from the table. The field information presented is delimited by a tab character, or ASCII 9.

The DDEInitiate Statement:

Notice the unique arguments used in the DDEInitiate command to establish the link with the remote share. The application name is the net DDE server application NDDE\$ located on the remote machine "\\RemoteMachineName". NDDE\$ is a system service application that is run automatically by Windows for Workgroups in the background. This application is always running and present when Workgroups is running and cannot be terminated.

When NDDE\$ receives the topic "Northwind Employees" it checks the list of available DDE shares on the local machine. If it finds one, it will validate any permissions on the share, prompting for a password if one is required. Once the [validation](#) is done, NDDE\$ establishes a local DDE link with the application name and topic assigned to the share name.

Once the link has been established, communication from the client

application proceeds in the same way as if it were communicating another application on the local machine. NDDE\$ becomes the liaison between the two applications across the [network](#).

For more information on NetWork DDE, see Chapter 11 "Network Dynamic Data Exchange" in the Windows for Workgroups Resource Kit manual.

[References](#)

INF: Removing alpha characters from an alpha/numeric field

Article Number: Q99938
CREATED: 10-JUN-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to create a Microsoft Access [function](#) called RemoveAlphas() to remove alpha characters from an alphanumeric [string](#). All non-numeric characters will be removed.

A practical example for RemoveAlphas() is to remove the parenthesis and dashes from a telephone number or social [security field](#). The following strings, "(206) 635-7050", "206-635-7050", "535-87-4529", which contain parenthesis, spaces, and dashes, when processed by RemoveAlphas(), will appear as "2066357050" and "535874529".

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications with the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming" manual.

MORE INFORMATION

The following instructions explain how to create the RemoveAlphas() function and how to use it in a Microsoft Access [update query](#) to cleanup all phone numbers from a [table](#).

Create the following Access Basic function.

1. Choose the Module button from the [Database window](#) and choose the New button.
2. In the [module](#) window type "Option Explicit" underneath the line that reads "Option Compare Database":
3. From the Edit menu, choose New Procedure.
4. Type RemoveAlphas in the Name box, choose Function for the Type and choose OK.
5. Type in the following function:

```
Function RemoveAlphas (AlphaNum)
    Dim Clean As String
    Dim Pos, A_Char$

    Pos = 1

    For Pos = 1 To Len(AlphaNum)
```

```

        A_Char$ = Mid(AlphaNum, Pos, 1)
        If A_Char$ >= "0" And A_Char$ <= "9" Then
            Clean$ = Clean$ + A_Char$
        End If
    Next Pos

    RemoveAlphas = Clean$
    AlphaNum = Clean$
End Function

```

5. Save the module with any unique name.

Using the RemoveAlphas function to clean up data entry

Suppose you want to ensure that whenever someone types in a telephone number into a field, that the field is cleaned of any parens or dashes that the user may accidentally type.

Given a field called "phone", add the following information in the AfterUpdate property of the phone textbox on a [form](#):

```
=RemoveAlphas([phone])
```

Using the RemoveAlphas function in an [update query](#)

Suppose you have a table with phone numbers in it where users have accidentally type parenthesis and dashes. The following query can be used to remove this information, leaving the phone numbers intact.

1. Create a new query and add the table containing the phone number field to it.
2. Place the phone number field in the first [column](#) of the query grid.
3. From the Query menu, choose Update. A new [row](#), "Update To:" will now appear in the query grid below.
4. In the "Update To:" row, type the following:

```
RemoveAlphas([phone])
```

NOTE: The name of your phone field should appear in the square brackets above.

5. Choose Run from the Query menu.

This article describes how to use the RemoveAlphas() function with an [expression](#). For more information on using functions with events, query on the following words in the Microsoft Knowledge Base:

function and expression and [event](#)

[References](#)

INF: Base Combo Box on Parameter Query to Filter Values

Article Number: Q99937
CREATED: 10-JUN-1993
MODIFIED: 09-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to filter values that appear in a combo box by basing it on a parameter query.

MORE INFORMATION

There may be situations in which you want to limit the values that appear in a combo box. For example, you may want to show only suppliers whose names begin with a certain letter. Although Microsoft Access does not allow filters on combo boxes, basing your combo box on a parameter query filters the values using a criteria clause.

This article describes the steps to create a combo box that displays only the suppliers that meet the criteria you specify. The sample database NWIND.MDB is used.

Steps to Reproduce Behavior

1. Create the following query to fill the combo box:

```
Query: Filtered Supplier List
-----
Type: Select
Field: Company Name
      Table: Suppliers
      Sort: Ascending
      Criteria: Like [My Criteria Box] & "*"
-----
```

2. Create the following macro to update the combo box with new criteria:

```
MacroName: CB Test Macro
-----
Action: Requery
ControlName: My combo box
-----
```

3. Create the following unbound form to test your controls:

```
Form: TestForm
-----
Record Source: <None>
-----
```

TestForm Controls

Combo Box: My combo box
ControlSource: <None>
RowSourceType: Table/Query
RowSource: Filtered Supplier List

Text Box: My Criteria Box
After Update: CB Test Macro

4. Open the form in Form view. Choose the arrow button. (Do not enter any value in the Criteria box.) Note that all suppliers are listed because you have not yet entered a filter criteria.
5. Type a single letter in the Criteria box (for example, "E").
6. Choose the arrow button. Note that only suppliers whose names begin with the letter "E" are displayed.
7. Give the Criteria box the focus. Type a new letter (for example, "P").
8. Choose the arrow button. Note that now only suppliers whose names begin with the letter "P" are displayed.

[References](#)

PRB: Error When Opening Attached Table in Shared .MDB File

Article Number: Q99936
CREATED: 10-JUN-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you attempt to open an [attached table](#) in a shared .MDB file, one of the following error messages may appear:

Couldn't find file "<drive>:\<path>\<filename>"

"<Drive>:\<path>\<filename>" isn't a valid path.

Couldn't open file "<drive>:\<path>\<filename>"

CAUSE

The path to an attached [table](#) is hard-coded in the [table properties](#). For a shared .MDB file, this path is most likely invalid.

RESOLUTION

To work around this problem, either change the path to the attached file so that it is valid or import the data.

MORE INFORMATION

To verify and correct the path for an attached table, use the following procedure:

1. Open the table in [Design view](#).
2. From the View menu, select Table Properties. The Description property contains the path to the attached table.
3. Place the .MDB file and its attached tables on a share on a dedicated server.
4. Use one of the following two naming conventions:
 - Choose a drive letter (for example, "u:") and make sure all users connect to the share using that drive letter. Use the same drive letter for the attached tables.
 - or-
 - Use the universal naming convention (UNC) for the attached tables (for example, \\<server>\<share>\<filename>).

NOTE: To attach a database using the UNC, you need to specify both the UNC name and the database name where you normally would specify the drive and database names. For example, instead of logging on to the network drive, typing the database name, and choosing Attach Table from the File menu, you need only to enter the UNC name, along with the database name, in the Attach dialog box, as follows:

\\OUTLAND\PUBLIC\MYDB.MDB

If the .MDB file is placed on a non-dedicated server (for example, in Microsoft Windows for Workgroups), you must maintain two separate .MDB files, one to share with other workstations and one for your local machine.

Maintaining two separate copies of the .MDB file is necessary because Windows for Workgroups does not allow a server machine to redirect to itself. For example, if the server name is SERVERNM, the SERVERNM machine cannot use the path \\SERVERNM\SHARE to access a file from its own hard disk.

References:

"User's Guide," version 1.0, pages 69-72

[References](#)

PRACC9306: Hidden Subform Maximized If Main Form Is Maximized

Article Number: Q99935
CREATED: 10-JUN-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In a main/subform combination where the main form is maximized and the subform is hidden, the subform returns maximized when it is made visible. This results in covering the entire main form.

RESOLUTION

Since the visible property is usually changed through a macro or function, work around this problem by adding steps to the process to restore the main form before the subform is made visible. Then, maximize the main form. An example of this workaround is detailed after the "Steps to Reproduce Problem" section below.

STATUS

Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

You can reproduce this problem by using the Orders form in the NWIND.MDB sample database. The following steps detail the macros necessary to reproduce this problem. The resolution to this sample problem is listed after the steps.

Steps to Reproduce Problem

1. Create a macro group that contains two macros. Name one "Hidden" and the other "Visible."

Macro Name	Action
Hidden	Setvalue
Visible	Setvalue

Hidden Macro Actions

Setvalue

Item: Forms![orders]![orders subform].Form.Visible
Expression: 0

Visible Macro Actions

```
Setvalue
  Item: Forms![orders]![orders subform].Form.Visible
  Expression: -1
```

2. Close and save the macro as "Show."
3. Open the Orders form and add two command buttons.
4. Set the following properties on the command buttons:

```
Command Button: Button1
  Caption: Hide
  OnPush: Show.Hidden
```

```
Command Button: Button2
  Caption: Show
  OnPush: Show.Visible
```

5. Choose View, then choose Form to open the Orders Form in Form view.
6. Maximize the Orders form by using the Maximize button in the forms upper-right corner.
7. Choose the Hide button to hide the subform.
8. Choose the Show button to show the subform maximized.

The main form is no longer visible.

To workaroud this problem, change the Visible macro actions as follows:

```
Visible Macro Actions
-----
Echo
  Echo On: No
Restore
Setvalue
  Item: Forms![orders]![orders subform].Form.Visible
  Expression: -1
Maximize
Echo
  Echo On: Yes
```

References:

"Microsoft Access User's Guide," Chapter 22, "Using Macros with Forms"

[References](#)

INF: Ordering Data in a Graph

Article Number: Q99934
CREATED: 10-JUN-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to order data in a [graph](#) more easily by changing the RowSource property to a [query](#) name, rather than to the [SQL](#) statement that Microsoft Access generates.

MORE INFORMATION

In the following steps, you will create a graph of all Northwind Traders orders over \$10,000, in descending order.

Steps to Reproduce Behavior

1. Open the sample [database](#) NWIND.MDB.
2. Create a new query based on the Orders [table](#), as follows:

Query: GraphQuery

Field name: Ship Name
Show: True

Field name: Order Amount
Sort: Descending
Show: True
Criteria: >10000

3. Use the [GraphWizard](#) to create a new [form](#) based on GraphQuery. Select both fields for the graph.
4. Switch to [Form view](#). Note that the orders are sorted by the Ship Name [field](#).
5. From the View menu, choose Form Design. Select the graph. From the View menu again, choose Properties.
6. Change the RowSource property of the object frame, as shown below:

Object: Object Frame

ControlName: Embedded0
RowSourceType: Table/Query
RowSource: GraphQuery

7. From the View menu, choose Form.

8. Switch to Form view. The orders are now sorted in descending order.

[References](#)

INF: Sending the Current Record to Word for Windows with DDE

Article Number: Q99570
CREATED: 02-JUN-1993
MODIFIED: 26-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes the steps for creating a [form](#) that allows you to press a button to send the [current record](#) to Microsoft Word for Windows. The data sent is merged into a prewritten letter and printed.

The article assumes that you understand dynamic data exchange ([DDE](#)), setting bookmarks in Word for Windows, and creating modules in Microsoft Access.

MORE INFORMATION

NOTE: In the sample code below, an underscore is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

Step One: Create the Word for Windows Document

1. Start Word for Windows and open a new document.
2. Type the following:

```
CompanyName  
Address  
City, Region, PostalCode  
Country
```

Dear ContactName,

```
NorthWind Traders would like to thank you for  
your business during the past year. Enclosed  
you will find several samples of new products  
that we are excited to announce.
```

Sincerely,
NorthWind Traders.

NOTE: Word for Windows will fail when you attempt to complete the merge if the [field](#) names contain spaces. If you copy text into a Word for Windows document, be sure to remove any and all tabs.

3. Save this document as DDEMERGE.DOC.
4. To create the bookmarks, select CompanyName and choose [Bookmark](#)

from the Insert menu. Name the Bookmark "CompanyName" (without the quotation marks).

5. Repeats these steps, creating bookmarks for the fields: Address, City, Region, PostalCode, Country, and ContactName.

Step Two: Create The Access Basic Modules

1. Open the sample [database](#) NWIND.MDB. (One of the following modules uses the [function](#) STARTAPP(), which is located in the "Introduction to Programming" manual.
2. Create a new [module](#) called Print Merge.
3. Place the following statement in the (declarations) section:

```
Global MergeChan As Integer
Const ID_YES = 6
```

4. Create a new function called Initiate_Word ().

```
Function Initiate_Word ()
    Dim Chan As Variant
    Dim WordTopics As Variant
    Chan = StartApp("Winword", "System")

    On Error GoTo AlertUser:
    WordTopics = DDERequest(Chan, "Topics")

    If InStr(1, WordTopics, "DDEMERGE.DOC") = 0 Then
        DDEExecute Chan, "[FILEOPEN("DDEMERGE.DOC")]"
    End If

    DDETerminate Chan
    Mergechan = DDEInitiate("Winword", "DDEMERGE.DOC")

Exit Function

AlertUser:
    MsgBox "Access is unable to initiate a DDE _
        channel with the document DDETEST.DOC"
    Resume Next
End Function
```

5. Create a new function called Send_Record().

```
Function Send_Record ()

    On Error GoTo CatchBlanks

    DDEPoke Mergechan, "CompanyName", _
        Forms![Customers]![Company Name]
    DDEPoke Mergechan, "ContactName", _
        Forms![Customers]![Contact Name]
    DDEPoke Mergechan, "Address", _
        Forms![Customers]![Address]
```

```

DDEPoke Mergechan, "City", _
Forms![Customers]![City]
DDEPoke Mergechan, "Region", _
Forms![Customers]![Region]
DDEPoke Mergechan, "PostalCode", _
Forms![Customers]![Postal Code]

DDEExecute Mergechan, "[FilePrint]"

Exit Function

```

CatchBlanks:

```

If MsgBox("Error sending one field, it may be blank._
Would you like to continue?", 52) = ID_YES Then
Resume Next
Else
Exit Function
End If

```

End Function

6. Create a function called Terminate_MergeChan().

```

Function Terminate_MergeChan ()

DDETerminate MergeChan

End Function

```

7. Choose Compile All from the Run menu and then close and save the module.

Step Three: Create the Form

1. Open the form [Customers] in Design view.
2. Set the OnOpen property of the form to:

Form Properties

```

-----
OnOpen:      =Initiate_Word()
OnClose:     =Terminate_MergeChan()

```

3. Add a new button to the Customers form.

Command Button

```

-----
Caption:     Print Letter
OnPush:     =Send_Record()

```

4. Save the form and switch to Form view. Choose the Print Letter button.

The current record will be sent to Word for Windows, merged into the document DDEMERGE.DOC and then printed.

References:

"Introduction to Programming," Chapter 9, "Dynamic Data Exchange"

"Word for Windows Technical Reference," pages 20-25

[References](#)

INF: How to Execute Macro Actions Using DDE

Article Number: Q99407
CREATED: 27-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.10

The information in this article applies to:

- Microsoft Access version 1.1
-

SUMMARY

Microsoft Access version 1.1 supports the execution of individual [macro](#) actions over a dynamic data exchange ([DDE](#)) channel.

MORE INFORMATION

The only macro actions that you can execute over a DDE channel to Microsoft Access are those supported by the DoCmd [function](#) in Access Basic. For this reason, the following macro actions cannot be executed:

- AddMenu
- MsgBox
- RunApp
- RunCode
- SendKeys
- SetValue
- StopAllMacros
- StopMacro

The [syntax](#) for macro actions executed through DDE is the same as the syntax for macro actions called by the DoCmd function, that is:

MacroActionName Argument1,Argument2,...,ArgumentN

You can't use the predefined constants that are available inside [Access Basic](#), such as A_NORMAL, as arguments when you are executing macro actions with DDE. Instead, you must use the actual values of the constants. The values of the constants are determined by using the following procedure:

1. Open a macro in [Design view](#).
2. Select the macro action you want to use from the list of macro actions in the Action [column](#).
3. Go to the Action Arguments section of the macro in Design view and locate the argument for which you need to supply a value.
4. In the [list box](#) associated with that argument, locate the option that you want to use.
5. Count down the list, starting at zero (0), until you reach that option.

The number you reach will be the number that you need to supply as the argument when executing the macro action over a DDE channel. For example, if you wanted to supply the third option on a list as your argument, you would use the number "2" (see step 5 above).

The following example creates a macro in Microsoft Excel that demonstrates how to execute macro actions in Microsoft Access.

NOTE: This macro will not work in Microsoft Access version 1.0.

Sample DDE Macro from Microsoft Excel to Microsoft Access

-
1. Open Microsoft Access. (This macro only works if Microsoft Access is already running.)
 2. Open a new macro sheet in Microsoft Excel. Enter the following macro:

Cell	Command
A1	MacroActionDDE
A2	
A3	chan=INITIATE("MSACCESS","SYSTEM")
A4	=APP.ACTIVATE("Microsoft Access",FALSE)
A5	=EXECUTE(chan,"[opendatabase nwind.mdb]")
A6	=EXECUTE(chan,"[beep]")
A7	=EXECUTE(chan,"[openform categories,,,,,2]")
A8	=EXECUTE(chan,"[beep]")
A9	=EXECUTE(chan,"[openform categories]")
A10	=EXECUTE(chan,"[close]")
A11	=EXECUTE(chan,"[closedatabase]")
A12	=TERMINATE(chan)
A13	=APP.ACTIVATE(,FALSE)
A14	=RETURN()

3. Before you run this macro, size your Microsoft Access and Microsoft Excel windows so that you can view both applications at the same time.
4. To run the macro, select cell A3, choose Run from the Macro menu, and then choose OK.

The commands in cells A3 and A4 initiate a DDE channel to Microsoft Access on the SYSTEM topic and activate the Microsoft Access application.

The command in cell A5 opens the sample database NWIND.MDB file in the current working directory using the OpenDatabase pseudo action. (Pseudo actions are actions that are only valid over the DDE channel.)

The commands in cells A6 and A8 execute the Microsoft Access macro action Beep.

The command in cell A7 executes the Microsoft Access macro action OpenForm to open the Categories form in a minimized state.

The command in cell A9 executes the Microsoft Access macro action OpenForm to open the Categories form in a normal state.

The command in cell A10 closes the currently active object in Access.

The command in cell A11 closes the currently opened database file using the CloseDatabase pseudo action.

The commands in cells A12:A14 terminate the DDE channel, activate the Microsoft Excel application, and end the macro.

References:

For more information about the DoCmd statement, search for "DoCmd," then "DoCmd Statement" using the Help menu.

For more information about sending macro actions to Microsoft Access, [query](#) on the following words in the Microsoft Knowledge Base:

access and dde and server and macro

"Language Reference," version 1.0, pages 149-150

[References](#)

PRB: Error Message When Importing Informix Delimited Text File

Article Number: Q99406
CREATED: 27-MAY-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you attempt to import an Informix [delimited text file](#) that uses the pipe (|) symbol as a delimiter, you receive the following error message:

Name " contains invalid characters

CAUSE

When you export in Informix as text delimited and you specify the pipe symbol as the delimiter, Informix adds an extra pipe symbol at the end of the line. When you import this file to Microsoft Access by specifying "First [row](#) contains [field](#) names", Microsoft Access assumes there is another field after the last pipe symbol. Since the value is null, Microsoft Access returns the above error message.

RESOLUTION

- Depending on the size of the file, you can open it in a text editor and add a dummy name after the last pipe symbol. After you import the file, you can delete this dummy field.
- Clear the First Row Contains Field Names [check box](#) when you import the file. After you import, you can manually delete the first row of the [table](#) (containing field names instead of data) and modify the field names with the table in [Design view](#).

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a text file in Microsoft Windows Notepad as follows

```
A|B|C|  
1|2|3|
```

Place a pipe symbol between each character. The pipe symbol is usually found on the backslash (\) key on your keyboard.

2. Save the file as INFORMIX.TXT.
3. Open a [database](#) in Microsoft Access.

4. From the File menu, choose Import.
5. Select Text (delimited) as your data source and choose Import.
6. Select the INFORMIX.TXT file that you created in step 1. Choose OK.
7. The Import Options dialog box appears. Select the First Row Contains Field Names check box, then choose the Options button.
8. In the Field Separator combo box, type the pipe symbol and choose OK. The above error message appears.

Informix is manufactured by Informix Software, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

INF: Part 2, DDE in Visual Basic to Request Data from Access

Article Number: Q99405
CREATED: 27-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This is the second in a series of two articles containing an advanced example that demonstrates how to use Visual Basic to request data from Microsoft Access by using dynamic data exchange ([DDE](#)).

MORE INFORMATION

The following Visual Basic example demonstrates how to use DDE to request information by using [SQL](#) statements and the rich set of items supported by Microsoft Access to navigate tables. Further, a subroutine that assists in parsing data from the tab-delimited records that are requested is provided.

Specifically, the example does the following:

1. Gets the number of records and fields in the [table](#) using the "FirstRow" and "FieldCount" items.
2. Creates an array to store the records when they are retrieved.
3. Requests the data from a table, one [record](#) at a time, using the "FirstRow" and "NextRow" items.
4. Requests a list of [field](#) names using the "FieldNames" item.
5. Uses a subroutine to [parse](#) each tab-delimited record requested from Access and places the data in the array mentioned above. (A separate array holds the field names).
6. This step is optional. Once all the data is placed in the array, a Visual Basic grid [control](#) will be populated with the array contents. Further, the grid columns will be populated with the names of the fields.

When you run this example, be sure that Access is running and the sample [database](#) NWIND.MDB is open.

To use this example, create a new Visual Basic program with a text box, a [command button](#), and a grid control with the following properties:

```
Text Box
-----
Name: Text1
```

Command Button

Name: Command1

Grid

Name: Grid1

NOTE: The grid control is available in the Professional Toolkit for Visual Basic version 1.0 and ships with version 2.0 of the Professional Edition of Microsoft Visual Basic for Windows.

Double-click the command button and type the code that follows between the Sub Command1_Click () and End Sub lines.

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
Dim CntRows, CntCols As Integer ' Row and column count in the table
Dim Row, Col As Integer        ' Current row/column being filled.
Dim Record As String           ' Record read from employees table.

' Get the number of records in the Employees Table
Text1.LinkTopic = "MSACCESS|NWIND;SQL Select Count(*) As _
    EmployeeCount From Employees;"

Text1.LinkItem = "FirstRow"    ' Setup to get the first row of data
Text1.LinkMode = 2             ' Establish a manual link
Text1.LinkRequest              ' Request the data into the Text1
Text1.LinkMode = 0             ' Terminate the link
CntRows = Val(Text1.Text)      ' Convert the result to a number

' Establish a DDE Link to the Employees table
Text1.LinkTopic = "MSACCESS|NWIND;SQL Select * From Employees;"
Text1.LinkMode = 2             ' Establish a manual link

' Get the number of columns in the Employees Table
Text1.LinkItem = "FieldCount"  ' Setup to get count of fields/cols
Text1.LinkRequest              ' Request the data into Text1
CntCols = Val(Text1.Text)      ' Convert the result to a number

' Create the arrays to hold employee information and field names.
ReDim Employees(CntRows, CntCols) As String
ReDim FieldNames(1, CntCols) As String

' Get the field names in the Employees table from Access
Text1.LinkItem = "FieldNames"  ' Setup to get the field names
Text1.LinkRequest              ' Request the data into Text1

' Parse the tab delimited list of field names and copy it into the
' .. FieldNames array. (ParseRecord subroutine is listed latter)
ParseRecord FieldNames(), 1, (Text1.Text)

' Request each tab delimited record of data one at a time.
For Row = 1 To CntRows
```

```

    If Row = 1 Then
        Text1.LinkItem = "FirstRow" ' Setup for the first record
    Else
        Text1.LinkItem = "NextRow" ' Setup for the next record
    End If
    Text1.LinkRequest ' Request the data into Text1

    ' Parse the tab delimited record and store it in Employees array
    ParseRecord Employees(), Row, (Text1.Text)
Next Row

Text1.LinkMode = 0 ' Terminate link, arrays are filled with data

' OPTIONAL: The following code populates a grid control with the contents
' of the Employees and FieldNames arrays.

' Setup the Grid with the correct number of Rows and Cols
Grid1.Rows = CntRows + 1 ' Add 1 for the row selector buttons
Grid1.Cols = CntCols + 1 ' Add 1 for the column header buttons

' Fill the Grid column header with the data in the FieldNames array
Grid.Row = 0 ' Move to the column header row
For Col = 1 To CntCols
    Grid.Col = Col ' Move to the column
    Grid.Text = FieldNames(1, Col) ' .. and fill it with data
Next Col

' Fill the Grid control rows with the data in the Employees array
For Row = 1 To CntRows
    Grid.Row = Row ' Move to the row
    For Col = 1 To CntCols
        Grid.Col = Col ' Move to the column
        Grid.Text = Employees(Row, Col) ' Fill it with array data
    Next Col
Next Row
End Sub

```

The following subroutine, ParseRecord, is used to extract the tab-delimited data from a record and place the data in an array in the specified row. The subroutine takes the following three arguments:

```

Array() - A two-dimensional array of strings to store data into.
Row      - The row position in the array to store the parsed data.
Record  - A string with tab delimited information to parse.

```

Create a new module in Visual Basic with the following subroutine:

```

Sub ParseRecord (Array() As String, ByVal Row As Integer, _
    ByVal Record As String)
    Dim Start As Integer ' Start position of field in record
    Dim TabStop As Integer ' Position in the record of the next tab.
    Dim CntCols As Integer ' The number of columns in the Array
    Dim Col As Integer ' The current column (field) being parsed

    CntCols = UBound(array, 2) ' Get count of columns in the array
    Start = 1 ' Start parsing with first character

```

```
' Find the first tab stop (ASCII character 9) in the record.
TabStop = InStr(Start, Record, Chr$(9), 0)

' Loop until we fill up all but the last column (Col < CntCols)
' .. or until there are no more columns to read (TabStop <> 0 means
' .. that no tab was found)
Col = 1
While Col < CntCols And TabStop <> 0
    ' extract the field from the string and store it in the array
    Array(Row, Col) = Mid$(Record, Start, TabStop - Start)

    Start = TabStop + 1          ' Increment start to next tab stop
    TabStop = InStr(Start, Record, Chr$(9), 0) ' Find next tab stop
    Col = Col + 1                ' Increment to next column in array
Wend
Array(Row, CntCols) = Mid$(Record, Start) ' Get last col of data
End Sub
```

[References](#)

PRACC9304: SQL Delete Triggers Function Incorrectly

Article Number: Q99404
CREATED: 27-MAY-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

Microsoft Access appears to delete a record in an attached SQL table, but the record is not actually deleted. The record will reappear when you requery the table.

CAUSE

The Microsoft SQL Server driver that comes with Microsoft Access version 1.0 incorrectly indicates that the result of the SQL delete action is successful.

RESOLUTION

To correct this problem, you must DROP the current trigger and CREATE a new delete trigger on the SQL Server that uses the RAISERROR command instead of the PRINT command. This method causes Access version 1.0 to not delete the record and display the following error messages:

ODBC - delete failed.

[Microsoft][ODBC SQL Server Driver][SQL Server] <MESSAGE> (#-31073)

where <MESSAGE> is the message you specified in the RAISERROR command.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

If you try to delete a record from an attached SQL table that has a delete trigger that performs a ROLLBACK and a PRINT action on the SQL Server, Microsoft Access version 1.0 appears to delete the record and then displays a message box asking you to verify your deletion. The record is actually not deleted and reappears when you requery the table by pressing SHIFT+F9, choosing Show All Records from the Records Menu, or closing and reopening the table.

In Microsoft Access version 1.1, the RAISERROR method works the same way that it does in version 1.0. The PRINT method works similarly to the way it does in version 1.0.

In version 1.0, the PRINT method appears to let you delete the record without a warning, then later the record reappears. In version 1.1, the PRINT method still does not display the message specified by the PRINT command; however, it does not remove the record, and it does display the following error message:

ODBC - call failed.

Steps to Reproduce Problem

1. Create a trigger on a SQL Server table that uses the SQL commands ROLLBACK and PRINT.
2. Attach to the SQL table that the delete trigger was created for by choosing Attach from the File menu.
3. Once the Table is attached, open it in [Datasheet view](#).
4. Select a record by clicking the gray [record selector](#) to the left of the record or by choosing Select Record from the Edit menu.
5. Delete the record by pressing the DEL key or by Choosing Delete from the Edit menu.
6. Choose Yes in the Confirmation [dialog box](#) that is now displayed.

The record appears to now be deleted from the table. Requery the table by pressing SHIFT+F9. The record reappears.

References:

"Microsoft SQL Server Language Reference," version 4.2, pages 88-93, 137, 203-204, and 214-216

[References](#)

INF: Sending a MAPI Mail Message from Microsoft Access

Article Number: Q99403
CREATED: 27-MAY-1993
MODIFIED: 17-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The following example includes a Microsoft Access [module](#) which demonstrates how to send a MAPI (Mail Application Programming Interface) message to multiple TO recipients, carbon copies (CC), and blind CC destinations with multiple attachments.

MORE INFORMATION

The following procedure explains how to use this sample module:

1. Create a [form](#) with the following controls:

```
Form: Test
-----
Caption: TestForm

Text box: Subject
  ControlName: Subject
Text box: To
  ControlName: To
Text box: CC
  ControlName: CC
Text box: BCC
  ControlName: BCC
Text box: Attach
  ControlName: Attach
Text box: Message
  ControlName: Message
Command button: Button0
  Caption: Send Message
  OnPush: =Mail()
```

2. Open either a new module or a previously created module and enter the following code:

NOTE: In the following sample code, an underscore (_) is used as a line continuation character for easier reading. Remove the underscore when re-creating this code in [Access Basic](#).

```
'*****
'Declarations section of the module.
'*****
```

Option Explicit

Option Compare Database 'Use database order for string comparisons

```
Type MAPIMessage
  Reserved As Long
  Subject As String
  NoteText As String
  MessageType As String
  DateReceived As String
  ConversationID As String
  Flags As Long
  RecipCount As Long
  FileCount As Long
End Type
```

```
Type MapiRecip
  Reserved As Long
  RecipClass As Long
  Name As String
  Address As String
  EIDSize As Long
  EntryID As String
End Type
```

```
Type MapiFile
  Reserved As Long
  Flags As Long
  Position As Long
  PathName As String
  FileName As String
  FileType As String
End Type
```

```
Declare Function MAPISendMail Lib "MAPI.DLL" Alias "BMAPISendMail" _
  (ByVal Session&, ByVal UIParam&, Message As MAPIMessage, Recipient _
  As MapiRecip, File As MapiFile, ByVal Flags&, ByVal Reserved&) As Long
```

```
Global Const SUCCESS_SUCCESS = 0
Global Const MAPI_TO = 1
Global Const MAPI_CC = 2
Global Const MAPI_BCC = 3
```

```
Global Const MAPI_LOGON_UI = &H1
```

```
'*****
' End of declarations section
'*****

'*****
' FUNCTION NAME: Mail
',
' PURPOSE:
' Passes information on the active forms To, Subject, CC, BCC,
' Attach, and Message text boxes to the SendMail function.
' It ensures that each box does not have a NULL value. It also
' displays an error message if SendMail fails.
' This function is called from the OnPush property of the form.
',
```

```

' INPUT PARAMETERS:
'   None
'
' RETURN
'   None
'*****
Function Mail ()
    Dim F As Form, Result
    Set F = Screen.ActiveForm

    ' Make sure user has something in the To: box
    If IsNull(F!To) Or F!To = "" Then Exit Function

    ' Make sure no Null values are in the other boxes
    If IsNull(F!Subject) Then F!Subject = ""
    If IsNull(F!CC) Then F!CC = ""
    If IsNull(F!BCC) Then F!BCC = ""
    If IsNull(F!Attach) Then F!Attach = ""
    If IsNull(F!Message) Then F!Message = ""

    ' Send the message, passing information from the form
    Result = SendMail((F!Subject), (F!To), (F!CC), (F!BCC), _
        (F!Attach), (F!Message))

    ' Test the result for any errors
    If Result <> SUCCESS_SUCCESS Then
        MsgBox "Error sending mail: " & Result, 16, "Mail"
    Else
        MsgBox "Message sent successfully!", 64, "Mail"
    End If
End Function

'*****
' FUNCTION NAME: SendMail
'
' PURPOSE:
'   This is the front-end function to the MAPISendMail function. You
'   pass a semicolon-delimited list of To, CC, and BCC recipients, a
'   subject, a message, and a delimited list of file attachments.
'   This function prepares MapiRecip and MapiFile structures with the
'   data parsed from the information provided using the ParseRecord
'   sub. Once the structures are prepared, the MAPISendMail function
'   is called to send the message.
'
' INPUT PARAMETERS:
'   sSubject: The text to appear in the subject line of the message
'   sTo:      Semicolon-delimited list of names to receive the
'            message
'   sCC:      Semicolon-delimited list of names to be CC'd
'   sBCC:     Semicolon-delimited list of names to be blind CC'd
'   sAttach:  Semicolon-delimited list of files to attach to
'            the message
' RETURN
'   SUCCESS_SUCCESS if successful, or a MAPI error if not.
'*****
Function SendMail (sSubject As String, sTo As String, sCC As String, _
    sBCC As String, sAttach As String, sMessage As String)

```

```

Dim i, cTo, cCC, cBCC, cAttach          ' variables holding counts
Dim MAPI_Message As MAPIMessage

' Count the number of items in each piece of the mail message
cTo = CountTokens(sTo, ";")
cCC = CountTokens(sCC, ";")
cBCC = CountTokens(sBCC, ";")
cAttach = CountTokens(sAttach, ";")

' Create arrays to store the semicolon delimited mailing
' .. information after it is parsed
ReDim rTo(0 To cTo) As String
ReDim rCC(0 To cCC) As String
ReDim rBCC(0 To cBCC) As String
ReDim rAttach(0 To cAttach) As String

' Parse the semicolon delimited information into the arrays.
ParseTokens rTo(), sTo, ";"
ParseTokens rCC(), sCC, ";"
ParseTokens rBCC(), sBCC, ";"
ParseTokens rAttach(), sAttach, ";"

' Create the MAPI Recip structure to store all the To, CC, BCC
' .. information to be passed to the MAPISendMessage function
ReDim MAPI_Recip(0 To cTo + cCC + cBCC - 1) As MapiRecip

' Setup the "TO:" recipient structures
For i = 0 To cTo - 1
    MAPI_Recip(i).Name = rTo(i)
    MAPI_Recip(i).RecipClass = MAPI_TO
Next i

' Setup the "CC:" recipient structures
For i = 0 To cCC - 1
    MAPI_Recip(cTo + i).Name = rCC(i)
    MAPI_Recip(cTo + i).RecipClass = MAPI_CC
Next i

' Setup the "BlindCC:" recipient structures
For i = 0 To cBCC - 1
    MAPI_Recip(cTo + cCC + i).Name = rBCC(i)
    MAPI_Recip(cTo + cCC + i).RecipClass = MAPI_BCC
Next i

' Create the MAPI File structure to store all the file attachment
' .. information to be passed to the MAPISendMessage function
ReDim MAPI_File(0 To cAttach) As MapiFile

' Setup the file attachment structures
MAPI_Message.FileCount = cAttach
For i = 0 To cAttach - 1
    MAPI_File(i).Position = -1
    MAPI_File(i).PathName = rAttach(i)
Next i

' Set the mail message fields
MAPI_Message.Subject = sSubject

```

```

    MAPI_Message.NoteText = sMessage
    MAPI_Message.RecipCount = cTo + cCC + cBCC

    ' Send the mail message
    SendMail = MAPI_SendMail(0&, 0&, MAPI_Message, MAPI_Recip(0), _
        MAPI_File(0), MAPI_LOGON_UI, 0&)
End Function

'*****
' SUB NAME: CountTokens
'
' PURPOSE:
'   Given a string of delimited items and the delimiter, the number
'   of tokens in the string will be returned. This function is useful
'   for dimensioning an array to store the delimited items prior to
'   calling ParseTokens.
'
' INPUT PARAMETERS:
'   sSource: A delimited list of tokens
'   sDelim: The delimiter used to delimit sSource
'
' RETURN
'   The number of tokens in sSource, which is the number of delimiters
'   plus 1. If sSource is empty, 0 is returned.
'*****
Function CountTokens (ByVal sSource As String, ByVal sDelim As String)
    Dim iDelimPos As Integer
    Dim iCount As Integer

    ' Number of tokens = 0 if the source string is empty
    If sSource = "" Then
        CountTokens = 0

    ' Otherwise number of tokens = number of delimiters + 1
    Else
        iDelimPos = InStr(1, sSource, sDelim)
        Do Until iDelimPos = 0
            iCount = iCount + 1
            iDelimPos = InStr(iDelimPos + 1, sSource, sDelim)
        Loop
        CountTokens = iCount + 1
    End If
End Function

'*****
' SUB NAME: GetToken
'
' PURPOSE:
'   Given a string of delimited items, the first item will be
'   removed from the list and returned.
'
' INPUT PARAMETERS:
'   sSource: A delimited list of tokens
'   sDelim: The delimiter used to delimit sSource
'
' RETURN
'   sSource will have the first token removed. The function

```

```

' returns the token removed from sSource.
'*****
Function GetToken (sSource As String, ByVal sDelim As String) _
As String
    Dim iDelimPos As Integer

    ' Find the first delimiter
    iDelimPos = InStr(1, sSource, sDelim)

    ' If no delimiter was found, return the existing string and set
    ' .. the source to an empty string.
    If (iDelimPos = 0) Then
        GetToken = sSource
        sSource = ""

    ' Otherwise, return everything to the left of the delimiter and
    ' .. return the source string with it removed.
    Else
        GetToken = Left$(sSource, iDelimPos - 1)
        sSource = Mid$(sSource, iDelimPos + 1)
    End If
End Function

'*****
' SUB NAME: ParseTokens
'
' PURPOSE:
' Extracts information from a delimited list of items and places
' it in an array.
'
' INPUT PARAMETERS:
' Array(): A one-dimensional array of strings in which the parsed tokens
' will be place
' sTokens: A delimited list of tokens
' sDelim: The delimiter used to delimit sTokens
'
' RETURN
' None
'*****
Sub ParseTokens (Array() As String, ByVal sTokens As String, ByVal _
                sDelim As String)
    Dim i As Integer
    For i = LBound(Array) To UBound(Array)
        Array(i) = GetToken(sTokens, sDelim)
    Next
End Sub

```

References:

"Microsoft Mail Technical Reference," chapter 4.

[References](#)

PRB: Incorrect Btrieve Error Message, [Btrieve ISAM] section

Article Number: Q99402
CREATED: 27-MAY-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The following incorrect error message may be returned when you attempt to attach or import a Btrieve file:

Invalid entries in [Btrieve ISAM] section in WIN.INI

CAUSE

This error message is generated when Microsoft Access tries to attach or import a Btrieve file and the settings in the [Btrieve] section are incorrect.

RESOLUTION

Search for the [Btrieve] section in the WIN.INI file instead of the [Btrieve ISAM] section.

References:

For more information, search on "Btrieve" and "WIN.INI" using the Help menu.

Btrieve is manufactured by Novell, Inc. a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

INF: How First Line of Data is Used to Import Delimited Text

Article Number: Q99401
CREATED: 27-MAY-1993
MODIFIED: 18-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The first line of data is extremely important when you are importing delimited text files. Microsoft Access uses the first line of data to determine the following:

- The number of fields
- The data type for each field

For this reason, it is important to verify that the first line of data correctly reflects the rest of the data in the table. This article discusses some of the common problems you may encounter when the first line of data is incorrect and how to work around these problems.

MORE INFORMATION

To specify the structure that Microsoft Access should use, you can use the following steps instead of relying on the first line of data:

1. Create a table in Microsoft Access that has the correct number of fields and the correct data type for each field.
2. From the File menu, choose Import.
3. Select Text (delimited) from the Data Source list and choose OK.
4. Select the text file and choose OK.
5. In the Import Text Options dialog box, choose Append To Existing Table and specify the table you created in step 1.
6. Choose OK to import the file.

Determining the Number of Fields

Microsoft Access scans the first line of data to determine the number of fields in the text file. If successive records contain more fields, these are ignored and no error message is given. For example, suppose you have a text file that looks like the following:

```
5959,"John Doe","123 Main Street"  
5960, "George Brown","55 Orange Avenue","MainTown","WA","USA"
```

Because the first line only contains three fields, Microsoft Access creates the following table:

Column 1	Column 2	Column 3
5959	John Doe	123 Main Street
5960	George Brown	55 Orange Avenue

Note that the additional fields were ignored and no error message was generated.

Determining the Data Types

Microsoft Access scans the first line to determine the data type for each field. If the data type in the field is invalid, a Type Conversion Failure error message is generated in the Import Errors table. For example, if the following text file is imported

```
5959,"John Doe","123 Main Street",8/12/90
5960,"George Brown","55 Orange Avenue",81290
ABCD,"Jane Bell","78 West Street",8/12/90
```

Microsoft Access detects four fields with the following data types:

Field	Data Type
1	Number
2	Text
3	Text
4	Date/Time

The second record fails to convert because the number 81290 in the fourth field is not a Date/Time data type, as determined by the first record value (8/12/90). The third record fails because "ABCD" in the first field is not a number. As a result, Microsoft Access reports type conversion errors in each case.

Other Workarounds

In addition to import appending the text file to an existing table, it is also possible to use a text editor to edit the first line of data to correctly reflect the number of fields and data types in the file.

If a field in the first row contains a number that is supposed to belong in a text field, type double quotation marks around the number in the first row. Microsoft Access will then correctly interpret the field as a text field.

REFERENCES

=====

"User's Guide," version 1.0, pages 72-79

[References](#)

INF: Using a Conditional Macro to Confirm Changes to a Field

Article Number: Q99400
CREATED: 27-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to prompt the user to confirm changes to a [field](#) using a conditional [macro](#).

MORE INFORMATION

This following procedure uses the Employees [form](#) in the sample [database](#) NWIND.MDB:

1. Create the following new macro by choosing the Macro button in the [Database window](#) and then choosing New:

Macro Name	Condition	Action	Defined Below
AreYouSure	MsgBox("Commit changes?",1)=2	CancelEvent	1.
	...	SendKeys	2.
AreYouSure Actions			

1. CancelEvent			
2. SendKeys			
Keystrokes: {esc}			
Wait: NO			

NOTE: The ellipses in the Condition [column](#) force Microsoft Access to perform the action on that line if the condition on the preceding line is true. Microsoft Access evaluates macro conditions as true or false. If the [expression](#) is true, Microsoft Access performs the action; if it is false, Microsoft Access ignores the action.

2. Save the macro as AreYouSure.
3. Open the Employees Form in [Design view](#) by choosing the Form button in the Database window, selecting the Employees form, and choosing the Design button.
4. From the View menu, choose Property to display the [property sheet](#).
5. Select the [Hire Date] [text box control](#) to display its properties.
6. Set the properties for the Employees form and [Hire Date] text box control as follows:

Form: Employees
Caption: Employees Form
ControlSource: Employees
Text box: Hire Date
ControlName: Hire Date
ControlSource: Hire Date
Before Update: AreYouSure

7. From the View menu, choose Form. When the date in the [Hire Date] field is edited, a confirmation message appears. Choose OK to confirm your change, or Cancel to cancel your change.

NOTE: This example uses the MsgBox statement instead of the MsgBox action. The MsgBox action has only an OK Button, whereas the MsgBox statement can also include a Cancel button.

References:

"Language Reference," version 1.0, pages 316-320

"User's Guide," version 1.0, pages 528-554

[References](#)

PRB: Erroneous Characters Appear in Imported dBASE IV Database

Article Number: Q99399
CREATED: 27-MAY-1993
MODIFIED: 15-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Erroneous characters, specifically Chr(236) and Chr(10), appear approximately every 65th character in a memo field of an imported dBASE IV database.

CAUSE

This behavior is inherent in the dBASE IV data. dBASE IV uses these characters so that its memo editor word wraps automatically. The dBASE IV memo editor has a fixed line length of 65 characters; the mechanism is stored in the data so the memo editor does not have to force the word wrap.

RESOLUTION

Using the code and procedure provided below, you can remove these characters without damaging the data, even if you want to eventually export the data back to dBASE IV.

MORE INFORMATION

1. Copy the following code into a new module in your database:

```
Sub DB4MemoFix (TableName As String, FieldName As String)
    Dim D As Database, T As Table
    Set D = CurrentDB()
    Set T = D.OpenTable(TableName)
    T.MoveFirst
    Do Until T.EOF
        If T(FieldName) <> "" Then
            oldmemo$ = Trim(T(FieldName))
            newmemo$ = ""
            For i = 1 To Len(oldmemo$)
                If Mid(oldmemo$, i, 2) = (Chr(236) & Chr(10)) Then
                    i = i + 1
                Else
                    newmemo$ = newmemo$ & Mid(oldmemo$, i, 1)
                End If
            Next
            T.Edit
            T(FieldName) = newmemo$
            T.Update
        End If
        T.MoveNext
    End Do
End Sub
```

```
Loop
D.Close
Debug.Print
Debug.Print "Done"
End Sub
```

2. From the File menu, choose Save.
3. From the View menu, choose Immediate Window.
4. Type the following

```
DB4MemoFix <tablename>, <fieldname>
```

where <tablename> is the name of the imported dBASE IV table and <fieldname> is the name of the memo field that contains the erroneous characters. For example:

```
DB4MemoFix "CUST", "NOTES"
```

Once the process is completed, the word "Done" will appear in the Immediate window and the erroneous characters will have been removed.

dBASE IV is manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

INF: Formatting a Label for Each Axis of a Graph

Article Number: Q99398
CREATED: 27-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The Microsoft Access [GraphWizard](#) does not give users the option to format a [label](#) for each axis of the [graph](#). You can modify the labels in Microsoft Graph, but this change is not permanent. The next time the graph is updated, the formatting is lost.

This article describes the steps to change the format permanently by modifying the information that Microsoft Access sends to Microsoft Graph.

MORE INFORMATION

When a graph is created using GraphWizard, a [SQL](#) statement is built. This statement is placed in the RowSource property of the graph object. Each time the [form](#) or [report](#) in which the graph resides is refreshed, the SQL statement is re-executed and the results are sent to Microsoft Graph.

This SQL statement can be modified by the user. We recommend that you copy it to a new [query](#), where you can test your changes.

The example below describes the steps to change the format of the labels in a graph, using the Orders [table](#) in the sample [database](#) NWIND.MDB:

Procedure A - Create the Graph

1. Open the sample database NWIND.MDB. In the [Database window](#), select the Orders table.
2. On the Toolbar, choose the Forms button to create a new form.
3. Choose FormWizards. Select Graph and then choose OK.
4. To accept the defaults, press Next when the following message appears: "Which type of graph do you want?"
5. Select the Order Date [field](#) from the Available Fields list and choose the right arrow to add it to the Fields for Graph list.
6. Repeat step 5 for Order Amount, then choose Next twice to accept the default (Group [Order Date] by: Year). Choose Open.

Note that the labels for the axes are: 1989, 1990, 1991, 1992

Procedure B - Changing the Label Format

1. Switch to [Design view](#). Select the chart object and display the [property sheet](#).
2. Place your insertion point in the RowSource property and press SHIFT+F2 to open the Zoom window.
3. Highlight the entire SQL Statement to select it. Press CTRL+C to copy it to the [Clipboard](#).
4. Create a new query by choosing the Query button on the Toolbar.
5. From the View menu, choose SQL.
6. Highlight all the text in the SQL [dialog box](#) to select it. Press CTRL+V to replace the existing text with the Clipboard contents.
7. Choose OK. The fields [Order Date] and [Order Amount] are placed in the query grid.
8. Change the field [Order Date] to Format([Order Date],"YY"). Execute the query to verify that the years in the Order Date [column](#) are formatted as follows: 89, 90, 91, 92.
9. Switch to Design view. Copy the contents of the SQL dialog box to the Clipboard.
10. Return to the form. Select the chart object and paste the contents of the Clipboard into the RowSource property.
11. Switch to [Form view](#). The labels of the axes should now read as follows: 89, 90, 91, 92.

[References](#)

PRB: Paradox Character Translation Different in Win 3.0 & 3.1

Article Number: Q99397
CREATED: 27-MAY-1993
MODIFIED: 25-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
 - Microsoft Windows versions 3.0 and 3.1
-

SYMPTOMS

When you are importing, attaching, or exporting Paradox tables, certain characters translate differently when you are using Microsoft Windows version 3.0 instead of version 3.1.

CAUSE

Windows 3.0 and 3.1 use different versions of the Windows application programming interface (API) [function](#) `OemToAnsi()`.

STATUS

This behavior is by design in Windows 3.0 and 3.1.

MORE INFORMATION

When importing Paradox data, Windows 3.0 translates ASCII characters 246, 251, and 256 differently than Windows 3.1. The differences are shown in the [table](#) below:

ASCII	Windows 3.0	Windows 3.1
246 ÷	95 _	247 ÷
249 ·	183 ·	149 o
254 o	168 İ	95 _

Paradox is manufactured by Ansa Software, a Borland company, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

INF: Calling RegQueryValue() to get an OLE Object Class Name

Article Number: Q99322
CREATED: 26-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

The RegQueryValue() [function](#), found in the Microsoft Windows SHELL.DLL file, uses an [OLE ClassKey](#) (such as PBRUSH or MSGRAPH) to obtain the information about that class from the REG.DAT file.

This article describes the way this function obtains the Object Class Name from the OLE Registration Database REG.DAT file, located in the WINDOWS subdirectory.

Though only the procedure to obtain the Object Class Name is described here, other information, including the current location of the OLE server and the types of <Primary VERBS> available to the object, is also available.

MORE INFORMATION

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
Declare Function RegQueryValue Lib "shell" (ByVal hkey&,_
    ByVal SubKeyStr$, ByVal QueryStr$, Buffsize&) As Long

Function GetObjectName$ (TheClassKey$)
    Dim ClassName$
    ClassName$ = Space$(255)
    X& = CLng(Len(ClassName$))
    cb = RegQueryValue(1, TheClassKey$, ClassName$, X&)
    GetObjectName$ = ClassName$
End Function
```

This function declares a [string](#) to pass as a parameter for a return value in the RegQueryValue() function. RegQueryValue() has four parameters:

HKEY:	The handle to a key that defines what type of information you want to retrieve
lpszSubKey:	A string that contains the OLE ClassKey that you want to look up
lpszValue:	A string in which to place the resulting value
lpcb:	The maximum length of the resulting string

To use the function mentioned above, create a control on a form with the following properties:

```
ControlName: ObjName  
ControlSource: =GetObjectName ("PBRUSH")
```

Any appropriate OLE ClassKey in the REG.DAT OLE database can be passed to the function. For information on finding the ClassKey of an embedded object, query on the article Q99319 here in the Microsoft Knowledge Base.

References:

"Microsoft Windows Programmer's Reference," version 3.1, Volume 2, page 282

[References](#)

INF: Query Optimizing Tips for Attached SQL Database Tables

Article Number: Q99321
CREATED: 26-MAY-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Special considerations must be made for performance optimization of queries built on attached [SQL database](#) tables. A SQL database, for this article, is defined as any client/server database that supports some level of ANSI SQL as an intrinsic part of the database's programming language. Optimal [query](#) strategy for these queries is to ensure that all query operations are performed on the server. This article outlines tips on how to ensure that queries against attached SQL database tables are performed on the server.

This article assumes a basic understanding of client/server computing environments and architectures.

MORE INFORMATION

The key to improving query performance on attached SQL database tables is to ensure that no data is filtered on the client. Filtering data on the client increases [network](#) traffic and does not allow for leveraging of advanced server hardware, essentially turning a client/server system into a file server system. To this end, keeping the query evaluation on the server reduces overhead and keeps an application running as fast as possible.

Generic query optimization techniques should not be ignored when you are using attached SQL database tables. "[WHERE clause](#)" restrictions, such as equality or range comparisons, and sorting should still be performed on indexed fields. For more information on query optimization, call Microsoft Consumer Sales at (800) 426-9400 to order a "white paper" on SQL Server optimization. Other SQL database vendors may provide the same information for their customers.

Use care when implementing intrinsic or user defined functions (UDFs) in query fields or when using [criteria](#) that are not supported on the server. Generally, SQL databases have functionality that corresponds to most standard Microsoft Access functions, but each server will be different.

Many intrinsic Microsoft Access functions have direct back-end correspondents. Microsoft Access asks the [ODBC](#) driver about intrinsic [function](#) support and performs the appropriate mappings.

You can use UDFs and Microsoft Access intrinsic functions without server equivalents when they are accompanied by server-capable restrictions that prune down the data. For example, the following query

Query1a: SELECT * FROM MillionRowTable WHERE Funk1(coll) = 10

returns the whole table and evaluates Funk1(coll) = 10 locally, whereas the following query

Query1b: SELECT * FROM MillionRowTable WHERE Funk1(coll) = 10
AND LastName BETWEEN 'g' AND 'h'

sends the BETWEEN 'g' AND 'h' restriction to the server, returns the qualifying rows, and evaluates Funk1(coll) on only those rows.

Non-remote-capable SELECT list items don't force a query to be executed locally, unless they are used with unique values or a totals query (DISTINCT/GROUP BY). For example, the following query

Query2a: SELECT * FROM MyTable WHERE Format(coll, ...) = 10

returns the whole table and causes the WHERE clause to be evaluated locally. However, the following query

Query2b: SELECT Format(coll,...) FROM MyTable WHERE col2 = 10

sends "SELECT col1 FROM MyTable WHERE col2 = 10" to the server, presumably returning far less data over the network. It then locally evaluates Format() on the coll values returned.

Of the following two queries, Query3a is sent completely to the server. Query3b sends "SELECT col1 FROM MyTable" and performs the Format() function, and therefore the DISTINCT clause, locally.

Query3a: SELECT DISTINCT col1 FROM MyTable
Query3b: SELECT DISTINCT Format(coll,...) FROM MyTable

The following two queries are performed as follows: Query4a is sent completely to the server. Query4b sends "SELECT col1 FROM MyTable" and performs the StdDev() aggregate function locally, since it's not a SQL standard function.

Query4a: SELECT Sum(coll) FROM MyTable
Query4b: SELECT StdDev(coll) FROM MyTable

Crosstab queries present unique restrictions, some pertinent to all queries, some to crosstab queries only:

1. Only standard aggregate functions, such as Count(), Sum(), Min(), Max(), and Avg(), can be used.
2. Aggregate functions can't be used as row or column headers.
3. Only one aggregate function can be used in the "value."
4. Nothing can be sorted.
5. If a fixed-value list of column headers is supplied, the value can't be embedded in an expression.

Restriction #1 applies to all queries against SQL databases from Microsoft Access. Except for #4, the others are uncommon and are crosstab/SQL database specific. Crosstab queries are not SQL standard, and are not generally supported on SQL database servers. However, if the rules above aren't violated (which is the case for most simple crosstab queries), Microsoft Access can reformulate the query as a standard GROUP BY query, send it remotely, return only the aggregation result, and transform it into a crosstab locally.

Open-ended restrictions do not use indexes on SQL databases. Typically, a SQL WHERE clause reading

```
WHERE col1 > 1000
```

will be slower than

```
WHERE col1 between 1000 and 1000000000
```

This is a server problem, not a Microsoft Access problem, but it can affect Microsoft Access performance when a SQL database is used as a back end.

When you are using wildcard characters, take special care to ensure that the correct wildcards are used. SQL Server supports the use of % and _ rather than ? and * for wildcards. For more information, query on the following words in the Microsoft Knowledge Base:

parameter and queries and wildcards and SQL

[References](#)

PRACC1405: System Hangs when Minimizing Access Window

Article Number: Q99320
CREATED: 26-MAY-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Your system stops responding (hangs) when you have switched away from Microsoft Access, started another application, switched back to Microsoft Access, and pressed the Minimize button.

This problem occurs only if Microsoft Access is running with Windows for Workgroups.

WORKAROUND

Clicking anywhere in the Microsoft Access window other than the Minimize button does not cause the system to hang. After clicking anywhere else, you can use the Minimize button without hanging the system.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here as it becomes available.

Steps to Reproduce Problem

1. Start Microsoft Access.
2. From the Help menu, choose [Cue Cards](#).
3. Select the first option, Design a Database with Tables.

NOTE: You must be at a Cue Card screen with a gray background.

4. Switch to Program Manger and start Calculator.
5. Press CTRL+ESC to bring up Task List and choose Microsoft Access.
6. Press the Minimize button.

Your system now stops responding; you must reboot your computer.

[References](#)

INF: Finding the OLE ClassKey of an Embedded Object

Article Number: Q99319
CREATED: 26-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access places a wrapper of information around an OLE object before it stores the object in a table. The Name and ClassKey of the object start at byte count 21 and are stored from there as two zero-terminating strings. By reading the field as a text field, you can parse out the ClassKey.

This article describes how to use Access Basic to find the ClassKey of an embedded object with an OLE Object data type.

MORE INFORMATION

When an OLE object is stored in a table, the ClassKey is the second Null-terminated string after the 20th byte in the object. It can be read as text with the following Access Basic code:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function GetObjectClass$ ()
  If IsNull(screen.activeform!Photo) Then
    GetObjectClass$ = "n/a"
  Else
    MyChunk$ = Mid(screen.activeform!Photo, 21, 40)
    NullOne% = InStr(1, MyChunk$, Chr$(0))
    NullTwo% = InStr(NullOne% + 1, MyChunk$, Chr$(0))
    GetObjectClass$ = Mid(MyChunk$, NullOne% + 1, _
                          NullTwo% - NullOne% - 1)
  End If
End Function
```

The GetObjectClass\$() function first checks to see if the object is null and returns the value "n/a" if it is. If the object is not null, it assigns bytes 21 through 40 of the OLE object to the string MyChunk\$ and then searches for the first Null character, the next Null character, and so forth. Finally, it sets the GetObjectClass\$() function equal to the string between the two Null characters.

Using the function provided above, a control on the Employees form of the sample database NWIND.MDB, with the properties below, displays the ClassKey for the Photo control:

ControlName: ObjClassKey

```
ControlSource: =GetObjectClass()
```

This example returns "PBRUSH" for the first photo in the Employees table.

There are several different ways to use this information. RegQueryValue(), a function contained in the Microsoft Windows SHELL.DLL file, uses the ClassKey to obtain information about that class from the REG.DAT file.

References:

For more information about calling the RegQueryValue() function to get the Class Name of an OLE object, [query](#) on article Q99322 here in the Microsoft Knowledge Base.

"Microsoft Windows Programmer's Reference," version 3.1, Volume 2, page 282

[References](#)

PRACC1115: ApplyFilter Action Causes Form to be Read-Only

Article Number: Q99318
CREATED: 26-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

Running a [macro](#) that contains an ApplyFilter action may cause a [form](#) to become [read-only](#). This problem occurs if the form is based on a [query](#) that contains two tables with a [one-to-many relationship](#) and only records from the one side are included in the output of the query.

RESOLUTION

There is no way to prevent the ApplyFilter action from making the form read-only. However, you can use one of the following methods to [filter](#) the records displayed in a form:

- Base the form on a [parameter query](#). The user will then be prompted for the [criteria](#) with which to filter the records each time the form is opened. For more information on parameter queries, refer to pages 178-181 in the "User's Guide."
- Place the form in another form as a [subform](#). Set up the LinkMaster and LinkChild properties so that one or more fields in the main form filter the data in the subform. For more information on subforms, refer to the pages 272-278 in the "User's Guide."
- Use [Access Basic](#) code to [control](#) the records displayed in the form. For more information, refer to pages 106-107 in the "Introduction to Programming" manual.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in version 1.1.

Steps to Reproduce Problem

1. In the sample [database](#), NWIND.MDB, create a new query based on the Customers and Orders tables. Join the tables on [Customer ID].
2. Drag all fields from the Customers [table](#) to the query grid. Drag down [Order ID] from the Order table. Add the criteria: >10080 and clear the Show box.
3. Save the query and create a new form based on it. Drag several fields from the [field list](#) to the form.

4. Create a macro with an ApplyFilter action. Set the macro's Where Condition argument to

[Company Name] Like "c*"

5. Save the macro and drag it onto the form.
6. Switch to Form view. Verify that you can edit fields.
7. Press the macro button to apply the filter.

You can no longer edit or add records.

[References](#)

INF: Exporting WordPerfect Secondary Files to Microsoft Access

Article Number: Q99317
CREATED: 26-MAY-1993
MODIFIED: 10-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to export data from a WordPerfect print merge secondary file to Microsoft Access.

MORE INFORMATION

A WordPerfect secondary file contains records, typically names and addresses, used in a print merge with a primary file, usually a [form](#) letter or mailing [label](#) document. To use data from a secondary file, the information must be saved in a comma or tab delimited (separated), unformatted text file. WordPerfect secondary files are neither comma nor tab delimited.

A typical comma [delimited text file](#) resembles the following:

```
"Joe","Smith","123 Main Street","Seattle, WA 98033"  
"Martha","Jones","4523 Williams Ave","","Redmond, WA 98052"
```

The equivalent WordPerfect secondary file resembles the following:

```
Joe{END FIELD}  
Smith{END FIELD}  
123 Main Street{END FIELD}  
Seattle, WA 98033{END FIELD}  
{END RECORD}  
<CRLF>  
Martha{END FIELD}  
Jones{END FIELD}  
4523 Williams Ave{END FIELD}  
{END FIELD}  
Redmond, WA 98052{END FIELD}  
{END RECORD}  
<CRLF>
```

where <CRLF> represents a hard [page break](#).

Problems: Secondary Files With Multi-Line Field Entries

WordPerfect allows you to merge multiple lines of data in a single [field](#). The following [record](#) has only two fields:

```
Joe Smith{END FIELD}  
123 Main Street
```

```
Suite 101
Seattle, WA 98033{END FIELD}
{END RECORD}
<CRLF>
```

Delimited text files use carriage returns to mark the end of a record; these cannot be included in the field information. Each line in the secondary file must be made a unique field before it can be prepared as a delimited file. When you add the {END FIELD} code to a record, the new field is represented in every record in your file. Each record must contain the same number of fields and each field should either contain the same type of data or remain empty.

Using the {FIELD NAMES} Section in a Secondary File

Fields in a secondary file can be referenced from a primary file by number. These fields are numbered from top to bottom, or by a name that is assigned in the optional {FIELD NAMES} section, which appears at the top of the secondary file. The {FIELD NAMES} section may look like this

```
{FIELD NAMES}
Name~
Address~
~{END RECORD}
```

or like this

```
{FIELD NAMES}Name~Address~~{END RECORD}
```

If referenced by number, the Name field is 1 and the Address field is 2. A hard page break falls between this section and the first record.

Preparing the Comma Delimited Secondary File

WordPerfect for Windows:

WordPerfect for Windows can save secondary files directly in a comma delimited format. However, if you have a {FIELD NAMES} section in the file and want to preserve these names in Microsoft Access, you must reformat the file so that each name is separated by a comma and not by a tilde (~) character. Also, the fields should not appear on separate lines, nor be followed by carriage returns. The following {FIELD NAMES} section

```
{FIELD NAMES}
Name~
Address~
Phone~
~{END RECORD}
```

should be editing as follows:

```
{FIELD NAMES}Name,Address,Phone{END RECORD}
```

Creating the Comma Delimited File:

1. From the File menu, choose Save As.
2. In the Save As box, type a unique file name, preferably with a .TXT extension.
3. In the Format box, select ASCII Delimited Text (DOS), then choose Save.

WordPerfect for MS-DOS:

WordPerfect for MS-DOS cannot save secondary files directly in a delimited format. To create the comma delimited file, you need to create a primary file and merge the secondary information into it. The primary document will contain fields already in comma delimited format. Once the data is merged, remove the hard page breaks separating each record. The resulting document can be saved as a text file that Microsoft Access can import.

Creating the Primary File:

1. Press F7, choose No, then choose No again to create a new WordPerfect document.
2. Type an opening quotation mark. Press SHIFT+F9, then 1 (Field).
3. In the Enter Field box, type the name of the first field if a {FIELD NAMES} section appears in the secondary file, or type "1" if there is no {FIELD NAMES} section.
4. Press ENTER. Type a closing quotation mark, followed by a comma.
5. Repeat steps 2-4 for each field in the secondary file, changing only the field name or field position number as appropriate. Do not type a comma after the last field.
6. Press ENTER to place a hard carriage return at the end of the record.
7. Press F10, type a unique filename in the Document To Be Saved box, and press ENTER.

The primary document will resemble the following:

```
"{FIELD}name~","{FIELD}address~","{FIELD}phone~"
```

NOTE: It is OK if the lines wrap on the screen.

Merging the Primary and Secondary Files:

1. Press F7, choose No, then choose No again to create a new WordPerfect document.
2. Press CTRL+F9, then 1 (Merge).
3. In the Primary File box, type the name of the primary file and

press ENTER.

4. In the Secondary File box, type name of the secondary file and press ENTER.

Removing the Residual Hard Page Breaks:

1. Press ALT+F2, then N to disallow replacements.
2. At the -> Srch prompt, press CTRL+ENTER to enter a [HPg] code.
3. Press ALT+F2 twice to pass the Replace With message.

Saving As an Unformatted Text File:

1. Press CTRL+F5, then 1 (DOS Text), then 1 (Save).
2. Type a valid document name, preferably with a .TXT extension, and press ENTER.

REFERENCES

=====

"WordPerfect for DOS Reference," version 5.1, pages 420-431
"WordPerfect for DOS Workbook," version 5.1, pages 237-246
"WordPerfect for Windows Reference," version 5.1, pages 300-308
"WordPerfect for Windows Workbook," version 5.1, pages 203-210

[References](#)

PRB: Paste Link to MS Excel Displays First Field Name Only

Article Number: Q99316
CREATED: 26-MAY-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
 - Microsoft Excel for Windows, versions 1.x, 2.x, and 3.0
-

SYMPTOMS

When you paste [link](#) a [table](#) from Microsoft Access to Microsoft Excel versions 3.0 and earlier, only the first [field](#) name of that table appears in the spreadsheet.

CAUSE

Microsoft Excel version 3.0 and earlier needs to know the exact dimensions of the spreadsheet area that you are pasting to before you choose the Paste Link command.

RESOLUTION

To paste link a table to Microsoft Excel versions 3.0 and earlier, first highlight the entire area on your spreadsheet. Another option is to use the Paste command, rather than the Paste Link command.

MORE INFORMATION

The following steps will result in an unsuccessful paste link using Microsoft Excel versions 3.0 and earlier. These same steps will successfully paste link a table to later versions of Microsoft Excel.

Steps to Reproduce Behavior

1. Start Microsoft Access and open the sample [database](#) NWIND.MDB.
2. In the [Database window](#), select the Customers table.
3. From the Edit menu, choose Copy.
4. Start Microsoft Excel and create a new worksheet.
5. From the Microsoft Excel Edit menu, choose Paste Link.

References:

"Microsoft Excel User's Guide," version 3.0, pages 318-319

[References](#)

PRACC9304: GPF Printing Subreport with Datasheet Default View

Article Number: Q98810
CREATED: 13-MAY-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

Reports do not have a [Datasheet view](#) as forms do; however, forms created with a default Datasheet view can be saved as reports. This can cause problems in Microsoft Access.

If you save a [form](#) with the DefaultValue property set to Datasheet as a [report](#) and that report is subsequently embedded as a [subreport](#), Microsoft Access generates a general protection (GP) fault when the report is previewed or printed.

This problem can occur only if you create the subreport using the Save As Report command on the File menu. (To see this command, you must be in [Design view](#) of the form.)

RESOLUTION

When using the Save As Report command to create a report from a form, be sure the DefaultValue property of the form is NOT set to Datasheet.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

INF: Operators as Field Names in Query Cause Error Message

Article Number: Q98809
CREATED: 13-MAY-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you use a logical operator as a [field](#) name in a [table](#) and then create a [query](#) that uses any of these fields, you receive the following error message:

Syntax error in query [expression](#) "<Table>.<Field>".

where <Table> is the name of the table you are using in the query and <Field> is the name of the field that is causing the problem.

MORE INFORMATION

To correct this problem, open the table in [Design view](#) and change the name of the field listed in the error message to a name that is not one of the logical operators.

The following are logical operators:

And, Between, Eqv, IIF, Imp, In, Is, Like, Mod, Not, Or, and Xor

For more information, search for "logical operators" using the Help menu.

[References](#)

PRB: <Expression> Not Part of Aggregate Function or Grouping

Article Number: Q98808
CREATED: 13-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If a select [query](#) that contains an [expression](#) is converted to a totals query, the following error message is displayed when the query is run:

<expression> not part of [aggregate function](#) or grouping.

CAUSE

When a [select query](#) is converted to a [totals query](#), all expressions are assumed to contain aggregate functions and cause the Total [field](#) to default to Expression.

RESOLUTION

To work around this problem, set the Total field to Expression and include an aggregate [function](#) as part of the expression for Field Name. If you cannot add an aggregate function, set Total to Group By as shown:

```
Query: Query Name
-----
Field Name: <expression>
Total: Group By
```

MORE INFORMATION

Steps to Reproduce Behavior

Using the sample [database](#) NWIND.MDB, complete the following steps:

1. Open the Employee List query in [Design view](#) by choosing the Query button in the [Database window](#), selecting the query name, and choosing Design.
2. From the View menu, choose Totals to convert the query to a totals query.
3. From the Query menu, choose Run. The following error message is displayed:

"[Last Name] & ", " & [First Name]" not part of aggregate function or grouping.

4. In the query grid, make the following changes:

Query: Employee List

Field name: Employee Name:[Last Name] & ", " & [First Name]

Total: Group By

REFERENCES

=====

"User's Guide," version 1.0, pages 141-149

[References](#)

PRB: Must Have Database Open to Use Database Analyzer Utility

Article Number: Q98807
CREATED: 13-MAY-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You may receive the following error message when attempting to start the Database Analyzer utility:

(2046) An error has occurred in Analyzer at:
Command not available: OpenForm

CAUSE

This behavior is by design.

RESOLUTION

Make sure that you always have a [database](#) open before starting the Database Analyzer utility. You will receive this error message only if you start Database Analyzer with no database open.

MORE INFORMATION

Steps to Reproduce Behavior

1. Add Database Analyzer to your Microsoft Access Help [menu bar](#).

For information about how to do this, [query](#) on the following words in the Microsoft Knowledge Base:

database and analyzer and help and menu

2. Open Microsoft Access, but do not open a specific database. You should see only a File menu and a Help menu.
3. From the Help menu, choose Database Analyzer. The above error message is displayed.
4. Open any database and repeat step 3. You do not get the above error message and you are able to use the Database Analyzer utility.

[References](#)

PRB: Seek Method Is Faster Than Find Method

Article Number: Q98806
CREATED: 13-MAY-1993
MODIFIED: 16-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you are using Access Basic to find a record in an indexed field, the Seek method may be faster than the Find method, especially in a large table.

CAUSE

This behavior is by design.

RESOLUTION

Use the Seek method on indexed fields to optimize your search speed.

MORE INFORMATION

When you perform a Seek, you are opening a table directly and moving to the record based on an index value. When you create a dynaset and perform a Find, you are checking the value of the field in every record until you find a match.

Steps to Reproduce Behavior

1. Open the sample database NWIND.MDB.
2. Create a new module and enter the following two functions:

```
'*****  
'Declarations section of the module.  
'*****  
Option Explicit  
  
'=====
```

```
'The following function uses the Seek method.  
'It quickly returns the Customer ID where the  
'   PrimaryKey field = 11070.  
'=====
```

```
Function Faster ()  
    Dim db As Database, tbl As Table  
  
    Set db = CurrentDB()  
    Set tbl = db.OpenTable("Orders")  
  
    tbl.Index = "PrimaryKey"
```

```
tbl.Seek "=", 11070
Debug.Print tbl("customer id")
tbl.Close
End Function
```

```
'=====
'The following function uses the Find method.
'It is slightly slower in returning the Customer
' ID where the PrimaryKey field = 11070.
'=====
```

```
Function Slower()
    Dim Criteria As String, MyDB As Database, Myset As Dynaset

    Set MyDB = CurrentDB()
    Set Myset = MyDB.CreateDynaset("Orders")

    Criteria = "[Order id] =" & 11070
    Myset.FindNext Criteria
    Debug.Print Myset("Customer id")
End Function
```

3. From the View menu, choose Immediate Window.

4. In the Immediate window, type

```
?Faster()
```

and press ENTER. Then type

```
?Slower()
```

and press ENTER.

The Faster() function, which uses Seek, is slightly faster than the Slower() function, which uses FindNext.

References:

"Microsoft Access Language Reference," version 1.0, pages 187-188 and 426-427

[References](#)

PRB: Internal Database Error (-5207) When Attaching Tables

Article Number: Q98805
CREATED: 13-MAY-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

When attaching to a French version Paradox file, the following error message may appear:

Internal Database Error -5027

CAUSE

The MSACCESS.INI file settings are incorrect for reading the international file.

RESOLUTION

Set CollatingSequence entry in the [Paradox ISAM] section of the MSACCESS.INI to International as follows:

```
[Paradox ISAM]
CollatingSequence = International
```

MORE INFORMATION

For more information search for "MSACCESS.INI," then "Customizing MSACCESS.INI Settings" using the Help menu.

[References](#)

INF: Cannot Use ANALYZER.MDB Filename for Output Database

Article Number: Q98804
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The product name Analyzer is reserved for the Database Analyzer utility that ships with Microsoft Access. You cannot use this name for an output [database](#) when you are running this utility.

MORE INFORMATION

You can create a database called Analyzer (ANALYZER.MDB) without any problems, but you cannot use this database as an output destination when running the Database Analyzer utility.

You will not receive any error messages if you attempt to use ANALYZER.MDB as an output database, but the object(s) that you selected will not be analyzed.

References:

For more information about the Database Analyzer utility, refer to Question #23 in the PSSKB.TXT file. This text file is located in your Access program directory and can be opened in Notepad.

[References](#)

PRB: Microsoft Access Cannot Find DESKEDIT.EXE File

Article Number: Q98803
CREATED: 13-MAY-1993
MODIFIED: 25-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Attempting to open the Microsoft Access README.TXT file or clicking the Microsoft Access Q&A icons may produce the following error message:

Cannot find DESKEDIT.EXE.

CAUSE

This error results when Norton Desktop for Windows, one of the Norton Utilities, has been deleted from your computer. This change causes Microsoft Windows to lose the association with the .TXT files.

RESOLUTION

In Windows File Manager, reassociate your text files using Notepad, as follows:

1. Open the Access subdirectory in File Manager.
2. Select the README.TXT file.
3. From the File menu, choose Associate.
4. In the Associate dialog box, choose Text File (NOTEPAD.EXE).
5. Press ENTER. Your text files are now reassociated.

MORE INFORMATION

Norton Utilities is manufactured by Peter Norton Computing, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

INF: Date/Time Data Type Provides Current Year Automatically

Article Number: Q98802
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you enter a partial date in a [field](#) formatted with the Date/Time [data type](#), Microsoft Access fills in the complete date automatically.

MORE INFORMATION

If your data is formatted with the Date/Time data type and you want to indicate the current year, it is not necessary to enter the year. This feature works for both tables and forms.

Steps to Reproduce Behavior

1. Create a new [table](#), as follows:

Table: DataTable

FieldName: My Date [[primary key](#)]
DataType: Date/Time

2. From the View menu, choose Datasheet.
3. Type 1/1, then press TAB to leave the field.

The value you entered is displayed as 1/1/93.

[References](#)

INF: How to Print a Single Record from a Form in a Report

Article Number: Q98801
CREATED: 13-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to select a single [record](#) on a [form](#) and print it in a [report](#).

If you want to print more than one record, page 558 of the "User's Guide" describes a way to restrict which records you print by changing either the Filter Name or Where Condition arguments in the [macro](#).

If you do not want to use a report, page 543 of the "User's Guide" describes a way to print the [current record](#) directly from the form using a macro.

MORE INFORMATION

The following procedure explains how to print only the record you select from the Customers form in the sample [database](#) NWIND.MDB:

1. Create a report called PrintRecordReport based on the Customers [table](#).
2. Create the following macro to open the report:

Macro: PrintRecordMacro

OpenReport

ReportName: PrintRecordReport

View: Print Preview

Where Condition: [Customer ID]=Forms![Customers]![Customer ID]

3. Add a [command button](#) with the following properties to the Customers form to run the macro:

Command Button

ControlName: PrintRecord

Caption: Print Record

OnPush: PrintRecordMacro

When you choose the command button on the form, you open the report and run the [query](#) based on it. The record selected by the cursor position on the form is returned by the query and is printed on the report.

[References](#)

INF: Tile/Cascade Menu Options Inconsistent with Prog Manager

Article Number: Q98800
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The order in which the Tile and Cascade options appear on the Window menu in Microsoft Access are not the same as in Microsoft Windows Program Manager. Also, SHIFT+F4 and SHIFT+F5 are not shown as shortcut key combinations for Tile and Cascade, respectively.

MORE INFORMATION

The Tile and Cascade menu options function in exactly the same way in Microsoft Access as they do in Program Manager.

For more information, search for "shortcut keys" using the Microsoft Access Help menu.

[References](#)

PRB: CTRL+C Does Not Copy SQL Statement in Windows 3.0

Article Number: Q98799
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Since you cannot choose options from the Edit menu when you have a dialog box open, if you want to copy the SQL text of a query, you must press CTRL+C instead of choosing Copy from the Edit menu. However, this key combination does not work if you are running Microsoft Access with Microsoft Windows version 3.0.

CAUSE

Even though Microsoft Access Edit menu displays CTRL+C as the equivalent for the Copy command, this key combination is defined by Windows and not by Microsoft Access. CTRL+C is a Windows 3.1 key combination. If you are running Windows 3.0, press CTRL+INS instead.

NOTE: You can use either key combination in Windows 3.1.

STATUS

This behavior is by design.

MORE INFORMATION

The following are the Edit menu options and the equivalent shortcut key combinations for Windows 3.0 and 3.1:

Edit Menu Options	Windows 3.0	Windows 3.1
Cut	SHIFT+DEL	CTRL+X or SHIFT+DEL
Copy	CTRL+INS	CTRL+C or CTRL+INS
Paste	CTRL+V or SHIFT+INS for both versions	

References

INF: Creating Reports to Mail Merge Microsoft Access Data

Article Number: Q98798
CREATED: 13-MAY-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

To execute a mail merge with Microsoft Access data, you can export or copy your data to a word processor. But you can also create a Microsoft Access [report](#) to generate simple mail-merge documents. This article explains the latter process.

MORE INFORMATION

The following example produces a business letter:

NOTE: All values shown in angle brackets (<>) are found in the underlying [table](#).

KB Advertising Agency

<February 5th, 1993>

<John Doe>

<123 E. Main Street>

<Middle Town, OH 44044>

Dear <Mr. Doe>,

We are pleased to announce our new line of office furniture. As a <valued> customer, you are entitled to a <10>% discount off our catalog prices.

Call us today toll-free at (800) 555-1212 and receive a free full-color catalog detailing all our top-of-the-line office products.

Sincerely,

J. Grant

Manager, Office Products

Example

1. Create the following table:

Table: Customer

FieldName: Salutation
DataType: Text

FieldName: First Name
DataType: Text

FieldName: Last Name
DataType: Text

FieldName: Address
DataType: Text

FieldName: City
DataType: Text

FieldName: State
DataType: Text

FieldName: ZIP
DataType: Text

FieldName: Customer Type
DataType: Text

FieldName: Discount Pct
DataType: Number

2. Enter the following records:

Salu- tation	First Name	Last Name	Address	City	State	ZIP	Cust Type	Disc Pct
-----	-----	-----	-----	-----	-----	-----	-----	-----
Mr.	John	Doe	123 Main	Middleton	OH	44044	Valued	10
Mrs.	Sally	White	52A Elm	Florence	KY	45123	Special	5
Dr.	Fred	Weiss	Box 456	Camden	SC	29332	Super	25

3. Create a new, blank report as follows:

NOTE: Do not use a Microsoft Access ReportWizard.

- From the Layout menu, turn off Report Hdr/Ftr and Page Hdr/Ftr.
- Add all the controls to the Details section.
- Delete the default text box labels.

Report: Merge Letter

RecordSource: Customer

Section: Detail

ForceNewPage: After Section
CanGrow: Yes

Control: Label

ControlName: Company Label
Caption: KB Advertising Agency

Control: Text Box

ControlName: Todays Date
ControlSource: =Date()
Format: Long Date

Control: Text Box

ControlName: Name Line
ControlSource: =[First Name] & " " & [Last Name]

Control: Text Box

ControlName: Address Line
ControlSource: =[Address]

Control: Text Box

ControlName: CSZ Line
ControlSource: =[City] & ", " & [State] & " " & [ZIP]

Control: Text Box

ControlName: Salutation Line
ControlSource: ="Dear " & [Salutation] & " " & [Last Name] & ", "

Control: Text Box

ControlName: Para 1
ControlSource: ="We are pleased to announce our new line of office
furniture. As a " & [Customer Type] & "customer,
you are entitled to a" & [Discount Pct] & "%
discount off our catalog prices."
CanGrow: Yes

Control: Label

ControlName: Rest of Letter
Caption: Call us today toll-free at (800) 555-1212 and receive a
free full-color catalog detailing all our
top-of-the-line office products.

Sincerely,

J. Grant
Manager, Office Products

4. Size and position the controls, then set the font styles and sizes.

NOTE: Press CTRL+ENTER to force new lines and paragraphs in controls.

5. Save the report. Open the report in Print Preview to see the results of the mail merge.

Summary and General Notes

- Fixed paragraphs should be entered in a [label control](#).
- Variable paragraphs should be entered in a text box control.
- Margins are determined by the width of the control. The text will wrap within the width of the control, not the width of the letter.
- The ControlName of a text box must be different from its [field](#) name in the table. If they are the same, the ControlName prints as a "#Error?" message.

Limitations

- You cannot change the formatting of individual words. You can change only the formatting of the control.
- You are limited to 255 characters in the ControlSource property of a text box. For long paragraphs, create one or more invisible text boxes to contain the bulk of the text, then concatenate them together in a visible text box.

[References](#)

INF: How to Import a Microsoft Windows Cardfile Data File

Article Number: Q98797
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access has no routines to import a Microsoft Windows Cardfile (.CRD) file. To import a .CRD file, you must use Access Basic. This article describes the [Access Basic function](#) you need to import a Windows .CRD file.

MORE INFORMATION

The function below can be used to import a Microsoft Windows Cardfile data file. To use this function, you need to add it to a new or existing Microsoft Access [module](#). You can run the code from the [Immediate window](#), or you can call it from a [command button](#) by setting the properties as follows:

```
Command Button  
-----  
OnPush: =Import_CRD()
```

In this function, you need to specify the correct path and filename of your actual .CRD file. As an alternative, you can pass the filename as a [string](#) to the function.

```
Option Explicit  
  
Type CRD_Header  
    Signature As String * 3 'always MGC  
    Number as Integer  
End Type  
  
Type CRD_Record_Type  
    Reserved As String * 6  
    Position As Long  
    Flag As String * 1  
    Index As String * 40  
    EndOfRecord As String * 1  
End Type  
Dim I As CRD_Record_Type  
  
Function Import_CRD ()  
    Const FIRST_REC = 6  
    Dim MyDB As Database  
    Dim MyTable As Table  
  
    Dim FileNum#
```

```

Dim NumberOfCards%, NextRecord%, I%
Dim CRD_Record As CRD_Record_Type
Dim Message$

Set MyDB = CurrentDB()
Set MyTable = MyDB.OpenTable("CardFile")

FileNum# = FreeFile
'Supply the actual path and name of the file you want to
'convert.
Open "msaccess.crd" For Binary As FileNum#

'The number of files in the .CRD file is at the 4 bytes offset.
Get #FileNum, 4, NumberOfCards%

NextRecord% = FIRST_REC
For I% = 1 To NumberOfCards%
    Get #FileNum, NextRecord%, CRD_Record
    NextRecord% = NextRecord + &H34
    Message$ = Space$(512)
    Get #FileNum, CRD_Record.Position + 5, Message$

    MyTable.AddNew
        MyTable.Title = Mid$(CRD_Record.Index, 1)
        MyTable.Comment = Mid$(Message$, 1, InStr(Message$, Chr$(0)))
    MyTable.Update
Next I%
Close
End Function

```

References:

For more information on the actual details of the .CRD file format, [query](#) on the following words in the Microsoft Knowledge Base:

cardfile and format and graphic and absolute and hexadecimal

[References](#)

INF: Data Type of Value Overrides Data Type of Variable

Article Number: Q98796
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access internally converts values with implicit [data type](#) definitions before assigning that value to a [variable](#). This may result in unexpected rounding of numbers, or in the following error message:

```
Invalid Number
```

MORE INFORMATION

You can implicitly define the data type of a value by appending a type-declaration character at the end of the value. For example, `x=1234&` implicitly defines 1234 as a [Long data type](#) (four-byte integer).

Steps to Reproduce Behavior

1. Open a new [module](#).
2. Add the following line underneath the Option Declare Database line:

```
Dim v as Currency
```

3. Press ENTER.
4. From the View menu, choose Immediate Window.
5. In the [Immediate window](#), type the following:

```
v=123456789123.3456789#
```

6. Press ENTER, then type the following line:

```
debug.print v
```

and press ENTER again.

The value of the variable <v> is now 123456789123.346. The number sign (#) implicitly defined the value as a double-precision value. Microsoft Access internally converted the value to that data type before assigning it to the currency variable.

7. Repeat steps 5-6, replacing # with other type-declaration characters (such as the ampersand [&], exclamation point [!], or at

[@] symbol).

The @ symbol defines the currency data type, which is limited to four decimal places. When you try to define a number with more than four decimal places as currency, an "invalid number" error message is displayed.

References:

"Language Reference," version 1.0, Part 2, pages 25-26

"Introduction to Programming," version 1.0, pages 49-51

[References](#)

PRB: Form May Lose the Focus When SHIFT+F8 Is Pressed

Article Number: Q98795
CREATED: 13-MAY-1993
MODIFIED: 25-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you press SHIFT+F8 to reverse an Extend mode selection, your insertion point may disappear and your [form](#) may no longer have the focus.

CAUSE

This scenario occurs only when you have selected a [record](#) by using the SHIFT+SPACEBAR key combination, rather than pressing the F8 key to turn Extend mode on.

When you press F8, Microsoft Access saves information regarding the previous location of the focus, but not when you press SHIFT+SPACEBAR. Therefore, Microsoft Access is unable to return the focus to your form when you press SHIFT+F8.

RESOLUTION

Press F8 instead of SHIFT+SPACEBAR to select the record, or click anywhere on your form to recover the focus.

MORE INFORMATION

This behavior appears only in [Form view](#), not in [Datasheet view](#).

Pressing SHIFT+SPACEBAR selects the [current record](#) or cancels your record selection.

Pressing F8 turns Extend mode on (EXT appears in the status bar). Once Extend mode is on, repeatedly pressing F8 extends the selection to the word, the [field](#), the single record, and finally to all records. Pressing SHIFT+F8 reverses the selection.

Steps to Reproduce Behavior

1. Open the sample [database](#) NWIND.MDB.
2. Open the Employees form.
3. From the View menu, choose Datasheet.
4. Press TAB to select the Last Name field (Davolio).

5. Press SHIFT+SPACEBAR to select the entire record.
6. Press SHIFT+F8 to cancel the selection.

Note that the focus remains on the Last Name field.

7. From the View menu, choose Form.
8. Repeat steps 5-6 above.

Note that the form no longer has the focus. Click anywhere on the form to restore the focus.

References:

For more information, search for "selection keys" using the Help menu.

[References](#)

PRACC9304: Invalid Results in Subreport When CanGrow = Yes

Article Number: Q98794
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

If a control in a subreport uses an expression or aggregate function and its CanGrow property is set to Yes, Microsoft Access does not calculate the value of the field correctly.

For example, assume your database contains a table named "Test" that has a text field called "TestColumn" containing the following values:

```
TestColumn
-----
Mickey
Donald
```

If you create a report with a single text box (Field1) containing the expression

```
=IIf(IsNull([TestColumn]),"N/A",[TestColumn])
```

and you set the CanGrow property for Field1 to Yes and then embed that report as a subreport, you get the following incorrect results:

```
Donald
Donald
```

If the calculated subreport field contains an aggregate function, instead of a text-based expression as above, the subreport displays incorrect numeric values.

RESOLUTION

Make sure that the CanGrow property of subreport fields that use expressions or calculations is set to No.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

INF: Compound Indexes Must Restrict First Indexed Field

Article Number: Q98793
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

For an [index](#) to be used, queries with restrictions on multiple-[field](#) (compound) indexes must restrict the first field of the index. If the restriction is placed on a field other than the first field, the [query](#) optimizer scans the [table](#), rather than using the indexes. Table scans are slower than index searches for most queries.

MORE INFORMATION

By default, when a compound index is created in Microsoft Access, no individual indexes are assigned to the fields included in the compound index. This behavior is by design. For the query optimizer to use the index, a comparison either of the complete index value or with the first field in the compound index must be used.

Example

Create a table called T1 with has two fields, key_part1 and key_part2, in the [primary key](#) and neither field indexed individually. Select * from T1 where key_part2 = <value> does not use the index, while Select * from T1 where key_part1 = <value> does use the index.

This principle also applies to using [criteria](#) in the Find methods in [Access Basic](#).

The above fields are not prohibited from having indexes on them; individual indexes can be built for each field, allowing comparisons on those fields with index searches. Be aware that the indexes take up as much, or more, space than the data.

[References](#)

INF: Use of Constants as Field Names in a Query

Article Number: Q98792
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you use a constant (Yes, No, True, False, On, Off, or Null) as a field name in a table and then create a query that uses any of those fields, you receive the following error message

Invalid use of '.', '!', or '()' in query expression <Table>.<Field>

where <Table> is the name of the table you are using in the query and <Field> is the name of the field in question.

A constant represents a numeric or string value that doesn't change. In some respects, a constant is like a variable or a reserved word. Unlike a variable, however, the value of a constant cannot change during program execution.

If you receive the above error message, you must change your field name at the table level.

References:

For more information search for "Constants" using the Help menu.

[References](#)

INF: Part 1: DDE in Visual Basic to Request Data from Access

Article Number: Q98791
CREATED: 13-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1,10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.10

SUMMARY

This article explains how to use Microsoft Visual Basic to request data from Microsoft Access using dynamic data exchange ([DDE](#)).

For information that goes into greater detail and demonstrates how to [parse](#) the requested data in a Visual Basic array and then populate a grid [control](#) with the data, [query](#) on the following words in the Microsoft Knowledge Base:

dde and vb and part two

MORE INFORMATION

This article provides a Visual Basic code example demonstrating how to request information from the Employees [table](#) in the sample [database](#) NWIND.MDB.

The following code example demonstrates how to fill a Visual Basic [text box](#) with the entire contents of an Microsoft Access table, including [field](#) names.

In Visual Basic, create a new [form](#), add the following controls, and set the appropriate properties for each control:

Text Box

Name: Text1
Multiline: True
Scrollbars: 3 - Both (vertical and horizontal)

Command Button

Name: Command1

NOTE: Size the text box large so that data can be seen and scrolled easily following the DDE request.

Double click the [command button](#) and enter the following code between the "Sub Command1_Click ()" and "End Sub" lines:

```
Sub Command1_Click()  
    Text1.LinkTopic = "MSACCESS|NWIND;TABLE Employees"  
    Text1.LinkMode = 2       ' Establish a manual DDE link to Access  
    Text1.LinkItem = "All"   ' Setup the DDE request item to get ALL the
```

```

                                ' .. data from the Employees table,
                                ' .. including field names.
Text1.LinkRequest              ' Request the data into the Text1 text box
Text1.LinkMode = 0            ' Terminate the DDE link to Access
End Sub

```

In this example, the first line specifies "MSACCESS" as the application to communicate with and "NWIND;TABLE Employees" as the topic of the DDE conversation. We could also specify an [SQL](#) statement as the topic of the conversation. The following line of code is functionally equivalent to the first line above:

```
Text1.LinkTopic = "MSACCESS|NWIND;SQL Select * From Employees;"
```

In this example, the item "All" tells Microsoft Access to supply Visual Basic with the entire contents of the Topic, including field names. Microsoft Access supports a rich set of other items and topics. For a list of supported topics and items, query on the following words in the Microsoft Knowledge Base:

dde and server and topics

When you run this example, be sure that Microsoft Access is running and has the NWIND database opened.

When you run this code, the text box will be filled with the entire contents of the Employees table. Each field will be delimited by a tab character, Chr\$(9), and each [record row](#) will end with a newline character (a carriage return followed by a linefeed character).

[References](#)

INF: Print a Sequential Number for Each Record/Group on Report

Article Number: Q98790
CREATED: 13-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can place a [text box control](#) that prints a sequential number for each [record](#) in the Detail section of a [report](#). This number can be reset so that a separate set of numbers is printed for each [group](#) of records. This article explains these procedures by modifying a report in the sample [database](#) NWIND.MDB.

MORE INFORMATION

If you want to print an incremental counter for each record in a report group, place an unbound text box control in the Detail section of the report. To begin the sequence with the number 1, set the ControlSource property to =1, then set the RunningSum property to Over Group.

Steps to Reproduce Behavior

1. From the [Database window](#), open the List of Products by Category report in [Design view](#).
2. Place the following text box control before the Product Name control in the Detail section:

Object: Text Box

ControlName: MyLineNumber
ControlSource: =1
RunningSum: Over Group

3. Preview the report. Note that each group's records are sequentially numbered, beginning with the number 1.

You can also increment the counter number for each group in the report by changing the RunningSum property to Over All. To do this, replace step 2 above with the following step:

Place the following text box control before the Category Name control in the Category Name Header section:

Object: Text Box

ControlName: MyLineNumber2
ControlSource: =1
RunningSum: Over All

Repeat step 3 above. Note that each individual record now prints a unique sequential number.

References:

"Microsoft Access User's Guide," version 1.0, page 575

[References](#)

INF: Nulls Allowed in Foreign Key with Referential Integrity

Article Number: Q98789
CREATED: 13-MAY-1993
MODIFIED: 10-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access allows a Null value in a field that is the foreign key in a relationship if referential integrity is enforced.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the sample database NWIND.MDB.
2. From the Edit menu, choose Relationships.
3. Under Primary Table, select Customers. Under Related Table, select Orders.

You will note the following:

- a. PrimaryKey Fields and Select Matching Fields both display "Customer ID".
- b. The Enforce Referential Integrity check box is selected.
- c. The Add button is unavailable.

These characteristics indicate an existing relationship between Customers and Orders tables; thus, you will not be able to make an entry in the Customer ID field of the Orders table if that Customer ID does not exist in the Customers table.

4. Close the Relationships dialog box and open the Orders table.
5. Change the Customer ID field of the first order to "XXXX" and press TAB to try to move to the next record. You will receive the following error message:

Can't add or change record. Referential Integrity rules require a related record in the table "Customers".

6. Choose OK.
7. Delete the entry in the Customer ID field. Make sure the field is completely empty. Press TAB to move to the next record.

REFERENCES

=====

"User's Guide," version 1.0, page 54

[References](#)

INF: Number of Times a Custom Function Executes in a Query

Article Number: Q98788
CREATED: 13-MAY-1993
MODIFIED: 10-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

A custom function called from a query will execute as follows:

- One (1) time per record if the expression does not reference a field
- One (1) time per record if the expression does reference a field
- Two (2) times per record if there are criteria on the expression

MORE INFORMATION

To optimize a query, Microsoft Access will not re-execute a custom function unless the value passed to it changes. If the function accepts a field as a parameter, Microsoft Access must re-execute the custom function for each record, since the data may change from record to record. If criteria are placed on the result of the function, Microsoft Access must re-execute the function when applying the criteria.

The examples below (Examples A, B, and C) demonstrate each of the three scenarios, using a custom function designed to number the records in a query.

NOTE: These examples are simplified to demonstrate the results you can expect when you use custom functions in a query. Many variables can affect the number of times Microsoft Access re-executes a custom function. For example, if you first minimize and then maximize Microsoft Access, the process of repainting the screen also includes re-executing any custom function that is part of the query result.

Complete the following steps to use Examples A, B, and C:

1. Create a new module called RecordNumbers in the sample database NWIND.MDB (Northwind Traders).
2. Add the following to the Declarations section:

Global RecordNum

Example A: An Expression That Calculates Only Once Per Query

1. Add the following function to the module you created in the previous procedure:

```

Function ShouldIncrement ()
    RecordNum = RecordNum + 1
    ShouldIncrement = RecordNum
End Function

```

2. Create a new query based on the Categories table. Drag the Last Name field to the query grid and add a column with the following expression:

```
RecordNumber: ShouldIncrement()
```

3. Run the query. You will obtain the following result:

Last Name	RecordNumber
Davolio	1
Fuller	1
Leverling	1
Peacock	1
Buchanan	1
Suyama	1
King	1
Callahan	1
Dodsworth	1

Note that the function IncrementRecNum() is executed only once.

Example B: An Expression That Calculates Once Per Record

1. Add the function DoesIncrement() to the module RecordNumbers you created in the first procedure:

```

Function DoesIncrement (AnyValue)
    RecordNum = RecordNum + 1
    DoesIncrement = RecordNum
End Function

```

2. Create a new query based on the Employees table. Drag the Last Name field to the query grid and add a column with the following expression:

```
RecordNumber: DoesIncrement([Employee ID])
```

3. Run the query. You will obtain the following result:

Last Name	RecordNumber
Davolio	1
Fuller	2
Leverling	3
Peacock	4
Buchanan	5
Suyama	6
King	7
Callahan	8

Example C: An Expression That Calculates More Than Once Per Record

1. Create a new query based on the Employees table. Drag the Last Name field to the query grid and add a column with the following expression:

```
RecordNumber: DoesIncrement([Employee ID])
```

NOTE: You will need to complete Step 1 of Example B in order to have the DoesIncrement() function available.

2. Open the module RecordNumbers in Design view. From the View menu, choose Immediate Window. Type the following statement to initialize the variable RecordNum:

```
RecordNum = 0
```

3. Run the query. You will obtain the following result:

Last Name	RecordNumber
Davolio	1
Fuller	2
Leverling	3
Peacock	4
Buchanan	5
Suyama	6
King	7
Callahan	8
Dodsworth	9

4. Repeat Step 2 in this example (Example C).
5. Add the following criteria under RecordNumber:

```
>=0
```

6. Run the query. You will now obtain the following result:

Last Name	RecordNumber
Davolio	10
Fuller	11
Leverling	12
Peacock	13
Buchanan	14
Suyama	15
King	16
Callahan	17
Dodsworth	18

In this case, Microsoft Access runs the expression twice, once to create the dynaset and again to check the criteria you specified.

For more information on creating custom counters for forms and queries, query on the following words in the Microsoft Knowledge Base:

custom and counter

[References](#)

PRACC9304: Query Progress Meter Reports Status Incorrectly

Article Number: Q98787
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

When running an [append query](#) that appends a large number of records to an existing [table](#), the progress meter may incorrectly reflect the status of the [query](#). The progress meter may increase quickly to 98 percent complete and then continue slowly until reaching 100 percent.

CAUSE

The progress meter does not take into account time for the sorting phase. If the sorting phase takes a long time, the progress meter does not correctly reflect the progress of the query.

STATUS

Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

This problem is highly dependent on the query and the sorting phase necessary for that query.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 7, "Designing Action Queries and Parameter Queries," pages 170-174

[References](#)

INF: SQL Server Views Read Only When Attached

Article Number: Q98786
CREATED: 13-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
 - Microsoft [SQL](#) Server versions 1.x and 4.2
-

Microsoft SQL Server supports updatable views. Microsoft Access can attach to these views as read only because SQL Server views do not have indexes. Microsoft Access requires a unique [index](#) on the attached object for [update](#)/delete/insert capability.

Please refer to Knowledge Base article Q90100 for more information on Microsoft Access and updating attached SQL Server tables.

[References](#)

INF: Maximum Database Size Increased in Version 1.1

Article Number: Q98675
CREATED: 11-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.10

The information in this article applies to:

- Microsoft Access version 1.1
-

SUMMARY

The maximum [database](#) size in Microsoft Access version 1.1 is 1 gigabyte (GB). In Microsoft Access 1.0, the maximum database size is 128 megabytes (MB).

This article explains how to convert your Microsoft Access version 1.0 databases into version 1.1 databases, thereby allowing you to take advantage of the new database size limit.

MORE INFORMATION

You can use Microsoft Access version 1.0 databases under Microsoft Access version 1.1. However, in order to take advantage of the new size limit, you must convert your version 1.0 databases.

To convert a Microsoft Access version 1.0 database to version 1.1 format, do the following:

1. Open Microsoft Access version 1.1.
2. From the File menu, choose Compact Database.
3. Select the database that you want to convert, and choose OK.
4. The Database to Compact Into dialog window appears. From the List Files of Type [combo box](#), choose Access V1.1 (*.MDB). Enter a new File Name, and choose OK.

NOTE: When you select a version 1.0 database, Access 1.0 (*.MDB) will be the default format. You will need to manually select Access 1.1 (*.MDB) in the List Files of Type combo box.

[References](#)

PRACC9305: Unable to Import a SQL Server View

Article Number: Q98674
CREATED: 11-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0
-

SYMPTOMS

You can attach to an [SQL](#) Server view. If you attempt to import this view, you receive the following error message:

Internal [database](#) error (-1603)

RESOLUTION

There is no workaround in Microsoft Access version 1.0.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem has been corrected in Microsoft Access version 1.1.

MORE INFORMATION

This is not consistent behavior, because the user interface for importing SQL tables should not differentiate between tables and views.

A SQL Server view is similar to a [dynaset](#). It is the result of running an SQL [query](#).

Steps to Reproduce Problem

1. Use another application, such as Microsoft Excel Q+E, to create and save a SQL Server view.
2. Start Microsoft Access and open a database.
3. From the File menu, choose Import.
4. Select SQL Database as the Data Source.
5. Select the Data Source you want to import.
6. Log on to the SQL Server by entering your password and choosing OK.
7. Choose the SQL Server view to import from the Import Tables dialog window.

In Microsoft Access version 1.0, you will get the above error message.
In Microsoft Access version 1.1, the view will import correctly.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 4, "Importing,
Exporting, and Attaching," pages 66-68

[References](#)

INF: Improved Print Merge Feature In Microsoft Access 1.1

Article Number: Q98673
CREATED: 11-MAY-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.10

The information in this article applies to:

- Microsoft Access version 1.1
-

SUMMARY

This article discusses a new feature of Microsoft Access version 1.1: the ability to directly use Microsoft Access tables in a Microsoft Word for Windows print merge.

MORE INFORMATION

To use data from a Microsoft Access [database](#) in Word for Windows, the data must be converted to a format that Word for Windows recognizes.

In Microsoft Access version 1.0, this is done by exporting the data from Microsoft Access as text, or copying tables and queries from Microsoft Access and pasting them into a Word for Windows document.

A new feature of Microsoft Access version 1.1 is the ability to select Word for Windows Merge directly from the Export Data Destination [dialog box](#).

The following steps, performed in Microsoft Access version 1.1, explain how to convert your [table](#) into a data document for Word for Windows:

1. From the [Database window](#), choose Export from the File menu.
2. Select Word for Windows Merge from the Export Data Destination Dialog box.
3. Select the Table or Query that contains the information you want.
4. Microsoft Access will enter a default file name for export. You can either accept the default filename or enter a new filename that you want to export to.
5. Microsoft Access will provide an Export Word Merge Options window that you may want to customize if you are exporting date/time fields.
6. Minimize Microsoft Access and start Word for Windows.
7. Open the main document in Word for Windows.
8. From the File menu, choose Print Merge.
9. Choose the Attach Data File button.

10. Select the exported text file that you created in step 4.

NOTE: Word for Windows will only show *.DOC files by default. You will need to specify *.TXT files in the List Files of Type combo box contained in the Attach Data File dialog box.

11. Use the Insert Merge Fields, Edit Data Field, Check, and Print Merge features of Microsoft Word for Windows to complete the document.

For more information on print merge capabilities in Microsoft Access version 1.0, [query](#) on the following words in the Microsoft Knowledge Base:

access and word and merge and 1.0

[References](#)

PRACC9305: GP Fault when Creating a Dynaset from Form Dynaset

Article Number: Q98672
CREATED: 11-MAY-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

When attempting to create a [dynaset](#) based on a [form](#)'s dynaset, you will get a general protection (GP) fault similar to:

Application Error

MSACCESS caused a General Protection Fault in [module](#)
MSACCESS.EXE at 00B0:09B5

CAUSE

This functionality is unsupported. Instead of returning an error message explaining this, Microsoft Access causes a GP fault. Microsoft Access does not support creating a dynaset based on a form's dynaset.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. The GP fault has been corrected in Microsoft Access version 1.1.

MORE INFORMATION

In Microsoft Access version 1.0, the following steps will reproduce a GP fault.

In Microsoft Access version 1.1, the following steps will correctly return the following error message, indicating that this functionality is not supported:

Can't perform operation; it is illegal

In the following example, the Clone [method](#) is used to create a duplicate dynaset based on a form's dynaset. This is incorrect usage of the Clone method.

Steps to Reproduce Problem

1. Open the Categories form in the sample [database](#) Northwind Traders (NWIND.MDB).
2. Create a new module, with the following [Declarations section](#):

```
'*****  
'Declarations Section of the module.  
'*****  
Option Compare database  
Dim DS as Dynaset, DS2 as Dynaset, F as Form
```

3. From the View menu, choose the [Immediate window](#).
4. Type the following lines in the Immediate window, pressing ENTER after each line:

```
Set F=Forms!Categories![Categories Subform].Form  
Set DS=F.Dynaset.Clone()  
Set DS2=DS.CreateDynaset()
```

The error will occur after the third line.

References:

For more information, search for "clone," then "Clone Method" using the Help menu.

[References](#)

PRACC9305: Setup Writes Over Newer CTL3D.DLL File

Article Number: Q98671
CREATED: 11-MAY-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

When installing Microsoft Access version 1.0, the file CTL3D.DLL may be replaced, even if you currently have a more recent version of the file.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem has been corrected in Microsoft Access version 1.1.

MORE INFORMATION

The CTL3D.DLL file provides the 3-D "look" for a Windows application's buttons. This file is normally stored in the WINDOWS\SYSTEM subdirectory.

[References](#)

PRACC9304: SQL 4-Byte Values Incorrectly Mapped to Text

Article Number: Q98670
CREATED: 11-MAY-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

When you try to edit data on an attached [SQL table](#), you receive the following error message:

[ODBC](#) Update Failed

CAUSE

When attaching to an SQL table in Microsoft Access version 1.0, 4-byte value fields were incorrectly mapped to the Text [data type](#).

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in version 1.1.

MORE INFORMATION

In Microsoft Access version 1.1, 4-byte [field data types](#) are mapped as follows:

SQL	ACCESS

4-byte Real	Number
SmallMoney	Currency
SmallDateTime	Date/Time

[References](#)

INF: Default Properties do not Carry over on Attached Table

Article Number: Q98669
CREATED: 11-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

When you attach to a Microsoft Access table in another database, the default values do not appear. For example, say your database DB1.MDB includes Table1, which contains a field with a Number data type set to the default value 5. When you attach to that table from the database DB2.MDB and try to add records to the attached table, 5 does not appear in new records. This limitation occurs because properties of an attached table are not read in Microsoft Access version 1.x.

References:

"Microsoft Access User's Guide," Chapter 4, pages 69-70

[References](#)

INF: How to Buffer SQL Strings Using DDE

Article Number: Q98668
CREATED: 11-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.10

The information in this article applies to:

- Microsoft Access version 1.1
-

SUMMARY

Microsoft Access version 1.1 can handle [SQL](#) strings larger than 255 characters through dynamic data exchange ([DDE](#)). To do this, break the SQL [string](#) into portions with fewer than 255 characters, and then send these strings to Microsoft Access using DDE. The separate portions are buffered until a semicolon is sent; at that point, Microsoft Access executes the [query](#).

MORE INFORMATION

The following steps walk you through creating a DDE [macro](#) in Microsoft Excel for Windows.

NOTE: This macro does NOT work with Microsoft Access version 1.0.

1. Open the sample [database](#) NWIND.MDB.
2. Open a new macro sheet in Microsoft Excel. Enter the following macro:

Cell	Command
B1	SQLDDEExample
B2	
B3	chan=INITIATE ("MSACCESS", "nwind.mdb;SQL")
B4	=POKE (chan, "SQLText", C3)
B5	=POKE (chan, "SQLText", C4:C5)
B6	QueryResult=REQUEST (chan, "All")
B7	=TERMINATE (chan)
B8	=FOR ("FldPos", 1, COLUMNS (QueryResult))
B9	=FOR ("RecPos", 1, ROWS (QueryResult))
B10	=FORMULA (INDEX (QueryResult, RecPos, FldPos), OFFSET (C7, RecPos, FldPos))
B11	=NEXT ()
B12	=NEXT ()
B13	=RETURN ()

3. Enter the following pieces of an SQL string on the macro sheet:

Cell	Command
C3	SELECT * From categories
C4	ORDER BY categories.[category id]
C5	DESC;

Important: Be sure to include one space in front of the words "ORDER" and "DESC."

4. To run the macro, select cell B3, choose Run from the Macro menu, and choose OK. The query then executes.

The results of the query are placed in cells D8:G16 on the Microsoft Excel macro sheet.

[References](#)

PRACC9304: Calculated Control on Subreport Displays Wrong Data

Article Number: Q98667
CREATED: 11-MAY-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

A calculated control on your subreport shows incorrect data for the first record of alternate pages in the report.

RESOLUTION

Change the CanGrow or the LinkChild and LinkMaster properties of the calculated controls so that either

- The control's CanGrow property is NOT set to Yes

or

- The control's LinkChild and LinkMaster properties are NOT blank

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

This problem occurs only with non-linked subreports when the calculated control's CanGrow property is set to Yes. This is not a problem with linked subreports.

The following example reproduces this problem in Microsoft Access version 1.0, using the sample database NWIND.MDB.

Steps to Reproduce Problem

1. Create a new report based on the Categories table. Add all the fields from the Categories table to the report and leave the report open in Design view.
2. Select the Category ID text box, then display its property sheet by choosing Properties from the View menu. Note that this control's ControlName property is currently Category ID. Modify this control's properties as follows:

Object: text box

ControlName: Category IDx
ControlSource: =[Category ID]
CanGrow: Yes

3. Save this report as "Sub" and close the report.
4. Create a new, unbound blank report and drag the Sub report from the Database window onto the new report. Save this new report as "Main."
5. Switch to Print Preview of the Main report. Notice the Category ID field of the first record on the Sub report shows the wrong Category ID for this record, but the Category Name field shows "Beverages" (the correct name for this record).

[References](#)

INF: "Form is read only" error

Article Number: Q98666
CREATED: 11-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

The following error message is displayed

Form is read only

when trying to update data, but the form property DefaultEditing is not set to Read Only.

CAUSE

If your form is based on a query that is Read Only, you will not be able to edit from a form based on that query.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new query based on the Order Details table in the example database, Northwind Traders (NWIND.MDB).

Query: Orders

Type: Select Query.

Join: None (only one table involved)

Field: Quantity

Total: Group By

Field: Unit Price

Total: Group By

Field: Expr1: [Unit Price]*[Quantity]

Total: Expression

2. Run query and try to change the Unit Price in one of the records. You get a beep. Notice that the status bar displays

Form is read only.

As expected, you get the same results when you use a form based on this query.

Click on the Totals button to remove the Total line in the OBE. Run the query. You may now change the records.

Workaround

A workaround would depend on what you are trying to accomplish by using the query. The Microsoft Access User's Guide contains information on designing queries which may be used to update records.

References:

Microsoft Access User's Guide, Designing Queries for Updating Records, pp. 156-157

Microsoft Access User's Guide, Creating a Query for Updating Two Tables, p. 158

[References](#)

INF: Sorting Titles Without Leading Articles (The, An, etc)

Article Number: Q98665
CREATED: 11-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

This article explains how to sort titles in text fields without including the leading articles "the," "an," or "a."

MORE INFORMATION

The following is a list of book titles:

Aztec Indians
The Ant People
Beast Man
The Baseball Club
An Attempt At Fun
Apple Valley

If you sort the above list using a custom Microsoft Access [function](#) in a [query](#), the following new list results:

The Ant People
Apple Valley
An Attempt At Fun
Aztec Indians
The Baseball Club
Beast Man

Create the following [Access Basic](#) function that returns a book title without the leading article:

```
Option Explicit

Function Gettitle (Titles) As String
    sp = InStr(1, Titles, Chr$(32))
    If sp > 0 Then
        Select Case Left(Titles, sp - 1)
            Case "An", "A", "The" ' add any other articles here!
                Gettitle = Mid(Titles, sp + 1)
            Case Else
                Gettitle = Titles
        End Select
    Else
        Gettitle = Titles
    End If
End Function
```

To sort a [table](#) on the Titles [field](#), create a query with the following two columns in the query grid:

Query

Field: Sorted List: Gettitle([Titles])

Sort: Ascending

Show: Yes

Field: [Titles]

Show: No

The first [column](#) is a calculated field that extracts the text of the Titles field, but omits the leading article. The results of this calculation are sorted in ascending order. The second Show [check box](#) is cleared so that the query result is not displayed.

References:

"Language Reference," version 1.0, pages 276-277, 404

[References](#)

INF: Sorting Titles Without Leading Articles (The, An, etc)

Article Number: Q98665
CREATED: 11-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

This article explains how to sort titles in text fields without including the leading articles "the," "an," or "a."

MORE INFORMATION

The following is a list of book titles:

Aztec Indians
The Ant People
Beast Man
The Baseball Club
An Attempt At Fun
Apple Valley

If you sort the above list using a custom Microsoft Access [function](#) in a [query](#), the following new list results:

The Ant People
Apple Valley
An Attempt At Fun
Aztec Indians
The Baseball Club
Beast Man

Create the following [Access Basic](#) function that returns a book title without the leading article:

```
Option Explicit

Function Gettitle (Titles) As String
    sp = InStr(1, Titles, Chr$(32))
    If sp > 0 Then
        Select Case Left(Titles, sp - 1)
            Case "An", "A", "The" ' add any other articles here!
                Gettitle = Mid(Titles, sp + 1)
            Case Else
                Gettitle = Titles
        End Select
    Else
        Gettitle = Titles
    End If
End Function
```

To sort a [table](#) on the Titles [field](#), create a query with the following two columns in the query grid:

Query

Field: Sorted List: Gettitle([Titles])

Sort: Ascending

Show: Yes

Field: [Titles]

Show: No

The first [column](#) is a calculated field that extracts the text of the Titles field, but omits the leading article. The results of this calculation are sorted in ascending order. The second Show [check box](#) is cleared so that the query result is not displayed.

References:

"Language Reference," version 1.0, pages 276-277, 404

[References](#)

INF: Deleting Duplicate Records from Two Tables

Article Number: Q98664
CREATED: 11-MAY-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

This article describes how to delete duplicate records from two tables and retain the most current information (records). In addition, the article shows how to merge the results of these two tables.

MORE INFORMATION

This article walks you through creating two tables, TEST-1 and TEST-2, with duplicate records. After adding additional nonduplicate records to both these tables, you will delete the duplicate records from the TEST-2 [table](#). Finally, you will [update](#) the TEST-1 table to include nonduplicate information from the TEST-2 table.

Steps to Reproduce Behavior

1. Open the sample [database](#) NWIND.MDB.
2. In the [Database window](#), highlight the Shippers table.
3. From the Edit menu, choose Copy.
4. From the Edit menu, choose Paste.
5. In the Paste Table As [dialog box](#), type TEST-1 in the Table Name [field](#). Choose OK.
6. Repeat steps 4 and 5, this time pasting the table as TEST-2.
7. Open the TEST-1 table and add the following two additional company names:

ABC Shipping
XYZ Express

Close the TEST-1 table.

8. Open the TEST-2 table and add the following two new company names:

Jiffy Ship
Hurry Package

Close the Shippers table.

9. From the Database window, select the Query button and then choose New.
10. Add the TEST-1 table and the TEST-2 table to the [query](#), then close the Add Table dialog box.
11. Join the tables on the Company Name field.
12. From the Query menu, choose Delete.

NOTE: For this example, the TEST-1 table will be considered the most current information; therefore, duplicate information will be deleted from the TEST-2 table.

13. From the TEST-1 table, drag Shipper ID and Company Name down to the [QBE](#) grid. Notice that the word "Where" appears on the Delete line in the QBE grid.
14. Bring the * down from the TEST-2 table. Notice that the word "From" appears on the Delete line in the QBE grid.
15. From the Query menu, choose Run.
16. You will receive a message that three rows will be deleted. Choose OK.
17. Open the TEST-2 table. The only two records that remain are the nonduplicate records that you added in step 8 above. All the duplicate records have been deleted.

The remaining steps append the remaining nonduplicate records from the TEST-2 table to the TEST-1 table.

19. Open the TEST-2 table.
20. From the Edit menu, choose Select All Records.
21. From the Edit menu, choose Copy.
22. Close the TEST-2 table and then open the TEST-1 table.
23. From the Edit menu, choose Paste Append. You will receive a message that two rows have been pasted into this table. Choose OK to complete the Paste Append.

Your most current information, without duplicates, is now in one table (TEST-1).

[References](#)

PRB: Exporting to Fixed-Width Text File Left Justifies Numbers

Article Number: Q98663
CREATED: 11-MAY-1993
MODIFIED: 25-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

When you export a table containing a Number or Currency field to a fixed-width text file, the numbers in the table become left justified.

CAUSE

Since the numbers are being exported to a fixed-width text file, Microsoft Access pads zeroes to the right of the number until it meets the proper field width.

RESOLUTION

To make the numbers to align with the decimal and pad zeros to the left instead of the right, you first must determine the maximum length of the Number or Currency field. Use this number of zeros in the Format function below. Next, you must determine the number of decimal places the field in question uses. Use this value to determine how many zeros to place after the decimal point in the Format function below. Finally, create a query based on the following Microsoft Access SQL statement:

```
SELECT DISTINCTROW
    Format([MyNumber],"0000.00")
AS [Expr1]
INTO [MyNewTable]
FROM [MyOldTable];
```

For this example, this SQL statement assumes that you have the following table:

```
Table: MyOldTable
-----
    fieldName: MyNumber
    DataType: Currency
```

It also assumes that the longest value in the MyNumber field is 7 characters in length (including the decimal and decimal places) and that each value has two decimal places. The above Microsoft Access SQL statement, when typed in the SQL window in the query-by-example (QBE) grid, makes a new table called MyNewTable with the MyNumber field padded to the left with zeros.

When you use the above Microsoft Access SQL statement in a query, the data from MyOldTable is formatted with decimal justification and

placed in a new table called MyNewTable.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create the following table:

Table: MyOldTable

FieldName: MyNumber
DataType: Currency

2. Enter the following numbers in the table:

34.5
123.56
4578.90

3. From the File menu, choose Export and select the Text (Fixed Width) option. Export MyOldTable.
4. Open the exported text file with a utility, such as Microsoft Windows Notepad, and view the numbers. They appear as follows:

34.5000
123.560
4578.90

[References](#)

PRB: Formatting Yes/No Data Type Example

Article Number: Q98662
CREATED: 11-MAY-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

If you type a format string in the Format property of a text box in which the control source is a Yes/No data type, in Form view, the format string appears in the text box instead of expected data. No error message is generated.

CAUSE

The Format property in the text box is incorrect. This can occur if you follow the illustration on page 219 of the "Language Reference." The illustration reads:

```
" ;Always [Blue] ;Never [Red]"
```

RESOLUTION

Instead, the illustration should read:

```
;"Always" [Blue];"Never" [Red]
```

REFERENCES

=====

"Microsoft Access Language Reference," version 1.0, page 219

[References](#)

INF: How to Create Synchronized Combo Boxes

Article Number: Q98660
CREATED: 11-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to create a [combo box](#) that is filtered to list only items related to an item selected in a prior combo box.

MORE INFORMATION

The following example was created using the sample [database](#) NWIND.MDB. Using the steps below, you can create an unbound [form](#) with two combo boxes on it. The first combo box lists the category names from the Categories [table](#). The second combo box lists the product names from the Products table. When you select a category name, you can see only the product names for that category.

1. Create a new [query](#) called Category Lister based on the Categories table:

Query: Category Lister

Field name: Category ID
Show: Yes

Field name: Category Name
Show: Yes
Sort: Ascending

2. Create a new query called Product Lister based on the Products table:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

Query: Product Lister

Field name: Product ID
Show: Yes

Field name: Product Name
Show: Yes
Sort: Ascending

Field name: Category ID
Show: Yes
Sort: Ascending

Criteria: IIF(IsNull([Forms]![Selector Form]![Category _
Selection]), [Category ID], [Forms]![Selector _
Form]![Category Selection])

Using the IIF() and IsNull() functions allows you to see a list of all products if no category is chosen.

3. Create a new macro called Reset Product Selection:

Macro Name	Action	Argument
Reset Product Selection	Requery	Product Selection
	GoToControl	Product Selection

4. Create a new, unbound, blank form called Selector Form.
5. Place two unbound combo boxes on Selector Form:

Object: Combo Box

ControlName: Category Selection
RowSourceType: Table/Query
RowSource: Category Lister
ColumnCount: 2
ColumnWidths: 0 in;1 in
BoundColumn: 1
After Update: Reset Product Selection

Object: Combo Box

ControlName: Product Selection
RowSourceType: Table/Query
RowSource: Product Lister
ColumnCount: 3
ColumnWidths: 0 in;1 in; 0 in
BoundColumn: 1

6. Save Selector Form, then switch to Form view and perform the following test:
 - a. In the Category Selection box, select a category.
 - b. In the Product Selection box, select a product. Note that only products for the selected category are listed.
 - c. In the Category Selection box, select a different category. Note that the Product Selection box is reset to reflect products in the new category.

References:

For more information, search for "requery" using the Help menu.

"Microsoft Access Users Guide," version 1.0, pages 237, 541

[References](#)

PRACC9302: Syntax Error When "(" is First Chr in Criteria Cell

Article Number: Q98301
CREATED: 02-MAY-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

You receive the error message "Syntax error" when you try to execute a [query](#).

CAUSE

A opening parenthesis is the first character in a [cell](#) in the Criteria [row](#) of the query grid.

RESOLUTION

Depending on the [WHERE clause](#) you are using, you can restructure the query so that there is no opening parenthesis in the Criteria row. See below in the "More Information" for an example.

You also can enter the WHERE clause directly in the [SQL dialog box](#). If Microsoft Access places the opening parenthesis in the Criteria row, the process is successful. However, the process works only until you make a change to the query. For example, if you remove a [column](#) from the query grid, even if it is unrelated to the [criteria](#), the error message is displayed.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

You may want an opening parenthesis in a Criteria cell if you are searching a Text or Memo [field](#) for several key words. For example, you may want to use a WHERE clause similar to the following:

```
(Textcol Like "*performance*" Or Textcol Like "*speed*") And  
(Textcol Like *start*" Or Textcol Like "*run*")
```

You can enter the WHERE clause either in a single cell in the Criteria row or as multiple rows in the Criteria section of the query grid. Place the Textcol column on the grid twice. In one or both instances of the column, the Show box may be cleared. The Criteria section of the grid should appear as follows:

Textcol

Textcol

Criteria: Like "*performance*"
or: Like "*speed*"

Like "*start" Or Like "*run*"
Like "*start" Or Like "*run*"

Steps to Reproduce Problem

-
1. Using the sample [database](#) NWIND.MDB, create a new query based on the Employees [table](#).
 2. Drag the Last Name and First Name fields from the table to the query grid.
 3. In the Criteria cell under Last Name, type the following:

(Like "*a*")
 4. Press TAB to move to the next cell. The "Syntax error" message appears.
 5. Choose OK. Note that the characters "*a*" are now highlighted.

[References](#)

INF: Determining Which User You Are Logged

Article Number: Q98300
CREATED: 02-MAY-1993
MODIFIED: 17-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes two quick ways to determine the account (user) under which you are logged in.

- In the Database window, select Change Password from the Security menu.

In the Change Password dialog box, Microsoft Access displays the account under which you are currently logged in.

-or-

- Call the User() function from Access Basic with the following steps:
 1. Create a new module in your database.
 2. From the View menu, choose the Immediate window.
 3. In the Immediate window, type "?User()" (without the quotation marks).

The user name is then displayed on the next line in the Immediate window.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 25,
"Administering a Database System"

"Microsoft Access Language Reference," version 1.0, page 489, "User Function"

[References](#)

PRB: Empty Subforms Are Not Visible in Print Preview

Article Number: Q98299
CREATED: 02-MAY-1993
MODIFIED: 12-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you preview a form containing a subform, the subform area is completely blank; however, the subform control does appear in the Design view of the main form.

CAUSE

If you have a subform that does not have any data in it, the subform appears in Form view with the structure only. However, no part of the subform is visible in Print Preview mode.

STATUS

This behavior is by design.

[References](#)

PRB: Empty Row of List Box Can Be Selected

Article Number: Q98298
CREATED: 02-MAY-1993
MODIFIED: 12-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

An Empty or Null Row of a list box can be selected.

CAUSE

Null is a valid selection in Microsoft Access.

STATUS

This behavior is by design.

MORE INFORMATION

It is also possible to select an Empty or Null value in a combo box.

Steps to Reproduce Behavior

-
1. Create a new form.
 2. Create a list box on the form with the following properties:

Object: list box

RowSourceType: Value List
RowSource: 1;2;;4

3. From the View menu, choose Form.

You can select the empty row between 2 and 4.

References

"Microsoft Access User's Guide," version 1.0, Chapter 9, "Designing Forms" pages 233-243

[References](#)

INF: How to Query for Duplicate Records

Article Number: Q98230
CREATED: 02-MAY-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you import data from other databases, records may contain duplicate information in the primary key fields. You can run a make-table query to eliminate the duplicate data, but if key fields for two or more records contain the same value, you may want to reconcile the information manually to make the records unique.

This article includes sample queries and shows how to list duplicate information for keys containing single fields or multiple fields.

MORE INFORMATION

Example 1 - A Single-Field Key

This example assumes you have a table called MyTable, with the primary key on the Name field.

1. Open MyTable in Design view and index the Name field, as shown below:

```
Table: MyTable
-----
Field: Name
Indexed: Yes (Duplicates OK)
```

2. Create a query called Find Duplicates.
3. Add MyTable to the above query.
4. From the View menu, choose View Totals.

```
Query: Find Duplicates
-----
Field: Name
Total: Group By

Field: Name
Total: Count
Criteria: >1
```

The equivalent SQL statement is as follows:

```
SELECT DISTINCTROW
```

```

        Name, Count(Name) As CountOfName
FROM
    MyTable
GROUP BY
    Name
HAVING
    Count(Name)>1
WITH OWNERACCESS OPTION;

```

5. Run the query. Duplicate key values are displayed in the first column and the number of times the value appears is displayed in the second column.

Example 2 - Multiple-Field Key

This example assumes you have a table called MyTable, with a double primary key on the First Name and Last Name fields.

1. Open MyTable in Design view and index the First Name and Last Name fields, as shown below:

```

Table: MyTable
-----
Field: First Name
    Indexed: Yes (Duplicates OK)

Field: Last Name
    Indexed: Yes (Duplicates OK)

```

2. Create a query called Find Duplicates.
3. Add MyTable to the above query.
4. From the View menu, choose View Totals.

```

Query: Find Duplicates
-----
Field: Last Name
    Total: Group By

Field: First Name
    Total: Group By

Field: First Name
    Total: Count
    Criteria: >1

```

NOTE: Select Group By for all fields in the key, but select Count only for the First Name field.

The equivalent SQL statement is as follows:

```

SELECT DISTINCTROW
    [Last Name], [First Name],
    Count([First Name]) As [CountOfFirst Name]
FROM

```

```
MyTable
GROUP BY
  [Last Name], [First Name]
HAVING
  Count([First Name])>1
WITH OWNERACCESS OPTION;
```

5. Run the query. Duplicate key values are displayed in the first two columns and the number of times the value appears is displayed in the third column.

For more information, query on the following words here in the Microsoft Knowledge Base:

[filter](#) and unique and query

[References](#)

INF: How to Determine the View in Which a Form Is Displayed

Article Number: Q98229
CREATED: 02-MAY-1993
MODIFIED: 12-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

Summary:

There are no internal functions or properties within Microsoft Access to determine if a [form](#) is in [Form view](#) or [Datasheet view](#). To do this, you must call a series of Windows application programming interface (API) functions.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

More Information:

To determine the current view of a form, call the [function](#) listed below and pass the form as an argument to the function. For example:

```
X% = Is_FormView (Forms!Form)
```

If the form is in Form view, the returned value is a nonzero number. If the form is in Datasheet view, zero is the returned value.

NOTE: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

NOTE: You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive the duplicate procedure name error, remove or comment the declare statement from your code.

Option Explicit

```
Declare Function GetWindowWord% Lib "User" (ByVal hwnd%, _  
    ByVal nIndex%)  
Declare Function GetWindow% Lib "User" (ByVal hwnd%, ByVal wCmd%)  
Declare Function IsWindowVisible% Lib "User" (ByVal hwnd%) As Integer  
Declare Function GetClassName% Lib "User" (ByVal hwnd%, _  
    ByVal lpClassName$, ByVal nMaxCount%)  
  
Const GW_SiblingHwnd% = 2  
Const GW_CHILD = 5  
  
Function Is_FormView (F As Form)  
    Dim SiblingHwnd%, CName$, CNameLen%
```

```

'get the first child window of the specied form
SiblingHwnd% = GetWindow(F.hwnd, GW_CHILD)

'enumerate through the forms child windows until exhausted
'or you find a visible sub-window with the classname of
'"OFormSub".

Do While SiblingHwnd%
  'allocate temporary space.
  CName$ = Space$(128)

  'get the class name of the specified window.
  CNameLen% = GetClassName(SiblingHwnd%, CName$, Len(CName$))
  'peel off the extra characters in the padded string.
  CName$ = Mid$(CName$, 1, CNameLen%)

  ' if the window is visible, and the classname is OFormSub
  ' then the form is either in Design view, or Form view and
  'not in Datasheet view.
  If IsWindowVisible(SiblingHwnd%) And CName$ = "OFormSub" Then
    Exit Do
  End If

  'get the next sibling window.
  SiblingHwnd% = GetWindow(SiblingHwnd%, GW_SiblingHwnd%)

Loop
Is_FormView = SiblingHwnd%

```

End Function

References

"Microsoft Windows 3.1 Programmer's Reference," Volume 2

[References](#)

INF: Cannot Create Relationships with Attached Tables

Article Number: Q98228
CREATED: 02-MAY-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

You cannot create a relationship between a Microsoft Access table and an attached table. Relationships can be created only between Microsoft Access tables. Any joins involving attached tables must be created using the graphical query by example (QBE) grid.

MORE INFORMATION

To establish a relationship between Microsoft Access tables, do the following:

1. Open a Microsoft Access database. By default, the Database window has the focus.
2. From the Edit menu, choose Relationships.
3. Select a table from the Primary Table list box. Note that no attached tables appear on the list.

REFERENCES

=====

For more information, search for "relationships between tables", then "Defining Relationships Between Tables" using the Microsoft Access Help menu.

[References](#)

INF: Explanation of Option Compare Database

Article Number: Q98227
CREATED: 02-MAY-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The manual describes Option Compare Binary and Option Compare Text. However, the manual's description of Option Compare Database is not so clear. The following is the description:

Comparison is based on the characters' relative position according to the sort order specified when the database was created or compacted.

MORE INFORMATION

Option Compare Database has to do with the value of one character compared to another as it is defined in the View->Options menu. For example, if you had Option Compare Binary, then the following expression...

```
"a" > "z"
```

...would be True because the ASCII value of "a" is greater than the ASCII value of "z". However, with Option Compare Text, it would return False because "z" comes after "a" in the alphabet. Option Compare Database means that the comparison depends on whatever setting you have for the 'New Database Order' option in the View->Options dialog.

The Option Compare setting may also affect the results of the Instr() Function. The following function finds the first tab character in a string:

```
Function FindTab (mystr$)  
    FindTab = Instr(1, mystr$, Chr(9))  
End Function
```

If mystr\$= "1<space>abc<tab>ABC" then the function will return the following (depending on the Option Compare setting):

Option Compare Database	2
Option Compare Text	6
Option Compare Binary	6

As another example, the following function returns the position of the first capital 'c' in the string:

```
Function FindEm (mystr$)  
    FindEm = Instr(1, mystr$, "C")
```

End Function

If mystr\$= "1<space>abc<tab>ABC" then the function will return the following (depending on the Option Compare setting):

Option Compare Database	5
Option Compare Text	5
Option Compare Binary	9

When you bring up the View->Options dialog, one of the items available is 'New Database Sort Order'. Option Compare Database makes it so that the order of one value compared to another depends on whatever is set for this option.

[References](#)

PRB: Err Msg: Macro " Not Found

Article Number: Q98226
CREATED: 02-MAY-1993
MODIFIED: 25-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you are entering data using a [form](#), you get the following error message:

```
Macro '' not found
```

CAUSE

This is may be caused by one or more spaces having been inadvertently typed in either the form's or a [control](#)'s Event properties, such as OnCurrent or AfterUpdate.

RESOLUTION

Open the form in design mode and remove the offending space.

MORE INFORMATION

Microsoft Access doesn't trim the [macro](#) name when trying to find the macro, but does trim it when displaying the error message, so you will just see two single quotes (no space separating them), which looks like one double-quote mark.

You should be able to determine whether the space is located in a control's [event](#) property or a form's event property based on what actions trigger the message.

For example, the error occurs in one of the following situations:

1. Opening the form.
 - Check the Form's OnOpen or OnCurrent property.
 - Check the OnEnter property for the first Control that receives the focus when the form is opened.
2. Typing in a control.
 - Check the control's BeforeUpdate or AfterUpdate property.
3. Saving a [record](#).
 - Check the form's BeforeUpdate or AfterUpdate property.

4. Moving to a new record.

- Check the form's OnCurrent property (also see #3).

Once you have determined which property is causing the problem, open the form in [Design view](#) and remove the space from that property. If you're still not sure which property, go to each property that is blank and press the Delete key.

NOTE: You may have spaces in multiple properties so it is a good idea to check them all.

Steps to Reproduce Behavior

This example uses the sample [database](#) NWIND.MDB.

1. Open the Employees form in Design view.
2. Display the [property sheet](#) by selecting Properties from the View menu.
3. Select the Last Name control and type a single space in the AfterUpdate property.
4. Change to [Form view](#) and type data in the Last Name control.

When you attempt to exit this [field](#) you will receive the error message.

[References](#)

INF: Use SendKeys Macro Action to Open Combo Box Automatically

Article Number: Q98225
CREATED: 02-MAY-1993
MODIFIED: 16-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you are entering a large amount of data and want to open a combo box, it is inconvenient to have to press additional keys or stop to use the mouse. This article presents a generic [macro](#) that opens a [combo box](#) when you set the focus on the [control](#) either by pressing the TAB key or by clicking the control.

MORE INFORMATION

A macro with a single action, SendKeys, is shown below. You can use this macro for all combo boxes on your [form](#).

1. Create a macro called Open Combo Box, as follows:

```
Macro: Open Combo Box
-----
Action: SendKeys
      Keystrokes: %DOWN}
      Wait: Yes
```

2. Open your form in [Design view](#).
3. Add the above macro to the OnEnter property of your combo box.

To manually open the combo box, type ALT+DOWN ARROW.

References:

"Language Reference," version 1.0, pages 433-435

[References](#)

PRB: ErrMsg: Couldn't Open Table MSysAccounts

Article Number: Q98224
CREATED: 02-MAY-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You receive the following error message when attempting to start Microsoft Access:

 Couldn't open Table 'MSysAccounts' in SYSTEM.MDA

CAUSE

The SystemDB entry in the [Options] section of the MSACCESS.INI file points to a file other than a valid SYSTEM.MDA.

RESOLUTION

Use the Change Workgroup Icon in the Microsoft Access Group to reset the MSACCESS.INI to the correct path for the SYSTEM.MDA file. Or you can open the MSACCESS.INI using NOTEPAD.EXE to correct the entry.

MORE INFORMATION

A correct entry in the MSACCESS.INI file would be:

```
[Options]  
SystemDB=C:\ACCESS\SYSTEM.MDA
```

[References](#)

PRACC9303: Password Is Case-Sensitive But Verification Is Not

Article Number: Q98223
CREATED: 02-MAY-1993
MODIFIED: 06-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

When you change your password by choosing Password from the Security menu, the Verify text box is not case-sensitive. However, the Password dialog box is case-sensitive. The next time you start Microsoft Access, the password you type must exactly match the new password you entered previously.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem has been corrected in Microsoft Access version 1.1.

MORE INFORMATION

Steps to Reproduce Behavior

1. From the Security menu, choose Change Password.
2. Type your old password, if you have one, or simply type a new password in the New Password box. Use only uppercase letters.
3. Type your new password in all lowercase letters in the Verify box. Choose OK.
4. Quit Microsoft Access, then restart it.
5. Type your password in all lowercase letters. The following error message appears:

Not a valid account name or password.

6. Type your password in all uppercase letters. Note that the error message no longer appears.

References:

"Microsoft Access User's Guide," version 1.0, page 624

[References](#)

PRACC9303: Multiple Users Adding Fields Causes Cross-Linking

Article Number: Q97995
CREATED: 28-APR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

Incorrect data appears in tables when multiple users load and modify the same database within minutes of each other.

The steps below describe a typical scenario in which this problem occurs:

1. User1 opens a table in a database.
2. User2 opens the same database and makes a change to the design of the table User1 has open.
3. Less than one minute later, User1 makes a change to the design of the same table.

Either user may then notice data corruption in the table. Corruption problems may be apparent in one of the following three ways:

- Data from one field in the table may be displayed in a new field.
- Data in a new or changed field may be displayed in a second field.
- Attempting to copy the table containing new or changed fields generates an error message.

CAUSE

This problem, called cross-linking, occurs because information about tables in a database is read from the system tables and is cached into memory by Microsoft Access when a user opens a database. The information in a user's cache may become obsolete if another person has just made a change and Microsoft Access does not read the new information before the first user makes changes. For example, in the situation above, User1 has cached the information for the table, User2 makes a change, and then User1 makes a change before the system tables are reread. If this occurs, the table can become corrupted.

It is not necessary for both users to open or change the same table for corruption to occur. The information for the table opened by User1 may be on the same page in the system table as the information for the table modified by User2. If that page is cached by Microsoft Access, table corruption may occur.

RESOLUTION

Information in the cache for tables not currently in use is discarded on a regular basis, so problems do not occur unless changes are made before Microsoft Access has had an opportunity to empty the cache. To ensure that information in the cache is emptied, wait a minimum of one to two minutes before opening the table in [Design view](#).

If the database is corrupt, compacting the affected database usually corrects the problem. To do this, close all databases and choose Compact Database from the File menu.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

INF: Executing an Access Macro from Visual Basic using DDE

Article Number: Q97776
CREATED: 21-APR-1993
MODIFIED: 17-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The following information provides some examples of how to use Microsoft Visual Basic to execute macros in Microsoft Access by using dynamic data exchange ([DDE](#)).

MORE INFORMATION

The following code demonstrates how to run a [macro](#) called "MyMacro" in Microsoft Access from a Visual Basic application.

```
Sub Command1_Click()  
    'the character between MSACCESS and System is the pipe symbol  
    Text1.LinkTopic = "MSACCESS\System"  
  
    ' Establish a manual DDE link to Access  
    Text1.LinkMode = 2  
  
    ' Run the macro called "MyMacro"  
    Text1.LinkExecute "MyMacro"  
  
    ' Terminate the DDE link to Access  
    Text1.LinkMode = 0  
End Sub
```

To use this code, create a new Visual Basic program with a [text box](#) and a [command button](#) with the following properties:

```
Text box  
-----  
    Name: Text1  
  
Command Button  
-----  
    Name: Command1
```

Double-click the command button and enter the code example listed above between the "Sub Command1_Click ()" and "End Sub" lines.

When you run this example, be sure that Microsoft Access is running and that a [database](#) containing a macro called "MyMacro" is open. For testing purposes, you may want to create a macro called "MyMacro" that contains a single MsgBox action.

NOTE: While the above example uses the Microsoft Access system Topic

listed in the LinkTopic line, Microsoft Access is capable of executing a macro with virtually every valid topic it supports. For a list of topics that Microsoft Access supports, [query](#) on the following words in the Microsoft Knowledge Base:

b_waccess and dde and server and topics

[References](#)

PRB: Layout Changes to Subform in Datasheet View Not Saved

Article Number: Q97775
CREATED: 21-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You have a form that contains a subform, and the subform is displayed in Datasheet view. Changes that you make to order and/or the width of the columns in the subform, may not show up in the main form.

CAUSE

The modified subform has not been reread by the main form.

RESOLUTION

There are two alternatives:

1. Close the main form, and then open the subform from the Database window. Make the desired layout changes, then close and save the subform. When you open the main form again, you will see the modified subform displayed.
2. Open the subform from the main form's Design view. Make the desired layout changes, then save the changes to the subform. When you are back in the main form, click your mouse pointer in the subform control and press ENTER. The changes to the subform will show in Form view.

MORE INFORMATION

Below are two methods you can use to commit subform changes to the main form. Follow these steps in the sample database NWIND.MDB.

Steps to Reproduce Behavior

1. Open the Categories form in Form view. The Categories Subform has a field called Product Name.
2. Modify the subform by decreasing the width of the Product Name field. Reverse the tab order of the Unit Price and Quantity Per Unit fields by choosing edit and then tab order.
3. Close and then reopen the Categories form in Form view. Notice that your changes to the Categories Subform do not show up.

The steps below show how to commit subform changes to the main form:

Method 1

1. Open the Categories Subform in Design view from the Database window.
2. Make the same changes you made in step 2 of 'Steps to Reproduce Behavior.'
3. Choose Save Form from the File menu.
4. Open the Categories form in Form view. Now the changes you made to the Categories Subform will show.

Method 2

1. Open the Categories form in Design view. Double-click the Categories Subform control to open the subform in Design view.
2. Switch the Categories subform to Datasheet view by choosing Datasheet from the View menu.
3. Make the same changes you made in step 2 of 'Steps to Reproduce Behavior'.
3. Choose Save Form from the File menu. Close the Categories Subform and go back to the Categories form (which should still be open in Design view).
4. If the Categories Subform control is not still selected, click one time on the Categories Subform control to select the subform control.
5. Click a second time to move the cursor into the subform control. You should see the cursor blinking inside the control at this point.
6. Press ENTER to cause the main form re-evaluate the subform.
7. Switch to Form view of the Categories form. Now the changes you made to the Categories Subform will show.

References:

"Microsoft Access Users Guide," version 1.0, Chapter 10, "Creating Forms Based on More than One Table," page 275

[References](#)

INF: Microsoft Graph Changes May Not Be Retained

Article Number: Q97774
CREATED: 21-APR-1993
MODIFIED: 17-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you create a [graph](#) on a [form](#) or [report](#) and later make some changes to the formatting or to the data itself within Microsoft Graph, these changes may not appear in the form or report.

This article discusses the different sources of information that Graph normally uses and which sources take precedence over others in the hierarchy of graph changes.

MORE INFORMATION

There are several sources of information for a graph: the Microsoft Graph application, the contents of the graph object's RowSource property, and the data in the underlying [table](#) or [query](#).

Microsoft Access regenerates the graph each time you choose Print or switch to [Form view](#) or Print Preview. When Microsoft Access regenerates the graph, it reads the [SQL](#) statement in the object's RowSource property. The name of the graph and the data to be graphed are obtained from this SQL statement. Microsoft Access obtains the format of the data from the table or query that is the source of the graph's SQL statement.

Which changes have precedence is determined as follows:

- Anything that you change in Microsoft Graph can be overwritten by the SQL statement in the graph's RowSource property or by the underlying data itself. (Examples of what you can change in Microsoft Graph include font, font size, font color, and the type of graph selected.)
- Anything that you change in the SQL statement of the graph's RowSource property can be overwritten by the underlying data. (Examples of what you can change in the SQL statement include the title and labels.)
- Anything that you change in the underlying data has precedence over other changes. (An example of what you can change in the underlying data is the actual format of the data. However, if the SQL statement contains a calculated [field](#), such as =Format([Start Date],"MMM"), the format is determined by the result of this calculation, not by the underlying data.)

For more information on customizing the title, labels, and data in a

graph, query on "Graph and Format and Label" or "Graph and Title."

References:

"Microsoft Graph User's Guide," version 3.0, Chapter 3, "Working with a Chart", pages 71-101

[References](#)

PRB: Export Table to Other Database Type Does not Create Index

Article Number: Q97773
CREATED: 21-APR-1993
MODIFIED: 25-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You have a Microsoft Access [table](#) with a [primary key field](#). When you export the table to any of the supported [database](#) formats the [index](#) is not created. Also, if you import a foreign database file that has an index or key, the Microsoft Access table will not have a primary key defined.

RESOLUTION

When you export a Microsoft Access table to an external database format you will need to use the native application for that file type to define the index or key fields. Likewise, after a foreign database file is imported to your Microsoft Access database, you will need to open the table in [Design view](#) and define the primary key field or fields.

CAUSE

The Microsoft Access export facility only exports data, not index definitions.

STATUS

This behavior is by design.

MORE INFORMATION

The following database formats are supported by the Import and Export facilities in Microsoft Access. You will have to use the native product to create indexes on the file after it is exported from Microsoft Access.

- FoxPro 2.x
- Paradox 3.x
- dBASE III +
- dBASE IV
- Btrieve
- [SQL](#) Server

[References](#)

PRB: Unable to Use Hexadecimal or Octal Values in Properties

Article Number: Q97772
CREATED: 21-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You cannot use hexadecimal or octal values referenced as &Hxxxx (for hexadecimal numbers) or &Oxxxx (for octal numbers) in the properties of a [control](#).

CAUSE

Most [control properties](#) expect either a [string expression](#) or a decimal number.

RESOLUTION

Use decimal numbers rather than hexadecimal or octal numbers.

To determine the decimal equivalent of a hexadecimal or octal number, enter that number in the BorderColor property of a control (first write down the original value of the property so that you can easily restore it). Microsoft Access converts hexadecimal and octal numbers (&H or &O format) entered in the color properties of a control to decimal numbers.

MORE INFORMATION

If you enter a hexadecimal or octal number in a control property that expects a string expression (for example, ValidationRule), Microsoft Access converts what you entered to a literal string.

If you enter a hexadecimal or octal number in a control property that expects a number with the Long Integer [data type](#) (for example, BackColor), Microsoft Access converts what you entered to a decimal number.

The following steps demonstrate how this applies to parameters that you may want to pass in a [function](#).

Steps to Reproduce Behavior

1. Open the sample [database](#) NWIND.MDB.
2. Open the Add Products [form](#) in [Design view](#).
3. Select the Product ID [text box](#).

4. From the View menu, choose Properties.
5. In the property sheet, enter the following sample function in the ValidationRule field:

```
=mytest(&HFF)
```

Microsoft Access converts this expression to:

```
=mytest("&HFF")
```

[References](#)

INF: Access Workstation Setup (/N) Instructions Are Confusing

Article Number: Q97771
CREATED: 21-APR-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you set up Microsoft Access to run from a [network](#) server, the on-screen instructions are misleading. This article explains what files are copied to your hard disk drive (or user directory) when you choose to run Microsoft Access from the network server.

MORE INFORMATION

The SETUP /N command for installing Microsoft Access on a workstation displays the following question:

Do you want to install Microsoft Access on your hard disk?

To run from the network server instead of your local hard disk drive, choose No. If you do so, Setup then presents the following message:

Setup will install Microsoft Access in the following directory, which will be created if it doesn't already exist.

This is confusing because it implies that a full copy of Microsoft Access will be installed on your local hard disk drive. This is not the case. Only the following files will be copied:

In the Access directory:

DETCMD.DLL
SETUP.INI
STFSETUP.EXE
STFSETUP.INF
SYSTEM.MDA <- Only if a workgroup is not joined

In the Windows directory:

MSACCESS.INI

WINHELP.EXE <- If not Windows v3.1 or above
WINHELP.HLP <- If not Windows v3.1 or above
WINHELP.INI <- If not Windows v3.1 or above

In the WINDOWS\SYSTEM directory:

CTL3D.DLL

COMMDLG.DLL <- If not Windows v3.1 or above

DDEML.DLL <- If not Windows v3.1 or above
OLECLI.DLL <- If not Windows v3.1 or above
OLESRV.DLL <- If not Windows v3.1 or above
SHELL.DLL <- If not Windows v3.1 or above
VER.DLL <- If not Windows v3.1 or above

These are the necessary files for running Microsoft Access from the network.

[References](#)

INF: How to Determine Number of Working Days Between Two Dates

Article Number: Q97757
CREATED: 20-APR-1993
MODIFIED: 20-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article explains how to calculate the number of working days between two dates.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to "Introduction to Programming."

MORE INFORMATION

There is no intrinsic [function](#) within Microsoft Access to determine the number of working days between two given dates. To determine this, you need to call an Access Basic function. The function below includes the start date and the end date, so the number of days between 02/02/93 and 02/03/93 equals two.

NOTE: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

Option Explicit

```
Function Work_Days (BegDate As Variant, EndDate As Variant) As Integer
    Dim WholeWeeks As Variant
    Dim DateCnt As Variant
    Dim EndDays As Integer

    BegDate = DateValue(BegDate)
    EndDate = DateValue(EndDate)
    WholeWeeks = DateDiff("w", BegDate, EndDate)
    DateCnt = DateAdd("ww", WholeWeeks, BegDate)
    EndDays = 0
    Do While DateCnt <= EndDate
        If Format(DateCnt, "ddd") <> "Sun" And _
            Format(DateCnt, "ddd") <> "Sat" Then
            EndDays = EndDays + 1
        End If
        DateCnt = DateAdd("d", 1, DateCnt)
    Loop
    Work_Days = WholeWeeks * 5 + EndDays
End Function
```

Example

To call the function, you can pass either a valid [string](#) or an actual date value. The following are two ways to call this function from the [Immediate window](#):

```
?Work_Days("01/01/93", "12/31/96")  
1043
```

```
?Work_Days(#03/05/93#, #04/06/93#)  
23
```

[References](#)

PRACC9304: Cue Cards Can't Be Started from Workstation

Article Number: Q97625
CREATED: 15-APR-1993
MODIFIED: 14-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you try to run Cue Cards from a workstation set up for a network installation of Microsoft Access, you receive the following error message:

Cue Cards Couldn't Be Started Because of Incomplete Setup.

CAUSE

Microsoft Access has been set up to run from a network drive and the path used to execute MSACCESS.EXE uses the universal naming convention (UNC).

RESOLUTION

For Cue Cards to run on a workstation, the path used to execute MSACCESS.EXE from Program Manager must use the complete conventional path, including the drive letter.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

To correct this problem, use the following steps:

1. In Program Manager, select the Microsoft Access icon.
2. From Program Manager's File menu, choose Properties.
3. In the Command Line text box, type the complete path, including the drive letter for MSACCESS.EXE, (for example, F:\ACCESS\MSACCESS.EXE).

[References](#)

INF: How to Synchronize Two Combo Boxes on a Form

Article Number: Q97624
CREATED: 15-APR-1993
MODIFIED: 16-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The information below describes how to create two combo boxes so that making a selection in one limits the list of available items in the other. For example, suppose a user needs to enroll in a school course. The user selects a class from one combo box and then selects from a second combo box a time and location from a list of available times for the class selected in the first combo box.

MORE INFORMATION

The following example uses the NWIND.MDB sample database supplied with Microsoft Access. One combo box will list the available product categories; the second will list the available products for the category selected in the first combo box.

1. Create a blank, unbound, form with two combo boxes where the properties are set to the following:

```
Combo Box 1
-----
ControlName:  Categories
RowSourceType: Table/Query
RowSource:    Categories
ColumnCount:  1
BoundColumn:  1
After Update: Refresh Products
```

```
Combo Box 2
-----
ControlName:  Products
RowSourceType: Table/Query
RowSource:    Category Combo Query
ColumnCount:  2
ColumnWidth:  1;0
BoundColumn:  1
```

2. Save this form as "Categories and Products".
3. Create a new query based on the Products table.
4. Add the Product ID, Product Name, and Category ID fields to the query grid.
5. For the Category ID field, clear the Show checkbox and type the

following in the criteria row:

```
Forms![Categories and Products]![Categories]
```

6. Save this query as "Category Combo Query".

7. Create a new macro as follows:

Action	Arguments
Requery	ControlName: Products

8. Save this macro as "Refresh Products".

Now, whenever a category is selected in the first combo box, the second box will list the available products for the selected category.

How This Technique Works

The second combo box is filled with the contents of Category Combo Query. This query finds all products that have a Category ID that matches the Category ID selected in the first combo box on the form.

Whenever a category is chosen from the first combo box, the AfterUpdate property executes the Refresh Products macro, which forces the second combo box to run Category Combo Query again. The macro forces the second combo box to refresh its list of products based on the new category selection. Without this macro, the user would have to force the second combo box to refresh itself by pressing the F9 function key.

[References](#)

PRB: Cannot Edit Access Data Using Query with Attached dBASE

Article Number: Q97623
CREATED: 15-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You are unable to edit fields in a Microsoft Access table through a query that joins an attached dBASE table to the Microsoft Access table. The query joins the tables on a key field in the Microsoft Access table.

CAUSE

Since relationships can't be created with attached tables using the Edit Relationship dialog box, joins must be performed manually in the OBE grid. When a join is created between an attached dBASE table and a Microsoft Access table on the Primary key field, Microsoft Access assumes the type of relationship is one-to-many with the Microsoft Access table representing the "one" side. You cannot edit fields that come from the "one" side of a one-to-many relationship.

RESOLUTION

Establish a main/subform setup with the subform based on the attached dBASE table and the main form based on the Microsoft Access table. You will be able to edit data from both tables through their respective forms.

MORE INFORMATION

Even if the data in the dBASE table actually has a one-to-one relationship with the Microsoft Access table, you will not be able to edit fields from the Microsoft Access table using a query joining these tables. If you create a single form based on this query, and set the form's AllowUpdating property to Any Tables, you will not be able to edit data in the Microsoft Access table.

References

PRB: Access Basic Does Not Support Yes and No Action Arguments

Article Number: Q97622
CREATED: 15-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you use the DoCmd statement in [Access Basic](#), Microsoft Access does not recognize a Yes or No argument.

RESOLUTION

Use True for the argument instead of Yes, and use False instead of No. True and False are reserved words in Access Basic.

STATUS

This behavior is by design.

MORE INFORMATION

Even though [macro](#) actions accept the Yes or No [action argument](#), Access Basic recognizes only True or False. However, the No argument will actually work because Microsoft Access does not recognize No. Therefore, a zero value is returned, which is interpreted as False. However, a Yes argument will also return a zero value, or False. Whether you use Yes or No, the argument will always be interpreted as False.

Steps to Reproduce Behavior

1. Create the following [module](#):

```
'*****  
'Declarations Section of the module.  
'*****  
Option Compare database  
  
'Do not include the Option Explicit or you will receive an error  
'message stating that you did not declare your variable "no".  
'This is because Access Basic does not recognize "no" as a  
'reserved word.  
  
'=====
```

'The following [function](#) will run a macro action that will turn
'warnings off, then attempt to turn them back on.
'=====

```
Sub SetWarnings()  
    DoCmd SetWarnings No      'Turn warnings off
```

```
    DoCmd SetWarnings Yes    'Turn warnings on
End Sub
```

2. Run the sub procedure from the [Immediate window](#) by typing the following in the window and pressing ENTER:

```
SetWarnings
```

3. Try to delete a [form](#), [report](#), or [table](#). Note that you did not receive any warnings. The sub procedure turned off the warnings, but did not recognize the Yes argument, so it did not turn them back on.

4. Change the sub procedure to:

```
Sub SetWarnings()
    DoCmd SetWarnings No
    DoCmd SetWarnings True
End Sub
```

5. Run the sub procedure from the Immediate window again.
6. Try to delete a form, report, or table. Note that you will receive a warning. The sub procedure recognized the True argument and turned the warnings back on.

[References](#)

PRACC9303: GPF Declaring New Object Variable During Break Mode

Article Number: Q97533
CREATED: 13-APR-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0
-

SYMPTOMS

A general protection (GP) fault occurs when you declare a new object variable, such as a form or a table, during break mode (when debugging Access Basic code,) and then attempt to recompile the code. The following are examples of object variable declarations:

```
Dim F as Form
Dim T as Table
```

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

Steps to Reproduce Problem

1. Load a database, or create a new one. From the database window, choose module and then new.
2. In the module window, enter this new function:

```
Function GP_Fault()
    z=Forms!Form!X
End Function
```

After entering the function, close and save the module.

3. Create a new form. In the form's OnOpen property, enter:

```
=GP_Fault()
```

Close the form and save it.

4. Open the form. You receive the error "Invalid reference to form 'Form'". Choose OK.
5. The cursor should now be in the module on the bogus "z=Forms!Form!X" statement. Move down a line and add the following statements to the GP_Fault function:

```
Dim T as Form
```

Set T = Forms![GPF]

6. Choose Compile All from the Run menu. Close the module window. Choose OK for the "Save Changes?" and the reset halted dialog boxes.

Result: You receive a GPF.

The order of events listed above must be followed in this order to generate the GPF.

[References](#)

PRACC9303: GPF Using Empty String with CVDDate() or IsDate()

Article Number: Q97532
CREATED: 13-APR-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

If either the CVDDate() or IsDate() function is called with an empty string as the expression, such as CVDDate(""), Microsoft Access causes a general protection (GP) fault.

RESOLUTION

Verify that you are passing valid date strings in the argument to CVDDate() or IsDate().

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

The CVDDate() function converts strings or numeric expressions to a variant of data type date. The IsDate() function returns a value indicating whether or not a variant argument can be converted to a date.

[References](#)

PRB: "Internal Database Error (-5001)" Msg Using Btrieve NLM

Article Number: Q97531
CREATED: 13-APR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You receive the following error message when you are using the Btrieve Network Loadable Module (NLM) with Microsoft Access:

Internal Database Error (-5001)

CAUSE

You do not have correct versions of the BREQUEST.EXE, WBTRCALL.DLL, and WBTRVRES.DLL files. In order for the Btrieve NLM to function properly with Microsoft Access, Microsoft recommends that the NLM requester (client) use BREQUEST.EXE, WBTRCALL.DLL, WBTRVRES.DLL version 6.0, and BTRIEVE.NLM version 5.15.

RESOLUTION

Use the following procedure to obtain the correct versions of these Btrieve files:

1. Sign on to CompuServe.
2. At the exclamation point (!) prompt, type "go novlib" (without the quotation marks).
3. Choose the BTRIEVE/XQL library section.
4. Download the BT515.ZIP file and unzip it using the PKUNZIP.EXE file.

Result: The BTRREQ.ZIP file is now available.

5. Unzip the BTRREQ.ZIP file.

The following files are now available.

WBTRCALL.DLL
WBTRVRES.DLL
BREQUEST.EXE

Follow the instructions in the README file to properly install and execute these files and for information on how to apply patches 1-65 to the BTRIEVE.NLM file on your server.

6. Add the following lines to your WIN.INI file:

[BREQUESTDPMI]
datalength=4096
tasks=10
local=no
chkparms=no

MORE INFORMATION

Btrieve and the Btrieve NLM are manufactured by Novell, Inc., a company independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

PRACC9303: GP Fault When RunningSum Tries to Evaluate #ERROR

Article Number: Q97530
CREATED: 13-APR-1993
MODIFIED: 14-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

If a report contains a control whose expression evaluates to a #ERROR message, changing the RunningSum property to Over Group or Over All causes a general protection (GP) fault.

CAUSE

This problem commonly occurs if you place the Sum() function in a text box that appears in the Page Footer section of your report. Aggregate functions, like Sum(), Count(), Avg(), and so forth, are not supported in report headers and footers and return a #ERROR message if placed there. Adding the RunningSum property to a control using the aggregate of the Sum() function causes the above GP fault, as does adding the RunningSum property to any control that returns a #ERROR message.

RESOLUTION

Either correct the expression in the ControlSource property so that it evaluates correctly, or move the location of the control on the report form.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

For an example of how to place totals in page headers and footers, see pages 576-579 of the "User's Guide" for version 1.0.

[References](#)

PRACC9303: GPF Using OpenForm Action to Hide Modal Pop-Up Form

Article Number: Q97529
CREATED: 13-APR-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

Using the OpenForm action to hide a modal pop-up form may cause a general protection (GP) fault.

RESOLUTION

Instead of using the OpenForm action to hide a modal pop-up form, use the SetValue action to set the form's Visible property to False.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

[References](#)

PRACC9210: Fast Search on Indexed Paradox Column Fails

Article Number: Q97528
CREATED: 13-APR-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

A Fast Search on an indexed Paradox column fails, even when there is data to match the search criteria.

CAUSE

This problem occurs when a Paradox column (field) name has the same name as the Paradox table it is contained in.

In Paradox, the primary index is always stored in a table named <TABLENAME>.PX, and secondary indexes are stored in tables named <COLUMNNAME>.NXX. For example, with a table named "Authors" that contains fields named "Authors", "Books", and "Publisher", the primary index file is named AUTHORS.PX, and the secondary indexes are named BOOKS.N01, PUBLISHER.N02, AUTHORS.N03.

When Microsoft Access performs a Fast Search, it ignores the file extensions for these indexes. As a result, if a column name is the same as a table name, Microsoft Access sees two indexes with the same name and assumes it to be a single multicolumn index instead of two separate single-column indexes. This results in strange behavior, such as not getting the results you would expect from a search or query.

RESOLUTION

Be sure that the field names in the Paradox table differ from the table name.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

PRACC9210: MS Access Ignores Paradox Private Directory Rules

Article Number: Q97527
CREATED: 13-APR-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Microsoft Access ignores Paradox private directory restrictions and allows you to examine and modify data in tables in them. Paradox, however, ignores any changes made by Microsoft Access, since it assumes the directories to be private.

RESOLUTION

If the directory or directories in question are set as working directories, they work correctly.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Paradox version 3.5 allows you to specify directories as "private directories" (Paradox Menu: Tools/Net/SetPrivate). Private directories are used to hold Answer tables and the results of queries, as well as other types of data, and cannot be examined or changed by other users. If another Paradox user attempts to do so, he or she receives the error message, "This is a private directory, you are not allowed access."

[References](#)

PRB: 'Circular Reference Caused by Alias' Error Message

Article Number: Q97526
CREATED: 13-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

An attempt to run a [query](#) fails and Microsoft Access displays the following message:

Circular reference caused by [alias](#) '[field](#) name' in query definition's SELECT list.

CAUSE

The alias, or [label](#), of a calculated field cannot be identical to any of the field names used to calculate the field.

RESOLUTION

Change the name of the calculated field or let Microsoft Access automatically assign a name for the field.

MORE INFORMATION

Perform the following seven steps to demonstrate this behavior.

Steps to Reproduce Behavior

-
1. Open the NWIND.MDB sample [database](#).
 2. In the [Database window](#), choose the Query button, then choose New. A query design screen appears, with a [dialog box](#) that prompts for a Table or Query to add to the query.
 3. Select Catalog, then choose the Add button.
 4. Choose the Close button to close the Add Table dialog window.
 5. In the first field of the Query design grid, enter the following:

unit price: [unit price]/100
 6. From the Query menu choose Run. The following error occurs:

Circular reference caused by alias 'unit price' in query definition's SELECT list.
 7. You can change the alias or label to resolve the error. For example,

replace the unit price cfield name with the following:

price: [unit price]/100

References:

For more information on the error, search for "error messages: reference" then search for the specific error message using the Microsoft Access Help menu.

For more information on renaming queries, search for "queries: renaming fields" then "Renaming a Field in a Query" using the Microsoft Access Help menu.

[References](#)

PRB: UndoCurrentRecord In BeforeUpdate Property Causes Error

Article Number: Q97525
CREATED: 13-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you create a [macro](#) that contains a DoMenuItem action that chooses Undo Current Record from the Edit menu, and you run the macro from a [form](#)'s BeforeUpdate property, Microsoft Access generates the following message when you edit data in the form:

Command not available: UndoCurrentRecord.

However, if you select the Edit menu, Undo Current Record is available and it works properly.

CAUSE

The Undo Current Record command is not available for use with a macro run from a form's BeforeUpdate property.

RESOLUTION

Use the SendKeys action to send an ESC key instead. (This has the same effect as running the Undo Current Field command.)

MORE INFORMATION

Page 48 of the "Microsoft Access Language Reference," version 1.0 or 1.1, discusses uses for this type of macro in the BeforeUpdate property, and indicates that calling Undo Current Record works correctly in such a macro. This is not the case. Calling Undo Current Record causes the error message above and should not be used.

REFERENCES

=====

"Microsoft Access Language Reference," versions 1.0 and 1.1,
"BeforeUpdate, AfterUpdate Properties", pages 47-48.

[References](#)

INF: Round or Truncate Values to Desired Number of Decimals

Article Number: Q97524
CREATED: 13-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

The Format property of a [control](#) can round a number or currency [field](#) to the desired number of decimal places. However, this does not change the underlying data, which may contain additional digits that the control does not display. If you add the values in this control, the sum is based on the actual values and not the displayed values, which can cause problem with perceived accuracy.

The text below presents four [Access Basic](#) functions that round or truncate data to two decimal places so the displayed and formatted value and the actual numeric or currency data are the same.

More Information:

This article assumes that you are familiar with Access Basic code.

The functions are presented in two styles. The first style is appropriate for the AfterUpdate property of a [form](#) control, to ensure that the data entered matches the data displayed. The second style is for use in expressions and calculated controls.

To round or truncate numbers to two decimal places, create a new [module](#) in Microsoft Access, create an appropriate declaration section, and add the following functions.

```
'*****  
'Declarations section of the module  
'*****
```

```
Option Explicit  
Const Factor = 100
```

```
'=====
```

```
' RoundAU and TruncAU are designed to be added to the  
' AfterUpdate property on a form control.  
'=====
```

```
Function RoundAU(X)  
    X = Int(X * Factor + 0.5) / Factor  
End Function
```

```
Function TruncAU(X)  
    X = Int(X * Factor) / Factor  
End Function
```

```
'=====
' RoundCC and TruncCC are designed to be used in
' expressions and calculated controls on forms and reports.
'=====
```

```
Function RoundCC(X)
    RoundCC = Int (X * Factor + 0.5) / Factor
End Function
```

```
Function TruncCC(X)
    TruncCC = Int (X * Factor) / Factor
End Function
```

Examples of Using the Round and Truncate Functions

The following examples use the NWIND.MDB sample [database](#) provided with Microsoft Access.

- Add the TruncAU [function](#) to the AfterUpdate property of the Unit Price field in the Products form, as follows.

```
Form: Products
-----
ControlName: Unit Price
    AfterUpdate: =TruncAU([Unit Price])
```

If the user accidentally enters \$23.055 instead of \$23.05, the TruncAu function catches the mistake and changes the value to \$23.05. Instead, if you install RoundAu, the function changes the value to \$23.06. If you use neither function, the value displays as \$23.06 but the entered value, \$23.055, is used in calculations.

- Add TruncCC or RoundCC to expressions in a [report](#) or a SetValue [macro](#) action. The following example demonstrates using RoundCC in a report [footer](#).

```
Report: Summary of Sales By Year
-----
ControlName: Total Sales for Quarter
    ControlSource: =Sum(RoundCC([Order Amount]))

ControlName: Total Sales for Year
    ControlSource: =Sum(RoundCC([Order Amount]))
```

If you use RoundCC, the report sums the values displayed in the report, even though the actual values may contain hidden digits.

NOTE: To change the number of decimal places used, change the value of the global [constant](#), Factor, as follows:

```
10 = 1 decimal place
100 = 2 decimal places
1000 = 3 decimal places, etc.
```

[References](#)

INF: Displaying Start, Finish, and Elapsed Times on a Form

Article Number: Q97523
CREATED: 13-APR-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

The text below demonstrates creating command buttons to enter start and finish times on a [form](#). Elapsed time is also displayed on request.

This article assumes that you are familiar with [Access Basic](#) and with creating applications using the programming tools provided with Microsoft Access.

More Information:

The following seven steps use macros and Access Basic code to display start and finish times on a form.

1. Create a new [module](#) called Module 1. Create the appropriate declaration section and add the following [function](#).

Note: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----  
' GLOBAL DECLARATION  
'-----  
Option Compare Database  
Option Explicit  
  
'=====
```

' The following function calculates elapsed time.
' This function displays the total time elapsed in the
' following format:
' <hours>:<minutes>:<seconds>
' For example:
' 48:57:04
'=====

```
Function TimePassed (Startup, Done)  
    TimePassed = Format(Int(Abs(Startup - Done) * 24), "###:") + _  
                  Format(Abs(Startup - Done), "nn:ss")  
End Function
```

2. Create a [macro](#) to set the elapsed time, as follows.

Macro Name	Action
-----	-----
Set Elapsed Time	Set Value

Set Elapsed Time Actions

Set Value

Item: [Elapsed Time]

Expression: TimePassed([Start Time],[Finish Time])

3. Create a macro to set the start and finish time, as follows.

Macro Name	Condition	Action
SetTime	IsNull(Forms![Time Form]![Start Time])	Set Value
	Not IsNull(Forms![Time Form]![Start Time])	Set Value

SetTime Actions

Set Value

Item: [Start Time]

Expression: Now()

Set Value

Item: [Finish Time]

Expression: Now()

4. Create a new blank, unbound form (do not select a [table](#) or [query](#)).
5. Add three [text box](#) controls to the form and assign them the following properties:

Object: Text Box

ControlName: Start Time

Enabled: No

Locked: Yes

Object: Text Box

ControlName: Finish Time

Enabled: No

Locked: Yes

Object: Text Box

ControlName: Elapsed Time

Enabled: No

Locked: Yes

6. Add three [command button](#) controls to the form, and assign them the following properties:

Object: Command Button

ControlName: Begin

Caption: Start

OnPush: SetTime

Object: Command Button

ControlName: Complete
Caption: Finish
OnPush: SetTime

Object: Command Button

ControlName: Time Elapsed
Caption: Time Elapsed
OnPush: Set Elapsed Time

Save the form as Time Form.

7. Choose [Form view](#) to test the example as follows.
 - a. Choose the Start button. The current date and time display in both the start time and finish time boxes.
 - b. Choose the Start button again. While the start time should not change, the finish time will [update](#) to the current data and time.
 - c. Choose the Finish button to set or reset the finish time.
 - d. Choose the Time Elapsed button to display the elapsed time in days, hours, and minutes.

References:

"Introduction to Programming", version 1.0, Chapter 3, "Access Basic Essentials".

[References](#)

INF: Sample DDE Macro Communicates with Microsoft Excel 4.0

Article Number: Q97522
CREATED: 13-APR-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
 - Microsoft Excel for Windows, version 4.0
-

SUMMARY

The text below demonstrates communication between two applications through dynamic data exchange (DDE). Specifically, one example shows a [macro](#) developed for Microsoft Excel version 4.0 for Windows that starts Microsoft Access, loads a [database](#) and runs a macro. A second example shows an [Access Basic function](#) that loads Microsoft Excel and runs a macro.

MORE INFORMATION

Running a Microsoft Access Macro from Microsoft Excel

Perform the following three steps to demonstrate this behavior. These steps assume that a "MESSAGE" macro already exists in Microsoft Access.

1. In Microsoft Excel, choose New from the File menu, select Macro Sheet and choose OK.
2. Enter the following macro into the macro sheet. Substitute the appropriate location for files on your computer.

```
=EXEC("c:\access\msaccess.exe c:\access\db4.mdb")
Chan=INITIATE("MSACCESS","system")
=APP.ACTIVATE("Microsoft Access")
=EXECUTE(Chan,"MESSAGE")
=TERMINATE(Chan)
=RETURN()
```

3. To run the macro, select the first [cell](#), select Run from the Macro menu and choose OK.

The EXEC function in the macro loads Microsoft Access minimized and loads the DB4.MDB database. The macro initiates a DDE channel and assigns it to the [variable](#) Chan. The APP.ACTIVATE switches to the Microsoft Access window to show the macro actions running.

The EXECUTE function runs a macro named "MESSAGE," as follows.

Name Summary:

Application Name : "MSACCESS:"

Window Title:"Microsoft Access"
Topic: "System"
Item: "Message"

Running a Microsoft Excel Macro from Microsoft Access

Perform the following two steps to demonstrate this behavior. These steps assume that a macro named "Message" already exists in a Microsoft Excel macro sheet named "MACRO1.XLM."

1. In Microsoft Access, create a new function that contains the following text. Substitute the appropriate location for files on your computer.

```
Function CallExcel ()  
    Dim Chan  
    x = Shell("c:\excel\excel.exe c:\excel\macro1.xlm", 1)  
    Chan = DDEInitiate("Excel", "System")  
    DDEExecute Chan, "[Run(""macro1.xlm!Message"")]"  
    DDETerminate Chan  
End Function
```

2. Run the macro. For more information on how to run the macro, see the "Macros: running" topic using the Help menu.

The Shell function loads Microsoft Excel and the "MACRO1.XLM" macro sheet full screen and leaves the focus on Microsoft Excel. The macro initiates a DDE channel and assigns it to the variable Chan.

The EXECUTE function runs a macro named "Message".

Name Summary:

Application Name : "Excel"
Window Title: "Microsoft Excel"
Topic: "System"
Item: "[Run(""macro1.xlm!Message"")]"

The [syntax](#) for these two macros demonstrates that the correct syntax for the "Item" depends on the target application.

References:

Microsoft Access README.TXT
Microsoft Excel "Function Reference" manual

[References](#)

PRACC9303: Hang Importing .DBF Files Longer Than dBASE IV Max.

Article Number: Q97521
CREATED: 13-APR-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

CodeBase++ version 1.07 allows you to create .DBF files where the **record** length exceeds the dBASE IV maximum record length. Microsoft Access may hang (stop responding) when trying to import this type of .DBF file.

RESOLUTION

Export the CodeBase++ .DBF file to delimited text, then import the ASCII file into Microsoft Access.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

INF: Maximum Number of Rows in Table When Counter is the Key

Article Number: Q97520
CREATED: 13-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The maximum value of a [Counter data type](#) is the same as the maximum value of a Long Integer. A Long Integer is 32 bits long and has a maximum positive value of 2,147,483,647 and a minimum negative value of -2,147,483,648. If you use a Counter as the [primary key](#), a [table](#) can contain up to four billion records. The maximum number of records in the table is also limited by the maximum size of a [database](#). For more information, see Appendix A in the "Microsoft Access User's Guide," version 1.0.

MORE INFORMATION

Values in a Counter [field](#) increment from 1 to 2,147,483,647. The next [record](#) receives the minimum negative value for a Long Integer, -2,147,483,648. The Counter value for each subsequent record is one greater than the previous value, until it reaches zero.

[References](#)

INF: Correct Syntax for TransferDatabase to dBASE and Paradox

Article Number: Q97519
CREATED: 13-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

Summary:

When you attempt to export a Microsoft Access table to dBASE or Paradox, you may encounter errors. The arguments for the TransferDatabase action are not clear in Microsoft Access Help or on pages 473-476 of the "Microsoft Access Language Reference."

This article explains the correct argument syntax for exporting the Categories table from the NWIND.MDB sample database to dBASE III using the TransferDatabase macro action. The equivalent Access Basic syntax is also explained.

More Information:

Note: Both of the following examples export the Categories table to a dBASE file in the C:\DBASE directory. You need to enter a valid directory in your macro or you may receive the following error message when you run the macro or function:

Not a valid path

Example 1 - Using the TransferDatabase Macro Action

Create a macro named Export1. This macro will export the Categories table to a dBASE file.

Macro Name	Action
Export1	TransferDatabase

Export1 Actions

TransferDatabase	
Transfer Type:	Export
Database Type:	dBASE III
Database Name:	C:\dbase
Object Type:	Table
Source:	Categories
Destination:	Test1.dbf
Structure Only:	No

Note: To use the TransferDatabase macro action to export to a Paradox table, you must choose "Paradox" for the Database Type argument and enter the Destination argument as "Test1.db".

Example 2 - Using an Access Basic Function

Create a new [module](#) and enter the following function:

```
' Note: The DoCmd statement should all be entered on one line.  
' The line continuation character '_' improves the readability  
' of the code.
```

```
Function ExportdBASE ()  
    DoCmd TransferDatabase a_export, "dBASE III", "c:\dbase", _  
        a_table, "Categories", "Test1.dbf"  
End Function
```

[References](#)

INF: How to Select More than One Item from a List Box

Article Number: Q97518
CREATED: 13-APR-1993
MODIFIED: 13-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The text below demonstrates how to create a pair of list boxes to imitate the multiple selection behavior used in AccessWizards.

MORE INFORMATION

The steps below create a new [form](#) that contains two list boxes and three command boxes. You can add or delete items from the list using either the Add Item or Delete Items buttons or the properties of the list boxes. Perform the following 7 steps.

1. Create the following [table](#):

Table: Table1

Field Name: List [Primary Key]
Data Type: Text
Field Size: 15 (This may vary)
Caption: Items that will be provided in list

Field Name: Selected
Data Type: Text
Field Size: 5
Caption: Indicates if the item has been selected

2. Choose [Datasheet view](#) and add five records. For each [record](#), place some data into the List [field](#) (for example, a list of colors or cities). Set the Selected field to the word "YES" (without quotation marks).

3. Create two queries based on Table1:

Query: Select Yes

Field Name: List
Show: True
Criteria: [selected] = "YES"

Query: Select NO

Field Name: List
Show: True
Criteria: [selected] = "NO"

4. Create the following three functions:

```
'=====
'The following function opens the table and changes the
'   selected value from YES to NO.
'Then the code executes the query for the two list boxes
'   so they display the updated values.
'=====
Function ADD ()
    Dim MyDB As Database
    Dim MyTable As Table
    Dim y As Control

    Set MyDB = CurrentDB()
    Set MyTable = MyDB.OpenTable("Table1")
    Set y = forms!SelectList.field0

    MyTable.index = "PrimaryKey"
    MyTable.Seek "=", y

    MyTable.Edit
    MyTable.[selected] = "no"
    MyTable.Update

    MyTable.Close

    DoCmd Requery "field0"
    DoCmd Requery "field1"
End Function

'=====
'The following function opens the table and changes the
'   selected value from NO to YES.
'Then the code executes the query for the two list boxes
'   so they display the updated values.
'=====
Function Del ()
    Dim MyDB As Database
    Dim MyTable As Table
    Dim y As Control

    Set MyDB = CurrentDB()
    Set MyTable = MyDB.OpenTable("Table1")
    Set y = forms!SelectList.field1

    MyTable.index = "primarykey"
    MyTable.Seek "=", y
    MyTable.Edit
    MyTable.[selected] = "yes"
    MyTable.Update
    MyTable.Close

    DoCmd Requery "field0"
    DoCmd Requery "field1"
End Function

'=====
```

```
'The following function sets all values in the field
'   "Selected" to yes.
'Then the code executes the query for the two list boxes
'   so they display the updated values.
```

```
'=====
```

```
Function Clear ()
    Dim MyDB As Database
    Dim MyTable As Table

    Set MyDB = CurrentDB()
    Set MyTable = MyDB.OpenTable("Table1")

    On Error GoTo erhandle
    MyTable.MoveFirst

    Do Until MyTable.EOF
        MyTable.Edit
        MyTable.[selected] = "yes"
        MyTable.Update
        MyTable.MoveNext
    Loop

erhandle:
    Resume Next

    DoCmd Requery "field0"
    DoCmd Requery "field1"
End Function
```

5. Create a blank unbound form named SelectList.
6. Add two list box controls to the form and assign the following properties:

```
Object: list box
-----
ControlName:   Field0
ControlSource:
RowSourceType: Table/Query
RowSource:     Select Yes
On Dbl Click: =ADD()
```

```
Object: list box
-----
ControlName:   Field1
ControlSource:
RowSourceType: Table/Query
RowSource:     Select No
On Dbl Click: =Del()
```

The first list box, Field0, displays the items to be selected. The second list box, Field1, displays items that are selected.

7. Add three command button controls to the form and assign the following properties:

```
Object: command button
```

```
-----  
ControlName: Button One  
Caption: Clear  
OnPush: =Clear()
```

```
Object: command button  
-----
```

```
ControlName: Button Two  
Caption: Add item  
OnPush: =Add()
```

```
Object: command button  
-----
```

```
ControlName: Button Three  
Caption: Delete item  
OnPush: =Del()
```

[References](#)

INF: Print Total Number of Pages on Each Page of Report

Article Number: Q97517
CREATED: 13-APR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can print the page number on each page of a [report](#) by using the Page property in a [text box control](#) in the page [footer](#). This article explains how to also print the total number of pages of the entire report on each page (for example, 1/20).

NOTE: This [method](#) will not work when placing the final text box control, PageNum, in a page [header](#).

This article assumes that you are familiar with [Access Basic](#) and how to create and use Access Basic procedures.

MORE INFORMATION

To print the total number of pages of a report at the bottom of each page, do the following:

1. Open a new [module](#) or a previously created module and enter the following code:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declarations section of the module.  
'*****  
Option Explicit  
Global TotalPages as Integer  
  
'=====   
' Create the following GetTotalPages() function in the Module  
'=====   
Function GetTotalPages()  
    GetTotalPages = TotalPages  
End Function  
  
'=====   
' Create the following GotoEnd() function in the Module  
'=====   
Function GotoEnd()  
    TotalPages = 0  
    SendKeys "%z{END}", False  
    ' SendKeys does not work if report is maximized
```

End Function

```
'=====
' Create the following SaveTotalPages() function in the Module
'=====
Function SaveTotalPages(TotPgNum As Integer)
    If TotalPages = 0 Then
        TotalPages = TotPgNum
        SendKeys "{HOME}", False
    End If
End Function
```

2. Open your report in [Design view](#).
3. From the View menu, choose Properties to display the property sheet. Add the following function to the report's OnOpen [event](#):

```
OnOpen: =GotoEnd()
```

4. Place a text box control called MyPage in the Page Footer.

```
Object: Text Box
-----
ControlName: MyPage
ControlSource: =Page
Visible: No
```

5. Change the Report Footer OnFormat property to the following:

```
OnFormat: =SaveTotalPages([MyPage])
```

6. Place a text box control in the Page Footer as follows:

```
Object: Text Box
-----
ControlName: PageNum
ControlSource: =Page & "/" & GetTotalPages()
```

When you print your report, this text box control will print the page number as 1/n where n is the total number of pages in the report.

[References](#)

PRB: Random Characters in Text Box Bound to OLE Object Field

Article Number: Q97516
CREATED: 13-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In Form view, when you set the ControlSource property for a text box to an OLE Object field, strange characters appear in the text box.

RESOLUTION

Change the ControlSource for the text box to a field that is not an OLE Object and use an OLE Object Frame to display the contents of the OLE Object field.

MORE INFORMATION

The text box displays random characters, such as vertical bar characters or apostrophes, when a text box is bound to an OLE Object field and the current record contains an OLE Object in that field. If the OLE Object field in the current record is empty, no characters appear in the text box.

[References](#)

INF: Passing a Null Pointer to an External DLL

Article Number: Q97515
CREATED: 13-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The null value is a valid or required parameter for some dynamic-[link](#) libraries (DLLs). To specify a null value, use 0&.

The text below assumes some familiarity with [Access Basic](#) and creating and using Access Basic procedures.

MORE INFORMATION

When your procedure calls a [function](#) and passes the [expression](#) 0&, the ampersand (&) specifies a 32-bit (far) null pointer. In a function declaration, an As Any parameter instructs Access Basic to omit type checking for that parameter and to pass the value to the called function.

The following code demonstrates the correct [method](#) to declare an Access Basic function that passes a null parameter to a function in a DLL. Open a new [module](#) or edit a previously created Module to contain the following code.

NOTE: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

NOTE: Some functions of the Microsoft Windows application programming interface (API) may be declared in an existing Microsoft Access library. If so, the declarations below are duplicates. If the duplicate procedure name error occurs, remove or comment out the offending declare statement from your code.

```
'*****  
'Declarations section of the module.  
'*****
```

Option Explicit

```
'=====
```

```
' Declare the WriteProfileString API function from the external  
' Windows dynamic-link library "kernel."  
'=====
```

```
Declare Function WriteProfileString% Lib "Kernel" (  
    ByVal lpApplicationName As Any,  
    ByVal lpKeyName As Any,  
    ByVal lpString As Any)
```

```
'=====
' Create the following FlushIniCache%() function in the Module. The
' following function calls the external function, specifying null for
' each argument. This causes WriteProfileString to flush its internal
' cache and writes to disk any changes to WIN.INI. NOTE: Using this
' function incorrectly can cause a general protection fault or modify
' your WIN.INI file.
'=====
Function FlushIniCache%()
    FlushIniCache = WriteProfileString% (0&, 0&, 0&)
End Function
```

[References](#)

INF: Writing Functions Called from Events or Expressions

Article Number: Q97514
CREATED: 13-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Functions can be used in a variety of places in Microsoft Access. How you write your functions depends on where the functions are going to be called from.

This article assumes that you are familiar with [Access Basic](#) and with creating applications for Microsoft Access using the programming tools provided with Microsoft Access.

MORE INFORMATION

There are two main styles for writing Access Basic functions:

- Functions can be called from [event](#) properties, such as the AfterUpdate property for a [control](#) on a [form](#). Typically, you would call the [function](#) with a parameter. The function then acts on or modifies the parameter.
- Functions can be used in expressions, such as calculated controls. The difference is in how the result is returned when the function exits.

The examples below use the Proper() function to illustrate the differences between the two function styles. Proper() converts the first letter of a word to uppercase and the other letters to lowercase.

Calling a Function from an Event Property

=====

The Proper() function can be written so it can be called from an event, such as the AfterUpdate property of a control on a form. In this example we will call it ProperAU() as a reminder that it should be called from the AfterUpdate property.

Enter the following function in a [module](#):

```
Function ProperAU(Field As Control)
    Field=UCase(Left(Field,1)) & LCase(Mid(Field,2))
End Function
```

NOTE: The result of the calculation updates the [field](#) that was passed as a parameter.

Example

1. Open the Customers form in [Design view](#).
2. View the Property sheet by choosing Properties from the View menu.
3. Add the following statement to the AfterUpdate property of the First Name field:

```
Object: Text Box
-----
ControlName: First Name
AfterUpdate: =ProperAU([First Name])
```

Now, whenever the employees name is typed into the Employee form, it will be converted to the correct format when the user presses TAB or ENTER.

Calling a Function from an Expression

The Proper() function can be written so it can be called from an [expression](#), or [calculated control](#). In this example we will call it ProperCC() as a reminder that it should be used in calculations.

Enter the following function in a module:

```
Function ProperCC(Field)
    ProperCC=UCase(Left(Field,1)) & LCase(Mid(Field,2))
End Function
```

NOTE: The result of the calculation is assigned to the function. This way, it can be used in an expression or calculated control.

Example

1. Open the Customers form in Design view.
2. Add the following calculated control to the form:

```
Object: Text Box
-----
ControlName: Proper Last Name
ControlSource: =ProperCC([Last Name])
```

Now when you type in the Last Name field, you will see the correct capitalization in the Proper Last Name field.

NOTE: You won't be able to type in the Proper Last Name field. ProperCC() doesn't change underlying data like ProperAU() does. For this reason, ProperCC() is useful in reports and expressions and can be used more places than ProperAU().

You can use ProperCC in the same manner as any of the built-in functions listed in the "Language Reference," such as UCase, LCase,

and so on.

Determining the Type of Function You Need

=====

Where used	Function style
AfterUpdate, BeforeUpdate, and so on	Event
RunCode <u>macro</u> action	Event
Calculated controls on forms and reports	Expression
Calculated fields in a <u>query</u>	Expression
SetValue macro action expression	Expression
Default values in a <u>table</u> or form	Expression
Called from another function or sub	Expression

[References](#)

PRB: 'Table Exclusively Locked' Error Switching to Form View

Article Number: Q97513
CREATED: 13-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

An attempt to switch from Design view to Form view fails and Microsoft Access generates the following message:

Table <tablename> is exclusively locked

CAUSE

The form's RecordSource property refers to a table that remains open in Design view.

RESOLUTION

Perform one of the following three procedures to address this behavior:

- Close the open table
- Change the open table to Datasheet view
- Open the form before opening the table (this opens the table in Design view for read-only access)

References

PRB: ValidationRule Parses Function Name w/o Parens to String

Article Number: Q97512
CREATED: 13-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you type an expression containing a function name that has no parentheses or parameters in the ValidationRule property of a text box, the expression is evaluated as a string instead of a function.

RESOLUTION

Put parentheses or parameters after all function names so they can be correctly evaluated as functions instead of text strings. For example, use

```
=Date()
```

instead of:

```
=Date
```

NOTE: For this example to work correctly, after putting parentheses after =Date, you also need to change the Format property of the text box to General Date.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Start Microsoft Access and open the NWIND.MDB sample database.
2. Create a new, unbound blank form.
3. Place a new, unbound text box on the blank form.
4. From the View menu, choose Properties so that you can change the properties of the form and objects on the form.
5. In the ValidationRule property of the text box, type the following and then press the ENTER key:

```
=Date
```

Note that Microsoft Access automatically changes the expression you entered from =Date to ="Date".

6. From the View menu, choose Form.
7. Type a date in the text box.

When you type a date such as 04/05/93 in the text box and then press the ENTER key, you get the message, "The value you entered is prohibited by the validation rule set for this field." However, the text box accepts the word "Date," which is the word changed by Microsoft Access in the ValidationRule property of the text box.

[References](#)

PRB: Unable to Add All Fields in Graph FormWizard

Article Number: Q97511
CREATED: 13-APR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When creating a [graph](#), you will receive the following error message if you try to include more than one date [field](#) in the graph:

You can select only one date field.

You will receive the following error message if you try to include more than three fields in the graph and more than one of those fields is a text field

You can only select one non-numeric field if you want to use <number> fields for your chart.

where <number> is the number of fields that you selected.

STATUS

This behavior is by design.

MORE INFORMATION

When you are creating a chart with the Graph FormWizard, the Add All (>>) button, even though it is displayed as a choice, will return an error message if your source [table](#) or [query](#) contains more than one date field. You will also receive an error message when using the Add All (>>) button if your source table or query contains more than one text field for a chart that has more than three fields.

Steps to Reproduce Behavior

-
1. Open the NWIND.MDB sample [database](#).
 2. Use FormWizards to create a new [form](#) based on the Orders table.
 3. When the FormWizard asks "Which AccessWizard do you want?" select Graph. Choose the Next button.
 4. When the FormWizard asks "Which type of Graph do you want?" choose the Next button.
 5. When the FormWizard asks "Which fields contain the data you want to graph and use for labels?" select all the fields. To select all the fields, choose the Add All (>>) button.

When you choose the Next button, you will receive an error message indicating that only one date field can be in the graph.

To receive the other error message, repeat steps 2-4, substituting the Customers table for the Orders table.

[References](#)

PRB: Only One Button Accepts Selection in Yes/No Option Group

Article Number: Q97510
CREATED: 13-APR-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When a database contains a form that has an option group bound to a Yes/No field and the option group contains two option buttons, one of which is assigned the option value 0 and the other the option value 1, only the second button can be selected.

CAUSE

In Microsoft Access, the value (0) represents FALSE and any other value represents TRUE. In a field of data type yes/no, TRUE values are assigned the value (-1). If you select an option button that has the value (1), the yes/no field to which the option group is bound is changed to (-1), or TRUE.

Actually, the option button attempts to put the value (1) into the yes/no field. But since the field is of data type yes/no, the field is set to (-1) for any value other than (0). The value of the option button is set to (1) and the yes/no field now contains a (-1), therefore the button does not appear to be selected.

RESOLUTION

Change the option value for the second button from 1 to -1.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. In a form, create an option group for a Yes/No field.
2. Add two radio buttons to the group.
3. Set the option value for Button1 to 0.
4. Set the option value for Button2 to 1.
5. Choose Form view. An attempt to select Button2 fails.

[References](#)

PRACC9303: Bad Results Searching Multicolumn Paradox Indexes

Article Number: Q97006
CREATED: 04-APR-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

Using the Find command on the Edit menu to search Paradox fields for specific values does not work properly if all of the following are true:

- The field being searched is part of a multicolumn index.
- The field contains NULLs.
- The Search Fields As Formatted option is disabled in the Find In Field dialog box on the Edit menu.

RESOLUTION

To work around this problem, check the Search Fields As Formatted option in the Find In Field dialog box so that the index is not used in the search. Similarly, when querying on Paradox fields that are part of a multicolumn index containing NULL values, use the LIKE operator in places where you would normally use the equivalence operator. For example, avoid using

```
SELECT *  
FROM TestTable  
WHERE ((FieldX=23));
```

and instead use the following:

```
SELECT *  
FROM TestTable  
WHERE ((FieldX Like 23));
```

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. We are researching this problem and will post new information here in the Microsoft Knowledge Based as it becomes available.

[References](#)

PRACC9303: Descending Sort Query Returns Wrong Number of Rows

Article Number: Q97005
CREATED: 04-APR-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

When sorting data in descending order on an indexed [column](#) containing over 506 duplicate entries, Microsoft Access may return the wrong number of rows to the user.

CAUSE

This occurs when the Index property for a column is set to "Yes (No Duplicates)" or "Yes (Duplicates OK)," but not when there is a [primary key](#) on the column. Nonprimary-key indexes can have multiple null entries.

RESOLUTION

Remove the [index](#) from the column being sorted. This causes the [query](#) to execute more slowly but gives you correct results.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

PRACC9303: GPF on Update Query Due to Query Design Option

Article Number: Q97004
CREATED: 04-APR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

Update queries do not work correctly if you use the Options command on the View menu to set the Query Design option, Restrict Available Fields, to No (the default is Yes). In most cases, Microsoft Access causes a general protection (GP) fault; however, there have been isolated cases where the [query](#) completed and incorrectly updated the wrong columns in the [table](#).

RESOLUTION

Set Restrict Available Fields to No on a query-by-query basis by clearing the Restrict Available Fields [check box](#) on the Query Properties [dialog box](#) instead of using the Options/View menu [method](#) to set this option on a global basis.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

PRACC9303: Can't Add Counter Columns to Tables w/ > 4 MB Data

Article Number: Q97003
CREATED: 04-APR-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You cannot add Counter data type columns to existing tables containing roughly greater than 4 megabytes (MB) of data.

CAUSE

This problem occurs because of a current limitation in the Microsoft Access transaction model (transaction size limited to 4 MB).

The problem occurs with Counter columns because the table data for columns is updated when the new column is added. The update takes place in a transaction, and with large tables (greater than 4 MB), the update encounters the Microsoft Access transaction size limit. When this happens, the transaction rolls back and the Counter column is not successfully added.

RESOLUTION

Use the following steps to add a Counter column to a large table:

1. Copy/paste the table's structure (not the table's data) to a new table.
2. Add a Counter data type column to the new table.
3. Create an append query that transfers data from the old table into the new table.
4. Verify that the new table has the correct data.
5. Delete the old table.
6. Rename the new table to the name of the old table.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

[References](#)

PRACC9303: GPF Referencing Form Dynaset with Form Variables

Article Number: Q97002
CREATED: 04-APR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

Form [dynaset](#) references do not always work properly when using [form](#) variables. If you reference a form dynaset with variables, you may receive a general protection (GP) fault.

RESOLUTION

Always refer to a form dynaset with explicit references through the form. The first code fragment below shows how to reference a form dynaset safely, and the second code fragment shows what most likely will cause the problem:

Safe

```
Forms!TestForm.Dynaset!Coll="MyVar"
```

May Cause Problems

```
Dim MyForm As Form  
Set MyForm = Form!TestForm  
MyForm.Dynaset!Coll = "MyVar"
```

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

PRB: 'Fieldname' Is not an Index in This Table

Article Number: Q97001
CREATED: 04-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you run an [Access Basic function](#) that attempts to perform a Seek [method](#) on a [table](#), you receive the following error message

```
'Fieldname' isn't an index in this table.
```

where 'Fieldname' is the name of a [field](#) in your table that has been defined as an index in your function. This field is also the primary key in the table.

CAUSE

When you set the Index property in your function and you are referring to a [primary key](#) field, you should refer to the field as PrimaryKey; that is, don't refer to the field by the name of the field.

RESOLUTION

Change the statement in your function that sets the current Index property. The correct [syntax](#) should resemble the following (note that the "Microsoft Access Introduction to Programming Guide incorrectly refers to PrimaryKey as one word):

```
EMPTable.Index = "PrimaryKey"
```

MORE INFORMATION

The following example is based on the sample [database](#) NWIND.MDB:

Steps to Reproduce Behavior

-
1. Create a new [module](#) and enter the following function:

```
'*****  
'Declarations section of the module.  
'*****  
Option Explicit  
  
'=====
```

```
' This FINDID function accepts one parameter.  
'=====
```

```

Function FINDID (EMPLOYID As String)
    Dim db As Database, EMPTable As Table
    Set db = CurrentDB()
    Set EMPTable = db.OpenTable("Employees")
    EMPTable.Index = "EMPLOYEE ID"
    EMPTable.Seek "=", EMPLOYID
    If EMPTable.NoMatch Then
        MsgBox "Not a valid ID. Try another"
    Else
        MsgBox "This Employee ID is in the table!"
    End If
    EMPTable.Close
End Function

```

2. From the View menu, choose the Immediate window.
3. Enter the following statement in the Immediate window:

```
? FINDID("1")
```

You will see the following error message:

```
'Employee ID' isn't an index in this table.
```

NOTE: The Employee ID field has been defined as the primary key in the Employees table.

4. Change the EMPTable.Index statement to the following:

```
EMPTable.Index = "PrimaryKey"
```

5. Repeat step 3.

You will see the appropriate message box indicating whether you entered a valid number.

References:

"Microsoft Access Introduction to Programming," version 1.0, page 114

[References](#)

PRB: Incorrect Results Received with DSUM Function

Article Number: Q97000
CREATED: 04-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Incorrect results received from DSUM function.

CAUSE

The DSUM() function returns a set of values in a specified set of records (domain). Typically you might use this function in a text box control on a report to total the values from a field in a table or query.

If you omit the surrounding quotation marks from the expression argument, you will not see the expected error message. Instead, you see a value that is actually derived by taking the last value of the field from the report and summing it x number of times (once for each record in the table or query).

RESOLUTION

Surround each argument in the DSUM function with quotation marks.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

The following example is based on the sample database Northwind Traders (NWIND.MDB):

1. Open the report Freight Charges.
2. Add a text box to the Company Name Footer section:

Object: Text Box

ControlName: MyDSUM

ControlSource: =DSum([Freight],"Freight Charges")

3. Preview the report. The amount showing in this text box is incorrect (compare it to the text box control in the report footer section named Grand Total). Notice that the optional criteria

expression is not included in this example. Therefore, you would expect to see the total of the Freight field from the Freight Charges query.

4. Change the ControlSource to:

```
=DSum("[Freight]","Freight Charges")
```

6. Preview the report. Now the amount in the text box is correct.

References:

"Microsoft Access Language Reference," version 1.0, pages 159-160

[References](#)

PRB: File Rename Dialog Still Enabled When No Filename Given

Article Number: Q96999
CREATED: 04-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you have the Database window open, if you choose Rename from the File menu and do not enter a name for the selected object, the OK button is still enabled. If you choose the OK button, you will receive the following error message:

Not a valid document name:

CAUSE

Dialog boxes that are provided with Windows disable the OK button if an expected entry is not made. This particular dialog box is not provided by Windows. Microsoft Access does not disable the OK button in this dialog box when there is no entry in the field.

RESOLUTION

You must enter a new name for the object.

STATUS

This behavior is by design.

[References](#)

PRB: Query Window Does not Always Scroll to Record

Article Number: Q96998
CREATED: 04-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you click on a [query's record selector field](#) and press ENTER, the screen does not always scroll automatically to display that record.

RESOLUTION

In order to get the window to scroll to a record, the selector field must first be altered.

STATUS

This behavior is by design.

MORE INFORMATION

If the record selector shows record 1, clicking on the record number and pressing ENTER sends the cursor to that record but doesn't scroll the screen to display record 1.

Steps to Reproduce Behavior

Follow these steps in the sample [database](#) NWIND.MDB:

1. Run the query named Customer List.
2. Using the [vertical scroll bar](#), scroll down the query. Notice that the record selector does not change from record 1.
3. Click the mouse just to the right of the record number 1 and press ENTER. The cursor disappears but the query window does not scroll to record 1.
4. Repeat steps 1-3, but this time erase the record number and retype it (1). Press ENTER.

Notice that this time the window scrolls to record 1. This is the normal behavior with any record number you enter.

[References](#)

PRB: Toolbar Disappears When Printing a Document

Article Number: Q96997
CREATED: 04-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you print a [form](#) or [report](#) from Print Preview, the [toolbar](#) disappears when the Print [dialog box](#) is displayed. The toolbar reappears after the print job has been sent.

STATUS

This behavior is by design. This behavior does not compromise any functionality of the toolbar because the toolbar is returned. This behavior may not be noticed unless you are processing a large print job.

MORE INFORMATION

Steps to Reproduce Behavior

1. Choose Print Preview for any report.
2. From the File menu, choose Print.

Notice that the Print dialog box appears and that the toolbar disappears. If you print the report, the toolbar returns when the print job has completed processing.

[References](#)

INF: How to Change the Font for the Zoom Box

Article Number: Q96996
CREATED: 04-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to modify the system UTILITY.MDA [database](#) to change the default font used in the Zoom box of Microsoft Access.

MORE INFORMATION

To change the font for the Zoom box, do the following:

1. Before you open Microsoft Access, copy the UTILITY.MDA file to MYUTIL.MDA. The UTILITY.MDA file is generally stored in the root directory for your Microsoft Access program files.
2. Start Microsoft Access. From the File menu, choose Open Database, select MYUTIL.MDA, and choose the OK button.

NOTE: When you open this database, you are prompted to choose the OK button in response to two error messages. These errors are caused by duplicate modules loaded from the UTILITY.MDA database. The error messages are:

Tried to load [module](#) with duplicate procedure definition: BuilderZoom

-and-

Tried to load module with duplicate procedure definition: DDB

3. In [Design view](#), open the ZoomForm [form](#). Select the unbound [text box control](#) Zoom, and change the following properties to the ones noted below or to a font of your choice.

FontName: Microsoft Sans Serif
FontSize: 12
FontWeight: Semi-bold

NOTE: Select a font and a size that suit your needs.

4. Close the database and save your changes to the ZoomForm form. Exit Microsoft Access.
5. Use any ASCII text editor, such as NOTEPAD.EXE, to modify the MSACCESS.INI file in your Windows directory.

Change the following line in your [Options] section from

```
[Options]  
UtilityDB=C:\ACCESS\UTILITY.MDA
```

-to-

```
[Options]  
;UtilityDB=C:\ACCESS\UTILITY.MDA
```

then add the new line so that the section reads:

```
[Options]  
;UtilityDB=C:\ACCESS\UTILITY.MDA  
UtilityDB=C:\ACCESS\MYUTIL.MDA
```

Save the MSACCESS.INI file.

When you next start Microsoft Access and use the Zoom box for editing, the font, size, and weight reflect the changes you made.

[References](#)

PRB: Stacker 3.0 CHECK.EXE Reports Database Files as Corrupt

Article Number: Q96995
CREATED: 04-APR-1993
MODIFIED: 06-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Running the Stacker version 3.0 CHECK.EXE program with the /=D and /F switches often results in erroneous reports that Microsoft Access .MDB files are corrupt and should be deleted. The /=D and /F switches are meant only for development and [debugging](#) purposes and can incorrectly identify files as corrupt.

RESOLUTION

Do not run the CHECK.EXE program with the /=D switch. If you do run CHECK.EXE with the /=D switch, choose No when you are prompted to delete "corrupt" files.

MORE INFORMATION

STAC Electronics has created a document explaining the "lost sector groups" reported by CHECK /=D. This file is called TEC042.DOC on CompuServe and the STAC BBS, and document #3003 on STAC FAX, STAC Electronics's automated fax response service.

Stacker is manufactured by STAC Electronics, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

PRACC9303: File Truncation on Password-Protected Servers

Article Number: Q96994
CREATED: 04-APR-1993
MODIFIED: 06-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Using an incorrect Microsoft Windows for Workgroups driver with Microsoft Access may cause file truncation. If you have an early version of the Windows for Workgroups WFWNET.DRV file (dated before 11/02/92), choosing the Network button in Microsoft Access dialog boxes to connect to password-protected servers may cause [database](#) files opened on those servers to become truncated.

NOTE: This problem was discovered before Microsoft Access was released. All copies of Microsoft Access include the updated Windows for Workgroups driver and installation instructions.

RESOLUTION

Update the WFWNET.DRV driver file, using the driver [update](#) program included with Microsoft Access. Complete instructions are contained in the flier included in your copy of Microsoft Access.

[References](#)

PRACC9303: GPF Using Invalid IIf() as ControlSource

Article Number: Q96993
CREATED: 04-APR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

If you use the IIf() [function](#) as the ControlSource property of a [form](#) or [report](#) and do not provide the necessary parentheses around the parameter list, Microsoft Access may cause a general protection (GP) fault instead of returning an error message.

For example, the following [syntax](#) is incorrect and will cause a GP fault:

```
=IIfIsNull([Fld]),"NULL","NOT NULL"
```

Note that the parentheses for the IIf() function are missing. As soon as you try to leave the property setting, the GP fault will occur.

RESOLUTION

Ensure that all IIf() functions include the required parentheses around the parameter list. The following syntax is a corrected version of the one displayed above:

```
=IIf(IsNull([Fld]),"NULL","NOT NULL")
```

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

PRACC9303: Setup Leaves MS-SETUP.T Directory and Files on Disk

Article Number: Q96992
CREATED: 04-APR-1993
MODIFIED: 08-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

If you request Microsoft Access SETUP.EXE to install SHARE.EXE and then reboot your machine when installation is complete, Setup does not properly erase the temporary directory MS-SETUP.T (and the files it contains) on your hard disk. This leftover directory contains data files totaling approximately 600K. These files occupy disk space, but otherwise have no affect on Microsoft Access or any other application.

RESOLUTION

Either do not choose Reboot Your Machine at the end of the installation process, or manually delete the temporary directory MS-SETUP.T and its files after you reboot.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem has been corrected in Microsoft Access version 1.1.

[References](#)

PRACC9303: GPF When Using ListParameters Method on Database

Article Number: Q96991
CREATED: 04-APR-1993
MODIFIED: 04-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

Illegally using the ListParameters [method](#) on a [database](#) object does not generate a compile-time error as expected, but rather a general protection (GP) fault.

RESOLUTION

The ListParameters method is intended only for use with the [QueryDef](#) object.

Use only methods legal for use on the database object:

Close	CreateDynaSet
CreateQueryDef	CreateSnapshot
DeleteQueryDef	ListTables
OpenQueryDef	OpenTable

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

In Microsoft Access version 1.1, you will receive the following descriptive compile-time error instead of a GP Fault:

Method not applicable for this object

Attempting to list parameters for a database object is illegal, but instead of generating the expected compile-time error, Microsoft Access version 1.0 generates a GP fault. For example, the following subroutine causes a GP fault in Microsoft Access version 1.0 instead of a compile-time error:

```
Sub Test
    Dim db As Database
    Dim Snap As Snapshot

    Set db = CurrentDB()
    Set Snap = db.ListParameters()
End Sub
```

[References](#)

PRACC9303: GPF Hiding Control in Form Footer in Datasheet View

Article Number: Q96990
CREATED: 04-APR-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

If you have a control in the footer of a form and you then hide that control in an event, such as OnCurrent, and then view the form in Datasheet view, Microsoft Access may cause a general protection (GP) fault.

CAUSE

In Datasheet view, Microsoft Access incorrectly assumes that a grid column is associated with the field being hidden. However, that is not true in this case, since the control is in the form's footer.

RESOLUTION

Since forms with controls in the footer are normally meant to be used in Form view, set the Form's ViewsAllowed property to "Form" so that the form cannot be switched to Datasheet view.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

INF: How to Create a "Please Wait" Message

Article Number: Q96989
CREATED: 04-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Sometimes you may want to display a status message to inform a user he or she must wait for an action to be completed. For example, you may want to display the text "Please Wait" while Microsoft Access is initializing a [database](#), running a long [query](#), and so on. You can do this by using a [form](#) and a [macro](#). This article presents a simple example using the sample database (NWIND.MDB).

More Information:

To display a status message in Microsoft Access, do the following:

1. Create an unbound blank form named PLEASEWAIT. Add a [label](#) with the message you want to display. For example, type "Please Wait" (without the quotation marks) as the caption. The [form properties](#) should be set to the following:

```
Form: PLEASEWAIT  
-----
```

```
ScrollBars: Neither  
Popup: Yes  
Modal: Yes  
RecordSelectors: No
```

2. Create the following macro named Test1:

Macro Name	Action

Test1	OpenForm RepaintObject OpenQuery Close

```
Test1 Actions  
-----
```

```
OpenForm  
Form Name: PLEASEWAIT  
View: Form  
Date Mode: Read Only  
Window Mode: Normal
```

RepaintObject
Object Type: Form
Object Name: PLEASEWAIT

OpenQuery
Query Name: <Daily Orders Total>
View: Datasheet

Close
Object Type: Form
Object Name: PLEASEWAIT

Note: The above argument for OpenQuery is just an example of what you may want to run; this can be anything. The main macro actions in Test1 are OpenForm, RepaintObject, and Close.

3. Run the macro. The PLEASEWAIT form should open. As soon as the query completes processing, the form disappears and the query results appear.

[References](#)

PRB: Not Enough Stack Memory When Opening Form

Article Number: Q96988
CREATED: 04-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you open a [form](#), it flashes repeatedly, and then the following message is displayed:

Not Enough Stack Memory

CAUSE

One possible cause is that a [macro](#) with a Requery action is assigned to the first [control](#) in the Tab Order on the form. Or, the macro is assigned to the OnCurrent property of the form. This creates an infinite loop, causing the form to requery until the stack memory is depleted.

RESOLUTION

If the macro must be assigned to the first control, assign it to the BeforeUpdate property of the control.

MORE INFORMATION

NOTE: In the sample [database](#) NWIND.MDB, use the steps below to re-create this behavior.

Steps to Reproduce Behavior

1. Create a new macro with a single Requery action. You do not need to fill in any arguments for this macro action. Save the macro as MyRequery.
2. Open the Categories form in [Design view](#).
3. Assign the macro MyRequery to the OnEnter property of the first control on the Categories form. This should be the [text box](#) control named Category ID.
4. Open the form in [Form view](#).

The form flashes several times. After several seconds, you receive the following error message:

Not Enough Stack Memory

References

PRB: Combo Box Displays Only 255 Characters of Memo Field

Article Number: Q96987
CREATED: 04-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When looking at the contents of a Memo field in a combo box, the entire contents of the Memo field are not visible. Instead, only the first 255 characters of the memo field are visible.

RESOLUTION

If you need to view the entire contents of a Memo field, use a text box instead of a combo box.

STATUS

This behavior is by design.

MORE INFORMATION

Most data types are limited to 255 characters. Memo data type fields are limited to 32,000 characters instead of 255 characters.

A combo box can display only 255 characters of data, regardless of the underlying field's data type.

[References](#)

PRB: NWIND Order Form Updates Order Amount Field with Zero

Article Number: Q96986
CREATED: 04-APR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

On the Orders form in the sample database NWIND.MDB, if you change the Quantity field for a product in the Orders subform, a macro assigned to the BeforeUpdate property of the Orders form resets the value of the Order Amount field in the Orders table. This macro incorrectly updates this field to a zero value.

CAUSE

The macro Orders.Write Order Amount sets the value of the Order Amount field in the table before the calculated field Orders Subtotal has calculated the new value on the subform.

RESOLUTION

Create a function that will cause a delay in the macro assigned to the AfterUpdate property. The Orders.Write Order macro will need to be modified. The first macro action in this macro will be a RunCode action that calls this new function.

MORE INFORMATION

In the sample database NWIND.MDB, you will create a new function that will cause a delay so the system can properly update the Order Amount field. This function will be added to the macro group called Orders in the macro named Write Order Amount.

1. Modify this macro:

Macro Name	Action
Orders.Write Order Amount	RunCode SetValue

Write Order Amount Actions

Runcode

Function Name: OrderSet()

SetValue

Item: Forms![Orders]![Order Amount]

Expression: Forms![Orders]![Orders Subform].Form![Order Subtotal]

2. Create the following function named OrderSet:

```
'=====
'The following Function OrderSet() runs a loop with the DoEvents
'command inside to allow the calculated field to update before the
'Setvalue runs. The number of times the function needs to loop may
'vary based on the speed of the machine.
'=====
```

```
Function OrderSet()
    For x = 1 to 100
        Doevents
    Next x
End Function
```

Steps to Reproduce Behavior

These steps demonstrate the problem that occurs in the Orders form.

1. Open the NWIND database.
2. Open the Orders form in [Form view](#).
3. Change the Quantity field for one of the products listed in the Orders subform (note the order ID of the order you are changing).
4. Open the Orders table and find the [record](#) with the same order ID.
5. Check the Order Amount field.

[References](#)

PRB: All Data 'Lost' When Doing Save As from Design View

Article Number: Q96985
CREATED: 04-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In Design view, you choose Save As from the File menu to create another table with the same structure. You name the new table and switch to Datasheet view. The new table contains no records.

CAUSE

Save As is designed to copy only the table structure, not the data.

RESOLUTION

To copy both the table structure and the data, do the following:

1. In the Database window, choose the Table button.
2. Select the table you want to copy.
3. From the Edit menu, choose Copy (or press CTRL+C).
4. Choose a destination for your new table. To paste the table into an existing database, choose Open Database from the File menu. To paste the table into a new database, choose New from the File menu.
5. From the Edit menu, choose Paste (or press CTRL+V). The Paste Table As dialog box is displayed.
6. Choose Structure And Data.
7. Type a name for the new table and choose the OK button.

References:

"Microsoft Access User's Guide," version 1.0, pages 32-33

[References](#)

PRB: Reordering Rows in Table or Macro Design Window

Article Number: Q96984
CREATED: 04-APR-1993
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you move the location of a [row](#) in a Table or Macro design window, Microsoft Access appears to insert the row at the wrong location.

CAUSE

This behavior is by design. While you are moving the row, Microsoft Access will display an insertion line immediately above the row selected by your mouse pointer. The row is always inserted immediately beneath the insertion line.

MORE INFORMATION

The following steps help demonstrate this concept, using a Macro design window.

Steps to Reproduce Behavior

1. Create a new [macro](#).
2. Create three macro actions: AddMenu, ApplyFilter, and Beep (in that order).
3. Click the [row selector](#) to select the first macro action, AddMenu.
4. Drag this action down until your mouse pointer is over the row selector for the Beep action. (Note that the insertion line is above the Beep action.)

When you drop the AddMenu action at this point, Microsoft Access will insert AddMenu immediately above Beep.

[References](#)

INF: Validating Data with Values From Another Table

Article Number: Q96973
CREATED: 01-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The text below presents two examples that demonstrate setting the ValidationRule property to use values in a different [table](#). You must use either the DCount() or DLookup() [function](#) to do this.

You can use a [validation expression](#) to compare the value entered by the user with a value in one or more other tables. You can create a [validation rule](#) to accept only values that exist in a second table or to reject values that exist in a second table.

MORE INFORMATION

Example 1: Using the DCount() Function

In the following example, the user can enter into "Table1" only those values that exist in the "Field" [field](#) of "Table2."

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
Table: Table1
-----
Field Name: Field Name
DataType: Text
Validation Rule: DCount("[Field]","Table2","[Field]='" &_
                    [Field in Table1] & "'") > 0
```

In the following example, the user can enter into "Table1" only those values that do not exist in the "Field" field of "Table2."

```
Table: Table1
-----
Field Name: Field Name
DataType: Text
Validation Rule: DCount("[Field]","Table2","[Field]='" &_
                    [Field in Table1] & "'") = 0
```

Example 2: Using the DLookup() Function

In the following example, the user can enter into "Table1" only those values that exist in the "Field" field of "Table2."

Table: Table1

Field Name: Field Name

DataType: Text

Validation Rule: DLookup("[Field]","Table2","[Field]='' &_
[Field in Table1] & ''")Is Not Null

In the following example, the user can enter into "Table1" only those values that do not exist in the "Field" field of "Table2."

Table: Table1

Field Name: Field Name

DataType: Text

Validation Rule: DLookup("[Field]","Table2","[Field]='' &_
[Field in Table1] & ''")Is Null

Replace "Field" with the appropriate field name in your database (be sure to preserve the quotation marks and square brackets). Replace "Table2" with the name of the validation table in your database and replace "Table1" with the name of the table in which you are entering data.

DCount() searches through the specified field in the validation table and counts all records where the field name is equal to the value entered in the table. If the function returns a value greater than zero, the function found one or more matching records.

References:

"Microsoft Access User's Guide," version 1.0, pages 45, 652.

[References](#)

PRACC9303: Access Hangs Importing Big Lotus 1-2-3 Spreadsheets

Article Number: Q96972
CREATED: 01-APR-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

The system stops responding ("hangs") when attempting to import a large spreadsheet (more than 64,000 cells) into Microsoft Access.

CAUSE

The Lotus 1-2-3 spreadsheet import option in Microsoft Access is limited to 64,000 formatted cells. By default, Lotus 1-2-3 formats all cells with the general format if they are not explicitly set to some other format by the user. As a result, if a Lotus 1-2-3 spreadsheet to be imported has more than 64,000 cells, then Microsoft Access cannot import it and the system may hang during the import process.

RESOLUTION

To import a Lotus 1-2-3 spreadsheet with more than 64,000 cells, you must first split the spreadsheet into separate spreadsheets so that the number of cells in each does not exceed 64,000.

The spreadsheet must be split--importing subranges from large spreadsheets does not eliminate the problem.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

INF: How Access Variable-Length Fields Export to dBASE

Article Number: Q96971
CREATED: 01-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access uses variable-length fields for tables. However, when you export a table to dBASE, which uses fixed-length fields, the field sizes in dBASE are set based on what the FieldSize property is set to in Microsoft Access. For example, if you have a table with a text field defined with a FieldSize property of 50, in the exported .DBF file, that field will have a field length of 50 in dBASE.

[References](#)

PRB: Weekday Disappears While Editing Long Date Format Field

Article Number: Q96970
CREATED: 01-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you edit a date field that is formatted as Long Date, the weekday disappears from the field. The weekday reappears when you exit the field.

CAUSE

A date is stored as a number that is parsed to display the format that you choose. All data types have certain formats that Microsoft Access can parse. The weekday is not parsed, and therefore it is removed from the date when you begin to edit the date field.

STATUS

This behavior is by design.

[References](#)

PRB: Zoom Box Accessible When Multiple Fields Are Selected

Article Number: Q96969
CREATED: 01-APR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You can open a Zoom box when multiple fields in a table are selected; however, the contents of only the first selected field are visible.

STATUS

This behavior is by design.

MORE INFORMATION

Although multiple fields in a table can be selected when opening a Zoom box, only one field is displayed in the Zoom box.

Steps to Reproduce Behavior

1. Create a table containing multiple fields and open it in Design mode.
2. From the Edit menu, choose Select All.
3. Press SHIFT+F2.

Notice that the data you see in the Zoom box comes from only the first selected item.

[References](#)

PRACC9210: Opening Objects in Transactions Can Cause Rollback

Article Number: Q96906
CREATED: 30-MAR-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

The following error message is displayed while attempting to open a second occurrence of a database object (such as a table, dynaset, form, and so forth) in a transaction:

Invalid database object

CAUSE

This error message results from a new occurrence of a database object overwriting a previous occurrence of a database object, which "orphans" the first object (that is, the first object's pointer to the database is destroyed).

When Microsoft Access detects that an object is orphaned, it attempts to close it. Because the object is in a transaction, however, the close fails, and therefore Microsoft Access rolls back all levels of nested transactions so that the close succeeds.

This implicit rollback causes all objects opened in transactions to be closed. As a result, any reference to an object closed by this process results in the "Invalid database object" error message.

RESOLUTION

This problem occurs only when objects are implicitly closed (by the process mentioned above), and does not occur when the object is explicitly closed in a transaction.

Thus, the keys to avoiding this problem are:

- Always open objects outside of transactions.
- Always explicitly close objects.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

```
Sub Sub1()  
  Dim db As Database, tb As Table  
  
  Set db = CurrentDB()  
  BeginTrans  
    Set tb = db.OpenTable("table1") 'Table opened in transaction  
    Call Sub2  
    Debug.Print tb.recordcount      'Result: "! Invalid database  
                                   'object."  
  
    tb.Close  
  CommitTrans  
  db.Close  
End Sub  
  
Sub Sub2()  
  Dim d as Database  
  Set d = CurrentDB()  
End Sub 'd loses scope, does implicit rollback and close of db.
```

[References](#)

PRB: Rounding Errors Importing/Exporting dBASE IV Values

Article Number: Q96905
CREATED: 30-MAR-1993
MODIFIED: 14-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Some floating-point values (single and double precision) have rounding errors when exported to or imported from a dBASE IV [database](#). For example, if the floating-point number .0807 is exported from a Microsoft Access database to a dBASE IV database and then imported back to Microsoft Access, after the import the value reads .08069.

CAUSE

This problem is caused by the floating-point number being converted from IEEE floating-point numeric format (used by Microsoft Access and several other Microsoft products) to ASCII character values required by dBASE IV. The Microsoft Access dBASE ISAM converts IEEE floating-point values to ASCII, and conversion of binary fractions to decimal is inherently inexact.

Typically, such rounding errors are not noticeable or relevant because they occur at digits far to the right of the decimal point and are not displayed. However, sometimes the rounding error may appear in the visible portion of the number.

STATUS

This behavior is by design.

[References](#)

PRB: Low Number of Share Locks Can Cause Error Message

Article Number: Q96904
CREATED: 30-MAR-1993
MODIFIED: 06-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

You receive the following error message

Can't save; record is locked by another user

in Microsoft Access, even when only one user (one session and machine) is using a database opened for shared access.

CAUSE

This error can be caused by an insufficient number of share locks.

RESOLUTION

Be sure that the number of share locks is set to 500 or more. The number of share locks is set by the /L:xxxx parameter when SHARE.EXE is invoked, usually in the AUTOEXEC.BAT file, as follows:

```
LOAD C:\DOS\SHARE.EXE /L:500
```

The Microsoft Access Setup program modifies your AUTOEXEC.BAT file and sets the number of share locks to 500, but some other program may alter that value.

Although 500 share locks are sufficient for most Microsoft Access applications, some applications may have problems related to the number of share locks. You may need to gradually increase the number of share locks until these problems disappear. Try to add as few share locks as possible, since each share lock uses memory.

[References](#)

PRACC9210: '#' Appears in Attached Table Name in Error Message

Article Number: Q96903
CREATED: 30-MAR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

In certain error messages, a number sign (#) instead of a period appears as a file extension [separator](#). For example, this message may appear:

```
Couldn't open file 'PDXSALES#DB'
```

instead of this message:

```
Couldn't open file 'PDXSALES.DB'
```

This problem occurs in errors that are generated while attempting to attach an [external table](#) that uses an [installable ISAM](#) driver (as seen in the [Installable ISAMs] section of the MSACCESS.INI file in the WINDOWS directory.)

CAUSE

Internally, an installable ISAM driver replaces the period in the [table](#) name with a number sign for [database](#) compatibility purposes.

The problem is caused when Microsoft Access uses this internal number sign-separated table filename in the error message, instead of the period-separated name.

This is a cosmetic problem only; it does not affect the operation of the installable ISAM driver or the attaching of external tables.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

[References](#)

PRACC9210: StDevP() with All NULL Records Returns #ERROR

Article Number: Q96902
CREATED: 30-MAR-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

When the StDevP() [function](#) is applied to a [database](#) containing only NULL records, #ERROR results. This problem does not occur with the StDev() function.

RESOLUTION

Make sure that at least one [record](#) of a database where the StDevP() function is used contains a non-NULL value.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

StDevP() and StDev() are functions that return estimates of the standard deviation for a population or a population sample, respectively.

Steps to Reproduce Behavior

1. Create and save the following [table](#):

```
Table: Table1
```

```
-----
```

```
FieldName: A
```

```
    DataType: Text
```

```
FieldName: B
```

```
    DataType: Number
```

```
    Default Value: (make sure to leave this property blank)
```

2. Add several records to the above table, entering text in [field](#) A and NOT entering any data in field B (leave field B NULL).
3. Create a blank, unbound [form](#).
4. Add a [text box](#) on this form, as follows:

```
Object: Text box
```

```
-----
```

```
ControlName: Field0
```

```
ControlSource: =DStDevP("B","Table1")
```

5. From the View menu, chose Form.

In Microsoft Access version 1.0 you will see #ERROR in this text box. In Microsoft Access version 1.1 this text box will correctly display no data (NULL).

[References](#)

INF: Limit to List Set to Yes Needs a Unique Entry

Article Number: Q96901
CREATED: 30-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

If you set the LimitToList property of a [combo box](#) to Yes, and type a portion of an entry into the combo box and press ENTER, you may receive the following error message:

The text you enter must match an entry in the list.

The error occurs because there are duplicates in the list. For example, if you enter the value "co" in the combo box and the values "conf" and "cond" exist in the drop-down list, the error occurs.

References:

"Microsoft Access User's Guide," version 1.0, chapter 9, "Designing Forms", pages 244-245

[References](#)

INF: How to Send Information to the Clipboard

Article Number: Q96900
CREATED: 30-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There is no Microsoft Access command to send information to the [Clipboard](#). To post information to the Clipboard, you need to define an [Access Basic function](#) that calls several Windows API functions. This article defines a function that copies text to the Clipboard.

For more information on how to retrieve information from the Clipboard, [query](#) on the following words in the Microsoft Knowledge Base:

Clipboard_GetData and GetClipboardData

This article assumes that you are familiar with Access Basic and with creating Access applications with the programming tools provided with Microsoft Access.

MORE INFORMATION

NOTE: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

NOTE: You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive the duplicate procedure name error, remove or comment the Declare statement from your code.

To call the function, you can make a [string](#) assignment to the returned value, such as:

```
X = ClipBoard_SetData("This string will go to the Clipboard!")

'=====
' General Declarations
'=====

Option Compare Database      'Use database order for string comparisons
Option Explicit

Declare Function kb_OpenClipboard% Lib "User" _
    Alias "OpenClipboard" (_
    ByVal hwnd%)
Declare Function kb_GlobalAlloc% Lib "Kernel" _
    Alias "GlobalAlloc" (ByVal wFlags%, ByVal wBytes&)
```

```

Declare Function kb_GlobalLock& Lib "Kernel" _
    Alias "GlobalLock" (ByVal hMem%)
Declare Function kb_lstrcpy& Lib "Kernel" _
    Alias "lstrcpy" (ByVal lpString1 As Any, _
    ByVal lpString2 As Any)
Declare Function kb_GlobalUnLock% Lib "Kernel" _
    Alias "GlobalUnLock" (ByVal hMem%)
Declare Function kb_CloseClipboard% Lib "User" Alias"CloseClipboard"()
Declare Function kb_EmptyClipboard% Lib "USER" _
    Alias "EmptyClipboard" ()
Declare Function kb_SetClipboardData% Lib "User" _
    Alias "SetClipboardData" (ByVal wFormat%, _
    ByVal hMem%)

```

```

Global Const GHND = &H42
Global Const CF_TEXT = 1
Global Const MAXSIZE = 4096

```

```

Function Clipboard_SetData (MyString$)
    Dim hGlobalMemory%, lpGlobalMemory&, hClipMemory%, X%

    '-----
    ' Allocate moveable global memory.
    '-----
    hGlobalMemory% = kb_GlobalAlloc(GHND, Len(MyString$) + 1)

    '-----
    ' Lock the block to get a far pointer
    ' to this memory.
    '-----
    lpGlobalMemory& = kb_GlobalLock(hGlobalMemory%)

    '-----
    ' Copy the string to this global memory.
    '-----
    lpGlobalMemory& = kb_lstrcpy(lpGlobalMemory&, MyString$)

    '-----
    ' Unlock the memory.
    '-----
    If kb_GlobalUnLock(hGlobalMemory%) <> 0 Then
        MsgBox "Could not unlock memory location. Copy aborted."
        GoTo OutOfHere2
    End If

    '-----
    ' Open the Clipboard to copy data to.
    '-----
    If kb_OpenClipboard(0&) = 0 Then
        MsgBox "Could not open the Clipboard. Copy aborted."
        Exit Function
    End If

    '-----
    ' Clear the Clipboard.
    '-----
    X% = kb_EmptyClipboard()

```

```
'-----  
' Copy the data to the Clipboard.  
'-----  
hClipMemory% = kb_SetClipboardData(CF_TEXT, hGlobalMemory%)
```

OutOfHere2:

```
  If kb_CloseClipboard() = 0 Then  
    MsgBox "Could not close Clipboard."  
  End If
```

End Function

References:

"Microsoft Windows 3.1 Programmer's Reference," Volume 2

[References](#)

INF: Importing Word Print Merge Data Files to Microsoft Access

Article Number: Q96899
CREATED: 30-MAR-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to import data from a Word for Windows print merge data file for use in Microsoft Access.

MORE INFORMATION

In order to use data from a Word for Windows print merge data file in Microsoft Access, the file must be saved as a comma or tab delimited unformatted text file.

Word for Windows can perform print merges on data contained in files that are comma or tab delimited, or on data formatted from a [table](#). If the data is already comma or tab delimited, you must take care to ensure the file is saved as an unformatted text file.

If the data is in table [form](#), the table must first be converted to a tab-delimited format before saving it as a text file.

NOTE: In Word for Windows, you can create multiline information within a table [cell](#) that contains carriage returns (paragraph marks). The carriage returns must be removed before converting the table to a tab-delimited format.

Converting Data from a Table into a Tab-Delimited Format

=====

Word for Windows 1.x

1. Open the data document in Word for Windows.
2. Position the insertion point anywhere inside the table of data.
3. Press ALT+Keypad 5 to select the entire table (make sure that NUM LOCK is turned off).
4. From the Insert menu, choose Table to Text.
5. In the Convert Table To box, select Tab Delimited, then choose OK.

Word for Windows 2.x

1. Open the data document in Word for Windows.

2. Position the insertion point anywhere inside the table of data.
3. From the Table menu, choose Select (or press ALT+Keypad 5).
4. From the Table menu, choose Convert Table to Text.
5. Choose the Tabs button from the Separate Text With group and choose OK.

Saving the Data File as an Unformatted Text Only File

Word for Windows 1.x

1. From the File menu, choose Save As, and enter an unique filename in the Save File Name box.
2. Choose the Options button.
3. From the File Format box, select Text Only, and choose OK.

Word for Windows 2.x

1. From the File menu, choose Save As, and type a unique filename in the File Name box.
2. In the Save File As Type box in the lower left corner, select Text Only, and choose OK.

Removing Unwanted Carriage Returns from a Table Formatted Data File

Word for Windows 1.X:

1. In the Search For box type "^p" (the caret symbol ^ is generated by pressing SHIFT+6)
2. In the Replace With box press the SPACEBAR 1 time.
3. Deselect the Confirm Changes checkbox and choose the OK button.

Word for Windows 2.x

1. From the Edit menu, choose Replace.
2. In the Find What box, type "^p" (the caret symbol ^ is generated by pressing SHIFT+6).
3. In the Replace With box, press the SPACEBAR once.
4. Choose the Replace All button.

Importing the Text Data File into Microsoft Access

1. From the File menu, choose Import.
2. Select Text (Delimited) from the Data Source list and choose OK.
3. Select the Word for Windows text file to import from the file list and choose the Import button.
4. Select "First Row Contains Field Names" if the first row in the Word data file contains a list of field names.
5. Choose the Options button.
6. Choose {tab} in the Field Separator box.
7. Choose the OK button to begin the import.

[References](#)

INF: Strange Stored Procedures Found in SYSOBJECTS Table

Article Number: Q96898
CREATED: 30-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Strange stored procedures are found in the SYSOBJECTS [table](#) in [SQL Server](#). Stored procedures with the general format [ODBC#xxxxnnnnn](#) are found when querying the SYSOBJECTS table. For example, if you issue the following [query](#) from SAF

```
SELECT * FROM SysObjects WHERE Name LIKE "ODBC%"
```

you may see stored procedures as follows:

```
ODBC#sa24518  
ODBC#sa2334  
ODBC#sa17854
```

This article describes these stored procedures and what you can do about them.

MORE INFORMATION

The stored procedures are created by ODBC for executing parameter queries. When Microsoft Access executes a query that is parameter, it calls the ODBC [function](#) SQLPrepare that creates these stored procedures that are compiled. These stored procedures pass the proper arguments to ensure proper execution.

The stored procedures are deleted under normal circumstances once the ODBC connection is closed. If a crash or abnormal termination of the ODBC connection occurs, it is possible that the stored procedures will remain on the server. In this case, the stored procedures can be removed without causing any problems.

To remove stored procedures manually, use SAF (SQL Administrator Facility) to drop each procedure. The command [syntax](#) is\

```
DROP PROCedure [owner.]procedure_name[, [owner.]procedure_name...]
```

where procedure_name refers to the procedure you are removing. For example,

```
DROP PROC ODBC#sa24518, ODBC#sa2334
```

would remove two stored procedures, ODBC#sa24518 and ODBC#sa2334.

[References](#)

PRB: Operation Stops when Editing Attached SQL Tables

Article Number: Q96897
CREATED: 30-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

Microsoft Access may return an inconsistent value for a floating point field in an SQL table when it checks the value of the field before editing and before updating. This can result in the error message

Data Has changed:Operation Stopped

when trying to update a record in an attached SQL table.

CAUSE

The problem occurs if an SQL field data type is set to Float and the table does not contain a TimeStamp field or when an SQL indexed field data type is set to either DateTime or Float.

RESOLUTION

To correct the problem:

1. Remove the index from any fields with the DateTime or Float data types.

To obtain information on any index within a SQL table, you can run the following system stored procedure within the System Administrator Facility (SAF):

```
sp_helpindex <Table Name>
```

To remove an index, use the following command:

```
DROP INDEX <Table_Name.Index_Name>
```

Note, you must be logged into the SQL server as either the Table Owner or the System Administrator.

2. If other fields in the table have the Float data type, you must insert a TimeStamp field in the table. If a field with this data type is present, Access returns a consistent value for fields of this data type and will use the value in this field to verify whether the record has been modified.

This can be accomplished by performing the following command within the SQL Server System Administrator Facility (SAF):

```
ALTER TABLE <Table Name> ADD TimeStamp timestamp
```

Please note that you must be logged in as either the [database](#) owner or the System Administrator to run the above command.

[References](#)

PRB: Raised and Sunken Check Box and Option Button Controls

Article Number: Q96896
CREATED: 30-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Raised check boxes and option buttons are indistinguishable from sunken check boxes and option buttons on both forms and reports.

STATUS

This behavior is by design.

MORE INFORMATION

You can specify three different appearances for controls on your forms and reports: Normal, Raised, and Sunken. Check box controls and option button controls allow you to specify these different appearances, but raised and sunken are displayed identically.

Steps to Reproduce Behavior

1. Create a blank unbound form.
2. In Design view add three check box controls to the form.
3. Add three option button controls to the form.
4. Select the Palette button and make one of the check boxes and one of the option buttons sunken. Leave one of each normal. Make one of each raised.
5. Switch to Form view. Compare the raised controls to the sunken controls.

Notice there is no difference between the controls.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 12, "Using Special Design Effects on Forms and Reports"

"Microsoft Access User's Guide," version 1.0, page 306

[References](#)

PRB: ORDER BYs Must Be Output Columns in DISTINCT Queries

Article Number: Q96895
CREATED: 30-MAR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you create a [query](#) and use the DISTINCT predicate in conjunction with the ORDER BY clause, the fields that are being ordered must be output columns. If the ORDER BY fields are not output columns, you will receive the following error message:

```
ORDER BY clause [Table Name].[Field Name] conflicts with DISTINCT
```

RESOLUTION

Use the DISTINCTROW predicate.

MORE INFORMATION

Steps to Reproduce Behavior

Follow these steps in the sample [database](#) NWIND.MDB:

1. Create a new query and add the Employees [table](#).
2. Place the Last Name and First Name fields in the query grid.

```
Query: MyQuery
```

```
-----  
Field Name: Last Name  
Sort: Ascending  
Show: False
```

```
Field Name: First Name  
Show: True
```

3. From the View menu, choose [SQL](#) command. Modify the SQL statement as follows:

```
SELECT DISTINCT  
    [Employees].[First Name]  
FROM [Employees]  
ORDER BY  
    [Employees].[Last Name];
```

4. Run the query.

The following error will be displayed:

ORDER BY clause (Employees.[Last Name]) conflicts with DISTINCT

To resolve this, you must Show the Last Name field.

[References](#)

PRB: Switching Modes Disables Undo

Article Number: Q96894
CREATED: 30-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you place a new control on a form and switch to Form view to see it, you cannot choose Undo Create from the Edit menu when you go back to Design view. Instead, Can't Undo is available, even though you have not made any changes since adding the control.

CAUSE

Switching to Form view is considered a change, thereby nullifying the Undo option.

STATUS

This behavior is by design.

MORE INFORMATION

If you create a form and add a control, you can choose the Undo Create command from the Edit menu to remove the control that you have just created. However, if you switch to Form view to see your modified form, and then switch back to design mode, the Undo Create option is no longer available. Immediately after you make a change, you can reverse it until you make another change. Switching to Form view is considered a change. Thus, you cannot undo the creation of the control, and the Edit menu list displays Can't Undo.

References:

For more information, search for "undo timeout," then "undoing changes" in the Microsoft Access Help menu.

[References](#)

PRACC9303: Setting OLE Object Property May Cause GP Fault

Article Number: Q96893
CREATED: 30-MAR-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

Microsoft Access will produce a general protection (GP) fault if the OLEDATA property is set to a null value or an invalid object.

CAUSE

OLEDATA is a property of an unbound embedded object used in either a [form](#) or [report](#). The property is not available in the Object Properties [dialog box](#); it is set by Microsoft Access and can only be modified using [Access Basic](#) code.

If the OLEDATA property is set to an invalid object, Microsoft Access will display the error "Not a valid [OLE object](#)." Microsoft Access will allow you to exit the dialog box and continue to work in design mode. The frame will remain in the form or report; however, since its [link](#) is broken, it will be dimmed. Access will produce a GP fault when leaving design mode to display the form in view mode, or the report in print [preview](#) mode.

If the OLEDATA property is set to null, Access will display the error "The setting you entered isn't valid for this property" and produce a GP fault upon exiting the dialog box.

RESOLUTION

Setting the embedded objects Visible property to False will suppress the display of the object.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

Any unbound embedded object inserted with the Paste, Paste Link*, or Paste Special* command is subject to the GP fault if the OLEDATA property is set to a invalid object.

Steps to Reproduce Problem

1. Using Paintbrush, open any .BMP file.

2. Select the object and copy it to the [Clipboard](#).
3. Exit Paintbrush.
4. Open Microsoft Access and create a new [database](#).
5. Create a new form.
6. In the form's [Design view](#), paste in the object.
7. Leave the form open and go to the [Database window](#).
8. Create a new [module](#).
9. Open the [Immediate window](#).

Option 1

10. Enter the following command:

```
Forms![Form1]![Embedded0].OLEDATA=NULL
```

Microsoft Access will display the following error:

```
The setting you entered isn't valid for this property.
```

11. Choose the OK button.

Microsoft Access will GP fault.

Option 2

12. Enter the following command:

```
Forms![Form1]![Embedded0].OLEDATA="X"
```

Microsoft Access will display the following error:

```
Not a valid OLE object
```

13. Choose the OK button.

14. Go back to Form1.

15. Switch to the form's View window.

Microsoft Access will GP fault.

[References](#)

INF: Page Locking, No Cluster Indexing Support

Article Number: Q96892
CREATED: 30-MAR-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Cluster indexing and row level locking are not possible with Microsoft Access version 1.x, unless you are using an attached table.

MORE INFORMATION

These features are not supported because of the variable-length architecture. When using an attached table, Microsoft Access uses the same locking scheme as does the native product. When you are using the installable ISAMs for dBASE, Paradox, and Btrieve, you will do record-level locking. The Paradox ISAM even understands the clustered indexes and takes advantage of these indexes in some queries.

The following products are manufactured by vendors independent of Microsoft: Btrieve by Novell, Inc., dBASE by Borland International, Inc., and Paradox by Ansa Software, a Borland company. We make no warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

PRB: TransferSpreadsheet Action Does not Add File Extension

Article Number: Q96891
CREATED: 30-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The TransferSpreadsheet action requires a file type and filename to export data to a spreadsheet. The filename tells Microsoft Access what to name the spreadsheet file that will hold the exported data. If you do not specify an extension in this filename, Microsoft Access will not add the appropriate extension for the file format to which you are exporting.

RESOLUTION

By not automatically writing a file extension, you are able to specify any extension you like (for example, .EXE, .DAT, and .INI).

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a Table called Table1 and add some fields and example data.
2. Create the following [macro](#):

Macro Name	Action
Transfer It	TransferSpreadsheet

Transfer It Actions

TransferSpreadsheet
Transfer Type: Export
Spreadsheet Type: Microsoft Excel
Table Name: Table1
File Name: TestIt
Has Field Names: No

3. Run the above macro.

A Microsoft Excel spreadsheet file will be created in the current directory and the file will be called TESTIT, not TESTIT.XLS.

References:

"Microsoft Access Language Reference," version 1.0, pages 476-479

[References](#)

PRB: System Defaults Not Showing in Imp/Exp Setup

Article Number: Q96890
CREATED: 30-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you choose the Imp/Exp Setup command from the File menu, the Leading Zeros in Dates and Four Digit Years check boxes do not take the defaults from the Microsoft Windows system settings found in Control Panel.

CAUSE

The Control Panel doesn't have a date setting that corresponds to the Leading Zeros setting used in Microsoft Access dates. The Windows setting has one for day and one for month, which do not have to be the same.

RESOLUTION

This behavior by design.

[References](#)

PRB: Page Hdr/Ftr Is Not Under View Menu

Article Number: Q96592
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Page 474 of the Microsoft Access "User's Guide" says to remove the page header and footer from a report by going to the View menu and choosing Page Hdr/Ftr. This option is not under the View menu, but rather the Layout menu.

RESOLUTION

The last two sentences of this paragraph read as follows:

Remove the page header and footer. (From the View menu, choose Page Hdr/Ftr.)

They should read as follows:

Remove the page header and footer. (From the Layout menu, choose Page Hdr/Ftr.)

References:

Microsoft Access "User's Guide," version 1.0, page 474

[References](#)

INF: Determine How Many Instances of Application Are Active

Article Number: Q96591
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can use Windows API calls in [Access Basic](#) to determine how many times, or instances, an application is running. You can use this to prevent re-entrance of an application that is already running, determine how many instances of an application are active, or start a loop in which you remain until the application quits.

This article assumes that you are familiar with Access Basic and how to create and use Access Basic procedures.

MORE INFORMATION

This article outlines three uses of two Windows APIs in Access Basic.

Open a new [module](#) or a previously created module and enter the following code:

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----  
'Global Declarations Section Of The Module.  
'-----  
Option Explicit  
  
Declare Function GetModuleHandle% Lib "Kernel" (ByVal lpModuleName$)  
Declare Function GetModuleUsage% Lib "Kernel" (ByVal hModule%)
```

1. Prevent re-entrance:

```
'-----  
' The following function will test for re-entrance and return  
' True if there is already an instance of the application active,  
' or it returns False. This function requires an argument that is  
' the name of the application.  
'-----  
Function TestForReentrance% (ApplicationName$)  
    If ApplicationName$ = "" Then Exit Function  
    If GetModuleUsage(GetModuleHandle(ApplicationName$)) Then _  
        TestForReentrance = True  
End Function
```

Example of Using this Code

You can use this function to keep from starting an application that is already running

```
Function RunMyApp$ (szAppToRun$)
  Dim x%
  If Not TestForReentrance (szAppToRun) Then
    x = Shell (szAppToRun)
    Exit Function
  End If
  RunMyApp = "AppToRun is presently active."
End Function
```

and a call to this in the [Immediate window](#) such as:

```
?RunMyApp$ ("Clock.exe")
```

2. Count the number of instances:

```
'-----  
' The following function will return the number of instances of an  
' application that are currently active.  
'-----
```

```
Function CountInstance% (szAppName$)
  CountInstance = GetModuleUsage (GetModuleHandle (szAppName))
End Function
```

3. Remain idle until an application quits:

```
'-----  
' The following function will attempt to start an MS-DOS application  
' and remain in a loop until that application quits.  
'-----
```

```
Sub RunDosAppUntil (szAppToRun$)
  Dim hMod%
  hMod% = Shell (szAppToRun, 1)
  If (hMod% > 32) Then
    While (GetModuleUsage (hMod%))
      DoEvents
    Wend
  Else
    MsgBox "Unable to start the Application"
  End If
End Sub
```

[References](#)

PRB: 'Couldn't Find Object [Object Name]'

Article Number: Q96589
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you get the following error message

 Couldn't Find Object [object name]

RESOLUTION

Use the following to troubleshoot the problem:

- First check to ensure that [object name] exists. It could be a [table](#), a [query](#), or an [SQL](#) statement. If it does not exist, you must remove all references to it from your application. Often, the RecordSource property of a [form](#) or [report](#) will contain a reference to an object whose name has changed or has been deleted.
- If [object name] exists, then it is likely that there is a [syntax](#) or spelling error in the reference to it.

[References](#)

PRB: Macro Condition Example Using COUNT() Is Incorrect

Article Number: Q96588
CREATED: 22-MAR-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

Page 519 of the "Microsoft Access User's Guide" gives examples of valid conditions to be used in the condition column in the Macro window. The last example shows the following:

```
COUNT(Order_ID) > 35
```

This condition will result in the error "Can't parse expression." You could correct the parsing error by adding square brackets and replacing the underscore with a space:

```
COUNT([Order ID]) > 35
```

You will now get this error message:

```
Function isn't available in expressions.
```

RESOLUTION

To see if the total number of entries in the Order ID field is greater than 35, you should use the DCOUNT() function in a macro expression.

```
DCOUNT("[Order ID]","Orders") > 35
```

If there are more than 35 entries in the Order ID field of the Orders table, the expression above will return TRUE.

References:

For more information, search for "DCOUNT" using the Microsoft Access Help menu.

"Microsoft Access User's Guide," version 1.0, Chapter 21, "Macro Basics," page 519

[References](#)

PRB: Join Is Broken by Value(s) in Fields '[Field Name]'

Article Number: Q96587
CREATED: 22-MAR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you add or edit records in a [form](#) or [query datasheet](#) and you attempt to commit the [record](#) by moving from the [current record](#) or by closing the [dynaset](#), the following error message is displayed:

Join is broken by value(s) in fields '<[field](#) name>'

where <field name> identifies the record that you attempted to edit.

CAUSE

You tried to enter or edit data through a query based on more than one [table](#). In a [one-to-many relationship](#), you entered or edited data on the "many" side, and a corresponding match based on the [join](#) field doesn't exist on the "one" side.

STATUS

This behavior is by design.

RESOLUTION

When you add or edit records in a query, if the [relationship](#) between two tables is a one-to-many relationship, only enter values in the linking field in the "many" table that have matches in the linking field in the "one" table.

MORE INFORMATION

Microsoft Access knows that a given relationship is a one-to-many relationship if the field on which the tables are linked or joined is a [primary key](#) (or Indexed - No Duplicates) in one of the tables, but not in the other table. (The linking field in the "many" table is also known as the "[foreign key](#).") If a field is updated on the "many" side of a relationship and does not have a match on the "one" side of the relationship, an "orphan" would be produced.

Steps to Reproduce Behavior

Follow these steps in the sample Northwind Traders [database](#) (NWIND.MDB):

1. Open the query named Order Information.

2. From the Records menu, choose GoTo, and then select New.
3. Input a new Order ID and Customer ID. Enter a value in the Customer ID field that does not exist in the Customers table.
4. Press the DOWN ARROW key to move to a new record.

The following error message is displayed:

Join is broken by value(s) in fields 'Customer ID'

REFERENCES

=====

For more information, refer to "Setting Relationships Between Tables," in Chapter 3 of the "Microsoft Access User's Guide" for version 1.0 or version 1.1.

[References](#)

PRACC9303: Orders Form Doesn't Print/Preview SubTotal & Total

Article Number: Q96586
CREATED: 22-MAR-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

You have a main form and a subform, and in the subform you have a control that uses an aggregate function, such as Sum(), to calculate a value. On the main form, you have a control set to this subform control. When the main form is displayed in Form view, the control on the main form displays the correct value. However, when you switch the main form to Print or Print Preview, this control is blank.

For example, look at the Orders form included in the sample database NWIND.MDB. Note the Subtotal and Total fields. In Form view, these fields display values. When you switch to Print Preview, these values do not appear.

CAUSE

If the subform is displayed in Datasheet view, any main form controls that refer to aggregate calculated subform controls do not get repainted.

RESOLUTION

Display the subform as a continuous form, or use a domain aggregate function in the subform control. (See below for more information on these workarounds.)

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

To work around this problem, you can either use a continuous form instead of a datasheet in the subform, or replace the aggregate function with a domain aggregate function in the subform control.

In the form named Orders Subform, a text box control named Order Subtotal is calculated by using the following aggregate expression:

```
=Sum([Extended Price])
```

On the form named Orders, the Subtotal control refers to this

calculated subform control. The Orders form displays the Subtotal value but does not display the value in Print Preview.

Workaround 1

1. Open the Orders Subform in [Design view](#) and change the ViewsAllowed property to Both and the DefaultView property to Continuous Forms.
2. Close and save the Orders Subform.
3. Open the Orders form. Note that the Subtotal and Total fields are correct. Switch to Print Preview and these amounts are now displayed.

Note: You will probably want to rearrange the detail section of Orders Subform so it displays properly in the Orders form.

Workaround 2

1. Open the Orders Subform in Design view.
2. Change the text box control named Order Subtotal to the following:

Note: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

Object: text box

```
-----  
ControlName: Order Subtotal  
ControlSource: =DSum("[Extended Price]", "[Order Details2]", _  
                    "[Order ID]=Forms![Orders].[Order ID]")
```

3. Close and save the form.
4. Open the Orders form in Form view.

Note how the Subtotal and the Total controls display the correct values in both Form view and Print Preview.

[References](#)

PRB: Form Opens Blank Except for Header

Article Number: Q96585
CREATED: 22-MAR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You open a [form](#) in [Form view](#) and it is completely blank. There are no controls or text; everything is blank except for the [form header](#).

CAUSE

The form [footer](#) may be too large for the form, and it may be covering the entire detail section when you view the form.

Also, if you have a form that is [read-only](#) and there is no data in the underlying [dynaset](#), no controls will be shown.

RESOLUTION

If the problem is with the [form footer](#), shorten the footer to allow only enough space for any controls that may appear there.

In the case of no data, controls in the form footer or header will be visible.

MORE INFORMATION

There are other reasons that a form may open and appear to be blank. Typically, this happens because the dynaset contains no records, or privileges have been limited. However, an oversized form footer can also cause this to occur.

Steps to Reproduce Behavior

The following steps are based on the sample [database](#) NWIND.MDB:

1. Open the Categories form in [Design view](#).
2. From the Layout menu, choose Form Hdr/Ftr to add a header and footer to the form.
3. Extend the length of the footer to 5 inches (place no controls in the footer).
4. View the form in Form view. You should see a "blank form" that is actually the entire empty footer section.

References:

For more information on why a form may appear blank when opened in Microsoft Access, [query](#) on the following words here in the Microsoft Knowledge Base:

completely and blank and form

[References](#)

INF: Sample Preview Ignores Query

Article Number: Q96584
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you choose Sample Preview from the File menu, Microsoft Access ignores any query that your report is based on. Use Sample Preview to check the general layout of your report, but use Print Preview to check the data in your report.

MORE INFORMATION

The Sample Preview option for reports provides an accurate view of the layout of the report. However, the Sample Preview option ignores any criteria or joins in the underlying query upon which the report is based. The option is designed as a speed enhancement to view the look and feel of your report.

The data that you will see displayed in this Sample Preview is based directly on the table used in the report's underlying query. If more than one table is in this query, the Sample Preview will erroneously combine records from the two tables.

References:

"Microsoft Access Users Guide," version 1.0, Chapter 17, "Report Basics", page 406

[References](#)

PRB: Unwanted Record Separator at the Bottom of a Subform

Article Number: Q96583
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In a main/subform, the bottom border of the subform appears to be painted even if the subform's Border property is set as "clear".

CAUSE

In actuality, what appears to be a border may really be a record separator.

RESOLUTION

To resolve this situation, choose Size to Fit from the Layout menu for the subform. This is done while you are in Design view of the main form and the subform control has been selected.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the form in Design view.
2. Click on the name of the embedded subform.
3. Choose Palette from the View menu, or click the Palette button on the toolbar.
4. You can check or clear the Border property Clear check box, but the border of the subform remains unchanged.

[References](#)

PRB: Alert Appears When it Should not When Running a Macro

Article Number: Q96582
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you create a [macro](#) but don't save it, and then try to run another macro from the File menu, the following message appears:

? Must save macro first. Save now?

The Help option states that if you choose Cancel, you will be returned to the current macro window; however, this does not happen. If you choose the Cancel option from the message box, the macro list appears and you can choose another macro to run, without saving the first one.

RESOLUTION

Microsoft recommends that you save your macro before you call another one to run.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a macro with a MsgBox action, close and save the macro.
2. Create another macro, but do not save it.
3. While you have the second macro open, choose Run Macro from the File menu.
4. When a message box prompts you to save the current macro, choose Cancel.

[References](#)

INF: Join Type Mismatch Errors Caught at Runtime

Article Number: Q96581
CREATED: 22-MAR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access will not detect a type mismatch error for joins created in a [query](#) until the query is actually run. This is illustrated in the following example.

MORE INFORMATION

Follow these steps in the sample [database](#) NWIND.MDB:

1. Create a new Query.
2. Add the Orders and Products tables from the Add Table dialog box.
3. Create a [join](#) between the Employee ID [field](#) in Orders and the Product Name field in Products.
4. Bring all fields from both tables into the Field Row of the [QBE](#) grid (refer to chapter 5 of the "User's Guide").
5. From the Query menu, choose Run.

The query will start to process, then return the following error message:

```
! TYPE MISMATCH
```

Choose OK to clear the box (choosing the Help button will give you information on the possible causes of this error).

It may not be obvious that the join is at fault because of the timing of the error message (especially when querying large amounts of data over a [network](#)).

This issue is one reason it is advantageous to create relationships between tables before creating queries (refer to chapter 3 of the "User's Guide" for more information).

[References](#)

PRB: Invalid Reference to Field '<Value>' Using SetValue

Article Number: Q96580
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If the expression argument of a SetValue Macro Action evaluates to a text expression beginning with a letter, you may receive the following alert message when a SetValue macro action attempts to set the value of a control

```
! Invalid reference to field 'A023bbb'
```

where "'A023bbb'" is a value in your text expression.

CAUSE

An expression argument of a SetValue Macro Action that begins with a leading equal sign (=) is understood to be a field name.

RESOLUTION

Remove the leading equal sign in the Expression argument.

Steps to Reproduce Behavior

1. Create a macro named My Macro with one action as follows:

Macro Name	Action
-----	-----
My Macro	SetValue

My Macro Actions

```
SetValue
  Item: Field0
  Expression: ="A023bbb"
```

2. Create a new unbound blank form.
3. Place a command button on the form by dragging the macro named My Macro from the Database window onto the form.
4. Create a text box on the form. Verify that the ControlName is Field0.
5. Switch to Form view.

6. Click the command button to run the macro.

You will receive the following error message:

Invalid reference to field 'A023bbb'

References:

"Microsoft Access User's Guide," version 1.0, pages 538-539

[References](#)

PRB: SendKeys Macro Does Not Run

Article Number: Q96579
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A [macro](#) including a SendKeys macro action does not run the SendKeys action.

CAUSE

If the Wait argument for the SendKeys action is not set to Yes, the macro action is run after some delay. Depending upon the following macro action, the keystrokes that are sent may not be valid. If the keystrokes are not valid, the SendKeys action appears not to have been run.

RESOLUTION

Set the Wait argument for the SendKeys action to Yes.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a macro named TestMacro.

Macro Name	Action
-----	-----
TestMacro	SendKeys MsgBox

TestMacro Actions

SendKeys
Keystrokes: %FX
Wait: No
MsgBox
Message: "HELLO WORLD"

2. Save the macro.
3. Run TestMacro.

The first macro action should exit Microsoft Access. However, the Wait argument is set to No and the message box is activated before the SendKeys action is run. When the keystrokes are processed, they

are not valid and you will hear beeping instead. If the Wait argument had been set to Yes, the SendKeys action would have been completed before moving to the next action, thus exiting Microsoft Access.

[References](#)

INF: Conversions for C Data Types in Access Basic

Article Number: Q96578
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes some of the conversions between [Access Basic](#) and C arguments.

MORE INFORMATION

The following are general [data type](#) conversions:

hWnd	ByVal var As Integer
LPSTR	ByVal var As String
hDC	ByVal var As Integer
WORD	ByVal var As Integer
DWORD	ByVal var As Long
INT	ByVal var As Integer
INT FAR *	var As Integer
CHAR FAR *	ByVal var As String
LONG	ByVal var As Long
LONG FAR *	var As Long
BYTE	ByVal var As Integer
BOOL	ByVal var As Integer
UNSIGNED SHORT	ByVal var As Integer
UNSIGNED CHAR	ByVal var As Integer - must convert with Asc(var)
UNSIGNED LONG	ByVal var As Long

The following list shows how to define the C data type in an Access Basic data type:

BYTE, BOOL	Var As String * 1	
LPSTR, CHAR FAR *	Var As Long	- Use API lstrcpy() to get the string pointer.
anything FAR *	Var As Long	- This does not give the correct data type but contains the correct number of bytes; you may have to convert.

The following list describes the differences returned from a [function](#):

LPSTR	Function As Long	- Use lstrcpy() to get the string from the pointer.
-------	------------------	---

[References](#)

PRB: Macro Condition Evaluates True if Control Is Null

Article Number: Q96577
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A macro that conditionally checks the contents of a control on a form always evaluates True if the control is null.

RESOLUTION

Include a StopMacro macro action with a condition to test the control for null. If the control is null, the macro will stop. If it is not null, the second macro action will occur. This will be the action you want to occur if your condition is met.

MORE INFORMATION

If you do not test for the null condition, the macro will evaluate to True.

Steps to Reproduce Behavior

1. Create a new unbound form named Form1.
2. Add a text box control to the form named Field0.
3. Create and save a new macro named Test.

Macro Name	Condition	Action
Test	[Field0] = "yes"	MsgBox

MsgBox Arguments

Message: The Answer is YES!
Beep: Yes

4. Switch back to Design view of Form1 and place a command button on the form. Set the OnPush property to the Test macro.
5. Switch to Form view and run the macro by choosing the command button. Note that Field0 is null. (The Msgbox should appear).
6. Type the word "NO" into Field0 and rerun the macro (no Msgbox will appear).
7. Type the word "YES" into the field and run the macro (Msgbox should

appear).

To correct this situation, change the Test macro so it includes the StopMacro action before the MsgBox action:

Macro Name	Condition	Action
Test	[Field0] Is null [Field0] = "yes"	StopMacro MsgBox

[References](#)

INF: Error Messages When Concatenating Variables/Controls

Article Number: Q96576
CREATED: 22-MAR-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you are concatenating variables or controls in a [function](#) or CreateDynaset [method](#), you may receive one of the following error messages:

Type Mismatch

-or-

1 parameter expected only 0 supplied

These errors can result if either of the following is true:

- You are including a [variable](#) or [control](#) name in the function [syntax](#) and Microsoft Access is unable to recognize the [data type](#).
- You are using incorrect syntax to concatenate the variables.

This article presents three examples in which these errors may occur and explains how to correct them.

MORE INFORMATION

"Type Mismatch" Error Message

You receive the "Type Mismatch" error message when you concatenate a variable or control in a method or function that has a [string](#) data type, which you are using as a numeric data type. For example, the following function produces the "Type Mismatch" error message:

```
Function MyFunction (DataToFind As String)
    Dim MyDB As Database, MySet As Dynaset
    Set MyDB = CurrentDB()
    Set MySet = MyDB.CreateDynaset("Employees")
    MySet.FindFirst "Title = " & DataToFind
```

The correct syntax for the last line of code above is as follows:

```
MySet.FindFirst "Title = '" & DataToFind & "'"
```

Keep the following requirements in mind:

- When the [criteria](#) for an [Access Basic](#) function or method is a string

data type, single quotation marks around the string variable are required.

- Numeric data types do not require delimiters.
- Dates require the number sign (#).

"1 Parameters Were Expected, But Only 0 Were Supplied" Error Message

You may receive this error message when you use the CreateDynaset method in Access Basic on an existing [query](#). If the query is a [parameter query](#), you need to explicitly declare the parameter and its data type and set the parameter value for that query in the function. (For more information, see page 137 of the "Introduction To Programming" manual.) The code that generates the error resembles the following:

```
Function TestQP ()
    Dim MyDB As Database, MySet As Dynaset
    Set MyDB = CurrentDB()
    Set MySet = MyDB.CreateDynaset("Query1")
    Debug.Print MySet![First Name]; Tab(10); MySet![Last Name]
End Function
```

When you refer to a parameter query, the correct syntax is as follows:

```
Function TestQP ()
    Dim MyDB As Database, MySet As QueryDef, MyDyna As Dynaset
    Set MyDB = CurrentDB()
    Set MySet = MyDB.OpenQueryDef("Query1")
    MySet![Enter a Name] = "Dav*"
    Set MyDyna = MySet.CreateDynaset()
    Debug.Print MyDyna![First Name]; Tab(10); MyDyna![Last Name]
    MyDyna.Close
    MySet.Close
End Function
```

The same error message also appears when you concatenate a variable in the [SQL](#) SELECT statement of a CreateDynaset method. The correct syntax is as follows:

```
Set MySet = MyDB.CreateDynaset("SELECT * FROM Employees WHERE _  
Employee ID] = " & Forms!Form1!Field0 & ";" )
```

NOTE: The preceding statement should be entered on one line. An underscore (_) is used as a line continuation character to improve the readability of the code.

This SELECT statement points to a control on a [form](#) for the WHERE clause.

[References](#)

INF: How to Use Access Basic to Derive Statistical Mode

Article Number: Q96575
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes an [Access Basic function](#) that creates a snapshot that determines the frequency of occurrences for all numbers in a [table](#).

The most frequently occurring number in a dataset is called the mode, which is a measure of central tendency (a "middle" measure of a dataset). The data set consisting of the numbers

```
{1,1,2,2,2,3,6,100}
```

has a mode of 2, because it occurs three times in the set. The data set consisting of the numbers

```
{1,1,1,2,2,2,3,6,10}
```

is bimodal, and has modes of 1 and 2 (both numbers occur three time in set).

This article assumes that you are familiar with Access Basic and how to create and use Access Basic procedures.

MORE INFORMATION

To create an Access Basic function that determines the mode, open a new [module](#) or a previously created module and enter the following code:

NOTE: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declarations section of the module.  
'*****
```

Option Explicit

```
'=====
```

```
' Create the following Mode() function in the Module
```

```
' The following function will initialize:
```

- ' - A [variable](#) for the [database](#) object.
- ' - A variable for the snapshot.
- ' - Sets the database object to the opened database.

```
' - Creates a snapshot based on the database object.
' - This function requires table and field name parameters where the
'   arguments are passed using "[" when the name includes spaces.
```

```
=====
Function Mode (tName$, fldName$)
    Dim ModeDB As Database
    Dim ssMode As Snapshot
    Dim ModalField1, ModalField2, ModalResult1, ModalResult2

    If tName$ = "" Or fldName$ = "" Then Exit Function

    Set ModeDB = CurrentDB()
    Set ssMode = ModeDB.CreateSnapshot("SELECT DISTINCTROW Count_
        (" & fldName$ & ") AS Mode, " & fldName$ & " _
        FROM " & tName$ & " GROUP BY " & fldName$ & " _
        ORDER BY Count(" & fldName$ & ") DESC;")
    ModalField1 = ssMode(fldName$)
    ModalResult1 = ssMode!Mode
    ssMode.MoveNext
    ModalField2 = ssMode(fldName$)
    ModalResult2 = ssMode!Mode

    If ModalField1 <> ModalField2 Then
        Mode = "The Result is Modal: " & ModalResult1
    Else
        Mode = "The Result is Bimodal: " & ModalResult1 & _
            " and " & ModalResult2
    End If
    ssMode.Close
    ModeDB.Close
End Function
```

Example of Using this Code

Create a [form](#) with [text box](#) controls that will reflect all measures of central tendency of a data set. In the ControlSource property for the text box [control](#), enter:

```
=Mode("TableName", "FieldName").
```

The value of this control will be the statistical mode of the data set. Another way to use this function is to call it from within another function that compares mode from different data sets. For example:

```
Function CompareModes()
    Dim MyDB as Database
    .
    .
    .
    X = Mode("[TableName]", "[FieldName]")
    Y = Mode("[Table Name]", "[Field Name]")
    If X > Y Then Debug.Print "The mode for X is greatest."
End Function
```

[References](#)

INF: General Number Format Ignores Decimal Number Setting

Article Number: Q96574
CREATED: 22-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you place a text box control on a form or report bound to a numeric field, if the control's Format property is set to General Number, the DecimalNumbers property setting will have no effect. This is because the General Number format displays only the number of decimal places necessary for each value.

References:

For more information, search for "General Number," then "Format Property - Number and Currency DataTypes" using the Microsoft Access Help menu.

[References](#)

PRB: Label Placed in, Not to Left of, Command Button on Form

Article Number: Q96573
CREATED: 22-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you place a command button on a form, a label is placed in the top left corner of the command button control, rather than to the left of the control.

RESOLUTION

You can change the default AutoLabel property of the command button to No before creating a new button from the toolbox. New command buttons will not have a label created when they are placed on the form. Alternately, you can move the label to a different location after the control has been created.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

Follow these steps in using the sample database NWIND.MDB:

1. Open the Suppliers form in Design view.
2. Choose the command button tool in the toolbox.
3. Place the control anywhere on the form. The command button will display with a label in the top left corner.

REFERENCES

=====

For more information on how to change the default properties of a control, see the "Microsoft Access User's Guide," version 1.0, page 430.

[References](#)

INF: Recovering MDB File Larger Than the 128-Megabyte Limit

Article Number: Q96550
CREATED: 21-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SUMMARY

A Microsoft Access version 1.0 [database](#) (.MDB file) is limited to a maximum size of 128 megabytes. However, you can exceed this limit by importing large amounts of data or by creating objects that cause the limit to be exceeded.

Microsoft Access version 1.1 supports databases up to 1 gigabyte.

MORE INFORMATION

In the [event](#) a Microsoft Access database exceeds the limit, here are two options you can use to attempt to remedy the situation:

1. If you are able to open the database, export some of the objects to another database, and then delete them from the [current database](#).
2. If you are unable to open the database, try to compact the database by using the Compact utility. It will work on files larger than 128 megabytes, unlike the Repair utility.

Unfortunately, there are situations where neither one of these options will work. If this is the case, you will be unable to recover the data using Access 1.0.

References:

"Microsoft Access User's Guide," version 1.0, Appendix A, page 663.

[References](#)

PRB: Changed Field Names In Table Are Not Reflected In Query

Article Number: Q96545
CREATED: 21-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you create a query using an existing table and then change a field name in that table, the new field name does not show in the query grid when the query is reopened. The old field name in the query grid has been made into an expression similar to this:

```
Expr1: TableName.[Old FieldName]
```

CAUSE

Changes to table field names do not propagate up through the queries based on that table.

RESOLUTION

Replace the new expression in the field cell of the query grid with the correct field name from the table. The field list in the upper half of the query window will show the current field names.

STATUS

This behavior is by design.

MORE INFORMATION

This behavior actually preserves the original query so you can see what was originally included in the query.

Steps to Reproduce Behavior

1. Create the following table. Save it, and enter a few records.

```
Table: Table1
-----
Field Name:  OldName
DataType:    Text
```

2. Create the following query adding the table that you created in step 1.

```
Query Name: Query1
-----
Field Name: OldName
```

Show: True

3. Run this query. Notice that data in the OldName field shows up in Datasheet view of the query.
4. Open Table1 in design view and change the field OldName to NewName. Close and save the table.
5. Open Query1 in Design view. Notice that the field cell now shows the following:

Expr1: Table1.[OldName]

If you run the query, this field will be blank. To correct this, choose NewName from the list in the field cell of the query.

[References](#)

INF: Using Microsoft Word to Edit and Spell Check Memo Field

Article Number: Q96544
CREATED: 21-MAR-1993
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can use a word processor to edit memo fields stored in Microsoft Access tables. Then you can check the spelling and grammar in your data. This article describes how to use Microsoft Word for Windows to edit memo fields. To make use of this article, you'll need to have some familiarity with [Access Basic](#).

MORE INFORMATION

Note the following information before trying the example:

- Microsoft Word for Windows must have the DOS TEXT Import/Export [filter](#) installed.
- Only the most basic formatting will be preserved because memo fields don't support tabs, margins, fonts, or other fancy formatting.
- Memo fields are limited to 32,000 characters (about 8-12 pages in Microsoft Word for Windows). Excess characters will be truncated.
- This example is presented in the most simplest terms. If you open the Microsoft Word document and return to Microsoft Access before you save the file, you will receive an "Access denied" error message when the Access Basic code tries to open the document that is already currently open in Microsoft Word.

The following example uses the Employees [table](#) in the NWIND.MDB [database](#).

1. Create the following [function](#) in a new [module](#).

```
*****  
'Declarations section of the module  
*****  
  
Option Explicit  
  
'=====
```

```
'The following function EditMemo calls Microsoft  
'Word for Windows  
'=====
```

```
Function EditMemo (MyEditBox As Control)  
Dim X%, LineInFile$, NewLine$, FileNum%, CarriageReturn_LineFeed$  
  
CarriageReturn_LineFeed$ = Chr$(13) & Chr$(10)
```

```

FileNum% = FreeFile

Open "TEMP.TXT" For Output As #FileNum

'save the text to temporary file for Microsoft Word for Windows

If Not IsNull(MyEditBox) Then
    Print #FileNum%, MyEditBox
Else
    MsgBox "There is no text to edit. Abort."
    Exit Function
End If

Close #FileNum%

X% = Shell("WINWORD.EXE TEMP.TXT", 3)

MsgBox "Press Any Key to Return" ' required to pause the function

'Assume completion of the edits in Microsoft Word, now read the text
'and place it back into the edit control

FileNum% = FreeFile
Open "TEMP.TXT" For Input As #FileNum%

NewLine$ = ""
Do While Not EOF(FileNum)
    Line Input #FileNum%, LineInFile$
    If NewLine$ <> "" Then
        NewLine$ = NewLine$ & CarriageReturn_LineFeed$
    End If
    NewLine$ = NewLine$ & LineInFile$
Loop
MyEditBox = NewLine$

Close #FileNum%

End Function

```

2. Create a new blank form and base it on the Employees table.
3. Place two text box controls (Last Name and Notes) on the form and assign the following properties:

```

Object: Text box
-----
ControlName: Last Name
ControlSource: Last Name

```

```

Object: Text box
-----
ControlName: Notes
ControlSource: Notes
OnDblClick: =EditMemo([Notes])

```

4. Save the form and switch to Form view.

5. Double-click the memo [field](#) to launch Microsoft Word for Windows. You will be presented with the Convert File dialog window. Select Text Only and then choose the OK button.
6. Perform the operations you want to do.
7. Exit from Microsoft Word for Windows and save the document.
8. Microsoft Access loads the changes back into the memo field.

For a more elegant solution, one that replaces the MsgBox routine, [query](#) on the following words in the Microsoft Knowledge Base:

```
b_waccess and waitshell
```

[References](#)

PRB: Print Preview Displays Only First Line of Toggle Button

Article Number: Q96543
CREATED: 21-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you preview a form with a toggle button that contains text without any spaces, if the text is longer than the size of the toggle button, characters on the first line of the toggle button are correctly displayed, but the remaining lines are either missing or truncated. This happens when you use a Hewlett-Packard (HP) LaserJet IIISi printer and you have selected either of the following screen fonts: System or Microsoft Sans Serif.

CAUSE

In print preview, the method to display some system fonts on toggle buttons is slightly different than the method used to actually print the buttons.

RESOLUTION

Change the format of the toggle button control text to a printer font or a TrueType font.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create an unbound form named Test.
2. Place a toggle button control on your form. Do not change the default size of the toggle button.
3. From the View menu, choose Properties.
4. Type the word "Caption" (without the quotation marks) in the Caption property.
5. Switch to Form view. (Notice that the toggle button correctly wraps the text.)
6. From the File menu, choose Print Preview. Print preview truncates the letter "n." When you print the form, however, the text appears

correctly.

[References](#)

PRB: Changes in Subform Design do not Update in Main Form

Article Number: Q96477
CREATED: 17-MAR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

After editing a subform with the main form in Design view, the changes you make to the subform are not committed until you close and reopen the main form.

STATUS

This behavior is by design.

RESOLUTION

After editing the subform with the main form in Design view, close and save the subform. Click in the subform control and press ENTER. Your main form now reflects the changes that you made to its subform.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the sample database NWIND.MDB.
2. Open the Categories form in Design view.
3. Select the Category ID control.
4. Double-click anywhere inside the Categories subform control to open it in Design view.
5. From the View menu, choose Field List. Drag the English Name field to the Category subform.
6. Exit and save the changes to the Categories subform. This action returns you to the the Categories form in Design Mode.
7. Switch to Form view.

Note that the field you added (English Name) is not in the subform. You must close the main form (Categories) and reopen it to see the field that you added to the subform, or follow the steps outlined below:

1. Repeat steps 1-7 above, dragging down the Supplier ID field instead of the English Name field.

2. Select the Category ID field in the main form.
3. Click in the Categories subform control and press ENTER.

NOTE: You cannot simply select the subform control; you must click in it.

4. Switch to Form view.

Note that the field you added (Supplier ID) appears on the subform without you having to close and reopen the main form.

References:

"User's Guide," version 1.0, page 274

[References](#)

INF: Create a Function to Capitalize Single Words and Phrases

Article Number: Q96476
CREATED: 17-MAR-1993
MODIFIED: 10-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There is no built-in format or [function](#) in Microsoft Access that capitalizes the first character of a word or set of words, as does the Proper function in Microsoft Excel. However, you can create your own Proper function in [Access Basic](#). This article provides the code to create a multiple-word Proper function.

There are two variations of the Proper function:

- One is used in the AfterUpdate property of a [control](#) on a [form](#), primarily for data entry.
- The other is used in calculated [query](#) fields, calculated controls on forms and reports, [macro](#) SetValue expressions, and Access Basic expressions.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

MORE INFORMATION

```
'*****  
'Declarations section of the module  
'*****  
    Option Explicit  
'=====
```

'The following function is designed for use in the AfterUpdate
' property of Form controls.
'Features:
' 1. Leading spaces do not affect the function's performance.
' 2. "O'Brian" and 'Wilson-Smythe' will be properly capitalized.
'Limitations:
' 1. It will change 'MacDonald' to 'Macdonald'.
' 2. It will change 'van Buren' to 'Van Buren'.
' 3. It will change 'John Jones III' to 'John Jones Iii'.
'=====

```
Function Proper (Field)  
  
    Dim Temp$, C$, OldC$, i As Integer  
  
    If IsNull(Field) Then  
        Exit Function
```

```

Else
    Temp$ = CStr(LCase(Field))
    OldC$ = " " ' Initialize this to a single space (Chr$(32))
                ' because the first letter needs to be capitalized,
                ' but has no preceding letter.
    For i = 1 To Len(Temp$)
        C$ = Mid$(Temp$, i, 1)
        If C$>="a" And C$<="z" And (OldC$<"a" Or OldC$>"z") Then
            Mid$(Temp$, i, 1) = UCase$(C$)
        End If
        OldC$ = C$
    Next i
    Field = Temp$
End If

End Function

```

How to Use This Function in the AfterUpdate Property of a Control

1. Create a form based on a [table](#) or query.
2. Type the following in the AfterUpdate property of a [text box](#) control:

```
=Proper(<control name>)
```

For example, if the ControlName is [Full Name], type the following:

```
=Proper([Full Name])
```

How to Modify This Function for Expressions and Calculated Controls

1. Enter the code shown above to create the Proper function.
2. Modify the following section of code

```
If IsNull(Field) Then
    Exit Function
```

to read as follows:

```
If IsNull(Field) Then
    Proper = Field
```

3. Modify the following section of code

```
Field = Temp$
End If
```

to read as follows:

```
Proper = Temp$
End If
```

Examples

The examples below all require the second version of the Proper() function.

1. Using Proper() in a Query:

```
Field: Full Name:Proper([Last Name] & " " & [First Name])
```

This will concatenate the first and last names and capitalize the first letter of each.

2. Using Proper() in a calculated control on a Form or Report:

```
TextBox
```

```
ControlName: AddressP
```

```
ControlSource: =Proper([Address])
```

NOTE: In calculated fields, the ControlName must be unique.

3. Using Proper() in a Macro:

```
Action: SetValue
```

```
Item: Screen.ActiveControl
```

```
Expression: Proper(Screen.ActiveControl)
```

NOTE: This macro can be called from the AfterUpdate property of a control on a form. It effectively does the same job as the first version of the Proper() function.

References:

"Access Basic Language Reference," version 1.0, part 2, "A-Z Reference"

For information on creating a single-word Proper function, query on the following words in the Microsoft Knowledge Base:

create and proper and function

[References](#)

INF: How to Convert to and from Unsigned Integers

Article Number: Q96475
CREATED: 17-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There are situations in which calling functions from external dynamic-link libraries returns a 2-byte unsigned integer. [Access Basic](#) does not support this [data type](#). Correctly evaluating this data type requires converting it from an unsigned integer to an Access Basic Long data type.

NOTE: This article assumes that you are familiar with Access Basic and how to create and use Access Basic procedures.

MORE INFORMATION

The Access Basic integer data type has a range of -32,768 to 32,767. An unsigned integer has a range of 0 to 65,536.

Access Basic uses the most significant bit to set the sign of the value. Therefore, when a value exceeds 32,767, bit 16 is set to reflect a negative number. To evaluate an unsigned integer, you must manually adjust bit 16.

There are two methods you can use to convert to and from the unsigned integer type to the Access Basic [Long data type](#). The first [method](#) uses basic arithmetic (65,536 is subtracted or added to the unsigned integer). The second uses bit-wise operators. The arithmetic method works as well as the bit-wise method; however, the arithmetic method may be more readable, and the bit-wise method may be faster when executed repetitively.

1. Open a new [module](#) or a previously created module and enter the following code:

```
*****  
'Declarations section of the module.  
*****
```

Option Explicit

```
'=====
```

```
' Arithmetic Method:
```

```
' -----
```

```
' Create the following ar_ConvertFromUnsignedInteger& (Uint%)  
' and ar_ConvertToUnsignedInteger% (Bytes&) function in the  
' Module. This first function reads in an unsigned integer  
' and returns the converted value as a long. The second
```

```

'   function reads in a long a returns an unsigned integer.
'=====
Function ar_ConvertFromUnsignedInteger& (Uint%)
  If Uint% < 0 Then
    ar_ConvertFromUnsignedInteger& = Uint% + 65536
  Else
    ar_ConvertFromUnsignedInteger% = Uint%
  End If
End Function

Function ar_ConvertToUnsignedInteger% (Bytes&)
  If Bytes& > 32767 Then
    ar_ConvertToUnsignedInteger% = Bytes& - 65536
  Else
    ar_ConvertToUnsignedInteger% = Bytes&
  End If
End Function

'=====
' Bitwise Method:
' -----
' Create the following bw_ConvertFromUnsignedInteger& (Uint%)
' and bw_ConvertToUnsignedInteger% (Bytes&) function in the
' Module. This first function reads in an unsigned integer
' and returns the converted value as a long. The second
' function reads in a long a returns an unsigned integer.
' The message box statement in the second function is used
' to prevent an overflow message when the value passed to
' the function is greater than 64 kilobytes.
' To illustrate what is taking place in the first bitwise function:
' Uint% equals -23584, a value returned from an external dynamic
' link library that is an unsigned integer and needs to be
' converted to an long:
'      1010001111100000 (-23584)
' AND   1111111111111111 (7FFF)
'      -----
'      10001111100000 (41952)
'=====
Function bw_ConvertToUnsignedInteger% (Bytes&)
  Dim x%
  If Bytes& > 65535 Then
    MsgBox "You passed a value larger than 65535"
    Exit Function
  End If

  x% = Bytes& And &H7FFF
  bw_ConvertToUnsignedInteger% = x% Or -(Bytes& And &H8000)
End Function

Function bw_ConvertFromUnsignedInteger& (Uint%)
  bw_ConvertFromUnsignedInteger& = Uint% And &HFFFF&
'-----
' The &HFFFF& requires the & at the end of the hex number. This
' qualifies the hex number as 32 bit versus 16 bit.
'-----
End Function

```

Examples of Using These Functions

An external dynamic-link library (DLL) requires and returns an unsigned integer. The Declare statement looks like the following:

```
Declare Function External_Call% Lib "your.dll" (ByVal ValueToPass%)
```

The Declare statement uses integer data types because Access Basic does not support unsigned integers. However, the external dynamic-link library does not know this and returns an unsigned integer that must be converted for use; therefore, the code may appear as follows:

```
x% = External_Call(bw_ConvertToUnsignedInteger%(41952))  
y& = bw_ConvertFromUnsignedInteger&(x%)
```

[References](#)

PRACC9303: Access Cue Cards Don't Use Standard Windows Menu

Article Number: Q96474
CREATED: 17-MAR-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When using [Cue Cards](#) in Microsoft Access, you might notice that the [Control menu](#) is not a standard Windows menu. For example, when you click a menu item to select it, it doesn't become highlighted.

RESOLUTION

Although the Cue Cards Control menu does not look like the standard Windows menu, its functionality is not impaired. You are still able to select the menu items with your mouse.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

The Control menu is the uppermost left-hand corner of a window. The standard Windows Control-menu commands allow you to resize, move, maximize, and minimize that window.

Steps to Reproduce Problem

1. From the Help menu, choose Cue Cards.
2. From the Cue Cards help screen, select "I'm Not Sure."
3. Click the Cue Cards Control-menu box (the upper-left corner of the Cue Cards window).
4. Select Minimize.

Although the word Minimize is not highlighted, the Cue Cards window will become minimized.

References:

For more information about the elements of a standard Microsoft window, see the "Microsoft Windows User's Guide." For Windows version 3.0, see Chapter 2, "Basic Skills," pages 21-22. For Windows version 3.1, see Chapter 1, "Windows Basics," pages 8-9.

[References](#)

PRB: Navigation Button Unavailable in Main/Subform Wizard

Article Number: Q96473
CREATED: 17-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you are using FormWizard to create a main/subform form, you cannot always use the FormWizard's navigation buttons to move backwards (to the previous Wizard screen).

RESOLUTION

If you need to change the table or query upon which the main form is based, choose Cancel. You will need to restart the FormWizard.

MORE INFORMATION

The FormWizard's navigation buttons (Back and Next) permit you to move forward and backward within the FormWizard.

Note that you will be able to navigate backward to change the subform source. It is only the main form source that you cannot navigate backward to change.

[References](#)

PRACC9303: Error: Not Enough System Resources in Access

Article Number: Q96471
CREATED: 17-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

Running a Microsoft Access [query](#) can result in this error:

Not enough system resources to [update](#) display.

CAUSE

Changing the Row Height and Font after initially running a query causes this error to occur the next time that query is run.

RESOLUTION

Choose the OK button when you get the above error message. The query runs and displays its results with the default settings. The next time you run the query, no error will appear.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

Steps to Reproduce Problem

1. Open the sample [database](#) NWIND.MDB.
2. Run the Category List query. To run a query, choose the Query button in the [Database window](#). Select the desired query, and then choose the Open button.
3. Choose Row Height from the Layout menu.
4. Change the setting to 12.0.
5. Choose Font from the Layout menu and change the font to Microsoft Serif and the font size to 10.
6. Switch the query to design mode and choose Run from the Query menu.
7. From the File menu, choose Close.
8. From the Database window, select the Category List query and choose Open.

The error message appears. After choosing the OK button (your only option), you will see the query with the default row height, font, and font point size. The next time you run the query, no error message will appear.

References:

For more information on changing the row height in a query, search for "Row Height command" then "Row Height Command (Layout Menu)" using the Help menu.

For more information on changing the font in a query, search for "Font command" then "Font Command (Layout Menu)" using the Help menu.

[References](#)

INF: How to Determine When a Shelled Process Has Terminated

Article Number: Q96470
CREATED: 17-MAR-1993
MODIFIED: 09-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

This article describes a [method](#) by which an [Access Basic](#) program waits until a shelled process has terminated. The Shell() [function](#) initiates a process and returns to Access Basic.

The shelled process continues indefinitely until you stop it; terminating Microsoft Access does not cause the shelled process to terminate. If you want Microsoft Access to wait until the shelled process has finished before continuing, you may not want to use this method.

MORE INFORMATION

By using the Windows application programming interface (API) functions GetActiveWindow() and IsWindow(), your program can monitor the status of a shelled process. The API function GetActiveWindow() should be called from within a loop, after the Shell() function, to get the handle of the main application window for the shelled process. This works correctly only if you invoke the Shell() function using a window style with focus (that is, window styles 1, 2, or 3) and a main application window is displayed for the shelled process.

This routine works when shelling to Windows or MS-DOS programs. It does not work for applications that do not display a window (such as applications that run invisibly in the background).

By continually calling the API function IsWindow() from within a While loop, you can make the Access Basic program wait until the shelled process has terminated. The Windows API function IsWindow() simply verifies that the window associated with the handle found with GetActiveWindow() is still a valid, existing window.

To start the example, create a sample [form](#) with a [command button](#). Set the properties as follows:

```
Control: Command Button
-----
OnPush: =Wait_Till_Shell_Done()
```

The steps below are necessary to build an Access Basic function called Wait_Till_Shell_Done(), which uses the Shell() function to execute the Windows Calculator accessory. The program is an example of how to use the Windows API functions GetActiveWindow() and IsWindow() to wait until a shelled process has finished before resuming execution.

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

NOTE: You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive the duplicate procedure name error message, remove or comment that particular declarations statement from your code.

```
'=====
' General Declarations Section
'=====

Option Explicit
Declare Function GetActiveWindow% Lib "User" ()
Declare Function IsWindow% Lib "User" (ByVal hWnd%)
Declare Function FindWindow% Lib "User" (ByVal lpClassName As Any, _
    ByVal lpWindowName As Any)

Function Wait_Till_Shell_Done ()
    Dim TimeOutPeriod%, r%
    Dim fTimeOut%, s!
    Dim AccessshWnd%, hWndShelled%
    Dim KeepWaiting%

    ' Maximum seconds to wait for application window to appear.
    TimeOutPeriod% = 5

    ' Must shell w/window style with focus, window style 1, 2, or 3.
    r = Shell("CALC.EXE", 1)

    fTimeOut% = False
    s! = Timer
    AccessshWnd% = kb_FindWindow("Omain", 0&)

    ' Wait for the application window of the shelled process
    ' to become active.

    Do
        ' Process Windows events for the shelled program.
        ' WARNING: Calling DoEvents() allows the user to select any
        ' window. If the user selects another window while in this
        ' routine, this routine will incorrectly treat that window as
        ' part of the shelled process.

        r = DoEvents()
        hWndShelled% = kb_GetActiveWindow()

        ' do while active window is the calling form...
        KeepWaiting% = hWndShelled% = Forms!form1.hWnd

        ' ...or do while Microsoft Access is the active window
        KeepWaiting% = KeepWaiting% Or (hWndShelled% = AccessshWnd)

        ' ...or do until exceeded the time-out waiting for program.
```

```
        KeepWaiting% = KeepWaiting% Or (s! < TimeOutPeriod)

Loop While KeepWaiting%

' If a time-out occurred, display a time-out message and terminate.
If fTimeOut% Then
    MsgBox "Timeout waiting for shelled application", 16
End
End If

' Wait until the shelled process has terminated.
While kb_IsWindow(hWndShelled%)
    ' Process pending Windows events.
    r = DoEvents()
Wend

MsgBox "Shelled application just terminated", 64
End

End Function
```

REFERENCES

=====

"Microsoft Windows 3.1 Programmer's Reference," Volume 2

[References](#)

INF: Short Description of a Dynaset

Article Number: Q96469
CREATED: 17-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

A [dynaset](#) is a Microsoft Access term for an updateable view. Rather than storing the actual data locally for the view, pointers to the data are stored locally in memory. Displaying the data in the view involves using the pointers to get the data from the underlying tables. Also, these pointers make it possible to [update](#) the data as well.

[References](#)

INF: Requirements to Connect to Sybase SQL Server

Article Number: Q96468
CREATED: 17-MAR-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The [SQL](#) Server [ODBC](#) driver that ships with Microsoft Access is designed to work with both Microsoft SQL Server and Sybase SQL Server. This article describes requirements for running Microsoft Access versions 1.0 and 1.1 with Sybase SQL Server.

MORE INFORMATION

The SQL Server ODBC driver works with both Microsoft and Sybase SQL Servers. However, the Microsoft applications that shipped with the SQL Server driver (Visual Basic version 2.0 and Microsoft Access) were tested and verified with Microsoft SQL Server only.

The following sections discuss the issues associated with using the Microsoft SQL Server ODBC driver with Sybase SQL Server.

ODBC Driver Requires Catalog Stored Procedures

The ODBC driver requires the catalog stored procedures that ship with SQL Server version 4.2a. Until the 4.9.1 Sybase release, these catalog procedures did not ship with Sybase servers. To install the procedures, you must run the INSTCAT.SQL script against the Sybase server.

NOTE: Sybase changed the system catalogs in their 4.8 release; therefore, Sybase SQL Server versions 4.8 and later require a modified script called INSTCAT.48. This revised script is available from Microsoft Product Support and CompuServe.

For more information, [query](#) on the following words here in the Microsoft Knowledge Base:

access and sybase and instcat.48

ODBC Driver Comes with Named Pipe Net-Library

The SQL Server ODBC driver that ships with Microsoft Access and Visual Basic 2.0 comes with the Named Pipe Net-Library. To connect to Sybase, you need to use the SQL Bridge product or buy the Net-Libraries from Sybase for each client machine. (The only Sybase Net-Library that has gone through interoperability testing is the Microsoft TCP/IP Net-Library.)

Both Microsoft Access and Visual Basic 2.0 use the network in a way that is atypical of Windows applications; they use multiple simultaneous connections and use asynchronous calls. For this reason, Sybase Net-Libraries may fail with these products in some network configurations. Microsoft and Sybase are discussing how best to perform interoperability testing with the various network configurations supported by Sybase (FTP, Wollongong, Novell and other TCP/IP, DECnet, and so forth).

The following is a list of Net-Libraries that Microsoft has confirmed to work Microsoft Access/ODBC and Sybase SQL Server:

- Named Pipe Net-Libraries 1.0
- Novell LAN WorkPlace TCP 4.0 with 1.0 Net-Library
- NetWare (SPX) 1.0 Net-Library
- FTP 2.05 with Net-Libraries 1.0 and 1.0.1
- FTP 2.10 with Net-Library 1.0.1 EBF 1285
- Microsoft TCP/IP 1.0 with Net-Library 1.0.1
- NetManage TCP/IP 3.06 Net-Library 1.0 EBF 1326
- PC-NFS Net-Library 1.0 EBF 1334

The following Net-Library works with Microsoft Access/ODBC and Sybase SQL Server under specific conditions:

- Wollongong Pathway Net-Library 1.0
 - Only if EBF # 1327 has been applied

NOTE: An EBF is an emergency bug fix, available from Sybase.

Some of the products included here are manufactured by vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

PRACC9301: Access GP Faults in Datasheet View of a Form

Article Number: Q96467
CREATED: 17-MAR-1993
MODIFIED: 14-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You may receive the following error when switching a [form](#) to Datasheet view:

An error has occurred in your application. If you choose Ignore, you should save your work in a new file. If you choose Close, your application will terminate.

You also receive the following message after the first one:

MSACCESS caused a general protection fault in MSACCESS.EXE.

CAUSE

A [macro](#) is assigned to the form's OnCurrent property and the macro includes a GoToControl action that sets the focus to a [control](#) located in the page [header](#) section. There is no visible [page header](#) section when you view the form in [Datasheet view](#).

RESOLUTION

If you are using a macro that includes a GoToControl action that sets focus to a control located in the page header section of the form, to avoid this error, do NOT switch to Datasheet view.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Create a GoToControl macro that goes to any control on the form and then assign it to the OnCurrent property of a form.
2. Place this control in the page header section of the form.
3. Switch to Datasheet view.

The above error appears.

[References](#)

PRB: Error When Installing Access on DEC Pathworks Network

Article Number: Q96466
CREATED: 17-MAR-1993
MODIFIED: 19-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

During a [network](#) server installation of Microsoft Access on a DEC Pathworks 4.x network, the following error message may be displayed:

You must run setup/a before setup/n

CAUSE

You probably attempted to install Microsoft Access to a Disk Services drive.

Microsoft Windows considers DEC Pathworks 4.x Disk Services to be a local disk drive. Windows considers DEC Pathworks 4.x File Services to be a network drive.

RESOLUTION

To perform a network server installation of Microsoft Access on a DEC Pathworks 4.x network, you must install to a File Services drive. You cannot successfully complete a network installation to a Disk Services drive.

[References](#)

PRB: OLE Excel and WinWord Do Not Close When Form Is Closed

Article Number: Q96465
CREATED: 17-MAR-1993
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Microsoft Access will not close some OLE servers when the user closes a table, form, or report while editing an OLE object.

STATUS

This behavior is by design.

MORE INFORMATION

Microsoft Access will not close Multiple Document Interface (MDI) applications, even if they were started as an OLE server for an object stored in Microsoft Access. An MDI application can edit and display multiple documents at once.

Microsoft Excel 3.0 and Microsoft Word for Windows 2.0 are examples of applications that will be left running if you double-click on an existing embedded object in Microsoft Access and then switch back to Microsoft Access and close the table, form, or report without closing the embedded object.

Steps to Reproduce Behavior

1. Open a form in Microsoft Access, and choose Insert Object from the Edit menu. In the list of available OLE servers, select Microsoft Excel Worksheet.
2. Create the object (worksheet) you want to embed. Once the object is completed, close Excel. Save the Access form and close it.
3. Reopen the form and double-click on the embedded object that you just created. Once Excel opens and the object is displayed, minimize Excel and close the Microsoft Access form. Excel will still be running. You must activate Excel and close it.

If you repeat these steps with applications such as Paintbrush, Microsoft Draw, and Note-It (these are not MDI applications), you will notice that these servers are closed when the object is closed.

[References](#)

INF: Using a Subquery to Simulate a SQL Sub-SELECT Statement

Article Number: Q96464
CREATED: 17-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

This article explains how to use a subquery to simulate a [SQL](#) sub-SELECT statement. The procedure below uses the sample [database](#) NWIND.MDB to generate a list of customers who have used all three shippers to deliver their orders.

MORE INFORMATION

You either can create a [query](#) (called Query1) based on the Customers and Orders tables and another query (called Query2) based on Query1, or you can enter the equivalent SQL statements. This article contains procedures for both methods.

Creating Query1

To create Query1 interactively, use the following steps:

1. In the [Database window](#), create a new query by choosing Query and then choosing New.
2. In the Table/Query box, select the Customers [table](#) and choose Add. Select the Orders table and choose Add again.
3. Choose Close. You have now added these two tables to the query. The tables are joined on the Customer ID [field](#).
4. Select the Customer ID field in the Customers table and drag it to the first [cell](#) in the Field [record](#) in the query grid. Select the Ship Via field from the Orders table and drag it to the second cell in the Field record.
5. Save this query as Query1 and choose Close.

The following is the SQL statement displayed for Query1 when you choose SQL from the View menu:

```
SELECT DISTINCT Customers.[Customer ID], Orders.[Ship Via]
FROM Customers, Orders,
Customers INNER JOIN Orders ON Customers.[Customer ID] =
Orders.[Customer ID];
```

Creating Query2

Create a second query (subquery) based on Query1.

To create Query2 interactively, use the following steps:

1. Create another new query, using step 1 in the previous procedure. Select Query1 as the source of your data.
2. Hold down the CTRL key and select the Customer ID and Ship Via fields from Query1. Drag these fields to the new query grid.
3. Choose the Sum button on the [toolbar](#).

NOTE: The Sum button looks like the Greek letter Epsilon.

4. In the Totals [list box](#) in the Customer ID field, select Group By; in the Ship Via field, select Count.
5. Clear the Show [checkbox](#) in the Ship Via field and type "=3" in the Criteria record.
6. Save this query as Query2. Choose Close.

The following is the SQL statement for the second query:

```
SELECT Query1.[Customer ID]
FROM Query1
GROUP BY Query1.[Customer ID]
HAVING ((Count(Query1.[Ship Via]))=3);
```

When you run Query2, a list of those customers who have used three different shippers is displayed.

[References](#)

INF: Using '<' and '>' and '=' Operators in Query Parameters

Article Number: Q96463
CREATED: 17-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

This article describes a simple, non-programmatic way to allow a user to type the comparison operators "<", ">", and "=" in [query](#) parameters. Due to the nature of the [method](#), query [criteria](#) should be kept as simple as possible. For more complex criteria, you will have to use [Access Basic](#) code.

MORE INFORMATION

1. Create a [table](#) called Names with the following fields:

Table: Names

Field: First Name
DataType: Text

Field: Last Name
DataType: Text

Field: Age
DataType: Number
Format: Single

2. Enter the following data:

First Name	Last Name	Age
Tom	Smith	35
Anne	Howard	7
Jim	Bowie	20
Sue	Thomas	44

3. Create the following query based on the Names table:

Query: List Names

Field: First Name

Field: Last Name

Field: Age
First Criteria Line: =Mid([Enter Age],2)
Second Criteria Line: <Mid([Enter Age],2)
Third Criteria Line: >Mid([Enter Age],2)

Fourth Criteria Line: <leave blank>
Show: True

Field: Expr1:Left([Enter Age],1)
First Criteria Line: "="
Second Criteria Line: "<"
Third Criteria Line: ">"
Fourth Criteria Line: Is Null
Show: False

The criteria are entered on successive lines (that is, "Criteria 1" is entered on the "Criteria" line, "Criteria 2" is entered on the "Or" line, and "Criteria 3" and "Criteria 4" are entered on the blank lines below that).

Save the query and run it. Sample output is given for each of the four types on input:

[Enter Age]	Output		
<leave blank>	Tom	Smith	35
	Anne	Howard	7
	Jim	Bowie	20
	Sue	Thomas	44
<21	Anne	Howard	7
	Jim	Bowie	20
>21	Tom	Smith	35
	Sue	Thomas	44
=7	Anne	Howard	7

Note: The query isn't designed to allow combinations of "<", ">", and "=", such as ">=20". You must always use one of the three operators, no more, no less, or you must leave the whole parameter blank.

The SQL for the query is given below:

```
SELECT DISTINCTROW
  [First Name], [Last Name], Age
FROM
  Names
WHERE
  (Age=Mid([Enter Age],2) AND Left([Enter Age],1)="=")
OR
  (Age<Mid([Enter Age],2) AND Left([Enter Age],1)="<")
OR
  (Age>Mid([Enter Age],2) AND Left([Enter Age],1)=">")
OR
  (Left([Enter Age],1) Is Null)
WITH OWNERACCESS OPTION;
```

[References](#)

INF: Misleading Documentation for Cancel Property

Article Number: Q96462
CREATED: 17-MAR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SUMMARY

The documentation for the Cancel property, as well as for the term "Cancel," is misleading. It implies that the purpose of the Cancel property is to allow you to select a Yes button to enable Cancel and thus undo the current record (just as if you chose the Undo Current Record command from the Edit menu).

This is not true; the Cancel property actually plays a much smaller role.

MORE INFORMATION

In Microsoft Access version 1.0 online help for "Cancel Property" states:

Use the Cancel property to give the user the option of canceling uncommitted changes and returning the form to its previous state.

The "Microsoft Access Language Reference" for versions 1.0 and 1.1 contains the same misleading wording on page 59.

NOTE: The help topic was corrected for version 1.1.

The actual purpose of the Cancel property is to execute the Cancel button's OnPush event property when you press the ESC key from anywhere on the form.

[References](#)

PRB: List Box Does Not Scroll when Scroll Box Is Dragged

Article Number: Q96461
CREATED: 17-MAR-1993
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you have a list box on a form with a vertical scroll bar, and you attempt to drag the scroll box up and down the list box, the list box contents will not scroll until the scroll box is dropped.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a form.
2. Place a list box control on the form with the following properties:

Object: Listbox

RowSourceType: Value List
RowSource: 1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0

3. View the form.
4. Drag the list box vertical scroll box up and down. The list box contents do not scroll until you release the scroll box.

[References](#)

PRB: Restore Macro Action Does Not Appear to Work

Article Number: Q96460
CREATED: 17-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In a [macro](#), the Restore action does not restore a window previously minimized by the Minimize action.

CAUSE

When an object is minimized in Windows, the focus is automatically shifted to the next window. The Restore action attempts to restore a window that is already in a restored condition.

RESOLUTION

To make Windows act on the minimized object, you must first select the object. To select the minimized object, put a SelectObject action in the macro immediately before the Restore action.

MORE INFORMATION

Steps to Reproduce Behavior

1. Start Microsoft Access and open a [database](#).
2. Create the following macro in a new macro [group](#) called Macro1:

Macro Name	Action
-----	-----
Test1	Minimize
	Restore

3. Create a new [form](#) called Form1 and place a [command button](#) on it. Give the command button the following properties:

Object: Command Button

OnPush: [Macro1].[Test1]

4. Open Form1 in [Form view](#) and push the command button.

The form will minimize as expected, but will not restore.

[References](#)

INF: Extend Mode and ESC in Form View

Article Number: Q96459
CREATED: 17-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The ESC key performs two different functions in Form view in Microsoft Access. If you are in Extend mode, pressing ESC will exit Extend mode.

While entering data in a control, pressing ESC activates the Undo feature and causes the data to be deleted. This can create a small inconvenience when you are in Extend mode and doing data entry simultaneously. Any effort to exit Extend mode by pressing ESC will erase your data in the current control (the one with the focus) in addition to leaving Extend mode.

MORE INFORMATION

Extend mode is used to select fields and/or records in Datasheet and Form view, and in a Report's Design view. It also can be used in grids to select cells and rows. To make a selection using Extend mode, position your cursor in a field or control, then press the F8 function key to turn Extend mode on. You will notice that EXT appears in the status bar. Once you have turned Extend mode on, you can press F8 repeatedly to select the word, the field, the record, and then all records.

To reverse the F8 selection use SHIFT + F8. The ESC key will exit Extend mode, as indicated in the summary, but it will also erase your data in the control with focus.

[References](#)

INF: Functions to Left and Right Pad Character Strings

Article Number: Q96458
CREATED: 17-MAR-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

An imported file may contain field values that have a combination of numeric and alphabetic characters. These fields do not sort in proper order if they contain variable numbers of characters. This article describes functions that you can use to pad these values with a selected character to produce values of consistent length.

Note: This article assumes that understand Access Basic code and know how to use Access Basic procedures.

More Information:

Suppose you have a table that contains Customer ID numbers with values entered as follows:

```
Customer ID
-----
123B
1231
1231B2
B123
```

In a query, these numbers would sort in ascending order as follows:

Unpadded	Right Padded	Left Padded
1231	123100	001231
1231B2	1231B2	00123B
123B	123B00	00B123
B123	B12300	1231B2

Right padding does not change the sort order; however, it is useful if you need to make all values a consistent number of characters. Left padding, however, will allow proper sorting.

To create a left-padding function and a right-padding function, enter the following code in a new or existing module in your database.

Note: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declarations section of the module.  
'*****
```

```
Option Explicit  
Dim x As Integer  
Dim PadLength As Integer
```

```
'=====
```

```
'The following function will left pad a string with a specified  
'character. It accepts a base string which is to be left padded with  
'characters, a character to be used as the pad character, and a  
'length which specifies the total length of the padded result.  
'=====
```

```
Function Lpad (MyValue$, MyPadCharacter$, MyPaddedLength%)
```

```
    Padlength = MyPaddedLength - Len(MyValue)  
    Dim PadString As String  
    For x = 1 To Padlength  
        PadString = PadString & MyPadCharacter  
    Next  
    Lpad = PadString + MyValue
```

```
End Function
```

```
'=====
```

```
'The following function will right pad a string with a specified  
'character. It accepts a base string which is to be right padded with  
'characters, a character to be used as the pad character, and a  
'length which specifies the total length of the padded result.  
'=====
```

```
Function Rpad (MyValue$, MyPadCharacter$, MyPaddedLength%)
```

```
    Padlength = MyPaddedLength - Len(MyValue)  
    Dim PadString As String  
    For x = 1 To Padlength  
        PadString = PadString & MyPadCharacter  
    Next  
    Rpad = MyValue + PadString
```

```
End Function
```

The following example shows an update query that would modify the Customer ID field by left padding the field with the 0 (zero) character. You will use the Lpad function that was created in step #2.

```
Update Query: Leftpad Customer Number  
-----  
Field name: Customer ID  
Update to: Lpad([Customer ID],"0",6)
```

[References](#)

PRACC316: Can't Paste Column If Focus Is on One Cell in Query

Article Number: Q96457
CREATED: 17-MAR-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you cut or copy a query grid column, you can paste it back in only if a column is selected. If there is no selection (that is, the focus is in a single cell), the Paste command is enabled and all information is erroneously pasted into one cell.

CAUSE

Currently Microsoft Access does not differentiate between a multi-line piece of text and a grid column in the Clipboard.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

[References](#)

PRB: Properties Do not Reset Between Crosstab and Make Table

Article Number: Q96456
CREATED: 17-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you choose Crosstab as the [query](#) type, by default all columns in the [QBE](#) grid have Group By in the Totals section. If you change the query type to Make Table, these [column](#) properties do not change.

CAUSE

When you change the query type, Microsoft Access removes the rows that are not allowed and displays rows that are required. Any other rows that are optional remain on the [QBE](#) grid. In this situation, the new Group By clause does not change the new [table](#).

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the sample [database](#) NWIND.MDB.
2. Create a new query.
3. Add the Employee Sales for 1991 table.
4. Bring the first two fields from the Field List down to the [QBE](#) grid.
5. From the Query menu, choose Crosstab.
6. From the Query menu, choose Make Table.
7. Type a table name in the Query Properties [dialog box](#).

Notice that the Group By attribute remains in the Total property.

[References](#)

PRB: Objects Cannot Be User-Defined Data Type Elements

Article Number: Q96420
CREATED: 16-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You will receive the following error message when attempting to use a Microsoft Access defined object data type as a user-defined data type element:

Expected: Integer or Long or Single or Double or Currency or String or Variant or Identifier.

CAUSE

Objects predefined by Microsoft Access are not supported as user-defined data type elements.

STATUS

This behavior is by design.

MORE INFORMATION

Microsoft Access defines eight object data types: Control, Database, Dynaset, Form, QueryDef, Report, Snapshot, and Table. You can use the Type statement in the Declarations section of a module to define your own data type. The user-defined data type cannot be one of the eight predefined data types.

Steps to Reproduce Behavior

1. Create a new module.
2. In the Declarations section, attempt to define a structure as follows:

```
Type ControlInfo  
    Ctl as Control  
End Type
```

3. From the Run menu, choose Compile All.

You will receive the above error message.

For more information, search for "Type Statement," then "Object Data Types" using the Help menu.

References:

"Microsoft [Access Basic](#): An Introduction to Programming," version 1.0,
Chapter 5, "Variable, Constants, and Data Types," page 43

[References](#)

PRB: NWIND.MDB Query with Criteria Displays Data in Odd Order

Article Number: Q96419
CREATED: 16-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you create a [query](#) to display the Category Name from the sample [database](#) NWIND.MDB Categories [table](#), the results appear in an order different than what is in the table. Even though the natural order of the Categories table is Beverages then Seafood, the query [datasheet](#) shows the opposite order.

CAUSE

The query will determine the fastest way to obtain the data. The order in which the records appear in the table may not be the logical order. This will result in an "odd" order returned by the query.

RESOLUTION

In the query you should choose the [sort order](#) you want to see the data displayed. Choose Ascending sort order for the Category Name [field](#) to have the names listed in alphabetical order.

STATUS

This behavior is by design.

MORE INFORMATION

Unless you specify the sort order in your query, there is the possibility that data will appear out of order.

Steps to Reproduce Behavior

Follow these steps in the sample database NWIND.MDB:

1. Create a new query based on the Categories table.
2. Drag the Category Name field onto the [QBE](#) grid.
3. Enter "Like BEV* OR SEAF*" (without the quotation marks) in the Criteria Row.
4. Run the query.

The resulting [dynaset](#) will put "Seafood" above "Beverages," even

though the table order is different.

[References](#)

INF: Example Has Wrong Number for Dlookup Ship Postal Code

Article Number: Q96116
CREATED: 08-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

The example given on page 145 of the "Microsoft Access Language Reference" shows a number for Ship Postal Code that does not exist in the sample [database](#) Northwind Traders (NWIND.MDB). The sentence reads as follows:

In the following example from the Orders [table](#) of the NWIND.MDB database, Ship Postal Code is the name of a [field](#), and 98008 is a [string](#) literal.

It should read:

In the following example from the Orders table of the NWIND.MDB database, Ship Postal Code is the name of a field, and 94117 is a string literal.

References:

"Microsoft Access Language Reference," version 1.0, page 145

[References](#)

INF: How to Use the Windows OpenFile Dialog Box

Article Number: Q96114
CREATED: 08-MAR-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to use the OpenFile [dialog box](#) from Access Basic.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications with the programming tools provided with Microsoft Access.

MORE INFORMATION

The OpenFile common dialog box can be implemented in Microsoft Access through the use of the Windows GetOpenFileName() application programming interface (API) [function](#). Using this function will simplify programming issues for the developer. This function creates a system-defined dialog box, familiar throughout Windows, that makes it possible for the user to select a file to open. This function will return a valid file to the programmer that is fully qualified with the path name. The developer can customize how the system will handle specific situations, such as specifying that the file must exist when the user wants to save a file through the use of flags.

To use the code below, paste the function and declarations into a [module](#).

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error, remove the Declare statement from your code or comment it out.

```
'-----  
' Global Declaration Section  
'-----  
Option Compare Database  
Option Explicit  
  
Type tagOPENFILENAME  
    lStructSize As Long  
    hwndOwner As Integer  
    hInstance As Integer  
    lpstrFilter As Long
```

```

    lpstrCustomFilter As Long
    nMaxCustFilter As Long
    nFilterIndex As Long
    lpstrFile As Long
    nMaxFile As Long
    lpstrFileTitle As Long
    nMaxFileTitle As Long
    lpstrInitialDir As Long
    lpstrTitle As Long
    Flags As Long
    nFileOffset As Integer
    nFileExtension As Integer
    lpstrDefExt As Long
    lCustData As Long
    lpfnHook As Long
    lpTemplateName As Long
End Type

Declare Function GetOpenFileName% Lib "COMMDLG.DLL" (_
    OPENFILENAME As tagOPENFILENAME)
Declare Function GetSaveFileName% Lib "COMMDLG.DLL" (_
    OPENFILENAME As tagOPENFILENAME)
Declare Function lstrcpy& Lib "Kernel" (ByVal lpDestString As Any, _
    ByVal lpSourceString As Any)

Dim OPENFILENAME As tagOPENFILENAME

Global Const OFN_READONLY = &H1
Global Const OFN_OVERWRITEPROMPT = &H2
Global Const OFN_HIDEREADONLY = &H4
Global Const OFN_NOCHANGEDIR = &H8
Global Const OFN_SHOWHELP = &H10
Global Const OFN_ENABLEHOOK = &H20
Global Const OFN_ENABLETEMPLATE = &H40
Global Const OFN_ENABLETEMPLATEHANDLE = &H80
Global Const OFN_NOVALIDATE = &H100
Global Const OFN_ALLOWMULTISELECT = &H200
Global Const OFN_EXTENSIONDIFFERENT = &H400
Global Const OFN_PATHMUSTEXIST = &H800
Global Const OFN_FILEMUSTEXIST = &H1000
Global Const OFN_CREATEPROMPT = &H2000
Global Const OFN_SHAREAWARE = &H4000
Global Const OFN_NOREADONLYRETURN = &H8000
Global Const OFN_NOTESTFILECREATE = &H10000

Global Const OFN_SHAREFALLTHROUGH = 2
Global Const OFN_SHARENOWARN = 1
Global Const OFN_SHAREWARN = 0

'-----
' Open Common Dialog Function
'-----
Function OpenCommDlg ()
    Dim Message$, Filter$, FileName$, FileTitle$, DefExt$
    Dim Title$, szCurDir$, APIResults%

    '*Define the filter string and allocate space in the "c" string

```

```

Filter$ = "Access (*.mdb)" & Chr$(0) & "*.MDB;*.MDA" & Chr$(0)
Filter$ = Filter$ & "Text (*.txt)" & Chr$(0) & "*.TXT" & Chr$(0)
Filter$ = Filter$ & "Batch (*.bat)" & Chr$(0) & "*.BAT" & Chr$(0)
Filter$ = Filter$ & Chr$(0)

'* Allocate string space for the returned strings.
FileName$ = Chr$(0) & Space$(255) & Chr$(0)
FileTitle$ = Space$(255) & Chr$(0)

'* Give the dialog a caption title.
Title$ = "My File Open Dialog" & Chr$(0)

'* If the user does not specify an extension, append TXT.
DefExt$ = "TXT" & Chr$(0)

'* Set up the default directory
szCurDir$ = CurDir$ & Chr$(0)

'* Set up the data structure before you call the GetOpenFileName

OPENFILENAME.lStructSize = Len(OPENFILENAME)
OPENFILENAME.hwndOwner = Screen.ActiveForm.hWnd
OPENFILENAME.lpstrFilter = lstrcpy(Filter$, Filter$)
OPENFILENAME.nFilterIndex = 1
OPENFILENAME.lpstrFile = lstrcpy(FileName$, FileName$)
OPENFILENAME.nMaxFile = Len(FileName$)
OPENFILENAME.lpstrFileTitle = lstrcpy(FileTitle$, FileTitle$)
OPENFILENAME.nMaxFileTitle = Len(FileTitle$)
OPENFILENAME.lpstrTitle = lstrcpy(Title$, Title$)
OPENFILENAME.Flags = OFN_FILEMUSTEXIST Or OFN_READONLY
OPENFILENAME.lpstrDefExt = lstrcpy(DefExt$, DefExt$)
OPENFILENAME.hInstance = 0
OPENFILENAME.lpstrCustomFilter = 0
OPENFILENAME.nMaxCustFilter = 0
OPENFILENAME.lpstrInitialDir = lstrcpy(szCurDir$, szCurDir$)
OPENFILENAME.nFileOffset = 0
OPENFILENAME.nFileExtension = 0
OPENFILENAME.lCustData = 0
OPENFILENAME.lpfHook = 0
OPENFILENAME.lpTemplateName = 0

'* This will pass the desired data structure to the Windows API,
'* which will in turn it uses to display the Open Dialog form.

APIResults% = GetOpenFileName(OPENFILENAME)

If APIResults% <> 0 Then

    '* Note that FileName$ will have an embedded Chr$(0) at the
    '* end. You may wish to strip this character from the string.
    Message$ = "The file you chose was " + RTrim$(FileName$)
Else
    Message$ = "No file was selected"
End If

MsgBox Message$

```

End Function

References:

"Microsoft Windows Software Development Kit Programmer's Reference,
Volume 2: Functions," version 3.1

[References](#)

PRB: RunCode Macro May Not Evaluate Conditions

Article Number: Q96113
CREATED: 08-MAR-1993
MODIFIED: 10-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you execute a RunCode macro with a condition that evaluates a control on a form, it runs the code specified regardless of whether the condition is met.

CAUSE

This will happen if the function specified in the action argument is preceded by an equal sign (=).

RESOLUTION

The macro action argument requires only the name of the Access Basic function procedure to execute and does not require an equal sign before the function name.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create an unbound form with two unbound text box controls named Field1 and Field2. Save the form as Form1.
2. In a new module, create a function named TestFunction.

```
'*****  
'Declaration section of the module.  
'*****
```

Option Explicit

```
Function TestFunction()  
    MsgBox "Test"  
End Function
```

3. Create a macro named Test1.

Macro Name	Condition	Action
Test1	Forms![Form1]![Field1]=2	RunCode

Test1 Actions

RunCode

Function Name: =TestFunction()

4. Open Form1 in Design view. Assign the macro named Test1 to the OnExit property of Field1.
5. Change to Form view and enter the value of 1 in Field1. Tab off the field and notice the message box pops up even though the condition was not met. This will happen regardless of whether the condition is met or not.

To evaluate the condition, remove the equal sign preceding the function name in the macro.

[References](#)

INF: Sorting String Values Based on their Numeric Values

Article Number: Q96112
CREATED: 08-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

To preface this article, I have a [database](#) that contains a product code [column](#). It is a text [data type](#) but I store numbers from 1 to 999 and each code can have a single letter for a suffix. For instance, I might have the following codes: 10a, 1d, 100b, 24c, 24a, 1, 89b, and 14. I want to view this data so that it is sorted on the code. It should appear like this: 1, 1d, 10a, 14, 24a, 24c, 89b, and 100b.

This article discusses one [method](#) for doing this. It requires basic knowledge of Microsoft Access queries and functions.

MORE INFORMATION

You need to use a [query](#) to view your data so that it is sorted on the code. This will work as long as the numbers do not exceed 1000 (if they do, you can add to this solution to accept larger numbers), there can be only one character digit after the number (that is, a, b, c; not aa, bc, cde), and numbers cannot have leading zeros (that is, 01, 02, 03, and so forth).

The query grid appears as follows:

```
Query: Query1
-----
Field: ID
      Show: Yes
Field: PRE: Val([ID])
      Sort: Ascending
Field: SUF: IIF(Val([ID])<10,Mid([ID],2,1),
              IIF(Val([ID]) between 10 and 99,
                  MID([ID],3,1),Mid([ID],4,1)))
      Sort: Ascending
```

This separates the numeric value (PRE) and the [string](#) value (SUF) into two separate (temporary) fields in the query. The query sorts on the individual pieces but shows only the true ID [field](#) (assuming your field name is ID).

[References](#)

INF: Using WIN.INI & ODBC.INI When Attaching SQL Server Table

Article Number: Q96110
CREATED: 08-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article discusses how the [ODBC](#) Driver manager and the [SQL](#) Server driver use the WIN.INI and ODBC.INI files during connection time.

MORE INFORMATION

For the purposes of this article, the following is assumed:

- You set up a data source for the SQL Server driver called mysqlsvr.
- The data source is located on a server called Unixmc.
- You are connecting to a Sybase SQL Server [table](#) on Unix whose address is 11.1.4.40,3180.
- The net-library you are using is wdbftptc.dll.

Given these assumption, if the setup was done correctly, the ODBC.INI file will have the following entry in the [ODBC Data Sources] section:

```
mysqlsvr=SQL Server
```

There will be a section called [mysqlsvr] in the ODBC.INI and it will have entries of the following [form](#):

```
DRIVER=c:\windows\system\sqlsrvr.dll  
Description=SQL\Server on server Unixmc  
OemToAnsi=No  
Network=wdbftptc  
Address=11.1.4.40,3180
```

The DRIVER entry points to the location of the SQL Server driver. It is located on the System subdirectory of the Windows directory.

The Description entry describes what kind of a data source this is. This is something that you can change when setting up the data source by using ODBC Administrator.

The OemToAnsi entry indicates whether conversion needs to be done from the OEM code page to the ANSI code page. Note that this is not the same as the translation DLL.

The Network entry indicates the name of the net-library.

The Address entry indicates the network address.

The WIN.INI file will have the following entry in its [SQLSERVER] section:

```
mysqlsrvr=wdbftptc,11.1.4.40,3180
```

When you run a SQLConnect or SQLDriverConnect command and choose to connect to the mysqlsvr data source, the Driver manager looks in the ODBC.INI file to find the name and location of the SQL Server driver. It looks for the DRIVER entry described above and loads it into memory.

Next, the SQL Server driver looks in the WIN.INI file's [SQLSERVER] section for mysqlsvr entry where it picks up the name of the net-library. Then it loads the net-library name (wdbftptc in this case) into the memory. Next, it looks at the entry again to get the address of the server which it passes on to the net-library. If there is no address following the name of the net-library, it passes the name of the data source to the net-library, as the network address.

Db-library users should note that unlike W3DBLIB.DLL, the SQL Server driver does not look at the dsquery entry unless dsquery is a data source in the ODBC.DLL file.

If you are using Microsoft Access, this sequence of operations take place when you attach a SQL Server table. If you are using Microsoft Visual Basic version 2.0, it takes place in response to the OpenDataBase call.

[References](#)

INF: Troubleshooting Access Setup and Installation

Article Number: Q96109
CREATED: 08-MAR-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article explains the following troubleshooting techniques that you can use to resolve Microsoft Access setup and installation problems:

- Checking your disks
- Checking the TEMP directory
- Checking what loads when you start Microsoft Windows
- Clean booting your system

Note that you can use many of these techniques to troubleshoot problems that may occur after installation.

MORE INFORMATION

Checking Your Disks

The following steps help determine whether your problem is related to bad disks. To check a specific disk:

1. Exit completely from Windows.
2. At the MS-DOS command prompt, type the following:

```
md \disktest
```

3. Put the suspect disk in your floppy disk drive, then type the following:

```
copy <a>:*.* c:\disktest
```

where <a> is the appropriate floppy drive letter.

4. If your disk is copied successfully, the MS-DOS command prompt is displayed and no error message appears. This means that this disk is not the problem, and you should continue troubleshooting.

If an error message is displayed, call Microsoft Customer Service at (800) 426-9400 and request replacement disks.

5. Delete the temporary DISKTEST directory and any files that you created in step 2 as follows:

- a. At the MS-DOS command prompt, type the following and then press

ENTER:

```
erase c:\disktest\*.*
```

- b. At the MS-DOS command prompt, type the following and then press ENTER:

```
rd c:\disktest
```

Checking the TEMP Directory

1. Exit completely from Windows.
2. At the MS-DOS command prompt, type the following:

```
set
```

3. Change to the directory pointed to by the TEMP [variable](#). For example, if you are at the root of your C drive when you type SET and "TEMP=C:\WINDOWS\TEMP" is displayed, type the following command:

```
cd \windows\temp
```

If an "invalid directory" message is displayed when you change to the TEMP directory, you must create the TEMP directory using the MS-DOS Make Directory (MD) command.

4. There should be no *.TMP files in your TEMP directory when Windows is not running. If there are *.TMP files in this directory, you should erase them. To see whether there are any .TMP files in your TEMP directory, type the following command:

```
dir *.tmp
```

For more information on the TEMP directory, [query](#) on the following words in the Microsoft Knowledge Base:

```
windows and temporary and files and definition
```

5. Use the MS-DOS CHKDSK command to check your hard drive. At the command prompt, type:

```
chkdsk
```

NOTE: Refer to your "Microsoft MS-DOS User's Guide and Reference" if you have lost allocation units, lost clusters, or cross-linked files.

6. Check the amount of free space on your hard drive by looking at the line that says "bytes available on disk." If this number is less than 4000000 bytes (4 MB), free up some additional hard drive space.

Checking What Loads When You Start Windows

For troubleshooting purposes, it is important that you have a clean

Windows configuration before you boot from a clean, bootable floppy disk (refer to "Clean Booting Your System" below). The following steps ensure a clean Windows configuration:

1. Using a text editor, such as the MS-DOS Editor or Windows Notepad, open the WIN.INI file. Check the following two lines:

```
load=  
run=
```

If there is anything to the right of the equal sign (=) on either of these two lines, place a semicolon (;) at the beginning of line. For example:

```
;load=c:\mydir\myprogram.exe
```

The semicolon causes Windows to ignore that line.

2. Save your changes and exit the WIN.INI file.
3. When clean booting, you do not want anything to load automatically from your startup group. To temporarily disable this functionality, hold down the SHIFT key as soon as you see the Windows logo screen. Keep the SHIFT key depressed until Windows is completely loaded.
4. Make sure that you are using a Windows video driver. To do this, exit Windows. At the C:\WINDOWS> prompt, type:

```
setup
```

Windows displays a System Information screen that indicates the type of display that you are using. If the Display field does not show EGA or VGA, follow the instructions on your screen to change the Display to VGA.

NOTE: If you receive a Welcome to Windows setup screen, you were not at the Windows prompt when you typed SETUP.

Clean Booting Your System

A clean boot eliminates many variables that may be related to your setup or installation problem. Follow these steps to clean boot your system.

1. Create a bootable floppy disk as follows:
 - a. Put a blank disk in your A drive. You must use your A drive.
 - b. At the MS-DOS command prompt, type

```
format a: /s
```

2. Use a text editor, such as MS-DOS Editor or Windows Notepad, to create a CONFIG.SYS file on the bootable floppy disk that you created in step 1.

A simple CONFIG.SYS contains the following:

```
files=50
buffers=20
device=c:\windows\himem.sys
<third-party disk partitioner>
<third-party disk compression driver>
<Other third-party driver if necessary to boot your computer>
shell=c:\<valid path>\command.com /p
```

NOTE: If you have the following line in your current CONFIG.SYS file, you should include in in your clean boot CONFIG.SYS file:

```
device=c:\windows\smartdrv.sys /double_buffer
```

For more on SMARTDRV.SYS, query on the following words in the Microsoft Knowledge Base:

SMARTDrive and 4.0 and design and overview

NOTE: If there is hard disk partitioning, disk compression, or other third-party software required to boot your computer, these files must be included in your CONFIG.SYS file.

3. Using a text editor, create an AUTOEXEC.BAT file on the bootable floppy disk that you created in step 1. A simple AUTOEXEC.BAT contains the following:

```
prompt $p$g
path=c:\windows
set temp=c:\<valid path>
```

4. Exit from the text editor.
5. With the clean, bootable floppy disk in drive A, reboot your system.
6. If you still experience setup problems, try starting Windows in standard mode by typing

```
win /s
```

at the A prompt. Windows does not try to create any virtual memory in standard mode. If running in standard mode resolves your problem, contact Windows Product Support for further assistance.

For more information on the clean booting your system, query on the following words in the Microsoft Knowledge Base:

clean and boot and windows

[References](#)

INF: Items in a Module Run Menu Are Unavailable (Grayed Out)

Article Number: Q96108
CREATED: 08-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you have a [module](#) window open and in focus, you may notice that many items in the Run menu are unavailable (grayed out):

MORE INFORMATION

The items that are unavailable are:

- Continue
- Single Step
- Procedure Step
- Set Next Statement
- Show Next Statement

This may be confusing because it seems that you ought to be able to do something similar to Single Step right from the code window.

You cannot begin stepping through code until you are in Break mode, which is why the items are unavailable. Once the code is executed, Break mode occurs when any of the following take place:

- A [breakpoint](#) that you set is encountered
- A Stop command is encountered
- CTRL+BREAK is pressed while the code is running
- An error occurs in the code

References:

"Microsoft Access Introduction to Programming," Chapter 4, "Debugging Your [Access Basic](#) Code," pages 29-42

[References](#)

PRB: Compile Error When Attempting to Use an AccessWizard

Article Number: Q96107
CREATED: 08-MAR-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

The following error appears when you try to use an AccessWizard:

Compile error

CAUSE

This occurs because there is a compile error in the code stored in the [database](#).

Microsoft Access must compile all code in memory to use any of it. This includes the code stored in the database and any libraries (.MDA files) that are loaded. When you attempt to use an AccessWizard, Microsoft Access must compile the [Access_Basic](#) code that is in the WIZARD.MDA file in addition to any code in your database. If there is an error in the code in your database, Microsoft Access will not be able to run the AccessWizard because the compilation failed.

RESOLUTION

To resolve this problem, follow these steps:

1. Open your database.
2. Open any [module](#) in [Design view](#).
3. Choose Compile All from the Run menu.

Microsoft Access will bring up the module window with the offending line of code. Upon correcting the error, and assuming there are no other errors, AccessWizards should now work.

If the Compile error message persists, repeat steps 2 and 3 until all errors in the code are corrected.

STATUS

This problem does not occur in Microsoft Access version 1.1.

[References](#)

INF: Accessing a Report Dynaset Requires a .Form Reference

Article Number: Q96106
CREATED: 08-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The text below discusses methods to access a [report's dynaset](#) and explains the apparent discrepancy in the documentation regarding the use of a [form](#) dynaset versus the use of a report dynaset.

MORE INFORMATION

Page 65 of the "Microsoft Access Introduction to Programming" manual for version 1.0 implies that both forms and reports have associated dynaset objects that are accessible and that [function](#) identically. This is basically true; once you access a report's dynaset, you can use the same procedures that you use with a form's dynaset. Any limitations in functionality relate only to the differences between an interactive object (the form) and a noninteractive object (the report).

According to page 162 of the "Microsoft Access Language Reference" manual for version 1.0, the Dynaset property applies only to forms. This is actually correct even though you can access a dynaset from a report in the same manner as you can access a dynaset for a form. The statement is correct because the Dynaset property is accessible only through a form reference. For example, if you have a form called "Form1," you can assess Form1's dynaset with the following reference:

```
Set MyDynaset = Forms!Form1.Dynaset
```

However, to access Report1's dynaset, use the following reference:

```
Set MyDynaset = Reports!Report1.Form.Dynaset
```

[References](#)

PRB: Leading Space in Property Sheet Entry May Cause Problems

Article Number: Q96105
CREATED: 08-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Valid values for properties are not recognized as valid. For example, you get a #Name value for a [control](#) that is bound to a [field](#) in the underlying [table](#) that definitely exists.

CAUSE

The problem may be caused by having a leading space as part of the entry for that property. For example, you may have a ControlSource property for a control that correctly identifies the name of a field for the control, but includes a leading space. In such a case, Microsoft Access attempts to include the leading space as part of the value you specified for ControlSource.

RESOLUTION

Remove the leading space.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new [form](#) in the sample [database](#) Northwind Traders (NWIND.MDB) bound to the Categories table.
2. From the fields list, drag the Description field onto the form.
3. With the Description field in focus, bring up the [property sheet](#) and put a space in front of the "Description" value that appears for the ControlSource property.
4. Switch from Design mode to Browse mode.

Notice that #Name appears for the Description field even though it is part of the underlying table.

[References](#)

INF: Requery, Repaint, and F9/Records Refresh

Article Number: Q96104
CREATED: 08-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Requery, Repaint, and F9/Records Refresh are different; however, they are all generally used to ensure that what you see is updated.

Requery

The Requery [macro](#) action updates controls, such as list boxes and combo boxes. It does not work for the [form](#) on which those controls exist.

Repaint

Repaint is not related to data, it merely redraws the object or [control](#). If there is data that has been updated in the underlying [dynaset](#) but is not reflected on the form, Repaint cannot be used.

F9

--

F9 requeries the form itself, as opposed to any control on the form.

[References](#)

INF: Macro to Enter Date Record Was Last Modified

Article Number: Q96103
CREATED: 08-MAR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article explains how to write a [macro](#) that will place the current date into a LastModified [field](#) in your [table](#) each time a [record](#) is modified. This [method](#) works when you are editing records from a [form](#).

MORE INFORMATION

When a form is used to edit records, you can date stamp the record by attaching a macro to the BeforeUpdate property of the form. For this example, we will use the Customers table in the sample [database](#) NWIND.MDB.

1. Add a new field to the Customers table.

Table: Customers

Field Name: DateModified
DataType: Date/Time

2. Create a macro called LastModified as follows:

Macro Name	Action
-----	-----
LastModified	SetValue
	Item: [DateModified]
	Expression: Date()

NOTE: There is no equal sign (=) in front of the Date() [function](#). If you put an equal sign in front of the Date() function, the macro will always put the date 12/30/99 into the [DateModified] [control](#) even though the macro step window shows the correct date in the argument section.

3. Create a form called Modify Customers based on the Customers table. Place a few of the fields on the form as follows:

Form Name: Modify Customers

ControlSource: DateModified
ControlSource: Customer ID
ControlSource: Company Name
ControlSource: Address

4. Enter the macro name LastModified into the BeforeUpdate property of

the Modify Customers form. Switch the form to Form view. Any time you modify the record, the current date will be placed in the DateModified control.

[References](#)

PRACC9303: User with Full Permissions Cannot Run Query

Article Number: Q96102
CREATED: 08-MAR-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You receive the following message even though you have full permission on the [table](#):

 Couldn't read; no read permission for table or [query](#) <query name>

You can view the table in Datasheet and [Design view](#), and you can open the query in Design view.

CAUSE

This occurs when a query has the property Run with Owner's Permissions checked and the user who created the query (the owner) is no longer on the system. Either the owner was deleted or the [database](#) was opened in a Microsoft Access session that was started with a different SYSTEM.MDA where the owner is not a included.

When the property Run with Owner's Permissions is checked, Microsoft Access checks the owner's [permissions](#). It does not compare the owner's permissions to those of the user running the query. Therefore, if the owner is not on the system, other users can't execute the query as long as the property Run with Owner's Permissions is checked.

RESOLUTION

To allow users to run the query with their own permissions, clear the Run with Owner's Permissions [check box](#) and save the query.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Start Microsoft Access. If you have activated [security](#), log in as a member of the [group](#) Admins.
2. From the Security menu, choose Users.

3. Press the New button and add a new user called John. Add another user called Jane. Do not assign either to any group other than Users.
4. If you have already activated security, skip this step. Choose Change Password... from the Security menu. Leave the Old Password blank, enter Admin for New Password and Verify. Press the OK button.
5. Exit and restart Microsoft Access. Log in as John leaving the password blank.
7. Create a new query based on any table. If you are in a secure system, base the query on a table that the Users group has full permissions to. From the View menu, choose Query Properties and select the Run with Owner's Permissions check box.
8. Exit and restart Microsoft Access. Log in as Admin using the password Admin.
9. Choose Users from the Security menu. Delete the user John.
10. Exit and restart Microsoft Access. Log in as Jane leaving the password blank.
11. Attempt to run the query.

At this point, you will see the error message "Couldn't read; no read permission for table or query '<query name>'".

12. Verify that Jane can open the query in Design view as well as open the table on which the query is based.

To avoid the problem, in Design view, the user Jane can choose Options from the View menu and clear the Run with Owner's Permissions check box. Depending on the security, that may have to be done by an Admin. After you clear the check box, the query will run.

[References](#)

INF: dBASE's REPORT FORM Command in Access Basic

Article Number: Q96101
CREATED: 08-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article discusses how to implement a dBASE-style REPORT FORM command in an [Access Basic](#) application.

MORE INFORMATION

In dBASE, you can create a [report](#) via the Report Generator, then print the report from a program using the REPORT FORM TO PRINT command. In Microsoft Access, you can print a pre-existing report programmatically using DoCmd OpenReport.

For example, suppose you have created a report in Microsoft Access called MyReport, and you want to print the report from an Access Basic procedure. Use the following command to accomplish this:

```
DoCmd OpenReport "MyReport"
```

If you omit the TO PRINT clause, which is optional in dBASE's REPORT FORM command, the report will print to the screen. This is equivalent to showing the report in Print Preview mode rather than sending it to the printer. This can be done with DoCmd OpenReport by adding the A_PREVIEW parameter:

```
DoCmd OpenReport "MyReport", A_PREVIEW
```

In Addition, you can open a report in Design mode by using the A_DESIGN parameter:

```
DoCmd OpenReport "MyReport", A_DESIGN
```

[References](#)

PRB: Text Label Truncated Printing Form on Epson 9-Pin Printer

Article Number: Q96100
CREATED: 08-MAR-1993
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When print a [form](#) or a [report](#) to an Epson 9-pin printer, the last two characters of the [label](#) controls are truncated. The label controls appear correctly in Print Preview.

CAUSE

This problem is caused by the difference in the sizes of the Epson printer driver fonts and the screen fonts.

RESOLUTION

Before adding any controls to your form or report, set the Layout for Print property to Yes.

MORE INFORMATION

Perform the following three steps to layout your form or report correctly.

1. Open the form or report in [Design view](#).
2. Choose the Properties button or choose Properties from the View menu.
3. Change the Layout for Print property to Yes.
4. Place fields into the form or report as desired. The label controls display and print correctly.

References:

For more information, [query](#) in the Microsoft Knowledge Base on the following words:

screen and fonts and printer and true and type

[References](#)

PRB: Incorrect Catalog Stored Procedures Error Message

Article Number: Q96013
CREATED: 03-MAR-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The following error message may be displayed when you attempt to attach a [SQL table](#) in Microsoft Access:

```
[Microsoft][ODBC SQL Server Driver] The ODBC catalog stored
procedures installed on your server <servername> are version
xx.xxxx; version xx.xx.xxxx is required to ensure proper
operation. Please contact your system administrator.
```

where <servername> indicates the name of your server.

CAUSE

To perform its functions, Microsoft Access requires certain information from a back-end [database](#). This information is provided by a series of catalog stored procedures that reside on the back-end database. You need to use the correct version of the catalog procedures for the version of Microsoft Access and SQL Server that you are running.

This error indicates that you have an incorrect version of the catalog stored procedures installed.

RESOLUTION

Reinstall the stored procedure file. Refer to the following paragraphs to determine which script file to use with your version of Microsoft Access and SQL Server.

Microsoft Access Version 1.0

If you run Microsoft Access version 1.0 with Microsoft or Sybase SQL Server, use the correct script file from the table below:

SQL Server And Version	Catalog Stored Procedure Script File Requirements
---------------------------	--

Microsoft SQL Server versions 1.x and 4.2	Use the INSTCAT.SQL file from the Microsoft Access ODBC disk.
--	--

Microsoft SQL Server version 4.2a	INSTCAT.SQL is not required, but some Microsoft Access problems are resolved with
--------------------------------------	--

it that are not in the Microsoft SQL Server 4.2a stored procedures.

Sybase SQL Server versions before 4.8 Use the INSTCAT.SQL file from the Microsoft Access ODBC disk.

Sybase SQL Server versions 4.8-4.9x Use the INSTCAT.48 file from the MSACCESS forum on CompuServe. Go to the ODBC/SQL Server library, and download the SYBASE.ZIP file.

Microsoft Access Version 1.1

In Microsoft Access version 1.1, both the Sybase and Microsoft SQL servers use the same INSTCAT.SQL file.

To install the catalog stored procedures, use the SQL Server ISQL (Interactive SQL) tool. You can run the ISQL tool from the MS-DOS or OS/2 [command prompt](#).

IMPORTANT: Do not use the SAF utility provided with Microsoft SQL Server. Microsoft SAF for MS-DOS and OS/2 is limited to 511 lines of code in an SQL script.

The [syntax](#) to install INSTCAT.SQL using ISQL is:

```
isql /U <sa loginname> /n /P <password> /S <SQL servername>
/i <drive:\path\INSTCAT.SQL> /o <drive:\path\outputfilename>
```

NOTE: Enter the preceding two lines as one line at the command prompt, and do not include the angle brackets < >.

```
/U Login name for the system administrator.
/n Eliminates line numbering and prompting for user input.
/P Password (case sensitive) used for the system administrator.
/S Name of the driver to set up.
/i Provides the drive and fully qualified path for INSTCAT.SQL.
/o Provides ISQL with an output file destination for results,
including errors.
```

Example

```
isql /U sa /n /P skier /S DUMMY_SERVER /i D:\SQL\INSTCAT.SQL /o
D:\SQL\OUTPUT.TXT
```

[References](#)

INF: Security Settings for Modifying Data in a Query Dynaset

Article Number: Q96012
CREATED: 03-MAR-1993
MODIFIED: 06-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

Summary:

You can affect a user's ability to edit data in a [query's dynaset](#) by altering the settings for the following two [permissions](#):

- Run with Owner's Permissions, for a query
- Modify Data, for a query and any underlying tables

More Information:

For a user to be able to edit a query's dynaset, one of the following must be true:

- The Run With Owner's Permissions [check box](#) must be selected. (Open the query in [Design view](#). From the View menu, choose Query Properties.)
- The Modify Data check box must be selected for both the query and the underlying tables. (From the Security menu in the Database window, choose Permissions. You must repeat this step for each object for which you want to assign the user permissions.)

The following [table](#) shows the possible combinations of these three settings and how each affects the customer's ability to edit the query's dynaset:

Permission to Modify Data in Table	Permission to Modify Data in Query	Run with Owner's Permission	Edit Query?
-----	-----	-----	-----
Yes	No	No	No
Yes	Yes	No	Yes
No	No	No	No
No	Yes	No	No
Yes	No	Yes	No
Yes	Yes	Yes	Yes
No	No	Yes	No
No	Yes	Yes	Yes

References:

"Microsoft Access User's Guide," version 1.0, pages 619-626

[References](#)

PRB: No System Table Permissions After Admin User Deleted

Article Number: Q96011
CREATED: 03-MAR-1993
MODIFIED: 28-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Your attempt to access the system tables MSysACES, MSysColumns, or MSysIndexes fails.

CAUSE

You deleted the Admin account from the list of valid users.

RESOLUTION

Before you delete the Admin account, you must assign [permissions](#) for the system tables to a new member of the Admins [group](#).

STATUS

This behavior is by design.

MORE INFORMATION

The text below explains how to assign permission to access the system tables to a new member of the Admins group.

If you create a new Admin account and delete the original Admin account from the Users group before you assign the new account access rights, you are unable to access the system tables. The one exception to this rule applies if the original Admin did not create the [database](#).

To assign permissions to the system tables, use the following steps:

1. Log on to Microsoft Access as the user Admin. If you do not receive a logon screen, you are automatically logged in as Admin.
2. From the Security menu, choose Users.
3. In the Users [dialog box](#), choose New.
4. Enter the name of the new administrator, assign the administrator a [PIN](#) (personal identification number), and choose OK.
5. In the Group Membership section of the Users dialog box, add the new administrator to the group Admins. To do so, select Admins in the Available Groups list and choose Add. Then choose Close.

6. From the View menu, choose Options.
7. In the Items box in the General category, set Show System Objects to Yes and choose OK.
8. From the Security menu, choose Permissions.
9. In the Object section of the Permissions dialog box, select the MSysACES table. In the Permissions section, select the Full Permissions check box. Microsoft Access sets all other permissions automatically.

Repeat step 9 for the MSysColumns and MSysIndexes tables.

REFERENCES

=====

"User's Guide," version 1.0, pages 616-623

For more information, search for "permissions," then "Adding Users to Groups," and "Assigning Permissions to Users" using the Help menu.

[References](#)

INF: Changing Area Codes Based on a Telephone Number Prefix

Article Number: Q96010
CREATED: 03-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Some states are adding new telephone area codes based on a phone number's prefix. This article describes how to change a phone number's area code in a [table](#) based on the prefix. This is accomplished by creating a table with the prefix to find and the new area code. An [Access Basic function](#) then reads this table into an array and searches the phone number table and changes the area code in the phone number table.

Before running this function, back up your [database](#) and make a copy of your table.

This article assumes that you are familiar with Access Basic and how to create and use Access Basic procedures and Microsoft Access tables.

MORE INFORMATION

1. Create a table that will contain the phone number prefix to search and the new area code:

Table: Area Codes To Change

Field Name: Prefix
Datatype: Text
Field Name: Area Codes
Datatype: Text

2. Enter in the phone number prefixes and area codes in the table Area Codes To Change. For example:

Prefix	Area Codes
635	207
634	208

3. Open a [module](#) or create a new one, and enter the following code:

Note: In the following sample code, an underscore (`_`) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declarations section of the module.  
'*****
```

Option Explicit

```
'=====
' Create the following ChgAreaCode() function in the Module
' The following function will initialize:
' - A variable for the database object.
' - Variables for the table objects and several other objects.
' - Sets the database object to the opened database.
' - Opens table "Area Codes To Change" and tPhName.
' - The action of this function is to locate the phone number
'   entries with certain prefixes and then change the
'   corresponding area code. This function operates on numbers
'   with the following format "(206) 635-7050". The table name
'   and field name containing the phone number must be passed
'   as arguments.
'=====
Function ChgAreaCode (tPhName, fldPhone)
    Dim PhoneDB As Database
    Dim tPhone As Table, tPrefix As Table
    Dim PCount%
    Dim i%
    Dim tPrefixName$
    Dim Prefix$
    Dim SpacePos%
    Dim HyphenPos%
    Dim PrefixLen%
    Dim PrefixToFind$
    Dim AreaCode$
    Dim Lastfour$
    tPrefixName$ = "Area Codes To Change"

    If tPhName = "" Or fldPhone = "" Then Exit Function

    Set PhoneDB = CurrentDB()
    Set tPrefix = PhoneDB.OpenTable(tPrefixName$, 1)

    tPrefix.MoveLast
    PCount% = tPrefix.RecordCount
    tPrefix.MoveFirst
    ReDim PrefixArray$((PCount% - 1), 1)

    For i% = 0 To PCount% - 1 Step 1
        PrefixArray$(i%, 0) = tPrefix.Prefix
        PrefixArray$(i%, 1) = tPrefix.[Area Codes]
        tPrefix.MoveNext
    Next i%
    tPrefix.MoveFirst
    tPrefix.Close

    Set tPhone = PhoneDB.OpenTable(tPhName, 1)
    Do Until tPhone.EOF
        SpacePos% = InStr(1, tPhone(fldPhone), " ")
        HyphenPos% = InStr(SpacePos% + 1, tPhone(fldPhone), "-")
        PrefixLen% = (HyphenPos% - SpacePos%) - 1
        PrefixToFind$ = Mid(tPhone(fldPhone), SpacePos% + _
            1, PrefixLen%)
```

```

For i% = 0 To PCount% - 1 Step 1
  If PrefixArray$(i%, 0) = PrefixToFind$ Then
    AreaCode$ = PrefixArray$(i%, 1)
    Prefix$ = Mid$(tPhone(fldPhone), 7, 3)
    Lastfour$ = Right$(tPhone(fldPhone), 4)
    tPhone.Edit
    tPhone(fldPhone) = "(" & AreaCode & ")" "
                        & Prefix & "-" & Lastfour
    tPhone.Update
  End If
Next i%

tPhone.MoveNext
Loop
tPhone.Close
PhoneDB.Close
End Function

```

When using this function you pass (tPhName), the name of the table containing the number to change as a string. Then you pass (fldPhone), the field name that contains the number to change as a string.

Example

This function can be called from the Immediate window for one time execution as follows:

```
?ChgAreaCode("Contacts", "Business Phone")
```

(Where "Contacts" is a table name and "Business Phone" is a field containing the phone number.)

References:

"Microsoft Access Basic: An Introduction to Programming," version 1.0

[References](#)

INF: Screen Fonts May Not Match Printer Output Fonts

Article Number: Q96009
CREATED: 03-MAR-1993
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Screen fonts and printer fonts can be different. When you create a new [form](#) or [report](#), the default font used to format the controls can be set up to use only printer fonts. Using printer fonts will ensure that the screen view and printed output are as close as can be to an exact match. If you use TrueType fonts (supplied with Windows 3.1) the screen view and printed output will also match. Forms that have been saved as reports may need to be modified to ensure that you get the correct output. By default, new reports use printer fonts but forms do not. You can [control](#) which fonts are used by setting the `LayoutForPrint` property in the form or report [property sheet](#).

MORE INFORMATION

When you choose a font for a form or report, the characters on the screen may not look exactly like those that appear on the printed page because printer fonts and screen fonts can differ. If you select a scalable font, such as a TrueType font, the screen and printer characters will be the same.

When you design a form or report, you can set the `LayoutForPrint` property to Yes in the form or report property sheet, and you will only see printer fonts listed in the [toolbar](#). The `LayoutForPrint` property determines whether a screen font or printer font is used. If `LayoutForPrint` is set to No, the fonts available on the toolbar include screen fonts.

If `LayoutForPrint` is set to Yes, the fonts and point sizes on the toolbar correspond to the fonts and sizes available on the printer you chose in the Print Setup command from the File menu.

For reports, this property is set to Yes by default. This property must be changed to Yes for forms.

In Microsoft Access, you can design a form and save it as a report. If `LayoutForPrint` is set to No when the form is originally designed, you may need to change the fonts from a screen font to a TrueType or printer font in the report design mode.

To convert a form to a report, open the form in [Design view](#) and choose Save As Report from the File menu.

References:

"Microsoft Access User's Guide," version 1.0, page 309

[References](#)

INF: Criteria Parameters Require Concatenated References

Article Number: Q95978
CREATED: 03-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access has both a [SQL](#) interpreter and an [Access Basic](#) interpreter. Because some Access Basic commands and functions require SQL [syntax](#) as a parameter, it may become confusing how the two work together.

For example, if you need to look up a description in a [table](#) called Categories where [Category ID] is equal to whatever value is currently in Forms!Form1!Field1, you cannot use:

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
DLookup("Description", "Categories", _  
    "[Category ID] = Forms!Form1!Field1")
```

Instead, you must concatenate the value of Forms!Form1!Field1 into the [criteria](#) parameter as shown below:

```
DLookup("Description", "Categories", _  
    "[Category ID = '" & Forms!Form1!Field1 & "'")
```

This means that if the value of Forms!Form1!Field1 is "BEVR," the criteria parameter in the [expression](#) above would evaluate to:

```
[Category ID] = 'BEVR'
```

MORE INFORMATION

When you make a [function](#) call such as:

```
DLookup("Description", "Categories", "[Category ID] = 'BEVR'")
```

Access Basic internally creates a SQL statement with the parameters you supply. In this case, the resulting SQL statement is:

```
SELECT Description FROM Categories WHERE [Category ID] = 'BEVR'
```

Access Basic sends this statement to the Access SQL engine, which returns the result of the SELECT statement back to Access Basic, and, in turn, back to your DLookup() expression.

Note that that a DLookup() coded as follows

```
DLookup("Description", "Categories", _  
    "[Category ID] = Forms!Form1!Field1")
```

would result in a [WHERE clause](#) that looks like:

```
[Category ID] = Forms!Form1!Field1
```

The SQL parser is designed to accept SQL syntax, and the Forms!Form1!Field1 reference is Access Basic syntax, so the [form](#) reference cannot be expected to work. Instead, the resulting evaluated criteria parameter must be done in such a way as to accommodate the syntax rules of the SQL parser.

Note that this also applies to other Access Basic methods and functions that require a criteria parameter. For example, Find methods use a criteria parameter as shown below:

```
Dim D As Database, S As Snapshot  
Set D = CurrentDB()  
Set S = D.CreateSnapshot("Categories")  
myvar = "BEVR"  
S.FindFirst "[Category ID] = '" & myvar & "'"
```

[References](#)

PRB: DateDiff 'W' Option Does Not Work

Article Number: Q95977
CREATED: 03-MAR-1993
MODIFIED: 20-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When calculating the number of weekdays between two dates, using the "w" Weekday option, DateDiff returns the number of weeks, not the number of work days. The "w" option is supposed to work the same as "d" for DateDiff. It is provided as an option for compatibility with the DatePart [function](#).

RESOLUTION

If you are using the DateDiff function to return the number of days, substitute "d" for "w". You can use the [Access Basic](#) code in this article to return the number of work days rather than the number of days.

MORE INFORMATION

The code below shows how to calculate the number of work days in a date range. To understand the following information, you need to be familiar with Access Basic, know how to create and use Access Basic procedures, and know how to use the [Immediate window](#).

Steps to Reproduce Behavior

In the Immediate window, type:

```
? DateDiff("W",#2/2/93#, #2/18/93#)
```

The result is 2 (the number of weeks), not 16 (the number of days) or 12 (the number of work days).

The following code provides a function, DateDiffW(), that calculates the number of work days between two dates.

NOTE: In the following sample code, an underscore () is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declaration section of the module  
'*****  
Option Explicit
```

```
Function DateDiffW(BegDate, EndDate)
```

```

Const SUNDAY = 1
Const SATURDAY = 7
Dim NumWeeks As Integer

If BegDate > EndDate Then
    DateDiffW= 0
Else
    Select Case Weekday(BegDate)
        Case SUNDAY : BegDate = BegDate + 1
        Case SATURDAY : BegDate = BegDate + 2
    End Select
    Select Case Weekday(EndDate)
        Case SUNDAY : EndDate = EndDate - 2
        Case SATURDAY : EndDate = EndDate - 1
    End Select
    NumWeeks = DateDiff("ww", BegDate, EndDate)
    If NumWeeks = 0 Then
        DateDiffW= Weekday(EndDate) - Weekday(BegDate) + 1
    Else
        DateDiffW= NumWeeks * 5 + _
            (Weekday(EndDate) - Weekday(BegDate) + 1)
    End If
End If
End Function

```

How to Use the DateDiffW() Function

Use the DateDiffW() function wherever you would use DateDiff().
Instead of

```
DateDiff("W",[StartDate],[EndDate])
```

use the following:

```
DateDiffW([StartDate],[EndDate])
```

Steps to Test the DateDiffW() Function

1. In the Immediate window, type:

```
?DateDiffW(#2/2/93#,#2/18/93#)
```

2. The result is 13, the number of work days.

References:

"Access Basic Language Reference," version 1.0, Part 2, "A-Z Reference" pages 102-107 and 492-493.

[References](#)

INF: How to Determine Whether a Menu Item Is Selected

Article Number: Q95935
CREATED: 01-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

Summary:

This article discusses how you can make application programming interface (API) calls to determine whether or not a menu item is checked (selected). This [method](#) can be used, for example, when you want to determine whether a [form](#) is in [Form view](#) or [Datasheet view](#). If a form is in Datasheet view, the Datasheet menu item on the View menu is selected. Thus, determining whether this menu item is checked allows you to determine whether the form is in Datasheet view.

The program code in this article comes from the ORDENTRY.MDB sample [database](#) and is presented in this article in a condensed version.

More Information:

To determine whether a menu item is selected, you can use the IsMenuChecked() [function](#) listed in the code at the end of this article. IsMenuChecked() accepts two parameters indicating the position of the desired menu item; IsMenuChecked() returns true if the item is selected and false if it isn't.

To find the position of the menu item you want to select, begin at 0 (zero) and count from the left-most menu item until you reach the item you want. This determines the first number. Then, from the selected menu, begin with zero and count down the [submenu](#) until you reach the desired submenu item. This determines the second number.

For example, suppose you have the following menu:

```
File   Edit   View
-----
Open   Cut     Form Design
Close  Copy    Form
Exit   Paste   Datasheet
```

To determine whether Datasheet is selected, you would call IsMenuChecked() with the values 2 and 2, because the View menu is in position 2 and the Datasheet item on the View submenu is in position 2. Note the same menu with numbered rows and columns to illustrate this further:

```
      0      1      2
File   Edit   View
-----
```

```

0  Open      Cut      Form Design
1  Close     Copy     Form
2  Exit      Paste    Datasheet

```

When determining the position of a menu command, [separator](#) bars count as commands. For example, to get to the Import menu command on the File menu in the [Database window](#), use the arguments 0 (for the File menu) and 7 (for the Import command). Positions 3 and 6 correspond to the separator bars on the File menu.

For an example of how to use this function, follow these steps in the sample database NWIND.MDB:

1. Copy the code at the bottom of this article to a new [module](#).
2. Open the Customers form in [Design view](#).
3. Click in the CustomerID [text box](#) to give it the focus.
4. Specify the following for the OnDblClick property in the property sheet:

```
OnDblClick: =MsgBox(Str(IsMenuChecked(2,2)))
```

5. Switch to Form view.

Notice that Datasheet on the View menu is not checked because the form is not in Datasheet view.

6. Double-click the CustomerID text box.

A zero appears indicating that the Datasheet item on the View menu is not checked.

7. Switch to Datasheet view.

Notice that the Datasheet item on the View menu is now checked because the form is in Datasheet view.

8. Double-click the CustomerID text box.

This time, a -1 appears indicating that the Datasheet item on the View menu is checked.

The following is a program listing:

```
Option Compare Database      'Use database order for string comparisons.
Option Explicit
```

```

Declare Function IsZoomed% Lib "User" (ByVal hWnd%)
Declare Function GetMenu% Lib "User" (ByVal hWnd%)
Declare Function GetSubMenu% Lib "User" (ByVal hMenu%, ByVal nPos%)
Declare Function GetMenuState% Lib "User" (ByVal hMenu%, ByVal idItem%,
      ByVal fuFlags%)
Declare Function GetActiveWindow% Lib "User" ()
Declare Function GetParent% Lib "User" (ByVal hwin%)

```

```
Declare Function GetClassName% Lib "User" (ByVal hwin%, ByVal stBuf$,  
    ByVal cch%)
```

```
Const WU_MF_BYPOSITION = &H400  
Const WU_MF_CHECKED = &H8  
' OMain is the predefined classname of the  
Global Const WU_WC_ACCESS = "OMain"
```

```
Function IsMenuChecked (iMenu%, iItem%) As Integer  
    Dim hMainMenu%  
    Dim hMenu%  
    Dim Flags%  
  
    '-----  
    'If the window is maximized, there is an additional  
    'system menu added to the MDI window.  
    '-----  
    If (IsZoomed(Screen.ActiveForm.hWnd)) Then  
        iMenu% = iMenu% + 1  
    End If  
  
    '-----  
    ' Get the appropriate handle to the menu item.  
    '-----  
    hMainMenu% = GetMenu(GetAccessHwnd())  
    hMenu% = GetSubMenu(hMainMenu%, iMenu%)  
    Flags% = WU_MF_BYPOSITION Or WU_MF_CHECKED  
  
    '-----  
    ' Call the API that returns the state of the menu.  
    '-----  
    IsMenuChecked = (GetMenuState(hMenu%, iItem%, Flags%) <> 0)  
End Function
```

```
Function StWindowClass (hWnd As Integer) As String  
    Const cchMax = 255  
    Dim cch%  
    Dim stBuff As String * cchMax  
  
    '-----  
    ' Get the class name of the window relating to hWnd  
    ' and return this as the function.  
    '-----  
    cch% = GetClassName(hWnd, stBuff, cchMax)  
    If (hWnd% = 0) Then  
        StWindowClass = ""  
    Else  
        StWindowClass = (Left$(stBuff, cch%))  
    End If  
End Function
```

```
Function GetAccessHwnd () As Integer  
    Dim hWnd%  
  
    '-----  
    ' Keep getting the handle of the parent, until the  
    ' ClassName = "OMain", which is the class name of
```

```
' the Microsoft Access window.  
'-----  
hWnd% = GetActiveWindow()  
While ((StWindowClass(hWnd%) <> WU_WC_ACCESS) And (hWnd% <> 0))  
    hWnd% = GetParent(hWnd%)  
Wend  
GetAccessHwnd = hWnd%  
End Function
```

For more information, [query](#) on the following word in the Microsoft Knowledge Base:

GetSubMenu

[References](#)

INF: How to Position Microsoft Access on the Screen

Article Number: Q95934
CREATED: 01-MAR-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

To position Microsoft Access on the screen, you need to call the Windows SetWindowPos API [function](#). The article gives a brief demonstration of how to position Microsoft Access on the screen.

More Information:

To position Microsoft Access on the screen, you will need to use the SetWindowPos() application programming interface (API) function included in the USER.EXE dynamic [link](#) library (DLL) included with Windows. Implement this function as follows:

1. Open or create a new [module](#) in Microsoft Access.
2. Within the [Declarations section](#), add the following declares, constants, and comments.

Note: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
'=====
' Global Declarations
'=====
Option Explicit

Declare Function FindWindow% Lib "user" (ByVal lpClassName As Any, _
    ByVal lpCaption As Any)
Declare Sub SetWindowPos Lib "User" (ByVal hWnd%, _
    ByVal hWndInsertAfter%, ByVal X%, ByVal Y%, _
    ByVal cx%, ByVal cy%, ByVal wFlags%)

'Inserts the window to precede the position window in the Z-Order.
'This parameter must be a window handle or one of the following
'constants.

Global Const HWND_TOP = 0
Global Const HWND_BOTTOM = 1
Global Const HWND_TOPMOST = -1
Global Const HWND_NOTOPMOST = -2

'values for wFlags%

Global Const SWP_HIDEWINDOW = &H80 'Hide the window
```

```
Global Const SWP_SHOWWINDOW = &H40 'Show the window
Global Const SWP_NOMOVE = &H2 'Ignores the x, y values
Global Const SWP_NOZORDER = &H4 'Ignores the hWndInsertAfter
Global Const SWP_NOACTIVATE = &H10 'Does not activate the form
```

3. Include the following function:

```
Function SizeAccess ()
    Dim cX%, cY%, cHeight%, cWidth%
    'get handle to Microsoft Access.
    h% = FindWindow("Omain", 0&)

    cX% = 0: cY% = 0
    cWidth% = 640, cHeight% = 480

    'position Microsoft Access
    SetWindowPos h%, HWND_TOP, cX%, cY%, cWidth%, cHeight%, _
        SWP_SHOWWINDOW
End Function
```

[References](#)

INF: Determining the Version of Catalog Stored Procedures

Article Number: Q95933
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to find the version of catalog stored procedures on a Microsoft [SQL](#) Server.

MORE INFORMATION

There are situations in which you want to determine the version of a stored procedure, such as when you are experiencing difficulties attaching to SQL Server. The first [method](#) does not use Microsoft Access.

Method 1

Using the System Administration Facility (SAF), you need to log onto SQL Server with the user's login ID and password (using the SQL administrator's login id (SA) is ideal but is not always readily available). Once logged on in SAF, the user needs to open the [query](#) window and issue the command USE MASTER. This tells it to use the Master [database](#) that contains the system tables. To determine the version, have the user enter the following SQL command:

```
SELECT Attribute_Value FROM MsServer_Info WHERE Attribute_ID = 500
```

Method 2

In a situation where you can successfully attach to SQL Server data, you can attach the MsServer_Info [table](#) from the Master database while using Microsoft Access, then run a query on it for the [record](#) where the Attribute_ID is equal to 500.

[References](#)

INF: How to Change the 'Microsoft Access' Window Caption

Article Number: Q95932
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There is no Microsoft Access property to set the main caption of the Microsoft Access window. To change the default "Microsoft Access" caption to your own text, you must call the Windows application programming interface (API) SetWindowText() [function](#).

MORE INFORMATION

To change the default "Microsoft Access" caption to your own text [string](#), create a new [module](#) in Microsoft Access and add the following function below with the appropriate declaration section.

You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error, remove the Declare statement from your code or comment it out.

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
'-----  
' GLOBAL DECLARATION  
'-----  
Option Compare Database  
Option Explicit  
Declare Function FindWindow% Lib "User" (ByVal lpClassName As Any, _  
                                     ByVal lpWindowName As Any)  
Declare Sub SetWindowText Lib "User" (ByVal hWnd%, ByVal lpString$)  
  
'-----  
' Function: SetCaption ()  
' This function will set the caption of  
' Microsoft Access's main window.  
'-----  
Function SetCaption ()  
    Dim hWnd%  
    hWnd% = FindWindow%("OMain", 0&)  
    Call SetWindowText(hWnd%, "This is cool!")  
End Function
```

To change the caption when Microsoft Access is started, you can execute the RunCode [macro](#) action with SetCaption() as an argument and save the macro as Autoexec. When you open this [database](#) up, this macro

will execute and change the "Microsoft Access" caption to "This is cool!"

References:

"Microsoft Windows Software Development Kit Programmer's Reference Volume 2: Functions," version 3.1

[References](#)

INF: How to Create a Query By Form

Article Number: Q95931
CREATED: 01-MAR-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

This article describes an approach to creating a query by form (QBF), which is common in many end-user applications. A QBF presents the user with several blank text boxes, each labeled with the name of the field in the table, then allows the user to enter search criteria into these text boxes, and performs a search only on those fields that have search criteria specified.

MORE INFORMATION

For example, a form might appear with the following text boxes:

First Name: _____
Last Name: _____
City: _____
State: _____
Zip Code: _____

The user could then enter any combination of criteria, such as a value for State and Last Name. The user would then click a button and get a set of records where the State and Last Name match their entries.

This can be accomplished by creating a parameter query. Each parameter in the query checks for a value on the form. If the value is there, the parameter returns the value; otherwise, the parameter returns the current value of the field so that every record satisfies the parameter value.

To illustrate this in the sample database Northwind Traders (NWIND.MDB), follow these steps:

1. Create a new query based on the Categories table and add the two fields to the query grid.

Query: Sample

Field: Category ID

Criteria: IIf(IsNull([Prompt1]), [Category ID], [Prompt1])

Field: Category Name

Criteria: IIf(IsNull([Prompt2]), [Category Name], [Prompt2])

2. Run the query.

Notice that the two parameter prompts appear. If you place a value in

the first prompt, such as BEVR, and leave the second prompt blank, only the first prompt is filtered. If you leave both blank, no filtering takes place and all records appear.

Note that these parameters do not have to be prompts. They can be references to text boxes on a form. For example, the parameters can be changed to:

NOTE: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
Query: Sample
-----
Field: Category ID
      Criteria: IIf(IsNull([Forms]![QueryForm]![Prompt1]),_
                  [Category ID], [Forms]![QueryForm]![Prompt1])
Field: Category Name
      Criteria: IIf(IsNull([Forms]![QueryForm]![Prompt2]),_
                  [Category Name], [Forms]![QueryForm]![Prompt2])
```

This would read the parameter from text boxes called Prompt1 and Prompt2 on a form called QueryForm. With this approach, the form can be based on the query described above, and a button can be added that runs a [macro](#) as shown below:

```
Action
-----
Requery
      Controlname: [nothing]
```

Every time the macro is executed, the parameters on the query [filter](#) the dataset according to any entries on the form. The text boxes and the button that runs the macro can be in the [header](#) or [footer](#) of the form, and the data itself can be located in the detail section.

[References](#)

PRB: DateAdd 'w' Option Doesn't Work As Expected

Article Number: Q95930
CREATED: 01-MAR-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When adding days using the "w" weekday option, the DateAdd() [function](#) adds the number of days instead of the number of work days.

CAUSE

"W" works the same as "d" for DateAdd(). It is provided as an option for compatibility with the DatePart() function. Using "w" with DatePart() returns the day of the week (Sunday-Saturday), not whether the day is a work day.

RESOLUTION

The functionality as described above is by design. You can use the [Access Basic](#) code in this article to return the number of work days rather than the number of days.

MORE INFORMATION

To understand the following information, you need to be familiar with Access Basic, how to create and use Access Basic procedures, and how to use the [Immediate window](#).

To reproduce this behavior, enter

```
? DateAdd("w",#2/2/93#,10)
```

in the Immediate window. The result is #2/12/93# (10 days), not #2/16/93# (10 work days).

The following code provides a function, DateAddW(), that will add work days to another date.

Note: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declarations section of the module  
'*****  
    Option Explicit  
'=====
```

' Create the following DateAddW() function in the module.

```

' This function provides a true work day substitute
'   for DateAdd("w", TheDate, Interval).
' This function performs error checking and ignores fractional
'   Interval values.
'=====
Function DateAddW (ByVal TheDate, ByVal Interval)

    Dim Weeks As Long, OddDays As Long, Temp As String

    If VarType(TheDate) <> 7 Or VarType(Interval) < 2 Or _
        VarType(Interval) > 5 Then
        DateAddW = TheDate
    ElseIf Interval = 0 Then
        DateAddW = TheDate
    ElseIf Interval > 0 Then
        Interval = Int(Interval)

'        Make sure TheDate is a work day (Round Down)
        Temp = Format(TheDate, "ddd")
        If Temp = "Sun" Then
            TheDate = TheDate - 2
        ElseIf Temp = "Sat" Then
            TheDate = TheDate - 1
        End If

'        Calculate Weeks and Odd Days
        Weeks = Int(Interval / 5)
        OddDays = Interval - (Weeks * 5)
        TheDate = TheDate + (Weeks * 7)

'        Take OddDays week-end into account
        If (DatePart("w", TheDate) + OddDays) > 6 Then
            TheDate = TheDate + OddDays + 2
        Else
            TheDate = TheDate + OddDays
        End If

        DateAddW = TheDate
    Else
        Interval = Int(-Interval) 'Interval is < 0
        'Make positive & subtract later

'        Make sure TheDate is a work day (Round Up)
        Temp = Format(TheDate, "ddd")
        If Temp = "Sun" Then
            TheDate = TheDate + 1
        ElseIf Temp = "Sat" Then
            TheDate = TheDate + 2
        End If

'        Calculate Weeks and Odd Days
        Weeks = Int(Interval / 5)
        OddDays = Interval - (Weeks * 5)
        TheDate = TheDate - (Weeks * 7)

'        Take OddDays week-end into account
        If (DatePart("w", TheDate) - OddDays) < 2 Then
            TheDate = TheDate - OddDays - 2

```

```
Else
    TheDate = TheDate - OddDays
End If

DateAddW = TheDate
End If

End Function
```

How to Use the DateAddW() Function

Use the DateAddW() function wherever you would use DateAdd. For example, instead of

```
DateAdd("w",[StartDate],10)
```

use:

```
DateAddW([StartDate],10)
```

To test the DateAddW() function, enter

```
? DateAddW(#2/2/93#,10)
```

in the Immediate window. The result is #2/16/93# (10 work days).

References:

Access "Language Reference," version 1.0, pages 102-107 and 492-493

[References](#)

PRB: Error in Database Analyzer: DTSysQuery Already Exists

Article Number: Q95929
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When using the Database Analyzer, you click the Analyze button and receive this error message:

DTSysQuery already exists

CAUSE

The Database Analyzer often creates a temporary [query](#) called DTSysQuery to complete certain tasks. When the Database Analyzer finishes using DTSysQuery or needs to create one, it first deletes the current copy. If it can't delete DTSysQuery, it attempts to create DTSysQuery anyway because it assumes DTSysQuery doesn't exist. As a result, an error occurs in the Database Analyzer and the error message you see is the result of a generic error trap mechanism that forwards the [Access Basic](#) error to the user.

One reason why the Database Analyzer fails to delete the existing copy of DTSysQuery is because the Database Analyzer is installed on a drive or directory that has [read-only permissions](#).

RESOLUTION

To prevent the error from occurring, you must follow these steps:

1. Open MSACCESS.INI in Notepad.
2. Locate the section called [Libraries].
3. Place a semicolon in front of the entry for ANALYZER.MDA.
4. Close and save MSACCESS.INI. Then restart Microsoft Access.
5. Open the [database](#) called ANALYZER.MDA.
6. In the database window, delete the DTSysQuery query.
7. Repeat steps 1 and 2. Then remove the semicolon from the front of the entry indicated in step 3.
8. Restart Access and try the Database Analyzer again.

If the Database Analyzer is located on a drive or directory that does not have modify permissions, you may not be able to complete step 6

successfully. Even if you can, the problem will likely recur. If it does, move the Database Analyzer to a drive or directory that has modify permissions.

[References](#)

INF: Microsoft Access Does Not Use Record Numbers

Article Number: Q95928
CREATED: 01-MAR-1993
MODIFIED: 16-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article discusses the reasons why Microsoft Access does not use [record](#) numbers, as do some other [database](#) products.

MORE INFORMATION

Many database products, such as Microsoft FoxPro, dBASE IV, and Paradox, use a record number that is physically associated with the record. For example, the first record in a [table](#) has record number 1, the second record has record number 2, and so on. The record numbers remain part of the record as long as the record exists. The first record entered into a table is always record number 1, even if the table is sorted and the record appears somewhere other than the first position in the table.

Microsoft Access does not use this approach. When you open a table in [Datasheet view](#), a number appears at the bottom of the [form](#). This number indicates the position of the record in the recordset, not the record's natural position in the table. For example, if a record with the key value "Anderson" is the first record in a recordset, the number 1 appears at the bottom of the form. However, if you sort the same records on a different key, the record may no longer appear as the first record and a different number may appear at the bottom of the form.

When you open a table, form, or [report](#), Microsoft Access creates a [dynaset](#), a dynamic copy of the appropriate recordset, and assigns numbers to the records therein. These numbers are used only by the user interface to provide a visually unique identifier for each record.

Microsoft Access uses this approach because assigning a physical record number to each record is not "relationally correct." According to [relational database](#) theory, data may appear in any order at any time. Therefore, the key value, not a random record number, provides the only reliable [method](#) to identify a record.

Because no record number is available, Microsoft Access provides another method to identify records in a dynaset. Using the [Bookmark](#) property, you can write a procedure to find a target record, store its bookmark value in a [variable](#), move to other records, and return to the original record. For more information about bookmarks, search on "Bookmark" using the Help menu.

dBASE IV is manufactured by Borland International, Inc. and Paradox is manufactured by Ansa Software, a Borland company, vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

INF: Currency Format Causes Report Alignment Problems

Article Number: Q95927
CREATED: 01-MAR-1993
MODIFIED: 13-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

In a [report](#), if you format numbers with a Currency format, you may have difficulty aligning the formatted numbers with numbers that are not formatted. For example, if you format totals in a detail section to Currency but do not format the numbers above or below the total, the unformatted numbers are not aligned with the formatted numbers.

MORE INFORMATION

To correct this problem, rather than selecting the Currency format for the [control](#) totals in your report, try using one of the Format functions listed in Microsoft Access Help. You can enter a Format [function](#) in the Format property in the [text box property sheet](#) for the appropriate control.

You can enter "\$#,##0.00" for a number when you want to display the dollar sign and use "#,##0.00" for a number when the dollar sign is not required (do not enter the quotation marks in either case). These examples format the numbers in your report to use the comma as the thousands [separator](#) and format each number to have two decimal places. Numbers formatted in this manner align correctly in the report.

For more information, please refer to the "Format, Format\$ Functions" topic in Microsoft Access Help. To do so, choose Index from the Microsoft Access Help menu, then choose Search. Enter "format" (without the quotation marks) and choose Show Topics. Select "Format, Format\$ Functions" in the topics [list box](#) and choose Go To.

References:

"Microsoft Access Language Reference," version 1.0, page 206

[References](#)

PRACC9303: Screen Redraw Problems When Reordering Groups

Article Number: Q95926
CREATED: 01-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

In a [report](#), when you change the order of two [group](#) levels and the headers or footers have different heights, Microsoft Access doesn't redraw the screen completely, making further editing of the sections difficult.

RESOLUTION

Close the report and reopen it to force Microsoft Access to redraw the screen correctly.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

The following Steps use the Employee Sales by Country Report from the NWIND.MDB sample [database](#). In this report, the Salesperson and Country footers do not have identical heights.

Steps to Reproduce Problem

In the sample database NWIND.MDB:

1. Open the Employee Sales by Country Report in [Design view](#).
2. Choose Sorting and Grouping... from the View menu to view the sorting and grouping levels.
3. Move Salesperson above Country so that the Report will group by Salesperson first, then Country.
4. Close the [Sorting and Grouping box](#).

Note that while the two group headers and footers changed places, Microsoft Access did not redraw the screen completely. As a result, you can't click the page [footer](#) because the Salesperson footer remains highlighted.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 19, "Sorting and Grouping Data," pages 450-464.

[References](#)

INF: OpenTable Action Argument Does Not List Attached Tables

Article Number: Q95925
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

One of the [macro](#) action arguments for the OpenTable action is Table Name. In the Action Arguments section of the [Macro window](#), the list of arguments for Table Name includes the tables in the [current database](#), but none of the attached tables are listed.

MORE INFORMATION

To use an [attached table](#) with the OpenTable macro action, you must manually type the attached [table](#) name. Or, you can drag the attached table from the [Database window](#) to the Table Name argument.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 21, "Macro Basics," pages 513-514

[References](#)

INF: How to Implement a Timer

Article Number: Q95924
CREATED: 01-MAR-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

There is no timer control in Microsoft Access. To implement a timer, you must create an Access Basic module that creates a timing sequence in your application.

MORE INFORMATION

To create a timer within your Microsoft Access application, you must call two functions. One function starts the sequence, and another function sets a global variable that tells the first function to stop performing the timing iteration.

In the first function, called StartTimer(), you execute an endless loop that performs the timing update and the DoEvents() function. The DoEvents() function instructs the Access Basic function to release the computer's CPU to other tasks, such as your form, other applications within Microsoft Windows, and most importantly the StopTimer() function. The StopTimer() function sets the global variable that the StartTimer() function checks to determine when to stop looping.

To create an example, follow the instructions below. Once these are done, open the form in Browse mode and press the Start Timer button. You should see the timing being performed by the updating in the text box. At any point you can stop the timing sequence by pressing the Stop Timer button.

1. Create the following controls on a form and set the appropriate properties:

Form: Form1

Control: Text Box

ControlName: Text1

Control: Command Button

Caption: Start Timer

OnPush: =StartTimer()

Control: Command Button

Caption: Stop Timer

```
OnPush: =StopTimer()
```

2. Create a new module and add the following code:

```
'-----  
'GLOBAL DECLARATIONS SECTION  
'-----  
Option Explicit  
Dim StopTheTimer      'if set to TRUE, then exit StartTimer()  
Const INTERVAL = 1    'determine how often in seconds to update  
                      'the screen  
  
'-----  
'This function will start the timer, update  
'the display with the correct time unit,  
'and look to see if the StopTheTimer is  
'set to TRUE so as to determine when to  
'stop the process.  
'-----  
Function StartTimer ()  
    Static InHere      'stop re-entrant code  
    Dim iLast, iNow, x%  
  
    'this stops the user from pressing the command button  
    'and re-entering this code again if it is already  
    'running.  
    If InHere = True Then Exit Function  
  
    iLast = Timer  
    Do  
        'StopTheTimer is set in StopTimer()  
        If StopTheTimer = True Then  
            StopTheTimer = False  
            Exit Function  
        End If  
  
        iNow = Timer  
  
        'need to update display only if the interval  
        'has been exceeded.  
        If (iNow - iLast) > INTERVAL Then  
            Forms!Form1!Field0 = iNow  
            iLast = iNow  
        End If  
  
        'release the CPU to other tasks, namely enable  
        'the user to enter the StopTimer() function.  
        x% = DoEvents()  
  
    Loop  
End Function  
  
'-----  
'This function will set the flag to let StartTimer()  
'stop execution.  
'-----  
Function StopTimer ()
```

```
    StopTheTimer = True  
End Function
```

[References](#)

INF: Making a Subform Uneditable and Unenterable

Article Number: Q95923
CREATED: 01-MAR-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

If a subform is uneditable, a user cannot make any changes to the subform. However, a user can give the focus to the subform by clicking it with the mouse. To prevent this from happening, you must make the subform unenterable. The text below describes the steps to do this. An uneditable and unenterable subform appears as a "label" for information.

More Information:

To make a subform uneditable and unenterable, use the following three steps:

1. Open the main form in Design view.
2. Click the mouse once on the subform object to display the Subform/Subreport property sheet.
3. Change the Subform/Subreport properties to the following:

Subform/Subreport

Enabled: No
Locked: Yes

References:

For more information, search for "locked," then "Enabled, Locked Properties" using the Help menu.

References

INF: How to Change Default Template for Forms and Reports

Article Number: Q95922
CREATED: 01-MAR-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access uses templates to define the characteristics of new forms and reports when AccessWizards are not used. The templates contain information about the default properties for the sections and controls the new [form](#) (or [report](#)) will contain.

There are two templates: a Normal Form template and a Normal Report template. You can change the defaults used in these templates by defining new templates or modifying the Normal templates.

MORE INFORMATION

The information in the Normal templates used by Microsoft Access includes the following:

- Default section dimensions
- Default [control properties](#)
- Options to include page headers/footers

To change the default settings, either modify a report and/or form and save it as Normal, or rename the templates.

Note that the names you enter for the form and report templates must actually exist in your [database](#). The SYSTEM.MDA file retains the template names that you define.

To change form and/or report template names, use the following steps:

1. Open a Microsoft Access database.
2. From the View menu, choose Options.
3. Under Category, select Form & Report Design.
4. Under Items, type the name of the form (or report) you want to use in the Form Template (or Report Template) [field](#).

To use your templates with other databases, you must copy or export the templates.

REFERENCES

=====

"User's Guide," version 1.0, pages 267-268 (forms) and pages 446-447

(reports)

[References](#)

PRB: Four Problems/Solutions to Blank Pages in Printed Reports

Article Number: Q95920
CREATED: 01-MAR-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you print a report generated in Microsoft Access, a blank page appears between every printed page of the report or a blank mailing label appears between every printed label.

CAUSE

Blank pages or mailing labels are generated when a report or label is wider or longer than the specified page, a condition caused by one or more of the following:

Cause 1

The total width of the form or report, plus the widths of the left and right margins, exceeds the page width (usually 8.5 inches).

Cause 2

The page size exceeds the user-defined paper size.

Cause 3

You have used the Groups/Totals ReportWizard, but did not select the "Fit all fields on one page" check box. Because of this, the ReportWizard may have located fields or controls wider than the paper size selected.

Cause 4

In the Print Setup box under Item Size, the Same as Detail check box has been cleared.

RESOLUTION

The following resolutions correspond to each of the four possible causes described above:

Solution 1

Reduce the width of the form or report, reduce the left or right margins, or reduce all three so that the total width of the form or report, plus the width of the margins, does not exceed the page width. Use the following formula:

$$(\text{Form or Report Width})+(\text{Left Margin})+(\text{Right Margin})\leq(\text{Page Width})$$

Use the form or report's [property sheet](#) to view or change the Width property. Before you reduce the width of a form or report, move the fields and controls on it to match the new size.

To view the left and right margin values, choose Print Setup from the File menu.

If problems persist, use the following procedure to check the Print Setup dimensions:

1. From the File menu, choose Print Setup.
2. Choose More.
3. Under Item Size, verify that the Same as Detail box is selected. If so, the width and height listed will match the dimensions of the Detail section of the report. If not, these settings will override the Detail settings of the report and may cause extra blank pages to print.

NOTE: Item Size is usually used for multi-[column](#) reports. You must ensure that all columns fit on the report form when you set these values.

Solution 2 -----

If the page size is user-defined, Microsoft Access reads the settings you specified earlier for that printer driver using the Printers utility in the Windows Control Panel. If your report page exceeds this user-defined paper size, your reports will include unwanted blank pages.

A new page size will not be available in Microsoft Access until you set one up in Control Panel.

To change the page size settings using the Printers utility:

1. Open Control Panel, found in the Main program [group](#) of Microsoft Windows Program Manager.
2. Double-click the Printers icon to start the Printers utility.
3. Select the printer driver you want to change and choose the Setup button.
4. Under Paper Size, select User-defined from the [list box](#). A dialog box is displayed in which you can adjust the settings.

NOTE: Only dot-matrix printer drivers have a user-definable

page option. Laser printer drivers, for example, do not.

Solution 3

When using the Groups/Totals ReportWizard, select the "Fit all fields on one page" check box on the last screen before you switch to Design view or Print Preview. Once you select this check box, the ReportWizard assumes a page 8.5 inches wide, then sizes the fields and controls accordingly. If the check box is cleared, the resulting report width may exceed the desired page size.

Solution 4

The Same as Detail option is useful when you create a detailed, multi-column report and want the Detail section to be a different size than the other sections. Clearing this box can cause blank pages if the width defined for the Detail section of the report is greater than the page size specified. This box, found under Item Size in the Print Setup box, is selected by default. The Width property is set to equal the report's Width property.

STATUS

This behavior is by design.

[References](#)

INF: Size To Fit a Label Without Using Layout Menu

Article Number: Q95919
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

One way to resize a Label control is to use the Size To Fit command. This command is available on the Layout menu when you are in Design view of a form or report. This article explains how to resize a Label control without using the menu.

MORE INFORMATION

To use Size To Fit on a Label control without using the Layout menu, first give the Label the focus by clicking the label. When a control has the focus, the sizing handles are visible. Once it has the focus, you can execute Size To Fit by double-clicking the left mouse button on any of the sizing handles except the top left-most handle. This handle is for moving the Label control. If you double-click any of the other handles, the Label control expands or contracts to the correct size.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 8 "Form Basics"

[References](#)

INF: Using Access Basic to Derive a Statistical Median

Article Number: Q95918
CREATED: 01-MAR-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article describes an [Access Basic function](#) that opens a [table](#), reads the data, and finds the statistical median. The median is a measure of central tendency, another "middle" of a data set. The data set consisting of the numbers 1, 2, 3, 6, and 100 has a median of 3, the middle of the set. The data set consisting of the numbers 1, 2, 6, and 10 has a median of 4, the middle of the set $(2+6)/2=4$.

This article assumes that you are familiar with Access Basic and how to create and use Access Basic procedures.

More Information:

To create an Access Basic function that determines the statistical median, open a new or previously created [module](#) and enter the following code:

Note: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declarations section of the module.  
'*****
```

Option Explicit

```
'=====
' Create the following Median() function in the Module
' The following function will initialize:
' - A variable for the database object.
' - A variable for the snapshot.
' - Sets the database object to the opened database.
' - Creates a snapshot based on the database object.
' - A variable for the record count, the for statement,
'   two field values and an offset. This function requires the
'   table name and field to be analyzed to be include as argument
'   strings.
'=====
```

```
Function Median (tName$, fldName$) As Single
    Dim MedianDB As Database
    Dim ssMedian As Snapshot
    Dim RCount%, i%, x%, y%, OffSet%
```

```

Set MedianDB = CurrentDB()
Set ssMedian = MedianDB.CreateSnapshot("SELECT " & fldName$ & _
    " FROM " & tName$ & " ORDER BY " & fldName$ _
    & ";")

ssMedian.MoveLast
RCount% = ssMedian.RecordCount
x% = RCount% Mod 2

If x% <> 0 Then
    OffSet% = ((RCount% + 1) / 2) - 2
    For i% = 0 To OffSet%
        ssMedian.MovePrevious
    Next i
    Median = ssMedian(fldName$)
Else
    OffSet% = (RCount% / 2) - 2
    For i% = 0 To OffSet%
        ssMedian.MovePrevious
    Next i
    x% = ssMedian(fldName$)
    ssMedian.MovePrevious
    y% = ssMedian(fldName$)
    Median = (x% + y%) / 2
End If
ssMedian.Close
MedianDB.Close
End Function

```

Example of Using this Code

Create a [form](#) with [text box](#) controls that will reflect all measures of central tendency of a data set. In the ControlSource property for the text box [control](#), enter

```
=Median("TableName", "FieldName")
```

The value of this control will be the median of the data set. Another way to use this function is to call it from within another function that compares the median from different data sets. For example,

```

Function CompareMedians()
    Dim MyDB as Database
    .
    .
    .
    X = Median("TableName", "FieldName")
    Y = Median("TableName", "FieldName")
    If X > Y Then Debug.Print "The median for X is greatest."
End Function

```

References:

"Microsoft Access Basic: An Introduction to Programming," version 1.0

[References](#)

INF: Concatenating Fields in a Text box to Remove Blank Lines

Article Number: Q95917
CREATED: 01-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article describes how to concatenate multiple fields for use in a [form](#) or [report](#). This is especially useful in a mailing [label](#) report when some fields are null or empty. [Null](#) fields can cause blank lines to be printed on a report. Additionally, from a form, you may want to be able to copy and paste an entire name and address into another application like Microsoft Word for Windows.

Using the IIF() and the IsNull() functions, you can determine if a [field](#) is blank. If a field is blank, it returns an "empty" value. Chr(13) and Chr(10) are used to add a return and a line feed to the [text box](#).

More Information:

The following steps can be used to create a concatenated field eliminating blank lines. In this example, the Employees Table from the sample [database](#) Northwind Traders (NWIND.MDB) is used.

1. Create a new form or a report. Create your report based on the [table](#) or [query](#) where the information resides.
2. Add a text box [control](#) to the form/report.

Note: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

Object: Text Box

ControlName: Full Address

ControlSource:

```
=IIf (IsNull([FIRST NAME]),"",[FIRST NAME] & " ") &_  
IIf (IsNull([LAST NAME]),"",[LAST NAME]&Chr(13)&Chr(10)) &_  
IIf (IsNull([ADDRESS]),"",[ADDRESS] & Chr(13) & Chr(10)) &_  
IIf (IsNull([CITY]),"",[CITY] & ", ") &_  
IIf (IsNull([REGION]),"",[REGION] & " ") &_  
IIf (IsNull([POSTAL CODE]),"",[POSTAL CODE])
```

Can Grow: Yes

Can Shrink: Yes

When the form or report is printed, there will be no blank lines.

For more information, search for "IIF," then "IsNull," then "Chr,"

then "Concatenation" using the Help menu.

[References](#)

PRB: ErrMsg: CREATE TABLE Permission Denied in Export to SQL

Article Number: Q95916
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you export a [table](#) from Microsoft Access to Microsoft [SQL](#) Server, the following error message is displayed

```
[Microsoft][ODBC SQL Server Driver][SQL Server] CREATE TABLE  
permission denied, database <database>, owner <owner> [#262]
```

where <database> is the database you were trying to export the table to and <owner> is the SQL Login ID of the owner of the database.

CAUSE

You do not have permission to use the SQL command CREATE TABLE in the database you are exporting the table to.

RESOLUTION

To correct this problem, the SQL [system administrator](#) or the owner of the database must grant you CREATE TABLE permission in the database you were trying to export the table to.

Note: Permissions cannot be granted through Microsoft Access.

[References](#)

PRB: Error Message: Must Run SETUP /A Before Running SETUP

Article Number: Q95915
CREATED: 01-MAR-1993
MODIFIED: 15-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you attempt to set up a workstation for Microsoft Access using the SETUP /N command, you receive the following error message:

You must run SETUP /A before running SETUP /N

CAUSE

When the [network](#) server was set up, the command SETUP /A was issued from a temporary installation directory on the harddisk, rather than from the installation floppy disks.

RESOLUTION

Run SETUP /A from the installation disks in the floppy disk drive to set up the network server. Then you can run SETUP /N from the workstations.

[References](#)

PRB: No Arrows on Scroll Bars if Window Is Larger than Form

Article Number: Q95914
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you create a [form](#) and set the form [property sheet](#) to allow scroll bars, you may not see the scroll bars when you switch to Form view.

CAUSE

If the details section of a form is designed so that all the controls fit within the form, Microsoft Access will not display the scroll bars. However, the scroll bars will display if the Form window is sized smaller than the details section.

STATUS

This behavior is by design.

MORE INFORMATION

The purpose of the scroll arrows is to view parts of the form that are sometimes not visible if the [Form window](#) is too small. You cannot scroll to the next [record](#) with scroll bars, only with record selectors.

Steps to Reproduce Behavior

1. Open a bound form in [Design view](#) and display the property sheet.
2. Set the DefaultView for the form to Single Form.
3. Set the Scroll Bars property to Both.
4. View the form in [Form view](#).
5. Size the form window so that all of the controls/fields are visible. The scroll bars should not be visible at this point.
6. Size the form so that you cannot see all of the controls/fields. The scroll bars should now be visible.

[References](#)

PRB: Graph Object Is Dimmed in Form View

Article Number: Q95913
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A Microsoft Graph object appears gray in [Form view](#).

CAUSE

If the RowSource for a Graph object is invalid, the [graph](#) will be dimmed.

RESOLUTION

To work around the problem, make the RowSource valid.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

-
1. Create a new unbound [form](#).
 2. From the Edit menu, choose Insert Object. In the Object Type box, select Microsoft Graph, then choose OK.
 3. Update the graph to Microsoft Access by closing Microsoft Graph and choosing Yes to [update](#) Microsoft Access.
 4. Change the following properties of the object frame:

```
Object: Object Frame
-----
Scaling: Scale
RowSourceType: Value List
RowSource: Title;AAA;H1;H2;H3;1;2;3
ColumnCount: 3
```

5. From the View menu, choose Form. Note that the Graph is gray.

The problem is that the RowSource contains eight values and the Column Count is 3. To correct the Graph, add another value.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 13, "Using Pictures, Graphs, and Other Objects," pages 327-330

[References](#)

PRB: DDE App Name for Access Is Incorrect in Manual

Article Number: Q95912
CREATED: 01-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You will not be able to initiate a [DDE link](#) with Microsoft Access if the application name is incorrectly specified as "ACCESS". An error message will be generated. For example, in Microsoft Word, the error generated is:

The remote nwind (System) is not accessible. Do you want to start the application Access?

CAUSE

On page 141 of the "Microsoft [Access Basic](#): An Introduction to Programming" manual, the app name for Microsoft Access is incorrectly listed as Access. It should be listed as MSACCESS.

RESOLUTION

The correct app name to use when referring to Microsoft Access in a DDE conversation is MSACCESS.

References:

"Microsoft Access Basic: An Introduction to Programming," version 1.0, page 141

[References](#)

INF: Calculated Fields Must Be Shown to Perform Calculations

Article Number: Q95911
CREATED: 01-MAR-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

If you have a calculated [field](#) in a [query](#) named Field1, and you have another field that is performing a calculation based on Field1, then Field1 must be checked to Show in the query grid.

MORE INFORMATION

The following example assumes that you have a [table](#) that contains a field named Field1.

1. Create the following query:

```
Query: QueryTest
-----
Field Name: Field1: (1+2)
  Show: False
Field Name: Field2: (Field1 + 5)
  Show: True
```

2. Run the query.

The calculation in Field2 will not result in any output. The Show box for Field1 must be checked True in order for the calculation in Field2 to be performed.

References:

Microsoft Access "User's Guide," version 1.0, Chapter 6, "Designing Select Queries," pages 148-149

[References](#)

PRB: Cursor Position Unchanged in Undo Paste

Article Number: Q95910
CREATED: 01-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you choose the Edit Undo Paste command in the Module window, the text is deleted but the cursor does not return to its original position.

CAUSE

The Undo [function](#) in Microsoft Access is very basic. It only remembers the changes to a line that have not been committed to a [module](#). It does not maintain information on cursor position.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the sample [database](#) NWIND.MDB.
2. Open the Introduction to Programming module. In the Procedure box, select ActionQuery.
3. Select any line of text within the Module window.
4. From the Edit menu, choose Copy.
5. In the Module window, place the cursor at the end of the line "Set db = OpenDatabase(dbName)". Choose Paste from the Edit menu.

Note the position of the cursor (at the end of the [string](#) of text).

6. From the Edit menu, choose Undo. The text will be deleted, but the cursor will not return to the end of the line where you started the paste. It will return to the beginning of the line on which the paste was made.

[References](#)

PRB: Error Message: 'Couldn't Find Input Table or Query'

Article Number: Q95909
CREATED: 01-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

When you run a [query](#), the following error message is displayed:

Couldn't find input [table](#) or query '[Object name]'

When you open the query in [Design view](#), the Field list is empty.

CAUSE

The query includes a deleted table or query.

RESOLUTION

Modify the query so that it contains tables or fields that do exist. This behavior is by design so that you can determine what was deleted, rather than losing your query completely. The [field list](#) for the table or query appears empty, but the fields from this table or query still appear in the [QBE](#) grid.

MORE INFORMATION

The query opens in Design view only. Running the query generates the following error message:

Couldn't find input table or query '[Object Name]'.

Steps to Reproduce Behavior

1. In the sample [database](#) NWIND.MDB, create a new query based on the Categories table.
2. Drag several fields from the Categories field list to the QBE grid.
3. Save the Query as Query1.
4. In the [Database window](#), delete the Categories table.
5. Open Query1 in Design view.

Note that the Categories field list is empty, but all the fields from step 2 still appear in the QBE grid.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 5, "Query Basics"

[References](#)

PRACC9303: SHIFT+HOME Does Not Select All the Text

Article Number: Q95908
CREATED: 01-MAR-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In the [property sheet](#), when you press SHIFT+HOME, Microsoft Access should highlight the text to the beginning of the line but doesn't. This problem occurs only when there is a large amount of text in the property. The same problem can occur with SHIFT+END.

CAUSE

The HOME and END keys do not work correctly with more than 20 inches of text in a [text box](#) or property box.

RESOLUTION

To work around the problem, press CTRL+SHIFT+HOME or CTRL+SHIFT+END.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Type a large amount of text into the ControlSource property of a text box [control](#).
2. Press SHIFT+HOME to highlight all the text to the beginning of the line.
3. Press SHIFT+LEFT ARROW, and note that more text can be highlighted.

[References](#)

INF: How to Get the Last Two Digits of the Year

Article Number: Q95907
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to display only the last two digits of the year. The [expression](#)

```
=Year(Now())
```

will display all four digits of the current year (for example, 1993). In order to display only the last two digits of the year (for example, 93), use the following expression:

```
=Right(Str(Year(Now())),2)
```

The breakdown of the above expression is:

- Year(Now()) returns the current year.
- Str(...) converts the date to a [string data type](#).
- Right(...,2) returns the right two characters.

References:

For more information, search for YEAR, RIGHT, STR, and NOW functions using the Microsoft Access Help menu.

[References](#)

PRB: Scroll Bars Displayed for a Subform in Datasheet View

Article Number: Q95906
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you set the ScrollBars property to Neither on a [subform](#), the scroll bars may not be disabled in [Datasheet view](#).

CAUSE

This will happen because you have more information than will fit in the frame created for the subform. Thus, Microsoft Access creates scroll bars so you can view all the data. If your frame is larger than the amount of data that will be shown in your subform, the scroll bars will be correctly disabled.

RESOLUTION

Make the frame for your subform large enough to hold all the data that will be displayed within the [datasheet](#).

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a subform.
2. Switch the DefaultView property to Datasheet, and set ScrollBars to Neither.
3. Add the subform to a main [form](#).
4. Switch to [Form view](#).

Notice the scroll bars still exist.

[References](#)

PRACC9303: Form Record Selector Buttons Disabled

Article Number: Q95905
CREATED: 01-MAR-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You are unable to choose the record selector buttons in the bottom, left-hand corner of the form.

RESOLUTION

You can design your own record selector buttons such as those used in the sample database ORDENTRY.MDB which is shipped with Microsoft Access.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Create a GotoControl macro that goes to any one of the controls on the form.
2. Assign the macro to the OnCurrent property of the form.
3. Open the form in Form view. The control that your macro refers to must be on the form.
4. From the Records menu, choose Edit Filter/Sort.
5. Use the mouse to drag any field onto the grid.
6. From the Records menu, choose Apply Filter/Sort.
7. Click the record selector buttons at the bottom of the form to see that these buttons are now disabled.

[References](#)

PRB: Error Message, Cannot Change Working Directory to Default

Article Number: Q95904
CREATED: 01-MAR-1993
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you start Microsoft Access, you may receive the following error message

 Couldn't change working directory to <Default Database Directory>.
 Click OK or press ENTER to continue.

where <Default Database Directory> is the path that appears as the Default Database Directory property in the Items box when you choose Options from the View menu.

Choosing OK or pressing ENTER opens Microsoft Access.

CAUSE

You receive this error message if any of the following is true about the Default Database Directory property of the Microsoft Access View menu:

- The property is not pointing to a valid path.
- The path is longer than 64 characters.
- If the property is a period (.), there is one space before it.

RESOLUTION

Make sure that the path shown in the Default Database Directory property does not exceed 64 characters and that no leading space precedes a period, if a period is present.

MORE INFORMATION

To check the default directory for Microsoft Access, complete the following steps:

1. Start Microsoft Access and open any [database](#).
2. From the View menu, choose Options.
3. In the Category box, select General. In the Items box, select Default Database Directory.

Normally, during Microsoft Access installation, the default directory

is set to ".", which represents the Microsoft Access Startup directory.

You can also set a default directory to ".\subdir" and it will set the default to a subdirectory under the Microsoft Access startup directory called 'subdir'. Or you can specify a complete path.

[References](#)

INF: Filling a Multiple-Column List Box by Using Access Basic

Article Number: Q95903
CREATED: 01-MAR-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can use [Access Basic](#) to fill in the information for a [list box](#) or a [combo box](#). On page 82-85 of the "Microsoft Access Introduction to Programming" manual, an example is given on how to fill in a single-[column](#) list box. You can also use Access Basic to fill in a multiple-column list box or combo box.

MORE INFORMATION

To create a multiple-column list box, you must set Code 4 in the Select Case statement to the number of columns that you want in your list box. In Code 6 of the Select Case statement, define the data that will display in each column and [row](#) of the list box. The following code is based on the example on page 84 of the "Microsoft Access Introduction to Programming" manual:

```
'*****  
'Declarations section of the module  
'*****  
Option Explicit
```

```
'=====
```

'The following [function](#) uses a select case statement to fill a two-column and four-row list box. The function fills the first column of the list box with the dates of the next four Mondays. The second column is filled with the dates of the next four Tuesdays.

```
'=====
```

```
Function ListMonTuesdays(fld As Control, id, row, col, code)  
    Dim offset  
    Select Case Code  
        Case 0  
            ListMonTuesdays=True  
            'Initialize.  
        Case 1  
            ListMonTuesdays=id  
            'Open.  
            'Unique ID number for control  
        Case 3  
            ListMonTuesdays=4  
            'Number of rows.  
        Case 4  
            ListMonTuesdays=2  
            'Number of columns.  
        Case 5  
            ListMonTuesdays=-1  
            'Column width.  
            'Use default width.
```

```
'=====
```

'In the next Case statement:

```
'  
' Offset is the formula for finding the next four Mondays.  
' If column=0, then fill in with the dates for the next four  
' Mondays in column 1. If column=1, then fill in with the dates  
' for the next four Tuesdays in column 2.  
'=====
```

```
Case 6                                'Get Date  
  Offset=abs((9-Weekday(Now))Mod 7)  
  If col=0 then  
    ListMonTuesdays=Format(Now()+offset+7*row,"mmm d")  
  Else  
    Offset=abs((10-Weekday(Now))Mod 7)  
    ListMonTuesdays=Format(Now()+offset+7*row,"mmm d")  
  End if  
End Select
```

End Function

References:

"Microsoft Access Introduction to Programming," version 1.0, pages 82-85

[References](#)

PRB: Two Accelerator Keys in Custom Menu Show Only First Key

Article Number: Q95902
CREATED: 01-MAR-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you create a custom menu, if you define two accelerator keys in the MacroName grid, the drop-down menu will show the second one as the accelerator, but the first one will actually run the macro action.

CAUSE

This behavior is a function of Windows. Microsoft Access does not do syntax checking on menu strings, so if Windows receives a menu string with extra accelerators, the above symptoms will result.

RESOLUTION

Only define one accelerator key for a menu item.

MORE INFORMATION

Steps to Reproduce Behavior

-
1. In the sample database NWIND.MDB, create a new macro called MyMenu with one following action:

Macro Name	Action	Argument
D&o &This	MsgBox	Message: Any text message here

2. Open the Orders macro, and create a new macro name called OnMenu with one action:

Macro Name	Action
OnMenu	AddMenu

Add Menu Actions

Menu Name: &File
Menu Macro: MyMenu

3. Open the Orders form. Attach the macro "Orders.OnMenu" macro to the OnMenu property of the form.
4. Close the Orders form, save it, and reopen it.

5. Press ALT+F to drop the custom File menu. Note that the "T" in "Do This" is shown as the accelerator key.
6. Press the T key. Notice that nothing happens.
7. Press the O key.

The message box will be displayed.

[References](#)

PRB: Error Occurs in Print Preview Without Driver

Article Number: Q95901
CREATED: 01-MAR-1993
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You may receive the following error message when choosing Print Preview with a form selected:

An error occurred while attempting to retrieve printer information for the (printer name) on (printer port).

CAUSE

The error occurs because the form was created with a specific printer driver and that printer driver no longer exists in the system.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new form.
2. Before saving the form, choose Print Setup from the File menu, and set Specific Printer to something other than default.
3. Close and save the form as Test Form.
4. Open Windows File Manager.
5. Locate the driver for the specific printer you selected in Print Setup (drivers are usually located in the \WINDOWS\SYSTEM subdirectory).
6. Select the driver and temporarily rename the file.
7. Select Test Form in the Database window.
8. From the File menu, choose Print Preview.

An error message about the missing printer driver will be displayed.

The problem is that normally COMMDLG.DLL performs a test to determine

if you have an incorrect printer. When you choose Print Preview, you do not go through COMMDLG.DLL. If, on the other hand, you choose File Print you will get the more descriptive error message produced by COMMDLG.DLL:

An error occurred while attempting to retrieve printer information for the (printer name) on (printer port).

[References](#)

INF: Checking for Last Or First Record in a Macro

Article Number: Q95807
CREATED: 25-FEB-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

In a [macro](#), you cannot check the result of an action just performed. As a result, you can run into a problem when using a macro to move to the next [record](#) or the previous record. There is no record previous to the first record and no next record following the last record. If you attempt to go to the previous or last record when there is none, Microsoft Access displays error messages that may confuse users of a custom application.

MORE INFORMATION

You can work around this on a [form](#) by using [Access Basic](#) functions to move from record to record. For example, use the following custom MoveToNext() [function](#) to move to the next record or previous record on a form:

```
Option Explicit
Function MoveToNext (DisplayMsg$, MoveForward%)
    Dim MyDyna As Dynaset
    Set MyDyna = Screen.ActiveForm.Dynaset

    MyDyna.Bookmark = Screen.ActiveForm.Bookmark
    If MoveForward% Then
        MyDyna.MoveNext
    Else
        MyDyna.MovePrevious
    End If
    If MyDyna.EOF Or MyDyna.BOF Then
        MsgBox DisplayMsg$
    Else
        Screen.ActiveForm.Bookmark = MyDyna.Bookmark
    End If
End Function
```

To use the MoveToNext function, specify the following:

- Name of the function.
- Message you want displayed if the [current record](#) is at the end or the beginning of the record set.
- 'True' if you want to move to the next record. 'False' if you want to move to the previous record.

Here is an example:

```
OnPush: =MoveToNext("You are at the end of the record set", True)
```

[References](#)

INF: How to Skip Used Mailing Labels and Print Duplicates

Article Number: Q95806
CREATED: 25-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to print multiple copies of the same mailing [label](#), and how to use a partially used page where only some of the labels are available.

To make use of this article, you need to be familiar with [Access Basic](#) and know how to design Microsoft Access reports.

MORE INFORMATION

Printing Multiple Copies of the Same Label

When you choose Print from the File menu, you can choose to print multiple copies of the same [report](#). But when you try to print a single mailing label 20 times, Microsoft Access prints one label on each of 20 pages.

On a dot matrix printer, using single [column](#) labels, you can work around this problem by defining each label as a separate page. However, you cannot use this [method](#) for laser printers or multiple-column labels. To solve this problem, use the step-by-step procedure described below.

Using Labels that Would Otherwise Be Wasted

After printing labels, you usually end up with a partially used last page. There is no built-in mechanism in Microsoft Access to use the remaining labels on a partially used page. Microsoft Access always starts on a new page. On a dot matrix printer, you can adjust the top of [form](#) manually. But you cannot do that on laser printers. To solve this problem, use the step-by-step procedure described below.

Step-by-Step Procedure to Solve Both Problems

The Microsoft Access report generator provides powerful hooks that allow [control](#) over the finished product. By calling a [function](#) from the OnFormat property of the report's detail section, you can alter the MoveLayout, NextRecord, and PrintSection properties to leave blank spaces or print multiple copies on the same page.

The code listed below is generic. You can attach it to any Mailing Label report to print multiple copies and to skip used labels if needed. To use the example, you need to have a mailing label report called MyLabels.

1. Create a new module, and place the following lines in the declaration section:

```
'*****
' Declarations section of the module.
'*****

Option Compare Database
Option Explicit

Dim LabelBlanks&
Dim LabelCopies&
Dim BlankCount&
Dim CopyCount&

'=====
' The following function will cause an inputbox to
' display when the report is run that prompts the user
' for the number of used labels to skip and how many
' copies of each label should be printed.
'=====

Function LabelSetup ()
    LabelBlanks& = Val(InputBox$("Enter Number of blank labels to skip"))
    LabelCopies& = Val(InputBox$("Enter Number of Copies to Print"))
    If LabelBlanks& < 0 Then LabelBlanks& = 0
    If LabelCopies& < 1 Then LabelCopies& = 1
End Function

'=====
' The following function sets the variables to a zero
'=====

Function LabelInitialize ()
    BlankCount& = 0
    CopyCount& = 0
End Function

'=====
' The following function is the main part of this code
' that allows the labels to print as the user desires.
'=====

Function LabelLayout (R As Report)
    If BlankCount& < LabelBlanks& Then
        R.NextRecord = False
        R.PrintSection = False
        BlankCount& = BlankCount& + 1
    Else
        If CopyCount& < (LabelCopies& - 1) Then
            R.NextRecord = False
            CopyCount& = CopyCount& + 1
        Else
            CopyCount& = 0
        End If
    End If
End Function
```

End Function

2. Open the report named MyLabels in Design view and add the following line to OnFormat property of the Detail section:

```
OnFormat: =LabelLayout(Reports![MyLabels])
```

3. Add the following line to the OnOpen property of the MyLabels Report:

```
OnOpen: =LabelSetup()
```

4. Although typically labels do not have a report header, choose Report Hdr/Ftr from the Layout menu to add a report header and footer. Then Add the following line to the OnFormat property of the report header:

```
OnFormat: =LabelInitialize()
```

5. Set the Height property for both the report header and report footer to 0.

When you print the report, the report calls the LabelSetup() function, which first asks you to enter the number of used labels to skip on the first page (BlankCount) and then asks how many of each label you want printed (CopyCount).

When the report header is formatted, it calls the LabelInitialize() function, so when you switch from preview to print, the BlankCount and CopyCount fields are set to zero. As each label is formatted, the LabelLayout() function adjusts the NextRecord and MoveLayout properties to skip used labels and print the desired duplicates.

References:

"Microsoft Access Language Reference," version 1.0, page 313, "MoveLayout, NextRecord, PrintSection Properties" section.

[References](#)

PRB: Unfreeze All Columns Does Not Restore Column Order

Article Number: Q95805
CREATED: 25-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you open a [database](#) and view a [form](#) in [Datasheet view](#) then choose Freeze Columns from the Layout menu, the [column](#) your cursor is in will be moved to the far left and separated by a bold black bar, as expected. However, if you then choose Unfreeze All Columns from the Layout menu, the order of the columns will not be restored.

CAUSE

Frozen columns are moved to the "frozen" area. They are not put back in their original position because many other things could have been changed because the column was originally frozen.

STATUS

This behavior is by design.

[References](#)

PRB: Edit Find Dialog Is Modal in Form View

Article Number: Q95651
CREATED: 18-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you are in the Edit Find [dialog box](#) in Microsoft Access, you cannot switch focus back to the [form](#). The Edit Find dialog must be cleared before you can continue working with the form.

CAUSE

The Edit Find dialog is [modal](#). When you open a modal form, you cannot access any other Microsoft Access objects until you close or hide the form.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Start Microsoft Access and open a [database](#).
2. Open a form with an editable [field](#) (for example, a [text box](#)) in [Form view](#).
3. From the Edit menu, choose Find.

At this point you will be unable to shift focus back to the form. You must choose the Close button in the Edit Find dialog box.

[References](#)

PRB: Short Label Autosizes to Cursor Size

Article Number: Q95649
CREATED: 18-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When in Design view you create a new label on a form and the label height is shorter than the cursor size, the control will autosize to the height of the cursor.

STATUS

The autosizing of the cursor in a label to a default height is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new database.
2. Create a new form.
3. Open the form in Design view.
4. Place a label on the form and size it to a height smaller than the height of the cursor.
5. Place the cursor in the text box to enter text.

The control will resize larger to the height of the cursor.

[References](#)

INF: Detailed Explanation of the KeepTogether Property

Article Number: Q95648
CREATED: 18-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Form and [report](#) sections have a KeepTogether property that you can set to Yes or No. The KeepTogether property setting determines whether Microsoft Access prints the controls located in that section on a single page or on more than one page.

The section's KeepTogether property affects the controls located in that section, for a single [record](#).

MORE INFORMATION

If the KeepTogether property is set to yes, Microsoft Access begins printing the controls located in that section at the top of the next page if it can't print all of the controls on the current page. If it is set to no, Microsoft Access prints as many controls as it can on the current page and continues printing them on the next page.

If all the controls in a section require more than one page for a single record, Microsoft Access continues printing on a second page regardless of the KeepTogether setting.

For example, say the detail section of a report has its KeepTogether property set to Yes and all the controls in that section won't fit on the current printed page. In this case, Microsoft Access prints none of the controls on the current page. Instead, it begins printing the controls at the top of the next page. If it turns out the section requires more than a single page, Microsoft Access ignores the KeepTogether property value and continues printing the controls at the top of the next page.

The KeepTogether property affects the controls in a [form](#) or report section for a single record -- not groups of records. The KeepTogether property won't keep all the grouped records together, and it will not avoid the problem of abandoned [group](#) headers left on the bottom of a printed page.

Several other articles can help you learn how to create a [label](#) that will print at the top of each report page when a group of records span more than one page. To find these articles, [query](#) on the following words here in the Microsoft Knowledge Base:

Access and continued and new and page

References:

For information on programatically controlling section printing, see the "Microsoft Access Language Reference," version 1.0, "MoveLayout, NextRecord, PrintSection Properties" section, pages 313-314.

[References](#)

INF: How to Play .WAV Sounds on Events in Microsoft Access

Article Number: Q95647
CREATED: 18-FEB-1993
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not have a built-in [function](#) to play sound files on events, such as when a [form](#) is opened or closed. However, you can use the Microsoft Windows 3.1 application program interface (API) through [Access Basic](#) code to create a user-defined function to play sound files.

This article assumes you are familiar with Access Basic and Windows APIs. In this article, the use of error trapping has been omitted to keep the information as clear and concise as possible.

MORE INFORMATION

Follow these steps to create a user-defined function to play sound files:

1. Create a new [module](#) with a single function named PlaySound().

Note: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declarations section of the module.  
'*****
```

Option Explicit

```
Declare Function sndplaysound% Lib "mmsystem" (ByVal filename$, _  
                                               ByVal snd_async%)
```

```
'=====
```

```
' The following function PlaySound calls the Windows API function
```

```
'=====
```

```
Function PlaySound (msound)  
    XX% = sndplaysound(msound, 1) ' play mmwave sound  
    If XX% = 0 Then MsgBox "The Sound Did Not Play!"  
End Function
```

2. Save the module as "Play It Sam."

The following steps use a [command button](#) to trigger the sound, but you can use other events to trigger the sound.

3. Add a command button to a form you want to play the sound from.

Object: Command Button

ControlName: Push_Button_1

Caption: Play Chimes

OnPush: =PlaySound("C:\WINDOWS\CHIMES.WAV")

(Where C:\WINDOWS\ is the location of the sound file and CHIMES.WAV is the sound file.)

When the form is open in [Form view](#), the sound is ready to play by clicking the command button. You can assign this function to the form's OnOpen [event](#) if you want chimes to play when you open a form.

NOTE: This example does not have error trapping. Unexpected results may occur if the sound file is not in the location specified or does not exist.

If the sound (.WAV file) is in the [table](#) as an [OLE field](#), the sound can be added to the form out of the [field list](#). You can then use a [macro](#) that does a GoToControl [olefield], DoMenuItem form-edit-object-<verb>. You can run this from a button on the form. It will go to the OLE field and edit the object (the default edit is play for a sound).

References:

"Microsoft Access Basic: An Introduction to Programming," Chapters 1-5

"Microsoft Windows Software Development Kit," Microsoft Press, 1992

"Programming Windows: the Microsoft Guide to Writing Applications for Windows 3", Charles Petzold. Microsoft Press, 1990

"Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference," Volumes 1 - 6, Microsoft Press, 1992

[References](#)

PRB: Setup Program STFSETUP.EXE Causes GP Fault

Article Number: Q95646
CREATED: 18-FEB-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The Microsoft Access Setup program displays the following error message immediately after trying to update an old version of the Windows help file (WINHELP.HLP):

General Protection Fault (GPF) in STFSETUP.EXE

CAUSE

The current WINHELP.HLP or WINHELP.EXE files in the Windows directory are corrupt. These corrupt files cause a general protection (GP) fault when you run the Microsoft Access Setup program to update the help file.

RESOLUTION

Rename the WINHELP.EXE and WINHELP.HLP files in the Windows directory. Then reinstall Microsoft Access. The Setup program will write new help files and place them in the Windows directory.

[References](#)

PRB: No AutoRepeat with GoToRecord/Next and DoMenuItem

Article Number: Q95645
CREATED: 18-FEB-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The AutoRepeat property for command buttons does not work in conjunction with the GoToRecord/Next [macro](#) action for moving to successive records on a [form](#). In other words, if your form uses command buttons instead of the [record selector](#) buttons to move from record to record, the AutoRepeat property does not repeatedly execute a macro that performs a GoToRecord/Next action.

Also, the AutoRepeat property fails under the same circumstances when using the DoMenuItem/Form/Records/GoTo/Next macro action. To re-execute the macro, you must click the [command button](#) again.

CAUSE

This is correct behavior for these macro actions. Here are two points that need clarification before the cause can be explained:

1. OnPush property: The macro under the OnPush property of a command button normally won't execute until you push and let up the command button with the mouse pointer in the Command Button's region. The exception is when the command button's AutoRepeat property is set to Yes. In this case, the macro under the OnPush property of the command button executes when the button is pushed and the mouse pointer is in the Command Button's region.
2. Focus: Several macro actions change the focus. For example, GoToControl, GoToRecord, OpenForm, OpenTable, and ApplyFilter all change the focus.

Therefore, if a command button's OnPush property macro changes the focus to, for example, the next record, then the command button has lost the focus. That is, when you click the command button, the click doesn't apply to that command button anymore because the focus changed. It's as if you had moved the mouse pointer off the command button while pressing the mouse button, as if the command button moved out from under the mouse pointer right before you pressed the mouse button.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Start Microsoft Access.
2. Open the sample NWIND.MDB [database](#).
3. Create a new macro with two actions (OpenTable and GoToRecord). Use the following as a guide to assign action arguments to each action:

OpenTable

Table Name: Categories
View: Datasheet
Data Mode: Edit

GoToRecord

Object Type: Table
Object Name: Categories
Record: Next

4. Close, save, and name the macro.
5. Open a new form and click the Blank Form button.
6. Drag the macro you just created onto the blank form to create a command button.
7. Set the AutoRepeat property of the command button to Yes. Notice the command button's OnPush property is set to the name of the macro.
8. Change to [Form view](#) to browse the form. As a result, you should see a single command button on the form.
9. Click the command button. Now you should see the Categories [table](#).
10. Size the Categories table so that both the form and the table fit on the screen and you can see the VCR button on the table.

Each time you click the command button, the record selector moves to the next record. When you click and hold down the command button, you would expect Microsoft Access to scroll rapidly to the last record in the table, but this doesn't happen. Instead, the record selector moves to the next record only, regardless of the AutoRepeat property setting of the command button.

However, if you place a macro that uses a different macro action such as Beep on the form, the AutoRepeat action works as expected. This is because Beep does not change the focus.

[References](#)

PRB: Form Handle Not Released When Closed with Basic Code

Article Number: Q95644
CREATED: 18-FEB-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

After closing a form (using the Close macro action) that was opened through Access Basic code, the file handle for the table that the form is based on is not released. Because the file handle is not released, no functions that require exclusive use on that table can be executed (for example, deleting the table). When a function that needs exclusive use of that table is executed, the following error will occur:

Couldn't lock table '<tablename>'; currently in use.

CAUSE

You are not guaranteed that the user interface (UI) will be synchronized with the underlying Microsoft Access code in all cases. In this case, the record locking is done behind the scenes. Microsoft Access releases the table when it hits its idle loop (that is, the UI optimization).

RESOLUTION

After the form is opened in Access Basic code, insert a DoEvents command before the function that needs exclusive use of the table, to cause Microsoft Access to yield execution to the operating system, so that it can process events. When the DoEvents command occurs it allows Microsoft Access to process the pending unlock sequence.

STATUS

This behavior is by design.

MORE INFORMATION

The following steps provide example code that reproduces this error. There is a comment in the Access Basic code identifying where to insert a DoEvents command to resolve this error.

Steps to Reproduce Problem

1. Open the sample database NWIND.MDB. Create a module and enter the following code:

```
'*****  
'Declarations section of the module.
```

```

'*****
Option Explicit

'=====
' Create and save the following TempCust() function in the Module
'=====
Function TempCust()
    DoCmd SelectObject a_table, "Customers", True
    DoCmd CopyObject , "Cust2"
    DoCmd OpenForm "Customers 2", a_normal, "", "", a_edit, a_dialog
    '**** This is where the DoEvents should be inserted to fix ****
    DoCmd SelectObject a_table, "Cust2", True
    DoCmd DoMenuItem 1, 1, 4
End Function

```

2. Open the Customers form in Design view, and change the RecordSource property to "Cust2."
3. Create a command button on this form with the following properties:

```

Object: Command Button
-----
ControlName: Close Form
Caption: Close Form
OnPush: ExitForm

```

4. Save the modified Customers form as "Customers 2".
5. Create and save the following macro:

```

Macro Name    Action
-----
ExitForm      Close

```

6. Create and save the following macro:

```

Macro Name    Action    Argument
-----
OpenCustForm  RunCode    TempCust()

```

7. From the Database window, run the OpenCustForm macro that you just created. The Customers 2 form will open.
8. Click the Close Form macro button on this form. You will receive a dialog box that asks:

```
Delete Table 'Cust2'?
```

9. Press OK. You will receive the following error message:

```
Couldn't lock table 'Cust2'; currently in use.
```

[References](#)

INF: Row Fix-up in the NWIND.MDB Order Form

Article Number: Q95643
CREATED: 18-FEB-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access provides an automatic "row fix-up" feature that uses links between tables to update (fix up) records in queries and forms. You can use this feature to enter a single value, and Microsoft Access automatically looks up and displays other values for that record.

Row fix-up allows you to update forms and queries without using macros or Access Basic code. This article uses the Orders form in the sample Northwind Traders database (NWIND.MDB) to demonstrate how to setup and use rowfix-up.

MORE INFORMATION

Prior to opening the NWIND.MDB database, refer to the Northwind Database Diagram in Appendix B of the "Microsoft Access User's Guide," for version 1.0. This diagram explains the database structure.

Then, proceed as follows:

1. Open the NWIND.MDB database.
2. Open the Orders form in Form view.
3. Open the Order Information query in Design view.

Reviewing the Form

Notice the following:

1. The RecordSource property of the Orders form is set to the Order Information query.
2. The Order Information query is based on three tables: Customers, Employees, and Orders.
3. The Join properties between the tables are as follows:

One	Many
-----	------

- a. CUSTOMERS.[customer id] ==> ORDERS.[customer id]
- b. EMPLOYEES.[employee id] ==> ORDERS.[employee id]

4. The Order ID field in the query grid comes from the Orders table,

which is on the many-side of the [relationship](#), and not from the Customers table, which is on the one-side of the relationship. (See 3a, above.)

5. The Bill To [combo box](#) is based on the Customer List query. Customer List has the following characteristics:
 - a. The query is based on the Customers table.
 - b. The query returns two columns: Company Name and Company ID.
 - c. The bound [column](#) is Company ID.
 - d. The visible column is Company Name.
6. Microsoft Access select queries are updatable. This means that when you make a change to a query's [dynaset](#), Microsoft Access can update the source tables with your changes. For more information on this, search for "updating underlying tables," then "When is Query Updatable" using the Microsoft Access Help menu.

Explanation of Row Fix-Up in the Orders Form

The Bill To combo box presents you with a list of company names from the visible column of the Customer List query. (See point 5, above.)

When you make a selection, the combo box stores the bound column--the Company ID field from the Customer List query--in the Customer ID field in the Orders table. (See points 4 and 5d, above.) It is important to note that the selection was stored in the many-side table. (See point 3a, above.)

After the combo box updates the Customer ID field in the Orders table (the many-side table), the field is changed so it no longer points to the same record in the Customers table (the one-side table). (See point 6, above.) Note that the Orders table is updated because the query's dynaset was changed.

Microsoft Access recognizes that the relationship no longer matches, so it automatically updates the relationship by linking the changed record in the Orders table with the appropriate record in the Customers table.

All the fields in the Orders form that come from the Customers table (the one-side table) are updated to show the values in record for the newly formed [link](#).

Example

When you make a change to the Bill To combo box, Microsoft Access uses row fix-up to update the following list of fields on the Orders form:

- [Address]
- [City]
- [Region]
- [Postal Code]
- [Country]

The following list of fields on the Orders form are not updated by Row fix-up. However they are updated by the Orders Fill In Ship To [macro](#) on the AfterUpdate property of the Bill To combo box.

- [Ship Name]
- [Ship Address]
- [Ship City]
- [Ship Region]
- [Ship Postal]
- [Ship Country]

REFERENCES

=====

Refer to "Creating a Many-to-One Form" in Chapter 10 of the "Microsoft Access User's Guide," version 1.0 or version 1.1.

Microsoft Access version 1.1 provides online help for row fix-up issues. The terminology "row fix-up" is synonymous with "dynamic lookup." For more information in Microsoft Access version 1.1, search for "dynamic lookup", then "Displaying Fields from Another Table or Query (Common Question)" using the Help menu.

For more information, query on the following words in the Microsoft Knowledge Base:

row and fix-up

[References](#)

INF: SHIFT+ENTER Key Combination vs. CTRL+ENTER

Article Number: Q95642
CREATED: 18-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

To insert more than one line in a [label](#), [field](#), or zoom box, use the CTRL+ENTER key combination.

The Microsoft Access "User's Guide" states on page 222 that in order to add more lines to a label, you should use the SHIFT+ENTER key combination. You can use this; however, CTRL+ENTER will also work and can be used to add more lines to text boxes and zoom boxes as well. The SHIFT+ENTER key combination will not work with zoom boxes or text boxes to add more lines. The SHIFT+ENTER key combination also has other uses with Microsoft Access. For example, if you are performing data entry, SHIFT+ENTER saves changes to the [current record](#). The CTRL+ENTER key combination is consistent throughout the program.

You can design a memo or text field so that the ENTER key alone creates a new line. For more information, [query](#) on the following words in the Microsoft Knowledge Base:

enter and getkeystate

References:

Microsoft Access "User's Guide," version 1.0, page 222

[References](#)

PRB: Toolbox Selection Lost When Closed

Article Number: Q95641
CREATED: 18-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The [toolbox](#) will not retain your selection when it is closed and then reopened.

RESOLUTION

The toolbox will always be reset to the pointer when you reopen it. If you want your selection to be saved, choose the Lock button at the bottom of the toolbox.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open a [form](#) in [Design view](#).
2. From the View menu, choose Toolbox.
3. Select an item on the toolbox other than the pointer.
4. From the View menu, choose Toolbox (this will close the toolbox). The mouse cursor still reflects the current tool selection as expected.
5. From the View menu, choose Toolbox to reopen the toolbox. The pointer will have the focus again.

[References](#)

INF: Converting Currency or Number Values into English Words

Article Number: Q95640
CREATED: 18-FEB-1993
MODIFIED: 25-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

By using four functions you can convert a currency or other numeric value into its equivalent in English words. For example, you can change 24.50 into the following text: Twenty Four DOLLARS and Fifty CENTS.

To understand this article, you need to be familiar with [Access Basic](#) and know how to create and use Access Basic procedures. Note that none of the words in the result are hyphenated. That is, the result shows Twenty Four, not Twenty-Four. Revise the code if hyphenation is important to you.

MORE INFORMATION

The following code creates four functions: NumToWord(), GetWord(), GetTens(), and GetDigit():

```
' Note: Enter each command on a single line. The underscore ( _ ) on
' the end of some lines is a line continuation character. It's there
' to improve readability. When you enter the line, omit the underscore
' and combine all continued lines into a single line.
```

```
'*****
```

```
' Declarations section of the module
```

```
'*****
```

```
Option Explicit
```

```
'=====
```

```
'*** This is the main function call
```

```
'=====
```

```
Function NumToWord(NumVal)
```

```
    Dim NTW, NText, Dollars, Cents, NWord, TotalCents As String
```

```
    DIM DecPlace, TotalSets, Cnt, LDollHold As Integer
```

```
    ReDim NumParts(9) As String    'Array for Amount (sets of three)
```

```
    ReDim Place(9) As String       'Array containing place holders
```

```
    Dim LDoll As Integer           'Length of the Dollars Text Amount
```

```
    Place(2) = " Thousand "      '
```

```
    Place(3) = " Million "       'Place holder names for money
```

```
    Place(4) = " Billion "        'amounts
```

```
    Place(5) = " Trillion "      '
```

```
    NTW = ""                     'Temp value for the function
```

```
    NText = Trim(Str(NumVal))     'String representation of amount
```

```
    DecPlace = InStr(Trim(NText), ".") 'Position of decimal 0 if none
```

```
    Dollars = Trim(Left(NText, IIf(DecPlace = 0, Len(NumVal), _
```

```
        DecPlace - 1)))
```

```

LDoll = Len(Dollars)
Cents = Trim(Right(NText, IIf(DecPlace = 0, 0, Abs(DecPlace - _
                               Len(NText))))))

If Len(Cents) = 1 Then
    Cents = Cents & "0"
End If
If (LDoll Mod 3) = 0 Then
    TotalSets = (LDoll \ 3)
Else
    TotalSets = (LDoll \ 3) + 1
End If
cnt = 1
LDollHold = LDoll
Do While LDoll > 0
    NumParts(cnt) = IIf(LDoll > 3, Right(Dollars, 3), _
                        Trim(Dollars))
    Dollars = IIf(LDoll > 3, Left(Dollars, (IIf(LDoll < 3, 3, _
                                                LDoll)) - 3), "")
    LDoll = Len(Dollars)
    cnt = cnt + 1
Loop
For cnt = TotalSets To 1 Step -1      'step through NumParts array
    NWord = GetWord(NumParts(cnt))    'convert 1 element of NumParts
    NTW = NTW & NWord                 'concatenate it to temp variable
    If NWord <> "" Then NTW = NTW & Place(cnt)
Next cnt                              'loop through
If LDollHold > 0 Then
    NTW = NTW & " DOLLARS and "      'concatenate text
Else
    NTW = NTW & " NO DOLLARS and "   'concatenate text
End If
TotalCents = gettens(Cents)          'Convert cents part to word
If TotalCents = "" Then TotalCents = "NO" 'Concat NO if cents=0
NTW = NTW & TotalCents & " CENTS"   'Concat Dollars and Cents
NumToWord = NTW                     'Assign word value to function
End Function      'End NumToWord function, Return to calling program

```

```

'=====
' The following function converts a number from 0 to 999 to text
'=====

```

```

Function GetWord (NumText)
    Dim GW As String, x As Integer
    GW = ""                                'null out temporary function value
    If Val(NumText) > 0 Then
        For x = 1 To Len(NumText) 'loop the length of NumText times
            Select Case Len(NumText)
                Case 3:
                    If Val(NumText) > 99 Then
                        GW = GetDigit(Left(NumText, 1)) & " Hundred "
                    End If
                    NumText = Right(NumText, 2)
                Case 2:
                    GW = GW & GetTens(NumText)
                    NumText = ""
                Case 1:
                    GW = GetDigit(NumText)
                Case Else
            End Select
        Next x
    End If
    GetWord = GW
End Function

```

```

        End Select
    Next x
End If
GetWord = GW 'assign function return value
End Function 'End function GetWord - Return to calling program

```

```

'=====
' The following function converts a number from 10 to 99 to text
'=====

```

```

Function GetTens (TensText)
    Dim GT As String
    GT = "" 'null out the temporary function value
    If Val(Left(TensText, 1)) = 1 Then ' If value between 10-19
        Select Case Val(TensText)
            Case 10: GT = "Ten" '
            Case 11: GT = "Eleven" '
            Case 12: GT = "Twelve" '
            Case 13: GT = "Thirteen" ' Retrieve numeric word
            Case 14: GT = "Fourteen" ' value if between ten and
            Case 15: GT = "Fifteen" ' nineteen inclusive.
            Case 16: GT = "Sixteen" '
            Case 17: GT = "Seventeen" '
            Case 18: GT = "Eighteen" '
            Case 19: GT = "Nineteen" '
            Case Else
        End Select
    Else ' If value between 20-99
        Select Case Val(Left(TensText, 1))
            Case 2: GT = "Twenty " '
            Case 3: GT = "Thirty " '
            Case 4: GT = "Forty " '
            Case 5: GT = "Fifty " ' Retrieve value if it is
            Case 6: GT = "Sixty " ' divisible by ten
            Case 7: GT = "Seventy " ' excluding the value ten.
            Case 8: GT = "Eighty " '
            Case 9: GT = "Ninety " '
            Case Else
        End Select
        GT = GT & GetDigit(Right(TensText, 1)) 'Retrieve ones place
    End If
    GetTens = GT ' Assign function return value.
End Function 'End function GetTens - return to calling program

```

```

'=====
' The following function converts a number from 1 to 9 to text
'=====

```

```

Function GetDigit (Digit)
    Select Case Val(Digit)
        Case 1: GetDigit = "One" '
        Case 2: GetDigit = "Two" '
        Case 3: GetDigit = "Three" '
        Case 4: GetDigit = "Four" ' Assign a numeric word value
        Case 5: GetDigit = "Five" ' based on a single digit.
        Case 6: GetDigit = "Six" '
        Case 7: GetDigit = "Seven" '
        Case 8: GetDigit = "Eight" '
        Case 9: GetDigit = "Nine" '
    End Select
End Function

```

```
Case Else: GetDigit = ""      '
End Select
End Function 'End function GetDigit - return to calling program
```

How to Use this Code

To change a currency value into English words, call the NumToWord() function giving the number as the argument. NumToWord() calls the other functions as needed and returns the result in English words.

There are two ways to call the NumToWord function. You can call it in Access Basic code as follows:

```
Amount = 24.50
WordValue = NumToWord(Amount)
```

Or you can call NumToWord() as a parameter in a SetValue action in a [macro](#). For example, set Amount to 24.50 and use the following in a SetValue action in a macro:

```
NumToWord(Amount)
```

[References](#)

PRB: Edit Menu Commands Unavailable in Design View

Article Number: Q95639
CREATED: 18-FEB-1993
MODIFIED: 12-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you press CTRL and click the mouse button in Design view, all Clipboard commands on the Edit menu are unavailable.

CAUSE

Dragging the mouse creates a contiguous selection. However, pressing CTRL and clicking the mouse button creates a discontinuous selection. The Cut, Copy, and Delete commands are unavailable for discontinuous selections, even if the objects you select are contiguous.

RESOLUTION

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open a table in Design view.
2. Select several fields by dragging the mouse on the row selectors.
3. Pull down the Edit menu. Note that the Cut, Copy, and Delete commands are available.
4. Close the Edit menu.
5. Select the same fields again by clicking the row selectors while holding down the CTRL key.
6. Pull down the Edit menu again. All the Clipboard commands are now unavailable.

References

PRB: Open Object Can Be Deleted by Another User

Article Number: Q95638
CREATED: 18-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

While one user has a [form](#) or [report](#) object open another user can delete the object with no warning.

CAUSE

Microsoft Access does not have a [record](#)-level locking mechanism for objects, therefore there is no way to prevent a user on one workstation from deleting an open object on another workstation. This problem does not occur with tables that have been opened exclusively.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

Use a shared [database](#) on two machines (A and B).

1. On machine A, open Formx for design.
2. On machine B, delete Formx.

Formx is deleted with no warning.

[References](#)

PRB: 'Unable to Load Communication Module'

Article Number: Q95612
CREATED: 17-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The following error message is displayed:

```
Unable to load communication module.  
Driver has not been properly installed.  
Unable to connect to data source.
```

CAUSE

The error occurs because the DBNMP3.DLL file you are using may be old, you may have duplicate DBNMP3.DLL files on your system and the wrong one is being used, or the DBNMP3.DLL file you are using may be damaged or corrupted.

This error can also be caused by an incorrect [SQLSERVER] entry in your WIN.INI file. This entry should look like:

```
[SQLSERVER]  
MY_SERVER=DBNMP3,\\MY_SERVER\PIPE\SQL\QUERY
```

If there are any spaces in the above entry, or if DBNMP3 is misspelled, you will receive the error.

RESOLUTION

In Microsoft Access version 1.0, the DBNMP3.DLL file installed by Microsoft Access version 1.0 is dated 9/15/92, has a size of 8241 bytes. In Microsoft Access version 1.1, the DBNMP3.DLL file is dated 5/26/93 with a file size of 9941 bytes. In both versions, this file should be installed within the \WINDOWS\SYSTEM directory.

MORE INFORMATION

It is possible that there are other situations that may indirectly cause the above error message to display. To troubleshoot this problem, do the following:

- Use File Manager to search for multiple, corrupted, or outdated copies of these files:

Filename	Date	Size

<u>ODBC</u> .DLL	10/16/92 or later	(44736 bytes)

SQLSRVR.DLL 10/16/92 or later (135792 bytes)
NETAPI.DLL depends on the [network](#) being used
COMMDLG.DLL 10/25/92 or later (89248 bytes)

Note: The sizes and dates referenced are correct if the files are installed by Access version 1.0.

- Make sure the files listed above are in their appropriate locations. All of the files except NETAPI.DLL are usually installed in the Windows \SYSTEM subdirectory. The NETAPI.DLL file is a network specific driver and is usually located in the network's program directory, which should be in the path.
- Use another application, such as Q+E, SAF, or PowerBuilder, on the same machine to attempt to attach to the same data source. If you cannot access the data source from another application, then it is probably a problem with the network, such as NETAPI.DLL, or a problem with the server, not with Microsoft Access or ODBC.
- Attempt to attach to another data source, if available, such as another SQL server. If you can attach to another server, then the original server may be down or you may need to reconfigure the entry for that server using the ODBC Administration Utility.
- Attempt to attach to the same data source from another machine. If you can't access the SQL server from another machine, it may indicate a network-wide problem or that the server is down or not communicating.
- Try increasing your ODBC timeout settings in the ODBC section of the MSACCESS.INI file.
- Check with the [system administrator](#) to see if the SQL server is up, available for transactions, and operating properly.
- Reinstall ODBC after performing a clean boot and renaming the ODBC.DLL, DBNMP3.DLL, and SQLSRVR.DLL files.

References:

Microsoft Access "User's Guide," version 1.0, pages 660-662

[References](#)

INF: Use a Query to Parse a Full Name into Two Fields

Article Number: Q95608
CREATED: 17-FEB-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

The following [query](#) parses a [field](#) containing a full name in which the last and first names are separated by commas. For example a name could be entered as: Smith, John. The query would return two columns named Last: and First:. The query uses the Left\$ and Right\$ functions and searches for the comma using the InStr [function](#).

Query: QueryTest

Field Name: Last: Left\$([Name],InStr(1,[Name],",")-1)
Show: True

Field Name: First: Right\$([Name],Len([name])-InStr(1,[Name],",")-1)
Show: True

You can modify the query to account for spaces in the name field. For example if the name field contained Smith,John with no spaces, you would remove the -1 for the first name field.

[References](#)

INF: Enforcing Transactions on Attached SQL Server Tables

Article Number: Q95607
CREATED: 17-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access will automatically support transaction processing on attached SQL Server tables using queries. However, explicit transactions in Access Basic require the use of dynasets on attached tables and further require a specific order in the creation of the dynasets and the implementation of transaction processing.

MORE INFORMATION

Microsoft Access will enforce transaction processing on an attached SQL Server table through the use of a dynaset created on that table. To get more information on how to create a dynaset on an attached SQL Server table, search for "CreateDynaset" using the Help menu. The key to making transaction processing work for attached SQL Server tables is to create and close the dynaset on the attached SQL Server table outside the transaction. Below are pseudo-code examples of the incorrect and correct methods of coding this process:

INCORRECT

```
Dim MyDyna As Dynaset
BeginTrans
    MyDyna = CreateDynaset("Table1")
    <misc.code such as Inserts/Updates/Deletes.>
    MyDyna.Close
CommitTrans/Rollback
```

CORRECT

```
Dim MyDyna As Dynaset
MyDyna = CreateDynaset("Table1")
BeginTrans
    <misc.code such as Inserts/Updates/Deletes.>
CommitTrans/Rollback
MyDyna.close
```

[References](#)

INF: Returning Case Sensitive Match in Queries

Article Number: Q95605
CREATED: 17-FEB-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access can differentiate between case sensitive text strings using the Asc() [function](#). Using this function in a select [query](#), Microsoft Access can locate an exact case sensitive match.

MORE INFORMATION

This technique is useful in a situation where the user needs to find all the records in a [field](#) that contain lowercase text rather than uppercase text for the first character. An example of this might be in the Employees [table](#) of the NWIND.MDB sample [database](#). Please note that the example below will find only the first character. If you need something more complex, you must call an [Access Basic](#) function to return the results.

Suppose the text field called First Name contains values such as the following:

```
First Name
-----
andrew
Andrew
Nancy
```

If you need to find all the records that contain the lower case "andrew" as compared to "Andrew," use the Asc() function. To do this, create a new [select query](#):

```
Query: QueryTest
-----
Field name: First Name
Show: True
Field name: Asc([First name])
Show: True
First Criteria Line: Like Asc("a")
```

This query will return only the [record](#) for "andrew," not "Andrew."

References:

Microsoft Access "Language Reference," version 1.0, page 40

Microsoft Access "User's Guide," version 1.0, chapter 6

[References](#)

PRB: Error Message: Query Must Have at Least One Output Field

Article Number: Q95573
CREATED: 17-FEB-1993
MODIFIED: 09-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

When you try to execute a [query](#), you receive the following error message even though the query clearly has fields in the query grid:

Query must have at least one output [field](#)

CAUSE

One of the fields in the query grid has a colon as the right-most character in its name (for example, LastName:). The query parser looks for colons in a name to denote calculated fields and misinterprets the field name as a result.

This happens only if the field name with the colon is the only field or if all fields have colons.

RESOLUTION

Open the source [table](#) in [Design view](#) and remove the colon from the field name. Changing the field name does not affect the data stored in the field.

OR

Place square brackets around the field name in the query grid.

STATUS

This problem does not occur in Microsoft Access version 1.1.

[References](#)

INF: Recent File List Displays Deleted .MDB Files

Article Number: Q95569
CREATED: 17-FEB-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

When you open a [database](#), Microsoft Access remembers that database name and will display the name in the Recent File List under the File menu. However, if you delete that database, the Recent File List will continue to show it as a valid database even though it doesn't actually exist. When you open other databases the list will be updated, and the deleted file will eventually disappear from the list.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new database.
2. Close the database.
3. Delete the database using File Manager.
4. Choose the File menu in Microsoft Access.

Notice the database you just deleted is still in the File menu list.

[References](#)

PRB: No Field List When Form Based on SQL Statement

Article Number: Q95566
CREATED: 17-FEB-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

An attempt to set the RecordSource for a [form](#) to an [SQL](#) statement fails and no fields appear in the Field List [dialog box](#).

CAUSE

Microsoft Access does not support using an SQL statement as the [record](#) source for a form.

RESOLUTION

Create a [query](#) based on the SQL statement and set the RecordSource for the form to the query.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the NWIND.MDB sample [database](#).
2. Open the Category List query in Design mode.
3. Choose SQL from the View menu to display the SQL statement for this query.
4. Select the contents of the SQL window. Copy the SQL statement to the [Clipboard](#) with the CTRL+C key combination.
5. Choose Cancel to close the SQL dialog box, then close the Category List query.
6. Open a new unbound form. Paste the SQL statement into the form's RecordSource property with the CTRL+V key combination.
7. To display the [field list](#), choose Field List from the View menu. Note that even though the SQL statement displays in the dialog box [title bar](#), no fields are listed to choose from.

[References](#)

PRB: Clicking Mouse in AccessWizard Background Changes Focus

Article Number: Q95565
CREATED: 17-FEB-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When use you the Form Wizard or the Report Wizard, you can change the input focus by clicking the mouse in different portions background in the Form Wizard or Report Wizard window. For example, if you click the mouse at the bottom of the Form Wizard (in the area that contains the "Cancel," "Back," and "Next" buttons), Microsoft Access gives the input focus to a control in that colored area.

CAUSE

The input focus changed because the Form Wizard and Report Wizard are forms that contain a Detail section and a Footer section. Clicking the mouse in the Footer section (that contains the buttons) moves the input focus to the first enabled control in this section (the tab order determines which control is first). However, the input focus does not move if you click the mouse in the section that contains a control that already has the input focus.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the NWIND.MDB sample database.
2. Choose Tables from the View menu.
3. Select the Categories table from the Database window.
4. From the File menu, choose New then choose Form.
5. Choose the FormWizards button. (If the FormWizard is not installed, the button is dimmed and unavailable.)
6. Select Single-column and then choose OK.
7. Note that the Next button does not have the focus.
8. Click the in the area at the bottom of the form (not on any of the buttons) and the focus shifts to the Next button.

References

INF: How to Send the Contents of a Control with SendKeys

Article Number: Q95455
CREATED: 09-FEB-1993
MODIFIED: 21-APR-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

Summary:

To specify the contents of a control or field in the SendKeys action argument, you must precede the control or field name with an equal sign (=) in the Keystrokes argument.

More Information:

The following example will perform a SendKeys action and send the contents of Field0 as the argument to the action.

Action	Keystrokes
SendKeys	=Forms!Form1!Field0

[References](#)

INF: Using Multiple Copies of the Same Table in a Query

Article Number: Q95454
CREATED: 09-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

When designing queries, you may at some point need to make multiple references (links) between a column in one table and different columns in another table. If you set up the links incorrectly, the query may return unexpected results or no results at all.

To build the links correctly, you should load multiple copies of the common table. Then you can reference columns from the other copies as needed. This is all possible because in Microsoft Access, you can add multiple copies of the same table to a query.

MORE INFORMATION

```
When you add a second copy of a table to a query, Micro<>UPDATE -  
  /TITLE="INF: Recent File List Displays Deleted .MDB Files  
[B_WAccess]" -  
  /UNIQUE_IDENTIFIER="Q95569" -  
  /BADGE=(AUTHOR="000162", ENTER="000000", MODIFY="000855", -  
    EDITORIAL_REVIEW="007006", TECHNICAL_REVIEW="007006") -  
  /NAME=(AUTHOR="LISAPR", ENTER="", -  
    MODIFY="SUSANCO", EDITORIAL_REVIEW="MARKM", -  
    TECHNICAL_REVIEW="MARKM") -  
  /DATE=(AUTHOR="17-FEB-1993", ENTER="17-FEB-1993 23:00:26.00", -  
    EXPIRE="1-JAN-2000", FLASH="17-NOV-1858", -  
    MODIFY="21-APR-1993 09:51:55.00", -  
    EDITORIAL_REVIEW="5-APR-1993", -  
    TECHNICAL_REVIEW="5-APR-1993") -  
  /MESSAGE="" -  
  /FLAGS=(CUSTOMER_READABLE, NODISPLAY_MESSAGE, -  
    EDITORIAL_REVIEWED, FIELD_READABLE, NOFLASH, -  
    TECHNICAL_REVIEWED, NOREADY)
```

1.00 1.10
WINDOWS
ENDUSER |

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

=====

When you open a [database](#), Microsoft Access remembers that database name and will display the name in the Recent File List under the File menu. However, if you delete that database, the Recent File List will continue to show it as a valid database even though it does not actually exist. When you open another database the list will be updated, and the deleted file will eventually disappear from the list.

MORE INFORMATION

=====

Steps to Reproduce Behavior

1. Create a new database.
2. Close the database.
3. Delete the database using File Manager.
4. Choose the File menu in Microsoft Access.

Notice the database you just deleted is still in the File menu list.

[References](#)

PRB: Moving Backward in Report Print Preview Window Is Slow

Article Number: Q95453
CREATED: 09-FEB-1993
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

In the Print Preview window in reports, you can move forward and backward page by page to view your [report](#) before printing it. When previewing a large report, you may notice it takes longer to move backward than it takes to move forward. The further along you are in the report [preview](#), the longer it takes to move backward.

CAUSE

Microsoft Access must reformat the report pages behind it when moving backward. In other words it reformats all the pages from page 1 to the current page minus 1.

As you preview a report in the Print Preview window, Microsoft Access formats the report pages. It keeps the page formats in memory until you move to a new page. Once you move past a page, all the formats constructed to show that page are lost. Therefore, previous pages must be reconstructed each time you move backward.

The more pages there are between page 1 and your current page location, the more time it takes Microsoft Access to reconstruct them. For example, it takes less time to go from page 3 to page 2 than it takes to go from page 300 to page 299. In the first example, only two pages need to be reformatted. In the second, 299 pages need to be reformatted.

RESOLUTION

Rather than previewing or printing a report in Microsoft Access, you might consider saving it in a file format that can be read by another program such as a spreadsheet or word processor and then use the spreadsheet or word processor to browse the report.

The ability to save a report in different file formats is not built into Microsoft Access. However, you can use the OutputAs utility that comes as two files -- OUTPUTAS.MDA and OUTPUTAS.DLL -- to save a report in other formats.

You can download OUTPUTAS.MDA and OUTPUTAS.DLL from the reports section of CompuServe library 5. At this time, this is the only place to get them. Please read the README.TXT file included with the OUTPUTAS.MDA and OUTPUTAS.DLL files for detailed instructions on how to use it.

After installing OUTPUTAS.MDA and OUTPUTAS.DLL, you will have a Save Report As command added to your Microsoft Access Help menu. You can choose Save Report As to save report output in the following file formats:

- BIFF (binary interchange file format) -- a Microsoft Excel version 3.0 and higher format.
- RTF (rich-text format) -- a Microsoft standard document interchange format.
- ASCII (American Standard Code for Information Interchange) -- a text-only format.

Once installed, you can choose the Output Report As command from the Help menu, select a report from the list it gives you, and then choose to output the report in RTF format. Then you can use Microsoft Word for Windows to examine and edit the resulting .RTF file.

[References](#)

PRB: Property List Boxes Show Two-Word Options on Two Lines

Article Number: Q95452
CREATED: 09-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

All the list boxes that show properties use two lines to display two-word options instead of putting each two-word option on a single line.

CAUSE

There is an incorrect List Separator in the International section of your WIN.INI file.

RESOLUTION

From the Main group in Program Manager, choose Control Panel. From there, choose the International icon. Change the List Separator to a comma (,). Now the Property list boxes in Microsoft Access will list two-word options correctly. Note that different List Separator entries will cause different results.

MORE INFORMATION

Steps to Reproduce Behavior

1. From the Accessories group in Program Manager, choose Notepad.
2. From the File menu, choose Open.
3. Type C:\WINDOWS\WIN.INI, and choose the OK button. If your Windows directory is different, modify the path accordingly.
4. From the Search menu, choose Find.
5. Type slist, and choose the Find Next button.
6. Change the line in the WIN.INI file to look like this:

```
slist=" "
```
7. Save the file, and restart Windows.
8. Start Microsoft Access and open any form in Design view.
9. Choose Default View and look at the valid choices in the list box by clicking the down arrow.

Your displayed choices will look like this:

```
Default View....Single
                  Form
                  Continuous
                  Forms
                  Datasheet
```

The list should look like this:

```
Default View....Single Form
                  Continuous Forms
                  Datasheet
```

[References](#)

PRB: Can't Change Combo Box Value If ControlSource Is Counter

Article Number: Q95450
CREATED: 09-FEB-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

After setting the ControlSource property of a combo box to a field that has a Counter data type, you try to change the value in the combo box and receive the following status bar message:

Control bound to counter column 'Field name'

CAUSE

The ControlSource of a combo box is a table field where Microsoft Access writes your combo box selection. By design, modifications to a Counter field are not allowed. Therefore, if the ControlSource property of a combo box is set to a Counter field, Microsoft Access cannot update the value in the Counter field, so you cannot change the combo box value. Microsoft Access gives you the message to tell you why it cannot update the combo box control.

RESOLUTION

Either change the data type of the field now in the ControlSource property of the combo box, or set the ControlSource property to a field that does not have a Counter data type.

MORE INFORMATION

Steps to Reproduce Behavior

1. Start Microsoft Access, open the NWIND.MDB sample database, and create a new table called MyTest.
2. Add the following three fields to the MyTest table:

Field Name	Data Type
MyCounter	Counter
MyData	Number
MyBucket	Number

3. Enter a few random numeric values in the MyData field. Notice that Microsoft Access automatically fills in the MyCounter field and puts zeroes in the MyBucket field as you enter values in the MyData field.
4. Create a new blank form.

5. In Design view, place a combo box on the form. Set its ControlSource property to the MyCounter field, and its RowSource property to the MyTest table.
6. Switch to Form view. Notice that the combo box values are the sequential numbers coming from the MyCounter field.
7. Drop down the combo box list and attempt to select a new value from it. You receive the following message on the status bar:

Control bound to counter column 'The Counter'.

The value in the combo box does not change.

8. Switch back to Design view and change the combo box ControlSource property from MyCounter to MyBucket.
9. Switch back to Form view and repeat step 7. Now you can select items from the combo box. If you look at the contents of the table, you can see the values you select come from the MyBucket field.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 9, "Designing Forms," pages 235-238.

[References](#)

PRB: Wait Argument in SendKeys May Not Stop Macro Processing

Article Number: Q95449
CREATED: 09-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When a [macro](#) calls the SendKeys action with the Wait parameter set to Yes, the macro will continue before the program that processed the keystrokes has finished performing the action that it was called to do.

RESOLUTION

To work around this issue, have the macro call an [Access Basic function](#) or procedure that checks to see if the other application is finished. To correct the situation described above, you can do the following:

Macro Action

- | | |
|---------------------|---|
| RunApp | - Start Excel |
| SendKeys | - Start the macro |
| RunCode | - Call a module that continues checking until the file exists |
| TransferSpreadsheet | - Import the spreadsheet |

MORE INFORMATION

For example, suppose you want to create a macro that will call Excel to make a file and then process that file once Excel has created it. The Access macro calls RunApp to start Excel, then calls SendKeys to start an Excel macro with the Wait parameter set to Yes. Access will wait until the key strokes are processed, then continue if the Excel macro has not finished creating the file. Access will NOT wait for Excel, and will fail when it tries to use the file.

The SendKeys action in Access has two arguments: the first argument indicates which keystrokes to send; the second argument causes Access to pause until the keystrokes are processed.

The description of the Wait argument on page 432 of the "Language Reference" is misleading. If Wait is Yes, Access will only wait until the keystrokes are processed. It will not wait until the application that the keystrokes were sent to is finished.

References:

Microsoft Access "Language Reference," version 1.0, pages 433-435

References

PRB: Bitmap May Not Display Correctly

Article Number: Q95448
CREATED: 09-FEB-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A [bitmap](#) that displays correctly in another application does not display correctly in Microsoft Access.

CAUSE

When Microsoft Access starts, it loads a fixed 256-color palette. This color palette supports the most common colors used in bitmaps and metafiles. All pictures and [OLE](#) Objects displayed in Microsoft Access are mapped to this color palette.

STATUS

This behavior is by design.

MORE INFORMATION

By using a fixed color palette, Microsoft Access can display multiple pictures and OLE objects on the screen at one time without each new object affecting the screen colors.

Windows uses only one palette at a time. This is usually not a problem because most applications, such as PBRUSH.EXE, display only one bitmap at a time, so they can use a single Windows palette to display the bitmap.

However, Microsoft Access can display multiple bitmaps at one time. Therefore, Because Windows can use only one palette at a time, Microsoft Access uses a common palette that hopefully will be common to all the bitmaps Microsoft Access might display.

This is why bitmaps and metafiles displayed in Microsoft Access may be colored differently than they were in the application in which they were created. Because Windows can use only one palette at a time, each object Microsoft Access displays must be mapped to the fixed color palette loaded by Microsoft Access. Microsoft Windows determines this mapping. You cannot modify it.

Steps To Reproduce Behavior

1. Start Microsoft Paintbrush (PBRUSH.EXE).
2. Choose Image Attributes and select Black and White under the Color option.

3. Create a simple picture using several different shades of gray.
4. Choose Save from the File menu.
5. Under the Save File as Type: option, select 256 Color Bitmap.
6. Start Microsoft Access, and open the example [database](#) NWIND.MDB.
7. Open the Categories [form](#) in [Form view](#).
8. Select the Picture Control.
9. Choose Insert Object from the Edit menu, and click the File button.
10. Choose the file you just created in Microsoft Paintbrush, and click the OK button.

Result: the file loads as a solid black picture. Microsoft Paintbrush uses a custom palette to offer a variety of shades of gray. But Microsoft Access does not have the ability to load or reproduce this custom color palette, so it requests that Windows map each of the colors in the bitmap to the custom palette that was loaded when Microsoft Access started.

This example is extreme. In most cases the color change will be slight.

[References](#)

PRB: '|' Isn't A Valid Path Error Message on Start

Article Number: Q95447
CREATED: 09-FEB-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

When trying to start Microsoft Access, you receive the following error message:

```
'|' Isn't a Valid Path
```

CAUSE

The drive that contains your TEMP directory does not have enough disk space, or there is no valid SET TEMP statement in the AUTOEXEC.BAT file.

RESOLUTION

Exit from Windows. At the MS-DOS prompt, type SET and press ENTER. This gives you a listing of the MS-DOS environment settings. Confirm that one of the settings is TEMP= followed by a valid path.

If you find the TEMP= setting, then check the amount of disk space available on the drive where your temporary directory resides. If you do not have enough disk space, delete unnecessary files from the drive or move the TEMP directory to a partition that has enough disk space and change the SET TEMP= line in the AUTOEXEC.BAT file.

STATUS

This problem does not occur in Microsoft Access version 1.1.

[References](#)

PRACC9302: Importing Delimited Text Files Strips Off Decimals

Article Number: Q95446
CREATED: 09-FEB-1993
MODIFIED: 31-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You import a delimited text file that contains numbers with decimal places, and the resulting table does not show the decimal places.

CAUSE

When Microsoft Access creates the table, it looks at the first record in the file to decide which data type to assign to the field. If the first record contains a .00 decimal value in a field, Microsoft Access gives the field a Number, Long Integer data type. Then it strips all the decimal values when importing the file because the Number, Long Integer data type does not support decimal values.

RESOLUTION

There are two workarounds. The first is to create a destination table that already has the fields defined. Choose Number, Double as the data type. It supports decimal places. When you import the text file, append it to the existing destination table rather than allowing Microsoft Access to create a new table.

Alternatively, you can work around the problem by making sure the number fields in the first record do not contain .00 as decimal values.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

If any of the fields in the first record contain numbers that have decimal values of .00, Microsoft Access strips the decimal places from all the records for the corresponding fields in the resulting table.

Steps to Reproduce Behavior

-
1. Use a text editor such as Notepad or MS-DOS Edit to create two text files (FILE1.TXT and FILE2.TXT) using the following as a guide:

```
FILE1.TXT
-----
234.50
350.75
```

```
FILE2.TXT
-----
100.00
276.93
```

2. Import the first file as a delimited text file and have Microsoft Access create a new table. Notice that the resulting table has the numbers in the correct format. The data type for this field was defined as Number, Double.
3. Import the second file as a delimited text file and have Microsoft Access create a new table. Notice that the resulting table imports the numbers as follows:

```
100
276
```

Microsoft Access stripped the decimal places from the second file because the data type for the field was defined as Number, Long Integer -- a data type that does not support decimal values.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 4, "Importing, Exporting, and Attaching," pages 72-78.

[References](#)

INF: Updating a Separate Table When a Value Changes on a Form

Article Number: Q95445
CREATED: 09-FEB-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article shows by example how to have Microsoft Access automatically update a field in a table when you use a form to change a related field in a different table. In other words, you can have Microsoft Access post your changes to a separate table that is not bound to the form.

To set this up, you need to:

- Add a text box control to the form.
- Create a new macro group (UpdateOther) that contains two macros (SaveValue and ChgValue) to run the update query.
- Create an update query (IDChgQuery in this example).
- Change several properties to execute the macros.

MORE INFORMATION

This example uses the Products form in the sample database NWIND.MDB. Here is the step-by-step procedure:

1. Open the NWIND.MDB database, and make a few modifications to set it up for this example. First, delete the relationship between the Products and Order Detail tables. Then, because you cannot update fields that have a Counter data type, change the following properties of the Product ID field in the Products table:

```
Table: Products
-----
Field Name: Product ID
  DataType: Number
  FieldSize: Long Integer
```

2. Open the Products form in Design view, and display the property sheet. Set the following form and control properties:

```
Form: Products
-----
OnCurrent: UpdateOther.SaveValue

Field: Product ID
-----
AfterUpdate: UpdateOther.ChgValue
```

Locked: No

These actions activate the macros, which in turn run the update query.

3. Add an unbound text box control to the form, and give it the following properties:

```
Control: Text box
-----
ControlName: Previous ID
Visible: No
```

4. Create a new macro group (UpdateOther) to hold two macros (SaveValue and ChgValue) by first choosing to create a new macro. Next, choose Macro Names from the View menu or click the Macro Names button on the toolbar. Microsoft Access displays the Macro Name column.
5. Enter the two macro names along with their actions and action arguments. Each macro in the macro group begins on the line that contains that macro's name. Use the following table as a guide:

Macro Name	Action and Action Arguments

SaveValue	SetValue Item: [Previous ID] Expression: [Product ID]
ChgValue	SetWarnings Warnings On: No OpenQuery Query Name: IDChgQuery View: Datasheet Data Mode: Edit SetValue Item: [Previous ID] Expression: [Product ID]

6. Save the macro group, and name it UpdateOther. Now UpdateOther appears in the list of macros in the Database window. You can use the following syntax to specify each macro in the macro group:

```
macrogroupname.macroname
```

For example, UpdateOther.SaveValue specifies the save value macro.

7. Create a new update query called IDChgQuery. Add the table Order Details to the query. In the Field cell, enter Product ID. In the Update To cell, enter Forms![Products]![Product ID]. In the Criteria cell, enter Forms![Products]![Previous ID]. Here is a SUMMARY

```
Query: IDChgQuery
-----
Tables: Orders Detail
Field: Product ID
Update To: Forms![Products]![Product ID]
Criteria: Forms![Products]![Previous ID]
```

Now, when using the Products form, which is bound to the Products table, if you enter a new value in the Product ID field, Microsoft Access updates the Product ID column in the Orders Detail table automatically. It does it by running the UpdateOther.SaveValue macro to save the previous value to use as criteria and then the UpdateOther.ChgValue macro to run the IDChgQuery update query. The IDChgQuery query updates the Product ID column in the Orders Detail table overwriting the previous old value with the new value.

[References](#)

INF: Creating Floating Popup Menus

Article Number: Q95444
CREATED: 09-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The text below discusses how to create "floating" popup menus using a combination of functions in the Microsoft Windows application programming interface (API) and [Access Basic](#) code.

MORE INFORMATION

To create a floating popup menu, perform the following eight steps.

1. Create a new [form](#) called "Popup Menu Form."
2. Add a List Box [control](#) called "Menu" to the form. Make the control large enough to display each of the strings it contains without displaying a scroll bar. Give the control the following properties:

List Box Properties

```
ControlName:  Menu
RowSourceType: Value List
RowSource:    String1;String2;String3;String4
AfterUpdate:  =ItemSelected([Menu])
Left:        0 in
Top:         0 in
FontName:    System
FontSize:    8
FontWeight:  Bold
```

NOTE: Specify the menu strings in the RowSource property. Separate succeeding [string](#) values with a semicolon (;) character.

3. Change the properties of the "Popup Menu Form" as indicated below.

Form Properties

```
ScrollBars:  Neither
Popup:       Yes
RecordSelectors: No
Width:       <Width of the list box control>
```

Section Properties

```
Height:     <Height of the list box control>
```

4. Save the form and close it.

5. Create a new module and place the following code into the Declarations section for the module.

```
'NOTE: Some of the following Windows API functions may be
'defined in an existing Microsoft Access library. If so, the new
'declarations would cause a duplication procedure name error. If
'this error occurs, remove the offending declare statement from
'your code or convert the declaration to a comment.
```

```
'NOTE: In the following sample code, an underscore (_) is used
'as a line continuation character. Remove the underscore when
're-creating this code in Access Basic.
```

```
Option Explicit
Type POINTAPI
    x As Integer
    y As Integer
End Type

Global Const GWL_STYLE = (-16)
Global Const WS_DLGFRAME = &H400000

Declare Sub GetCursorPos Lib "User" (lpPoint As POINTAPI)
Declare Function GetWindowLong& Lib "User" (ByVal hWnd%, _
    ByVal nIndex%)
Declare Function SetWindowLong& Lib "User" (ByVal hWnd%, _
    ByVal nIndex%, ByVal dwNewLong&)
```

6. Add the following functions to the module.

```
Function ShowPopup ()
    Dim coord As POINTAPI
    Dim attr&

    GetCursorPos coord
    DoCmd OpenForm "Popup Menu Form"

    attr& = GetWindowLong(Forms![Popup Menu Form].hWnd, GWL_STYLE)
    attr& = SetWindowLong(Forms![Popup Menu Form].hWnd, GWL_STYLE, _
        attr& And Not WS_DLGFRAME)

    DoCmd MoveSize (coord.x * 14), (coord.y * 14), , 1100
End Function

Function ItemSelected (WhichItem As String)
    DoCmd Close
    MsgBox "The selected item was " & Trim(WhichItem)
End Function
```

7. Choose the form in which you want the popup menu to appear. Open the form in Design mode.
8. Select the event property that you want to use to activate the popup menu and specify the following function call:

```
=ShowPopup()
```

When you open your form and cause the event chosen in Step 8 above, the popup menu appears at the current mouse position. The menu remains on the screen until you select an item from the popup menu.

[References](#)

PRB: Focus Returns to Control After OnDblClick

Article Number: Q95443
CREATED: 09-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you assign an [event](#) ([function](#) or [macro](#)) to the OnDblClick property of a [control](#), the event is processed and the focus is returned to the control you double-clicked on. Even if the event contains a GoToControl action, the focus will still be returned to the control that you double-clicked.

CAUSE

This behavior is by design.

RESOLUTION

If you would like the focus to move to another control, you must use the CancelEvent action after the GoToControl action in the macro or function.

MORE INFORMATION

The following is an example of a macro that can be attached to the OnDblClick property of a control (Field1) and used to move the focus to another control (Field2).

Macro Name	Action	Argument
Test1	GoToControl	Field2
	CancelEvent	(no arguments)

The GoToControl action moves the focus to Field2, and the CancelEvent action cancels the double-click action. Thus, the focus remains on Field2.

[References](#)

PRB: Valid Sum Function Returns '#Name?' in Calculated Control

Article Number: Q95442
CREATED: 09-FEB-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you open a [form](#) in [Form view](#) and have two or more calculated controls, you may get "#Name?" in each of the calculated controls on the form, even if the [expression](#) for that [control](#) is valid.

CAUSE

All of the [domain](#) functions are based on the same [query](#) (over the underlying [dynaset](#)). If any of the bound functions on the form has a binding error, all of the functions on the form return an error.

RESOLUTION

Find the control(s) that uses an invalid expression. Either remove the control or correct the expression being used for the control.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new form based on the Orders [table](#) from the sample [database](#) NWIND.MDB.
2. Create a [text box](#) bound to the Order Amount [field](#) of this table. Set the following properties:

```
Text Box
-----
ControlName:  Order Amount
ControlSource: Order Amount
```

3. Create an unbound text box on the form. Set the following properties:

```
Text Box
-----
ControlName:  Test1.
ControlSource: =Sum([Order Amount])
```

4. Create another text box on the form. Set the following properties:

```
Text Box
```

```
-----  
ControlName:    Test2.  
ControlSource:  =Sum([Test1])
```

5. View the form in Form view.

#Name? will appear in the second and third text boxes (Test1 and Test2). Because you cannot use the Sum function on a calculated expression, Test2 is invalid. This causes all calculated expressions on the form to return "#Name?."

If you remove Test2 from the form, Test1 will display the correct value.

[References](#)

PRB: Bound SubReport Does Not Show DDE Data

Article Number: Q95441
CREATED: 09-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The result of a DDE expression in a subreport does not display in Print Preview or when a report is printed.

CAUSE

If the subreport is bound, there must be at least one row of data in the underlying table or dynaset before the subreport's DDE data will display.

RESOLUTION

There are three possible resolutions:

- Hide the DDE expression if the dynaset for the subreport is empty.
- Place the DDE expression in an unbound subreport.
- Place the control that contains the DDE expression in the main report.

MORE INFORMATION

Steps to Reproduce Behavior

1. Start Microsoft Excel and create a new worksheet. Place the text "This is a test" in row 1, column 1. Save this worksheet as DDETEST.XLS, and leave this worksheet open.
2. In Microsoft Access, create a new table:

 Table: Table1

 Field Name: TXT
 DataType: Text
3. Create a new report bound to [Table1].
4. Add a text box bound to the field [TXT] by dragging [TXT] from the Field List to the report.
5. To add an unbound text box, select the text box tool from the

ToolBox and click on the report. Put the following expression in the ControlSource property of this unbound text box:

```
=DDE("Excel","DDEtest.xls","R1C1")
```

6. Save this report as "SubRep" (without the quotation marks).
7. View [SubRep] in Print Preview. "This is a test" displays correctly.
8. Close [SubRep] and create a new report. Drag the report, [SubRep], from the Database window onto this new report. Save this report as [MainRep].
9. View [MainRep] in Print Preview. "This is a test" is not visible on the report. Close [MainRep].
10. Open [Table1] and add some data to it. View [MainRep] in Print Preview. Notice that "This is a test" is now visible.

[References](#)

INF: Eliminating White Space in Reports with CanShrink and

Article Number: Q95390
CREATED: 08-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Before printing a [form](#) or [report](#), you can save space by conditionally reducing the sizes of the controls on the form or report. For example, you might want to reduce the size of text boxes that contain varied-length text or reduce the size of text boxes bound to fields that could be empty.

Note the following information before you try to reduce the size of controls in reports:

- Overlapping controls will not shrink, even when the CanShrink property is set to Yes.
- Controls shrink line by line (vertically). This means, for example, that if there are a [group](#) of controls placed on the left side of the page and a large [control](#) (for example, an [OLE](#) picture) on the right side of the page, the controls on the left side of the page will not shrink, unless the picture is blank and hidden.
- Space between controls is not affected by the CanShrink or CanGrow properties.
- Controls located in the Page Header or Page Footer will not shrink or grow.

MORE INFORMATION

A control containing null information automatically shrinks to nothing and disappears when its CanShrink property is set to Yes, but the space between the controls is not affected by the CanShrink property.

For example, say you have 11 controls in a [column](#) of a report, and there is .1 inch of space between each control. Even if all 11 controls contain null information and all 11 have their CanShrink property set to Yes, the space between the controls adds up to 1 inch. When you print the report, even though the 11 controls do not print, the first control that does contain text will print 1 inch lower than you might expect because of all the accumulated space.

A good way to manage the size of your controls is to have as few as possible. This will minimize the white space between the controls.

The following example demonstrates how to print mailing labels by using an [Access Basic function](#) to minimize the number of controls and give you nice, even spacing. Say you want to create mailing labels containing a

name, business name, address, city, state, and zip code. You could use several different fields, but it would be better to use a single [text box](#), as in the following step-by-step example:

1. Create a single text box control and set its ControlSource property to this [expression](#):

```
=AddressBlock([Name],[Business Name],[Address],[City],[State],[ZipCode])
```

2. Create a new Access Basic [module](#), and type the following code into it:

```
Option Explicit
Function AddressBlock$ (AName, Addr1, Addr2, City, State, Zip)
    Dim A1$, A2$, A3$, A4$, CR$

    CR$ = Chr(13) & Chr(10) 'Carriage return and line feed

    A1$ = IIf(ISB(AName), "", AName & CR$)
    A2$ = IIf(ISB(Addr1), "", Addr1 & CR$)
    A3$ = IIf(ISB(Addr2), "", Addr2 & CR$)
    A4$ = City & ", " & State & " " & Zip

    AddressBlock = A1$ & A2$ & A3$ & A4$ 'Concatenate the strings.
End Function

Function ISB (V) As Integer
    If IsNull(V) or V = "" Then ISB = True Else ISB = False
End Function
```

Now you can run the module to print the labels and keep wasted space to a minimum.

[References](#)

PRB: Error Using Edit Find with Yes/No Data Type

Article Number: Q95389
CREATED: 08-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you are using the Find command from the Edit menu on a form with a check box control bound to a field with a Yes/No data type, the Find will not work correctly. When you enter Yes or No in the Find What box and execute the Find, the following message is displayed:

Microsoft Access has reached the end of the dynaset.

CAUSE

Although it appears that the check box is receiving the focus, the actual focus is on the label for the check box. The check box will never receive the focus, so you cannot use the Find function on the check box. The Find function will work correctly if the control is not a check box.

RESOLUTION

Do not use a check box if you want the ability to search on this field. You can search on a Yes/No data type if it is a text box.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a form with a check box control using the Yes/No data type.
2. Switch to Form view, and click in the new check box.
3. From the Edit menu, choose Find.
4. Enter Yes or No in the Find What box. Choose the Find Next button.

The above error message will be displayed without any values being located.

[References](#)

INF: Doc Err: dBASE SET DELETED Characteristic in MS Access

Article Number: Q95388
CREATED: 08-FEB-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you delete a record in a dBASE or Microsoft FoxPro database, the record is actually only marked for deletion. It is not permanently removed until the database is packed in dBASE or FoxPro.

This characteristic can produce confusing effects when you open a table in Datasheet view.

MORE INFORMATION

In dBASE or FoxPro, records that are marked for deletion still appear in any output form, such as a table in Datasheet view or a report in Print Preview, unless you SET DELETED to ON. Because of this, Microsoft Access displays such records in an attached dBASE or FoxPro table, even if the records are marked for deletion.

To reproduce this characteristic in Microsoft Access, use the following steps:

1. Attach the dBASE table called NEWCUST.DBF. This table should be present in your Access program directory if you installed the sample databases with Microsoft Access.
2. Open the NEWCUST table in Datasheet view. Note that there are 13 records in this table.
3. Select the last record in the table by clicking on the record selector on the leftmost side of the datasheet.
4. Press the DEL key. Note that the record disappears from the datasheet, leaving 12 records in this table.
5. Close the Datasheet window and reopen it. Note that all 13 records reappear.

It may seem that Microsoft Access deleted the record, then undeleted it when you reopened the table. In reality, Microsoft Access marked the record for deletion, but because the marked record still existed in the table, Microsoft Access continued to display it in datasheet.

To prevent this problem from occurring, Microsoft Access provides functionality identical to the dBASE and FoxPro SET DELETED command. To take advantage of this functionality, be sure that you set Deleted to On if you do not want records that have been marked for deletion

to show. Use the following procedure to do this:

1. Open MSACCESS.INI in Windows Notepad.
2. Find the section called [dBASE ISAM].
3. Change the setting...

Deleted=Off

-to-

Deleted=On

4. Save the MSACCESS.INI file and restart Microsoft Access.

When you view the dBASE data, records marked for deletion will no longer appear. To permanently remove records marked for deletion, you must open the dBASE table in dBASE format and run the PACK command.

dBASE is manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

PRB: CTRL+HOME in List Box Moves to First Item in List

Article Number: Q95387
CREATED: 08-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you press CTRL+HOME in a list box, the cursor moves to the first item in the list box, not the first field in the first record.

However, when you search for "navigation keys," then "Navigation Keys -- Datasheets and Grids" using the Microsoft Access Help menu, you see that Help says CTRL+HOME has two purposes. You can press it to move to the first field in the first record, or in a multiple-line text box, you can press it to move to the beginning of the first line.

CAUSE

The key combination CTRL+HOME is an edit key combination, so in a list box, the first use (moving to the beginning of the field) has precedence over the second use (moving to the first field in the first record). Because of this, the focus remains in the list box rather than moving to other fields or controls.

STATUS

This behavior is by design.

[References](#)

INF: How to Determine Whether a Form Is Minimized

Article Number: Q95329
CREATED: 07-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not include an intrinsic [function](#) to indicate that a [form](#) is minimized. Use the IsIconic() function in the Microsoft Windows Application Programming Interface (API) to determine that a form is minimized.

MORE INFORMATION

The following code example demonstrates declaring and calling the IsIconic() function from [Access Basic](#). The argument to IsIconic() is the handle of the form the state of which you would like to know. IsIconic() returns 1 if the form is minimized or 0 (zero) if the form is not minimized.

```
'=====
'GLOBAL DECLARATION SECTION
'=====
Option Explicit
Declare Function IsIconic% Lib "User" (ByVal hwnd%)

'=====
' w_IsIconic()
'
' This function returns the window state of a form.
'
' Return value:
' 1 - Form is minimized.
' 0 - Form is not minimized.
'=====

Function w_IsIconic ()
    w_IsIconic = IsIconic(Forms!Form1.hwnd)
End Function
```

References:

Microsoft Windows Software Development Kit, "Programmer's Reference, Volume 2: Functions," version 3.1, pages 555-556.

[References](#)

PRB: User with Full Permissions Cannot Change a Query

Article Number: Q95328
CREATED: 07-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

After changing a [query](#) definition, you try to save it but cannot even though you have full [permissions](#), including the fact that you are a member of the Admins [group](#).

CAUSE

Queries have a property called Run with Owner's Permissions. You can see it in the Query Properties [dialog box](#). If it is checked, only the owner of the query can save it. Allowing someone else to make changes would be a [security](#) violation.

RESOLUTION

Nobody other than the query owner (the person who created it) can save the query as long as this property is set. To allow anyone with Modify Definition permissions to save changes to a query, you need to clear the Run with Owner's Permissions [check box](#).

References:

"Microsoft Access User's Guide," version 1.0, Chapter 25,
"Administering a Database System," pages 625-629.

[References](#)

PRACC9302: Error When Import dBASE, FoxPro, or Paradox Table

Article Number: Q95327
CREATED: 07-FEB-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you import a dBASE, FoxPro, or Paradox [table](#) without including the file extension, Microsoft Access displays this error:

```
Couldn't find object '<filename>'
```

RESOLUTION

Rename the file so that it has an extension. It does not have to have the correct extension, such as .DBF for a dBASE table, but it must have some sort of extension.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Rename a dBASE, Paradox, or FoxPro file so that it has no extension. For example, rename CUSTOMER.DBF to CUSTOMER.
2. Open a Microsoft Access [database](#).
3. From the File menu, choose Import...
4. In the Import [dialog box](#), select the appropriate data source. For example, select dBASE III or IV to import a dBASE file.
5. Under List Files of Type, select All Files (*.*) .
6. Locate and select the import file created in step 1 without an extension.
7. Choose the Import button.

You will see the following error message:

```
Couldn't find object '<filename>'.
```

[References](#)

INF: Using a Subtract Query to Find Unmatched Records

Article Number: Q95326
CREATED: 07-FEB-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to create a Subtract [query](#) to compare two tables and return a [dynaset](#) that includes only those records from the first [table](#) that do not have matching records in the second table.

For example, say you have two tables (Table1 and Table2) joined on a common [field](#) (Customer ID). Your goal is to find the records in Table1 that do not have a matching Customer ID in Table2.

MORE INFORMATION

To return a dynaset of records that do not match on the joined field, create a query to [join](#) the two tables and subtract the matching records. This article assumes the two tables are named Table1 and Table2. The steps listed below outline the process.

NOTE: Selecting Table Names from the View menu (when in Query Design view) prevents confusion when referring to the same field from two or more tables.

1. In [Design view](#), create a new query and add Table1 and Table2.
2. Join the two tables on the appropriate field. Microsoft Access joins the tables automatically if there is an underlying [relationship](#) between them.
3. Double-click the join line. Convert the join from its default of type 1 to either type 2 or type 3. Use the join information to decide which type of join you want. For this example, you want all the records from Table1 and only those that match from Table2 -- a Type 2 join.
4. Drag the joined field from Table2 down to the query grid and set the [criteria](#) for this field to "Is [Null](#)." Uncheck the Show box so that Microsoft Access will not display this field.
5. Drag any other needed Table1 fields, such as the [primary key](#) field, to the query grid. These are the columns that identify unmatched records.
6. Run the query. The Is Null condition ensures that the dynaset contains only those records from Table1 that do not have a match on joined field in Table2.

The following is a sample Select statement that uses the Northwind Traders (NWIND.MDB) sample [database](#). It returns all the customers who have not placed an order.

```
SELECT DISTINCTROW Customers.[Company Name]
FROM Orders, Customers,
Customers LEFT JOIN Orders
ON Customers.[Customer ID] = Orders.[Customer ID]
WHERE ((Orders.[Customer ID] Is Null));
```

[References](#)

PRB: Error in Column Property Help Menu Topic

Article Number: Q95325
CREATED: 07-FEB-1993
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you search for "Column," then "Column Property" using the Microsoft Access Help menu, some of the remarks are not correct.

RESOLUTION

When you use the Column property to reference a specific column in a combo box or list box, use Column(0) to refer to the first column, Column(1) to refer to the second column, and so on. In other words, the numbering begins with 0, not 1.

Incorrect Statement in the Column Property Help Topic

You can use the Column property to refer to a particular column in a multiple-column combo box or list box. Use 1 to refer to the first column, 2 to refer to the second column, and so on.

Corrected Version

You can use the Column property to refer to a particular column in a multiple-column combo box or list box. Use 0 to refer to the first column, 1 to refer to the second column, and so on.

MORE INFORMATION

To create a multiple-column combo box, set the RowSource property to include more than one column, and set the ColumnCount property to a number greater than one. To refer to each column in the combo box, use the Column property and include a reference number. Here is the syntax:

```
Forms![Form Name]![Control Name].Column(#)
```

Replace the pound sign (#) with the reference number. The reference number is 0 for first column in the control, 1 for the second column, 2 for the third, and so on.

Here is a step-by-step example that demonstrates how to use the Column property reference number correctly:

1. Open the NWIND.MDB sample database, and create a new, blank (unbound) form.
2. Add a combo box and a text box to the new form. Use the following as a guide to set the properties for each control:

Object: Combo Box

ControlName: My Combo Box
RowSource: Employees (a [table](#))
ColumnCount: 3
ColumnWidth: 0,1,1
BoundColumn: 1

Object: Text Box

ControlName: My Text Box
ControlSource: =[Forms]![Column Example]![My Combo Box].Column(2)

Because you set ColumnCount to 3 and RowSource to the Employees table, Microsoft Access loads the combo box with values from the first three fields in the Employees table: Employee ID, Last Name, and First Name. Because you set the width of the first column to 0 in the ColumnWidth property, the first column (Employee ID) does not show up in the combo box. The 2 in the Column(2) in the ControlSource setting for the text box, refers to the third column (First Name), not the second (Last Name).

3. Save the form, and name it Column Example.
4. Switch to [Form view](#), and select a name from the combo box list.

The combo box list displays the Last Name and First Name fields for each employee in the Employees table. When you select a name from the list, Microsoft Access displays the last name in the combo box and the first name in the text box.

[References](#)

PRB: Current Record Not Immediately Updated with Replace All

Article Number: Q95324
CREATED: 07-FEB-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you choose the Replace All button and the focus is on a particular field in a record, that record is not changed until all other records are replaced.

STATUS

This behavior is by design.

MORE INFORMATION

The record that has the focus is refreshed only after all other records in the dynaset have been updated.

Steps to Reproduce Behavior

1. Open the sample database NWIND.MDB.
2. Create a query based on the Orders table, as follows:

 Query: QueryTest

 Field Name: Ship City
 Criteria: ="seattle"
3. Place the indicator in the first field, between the "e" and "a" in "Seattle".
4. Execute the query.
5. From the Edit menu, choose Replace. Type "LE" in the Find What box. Type "LEFOO" in the Replace With box. Select the Current Field button and clear the Match Case and Match Whole Field check boxes.
6. From the Edit menu, choose Replace All. Note that the first field does not immediately display the change, but does as soon as all other records are updated.

[References](#)

PRB: Paste Unavailable After Record Is Copied to Clipboard

Article Number: Q95323
CREATED: 07-FEB-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If two users are simultaneously modifying the same [record](#) in the same [table](#), the user who attempts to save the record last receives the following error message:

Data has changed; Operation Stopped

When you choose the OK button to close this message, you receive the following error message:

This record has been changed by another user since you started editing it. If you save the record you will overwrite the changes the other user made. Copying the changes to the [Clipboard](#) will let you look at the values the other user entered, and then paste your changes back in if you decide to make changes.

You are given the option to copy the changes the other user made to the Clipboard and then view the contents in the Clipboard; however, the Paste command is unavailable in Microsoft Access.

CAUSE

Microsoft Access does not support pasting records when the focus is in a [control](#).

RESOLUTION

You can choose Paste Append to append the record to the table; or, if you select the record you want to paste to, you can then use the Paste command (or CTRL+V).

STATUS

This behavior is by design.

[References](#)

PRB: Stored procedure 'PROCEDURE' not found -- Error Message

Article Number: Q95061
CREATED: 31-JAN-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

You receive this error:

Stored procedure 'PROCEDURE' not found

The 'PROCEDURE' noted in the error message is the name of a stored procedure that cannot be found but is necessary to gain access to the [SQL](#) server.

CAUSE

The SQL server you are attempting to access is not set up correctly. It is missing the required stored procedures.

RESOLUTION

To correct this situation, install the necessary stored procedures on the SQL Server by using the INSTCAT.SQL SQL script file supplied with the Microsoft Access disk set.

MORE INFORMATION

Run INSTCAT.SQL to configure Microsoft SQL Server for use with [ODBC](#), the open [database](#) connectivity protocol used by Microsoft Access to attach SQL Server tables. INSTCAT.SQL is the SQL Script file that ships with with ODBC. You need to run it to set up the stored procedures that provide catalog information used by ODBC.

To install the catalog storedprocedures using INSTCAT.SQL, run INSTCAT.SQL using the SQL Server facility ISQL (Interactive SQL) from the MS-DOS [command line](#).

Following is the [syntax](#) for this procedure. Note that you need to enter the two lines as one continuous line, and do not include the angle braces <> in the command:

```
ISQL /U <sa login name> /n /P <password> /S <SQL server name>  
/i <drive:\path\INSTCAT.SQL> /o <drive:\path\output file name>
```

Following is a description for each switch:

```
/U Gives the system administrator's login name  
/n Eliminates line numbering and prompting for user input.  
/P Gives the system administrator's Password -- case sensitive
```

/S Gives the name of the server to set up.
/i Gives the drive and fully qualified path for INSTCAT.SQL
/o Provides the output file destination for results of the process including errors.

Here is a sample ISQL command line:

```
ISQL /U sa /n /P SA_Password /S SQL_SERVER /i d:\SQL\INSTCAT.SQL  
/o d:\SQL\output.txt
```

After running INSTCAT.SQL, run the RECONFIGURE command against the MASTER database by using the SQL Administration Facility (SAF) from an MS-DOS client or the server, or run the SQL Administrator program from a Microsoft Windows client.

Other Possible Causes for this Error Message

It is possible that other situations can indirectly cause this error message. Use the following steps to troubleshoot the problem:

1. Use File Manager to search for multiple, corrupted, or outdated copies of these files:

ODBC.DLL - dated on or after 10-16-92 containing 44736 bytes
DBNMP3.DLL - dated on or after 9-15-92 containing 8241 bytes
SQLSRVR.DLL - dated on or after 10-16-92 containing 135792 bytes
NETAPI.DLL - depends on the [network](#) being used
COMMDLG.DLL - dated on or after 10-25-92 containing 89248 bytes

NOTE: The sizes and dates referenced are correct if the files are installed by Microsoft Access version 1.0.

2. Make sure the files listed above are in appropriate locations. All the files except NETAPI.DLL are usually installed in the Windows System directory. The NETAPI.DLL file is a network specific driver that is usually located in the network's program directory, which should be in the path.
3. Use another application, such as Q+E, SAF, or PowerBuilder, on the same machine to attempt to attach the same data source. If you cannot access the data source from another application, there is probably a problem with the network. That is there may be a problem with the NETAPI.DLL file or a problem with the SQL database server, not with Microsoft Access or ODBC.
4. Attempt to attach another data source such as another SQL server. If you are successful, the original server may be down or you may need to reconfigure the entry for that server by using the ODBC Administration Utility.
5. Attempt to attach to the same data source from another machine. If you cannot gain access to the SQL database server from another machine, there could be a network wide problem or the server may be down or not communicating.

6. Try increasing your ODBC timeout settings in the ODBC section of the MSACCESS.INI file.
7. Check with the system administrator to see if the SQL database server is up, available for transactions, and operating correctly.
8. Reinstall ODBC after restarting the server and renaming the ODBC.DLL, DBNMP3.DLL, and SQLSRVR.DLL files.

References:

"Microsoft Access User's Guide," version 1.0, Appendix D, "Setting up Microsoft Access on a Network," pages 660-662.

"Microsoft SQL Server Administrator's Guide," version 4.2, pages 205-212.

[References](#)

PRB: Connection Not Open Error Message

Article Number: Q95060
CREATED: 31-JAN-1993
MODIFIED: 18-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

You receive the following error message:

Connection not open.

CAUSE

The most likely cause of this error message are missing, damaged, or incorrect versions of the SQLSRVR.DLL or [ODBC.DLL](#) files.

Other situations may indirectly cause this error message. For more information, [query](#) on the following words here in the Microsoft Knowledge Base:

[debugging](#) and troubleshooting and tips and Q102437

RESOLUTION

To solve this problem, verify the following:

- Single copies of SQLSRVR.DLL and ODBC.DLL both exist in your Windows SYSTEM directory.
- You have the correct versions of these two files. The SQLSRVR.DLL file installed by Microsoft Access version 1.0 is dated 10-16-92 and has a size of 135792 bytes. The ODBC.DLL file installed by Microsoft Access version 1.0 is dated 10-16-92 and has a size of 44736 bytes.

[References](#)

INF: Changing the Relationship Between Two Existing Tables

Article Number: Q95058
CREATED: 31-JAN-1993
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can change an ID field from a Long Integer or other data type to a Counter data type in the table on the one-side of a one-to-many relationship and change the ID values in the corresponding records in the many-side table while still maintaining the links between the two tables.

MORE INFORMATION

The following example uses the Northwind Traders (NWIND.MDB) sample database to show you how to change the ID values in the Customers table (the one-side table) and update the values in the Orders table (the many-side table) so that the records in the two tables are still linked.

1. Add the following fields to the Customers and Orders tables:

Table: Customers

Field name: New ID
Data Type: Counter

Table: Orders

Field name: New ID
Data Type: Number
Field Size: Long Integer

2. Create a new query. Add the Customers and Orders tables to the query. Make sure the tables are joined on Customer ID.
3. From the Query menu, choose Update.
4. Add the following field and properties to the query grid:

Update Query: Query1

Field name: Orders.New ID
Update To: [Customers].[New ID]

When you type Orders.New ID into the Field name cell, the Orders table name disappears. From the View menu, choose Table Names to see all the table names displayed on the grid.

6. To run the query, click the Run button on the tool bar.

7. Close the query. You do not need to save it.

If there is a relationship already defined for the two tables, you need to remove it. For example, the NWIND database has a defined relationship, so you need to remove it.

To remove the original relationship between tables:

1. Press F11 to give the [Database window](#) the focus. From the Edit menu, choose Relationships.
2. Select Customers for the Primary Table and Orders for the Related Table. Customer ID automatically appears in Select Matching Fields.
3. Choose the Delete button to remove the relationship. Then close the [dialog box](#).

Next, remove the original ID field and rename the new one in the Customers table:

1. Open the Customers table in [Design view](#).
2. Delete the Customer ID field.
3. Change the name of New ID to Customer ID and make it the [primary key](#).
4. Close the table, and save the changes.

Then remove the original ID field and rename the new one in the Orders table:

1. Open the Orders table in Design view.
2. Delete the Customer ID field.
3. Change the name of New ID to Customer ID.
4. Close the table, and save the changes.

Now you can redefine the relationship between the tables. From the Edit menu, choose Relationships.

[References](#)

PRB: Driver Cannot Be Found When Attaching to SQL Server

Article Number: Q95054
CREATED: 31-JAN-1993
MODIFIED: 18-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When trying to attach a [SQL](#) Server [table](#), or open a previously [attached table](#), you receive the following error message:

Driver specified by data source could not be loaded.

CAUSE

The most likely cause of this error message is a missing, damaged, or incorrect version of the SQLSRVR.DLL file.

Other situations may indirectly cause this error message. For more information, [query](#) on the following words here in the Microsoft Knowledge Base:

[debugging](#) and troubleshooting and tips and Q102437

RESOLUTION

To correct this problem, verify the following:

- There is a single copy of the SQLSRVR.DLL file in your Windows SYSTEM subdirectory.
- You have the correct version of the SQLSRVR.DLL file. The version installed by Microsoft Access version 1.0 is dated 10-16-92 and has a size of 135792 bytes. You need either this file or a more recent version.

[References](#)

PRB: Query Fails When Linked SQL Server Table Cannot Be Found

Article Number: Q95053
CREATED: 31-JAN-1993
MODIFIED: 18-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When attempting to use an attached [SQL](#) Server [table](#), you receive the following error message

```
    Couldn't execute query; couldn't find linked table.  
    Invalid object name <TABLE_NAME>.
```

where <TABLE_NAME> is the name of the table you are attempting to use.

CAUSE

The most likely cause of this error is an incorrect entry in the [SQLSERVER] section of your WIN.INI file.

The [SQLSERVER] section of your WIN.INI file should look like the following:

```
[SQLSERVER]  
MYSERVER=dbnmp3,\\MYSERVER\PIPE\SQL\QUERY  
YOURSRVR=dbnmp3,\\YOURSRVR\PIPE\SQL\QUERY
```

The error is caused by an incorrect server name (MYSERVER or YOURSRVR) or additional spaces in the entry.

Other situations may indirectly cause this error message. For more information, query on the following words here in the Microsoft Knowledge Base:

[debugging](#) and troubleshooting and tips and Q102437

RESOLUTION

Verify that the [SQLSERVER] section of your WIN.INI file contains the correct entry for the server you are attempting to attach.

[References](#)

PRACC9302: Running Delete Query Gives Misleading Error Msg

Article Number: Q95052
CREATED: 31-JAN-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you attempt to run a Delete [query](#) that would violate referential integrity, Microsoft Access displays the following misleading error message:

Errors were encountered: The contents of fields in 0 [record](#)(s) were deleted, XXX record(s) were lost due to key violations, and 0 record(s) were locked and couldn't be modified. Proceed anyway?

CAUSE

Microsoft Access uses the same error message for all action queries.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Microsoft Access enforces [referential integrity](#) if relationships were defined between tables with this option. You will receive an error message if you attempt to breach the rules of referential integrity when running a Delete Query or when you delete fields that would effect the one-side of a [one-to-many relationship](#).

However, as you can see in the SYMPTOMS section above, when you run the Delete query, the error message displayed is misleading.

When you try to run a Delete query on the one-side of a one-to-many [relationship](#) that has Referential Integrity enforced, after you choose the OK button in response to this message: "XXX [row](#)(s) will be deleted," Microsoft Access gives you the following error message:

Errors were encountered: The contents of fields in 0 record(s) were deleted, XXX record(s) were lost due to key violations, and 0 record(s) were locked and couldn't be modified. Proceed anyway?

If you proceed by choosing OK, nothing happens.

If you attempt to delete a [field](#) from a [form](#) or [table](#) that is on the one-side of a one-to-many relationship, Microsoft Access gives you

the following error message:

Can't delete or change record. Since related records exist in table 'The table that is on the many side', referential integrity rules would be violated.

In this case, the two preceding error messages mean the same thing.

[References](#)

PRB: Unable to Connect to Attached SQL Server Table, Error Msg

Article Number: Q95051
CREATED: 31-JAN-1993
MODIFIED: 18-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you try to use an attached [SQL](#) Server [table](#), you receive the error message:

Unable to connect to data source.

CAUSE

The following two conditions are most likely to cause this error message:

- There is an incorrect entry in the [SQLSERVER] section of your WIN.INI file.

The [SQLSERVER] section of your WIN.INI file should look like the following example:

```
[SQLSERVER]
MYSERVER=dbnmp3,\\MYSERVER\PIPE\SQL\QUERY
YOURSRVR=dbnmp3,\\YOURSRVR\PIPE\SQL\QUERY
```

This error is caused by an incorrect server name (MYSERVER or YOURSRVR) or extra spaces in the path.

- The SQL [database](#) server is overloaded or unavailable.

Other situations may indirectly cause this error message. For more information, [query](#) on the following words here in the Microsoft Knowledge Base:

[debugging](#) and troubleshooting and tips and Q102437

RESOLUTION

Verify that the [SQLSERVER] section of your WIN.INI file has the correct entry for the server you are attempting to attach. Then check with the [system administrator](#) to see if the SQL database server is available for transactions, operating properly, and not overloaded.

References

PRB: Data Source Not Found, No Default Driver Specified Error

Article Number: Q95050
CREATED: 31-JAN-1993
MODIFIED: 18-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When trying to attach a [SQL](#) Server [table](#), or use a table that has already been attached, you may receive the following error message:

Data source not found and no default driver specified.

CAUSE

You may have a missing, damaged, or incorrect [ODBC](#).INI file.

Other situations may indirectly cause this error message. For more information, [query](#) on the following words here in the Microsoft Knowledge Base:

[debugging](#) and troubleshooting and tips and Q102437

RESOLUTION

Verify the following:

- There is an ODBC.INI file in your WINDOWS subdirectory.
- There is only one copy of the ODBC.INI file available on the machine you are using.
- The entries in the ODBC.INI file are correct for the SQL [database](#) server to which you want to connect.

MORE INFORMATION

The contents of your ODBC.INI file should resemble the following:

```
[ODBC Data Sources]
MYSERVER=SQL Server
YOURSRVR=SQL Server
```

```
[MYSERVER]
Driver=C:\WINDOWS\SYSTEM\SQLSRVR.DLL
Description=SQL Server on server MYSERVER
OemToAnsi=No
Network=dbnmp3
Address=\\MYSERVER\PIPE\SQL\QUERY
```

```
[YOURSRVR]
```

```
Driver=C:\WINDOWS\SYSTEM\SQLSRVR.DLL
Description=SQL Server on server YOURSRVR
OemToAnsi=No
Network=dbnmp3
Address=\\YOURSRVR\PIPE\SQL\QUERY
```

You will receive the error message if there is no section that refers to the SQL database server to which you are trying to attach or if any of the following are true:

- The [MYSERVER] section is missing.
- The Driver= line is missing.
- The .DLL file on the Driver= line is missing its .DLL extension.
- The .DLL file is missing, damaged, or outdated.

[References](#)

INF: Use Row Fix-up Technique to Look Up Info Automatically

Article Number: Q95048
CREATED: 31-JAN-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can create a [query](#) or [form](#) that automatically looks up information in the one-side [table](#) of a [one-to-many relationship](#) based on an entry in a many-side [field](#). This process is sometimes called [row](#) fix-up.

You can use row fix-up to have a query automatically look up information in other tables and display it on a form or [report](#).

You can look up information by using any one of the following three techniques:

1. Use row fix-up. For more information on row fix-up, see the "More Information" section below.
2. Use multiple DLookup() functions in forms and reports. For more information on the DLookup() [function](#), query on the following words here in the Microsoft Knowledge Base:

DLookup and forms and reports

3. Use the Column property of a multiple-[column combo box](#) to [update](#) a [text box control](#) with new information as focus moves from row to row in the combo box. For more information on this, query on the following words here in the Microsoft Knowledge Base:

Column and property and combo and Update

MORE INFORMATION

Use row fix-up when you want Microsoft Access to automatically look up information in the one-side table based on an entry in a many-side field. The row fix-up technique works in a query or in a form.

The most typical table [relationship](#) that can take advantage of row fix-up is a one-to-many relationship. However, row fix-up also works with tables that have a [one-to-one relationship](#) if you use a left outer [join](#). For more information on what data is and is not updatable, search for "Underlying table or query," then for "When is a query updatable?" using the Microsoft Access Help menu.

For example, the Categories and Products tables in the sample [database](#) NWIND.MDB have a one-to-many relationship. Each category may appear several times in the Products table. The Products table contains the Category ID field, which is the [foreign key](#) that identifies the

category for a product.

When row fix-up updates records, Microsoft Access automatically recalculates any totals or expressions that are dependent on the updated information.

Microsoft Access version 1.1 provides additional online Help topics for row fix-up issues. In this context, the term "row fix-up" is synonymous with the term "dynamic lookup." For more information, search for "dynamic lookup", then for "Displaying Fields from Another Table or Query (Common Question)" using the Help menu.

The following two examples demonstrate how row fix-up works:

Example 1: Create a Query That Uses Row Fix-Up

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new query. Add the Products and Categories tables. Products is the many-side table and Categories is the one-side table. The two tables have a many-to-one relationship based on the Category ID field (meaning, there are many identical Category ID values in Products for each unique Category ID value in Categories).
3. Add the following fields from the Products table:

Product ID
Product Name
Category ID
4. Add the following fields from the Categories table:

Category Name
Description
5. Save the query as Row Fix-up Example.
6. View the results of the query. To verify that the query performs row fix-up, move to the end of the query and create a new record. Enter a valid Category ID value in the Category ID field. Microsoft Access will automatically look up the values associated with that Category ID value in the Categories table.

This process works because the Category ID field located in the query comes from the many-side table (Products). Microsoft Access performs row fix-up on one-side tables when you enter identifying data in the many-side table. When you enter a valid value in the Products table's Category ID field, Microsoft Access knows that Category ID is the key field for the one-side table (Categories), so it looks up the information from Categories based on the Category ID value and automatically displays it in the query.

Example 2: Create a Form That Uses Row Fix-Up

1. Create a new form based on the query that you created in Example 1.
2. Locate the fields from the Products table on "one" side of the form and those from the Categories table on the "many" side of the form.
3. Create a combo box bound to the Category ID field. Set the RowSource property to the Categories table. Display a single column, [Category ID], in the combo box.
4. Switch to Form view. When you enter new records and select a category from the Category ID combo box, all related information about that Category is displayed.

Note that if you had included the primary key value from the one-side table (Categories), you would not have been allowed to add new records to the query or to change the Category ID value. For this reason, you must drag the Category ID field from the Products table, rather than from the Categories table.

REFERENCES

=====

"Microsoft Access User's Guide," version 1.0, Chapter 10, "Creating Forms Based on More than One Table," page 282

[References](#)

INF: Using SHARE.EXE and VSHARE.EXE with Microsoft Access

Article Number: Q95047
CREATED: 31-JAN-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

MS-DOS, Microsoft Windows, and Windows-based applications use SHARE.EXE to provide a file locking mechanism so that Windows-based tasks can communicate with each other. The tasks use SHARE.EXE to sort out file read/write conflicts. Therefore, it is important that you load SHARE.EXE before starting Windows. Without SHARE.EXE loaded, you may experience data corruption in applications.

Using SHARE.EXE with Microsoft Access allows multiple users to access a database at the same time while protecting them from updating the same record at the same time. If a database is opened exclusively by one user, no other users will be able to use the database until the first user releases it. However, if all users open the database in a shared mode, SHARE.EXE allows all users to read and write to the database simultaneously, with the exception of locked records. For example, if a user has record 100 open on a particular database, that record is considered locked, and no other users will be able to change any data in record 100 until the first user finishes changing it.

For this reason, while it is possible to run Microsoft Access without first loading SHARE.EXE, Microsoft does not recommend it. If you choose not to use SHARE.EXE, you must ensure that only one Windows-based task writes to a database record at any given moment.

MORE INFORMATION

MS-DOS uses SHARE.EXE to lock files as requested by applications. SHARE.EXE controls updates and prevents data corruption in files.

An application can request a deny-read lock (exclusive), a deny-write lock, or both. Deny-write locks allow other applications to read data, but prevent two different applications from updating the same file at the same time.

SHARE.EXE also provides byte-range locks. When a range of bytes is locked, other applications cannot read, write, or lock those bytes. Other, unlocked bytes, in the file are not affected. An application can use byte-range locks to place exclusive locks on records for record-level locking.

In single-user environments, SHARE.EXE prevents file contention accidents when multiple instances of a program attempt to write to the same database or database record at the same time. In multiple-user environments such as networks, SHARE.EXE prevents possible database

corruption when several different network users attempt to write to the same database or database record at the same time.

A Single Exception

The only time you do not have to use SHARE.EXE is when you run Microsoft Windows for Workgroups in Enhanced mode. Microsoft Windows for Workgroups in Enhanced mode loads and uses a file-sharing program called VSHARE.386, which is a replacement for SHARE.EXE. There is an entry for VSHARE.386 in the SYSTEM.INI file.

In fact, if you plan on running Windows for Workgroups exclusively in Enhanced mode and do not run other applications that require SHARE.EXE, you can save approximately 5K of conventional memory by not loading SHARE.EXE. To do this, remove SHARE.EXE from your AUTOEXEC.BAT file and restart your computer.

SHARE.EXE is likely to be in your AUTOEXEC.BAT file because during the Microsoft Access installation process, the SETUP.EXE program automatically inserts the following MS-DOS command in your AUTOEXEC.BAT file:

```
<MS-DOS directory>\SHARE.EXE /L:500
```

If SHARE.EXE is already present before running Microsoft Windows for Workgroups in Enhanced mode, VSHARE.386 loads in addition to SHARE.EXE, temporarily disabling SHARE.EXE. Then VSHARE.386 takes over file sharing tasks until you exit Microsoft Windows for Workgroups. After you exit Windows for Workgroups, VSHARE.EXE passes file-sharing control back to SHARE.EXE for use by MS-DOS applications that require it.

Microsoft Access runs correctly with either SHARE.EXE or VSHARE.386, but SHARE.EXE limits the number of available locks to the number specified when SHARE.EXE was loaded. The /L parameter specifies the number of share locks; the default is 20. VSHARE.386, on the other hand, dynamically allocates the number of locks available based on the demand for locks. The number of locks available is especially important if the computer running Microsoft Windows for Workgroups acts as a server in a client-server environment.

[References](#)

INF: Attaching Clipper Tables to Microsoft Access Databases

Article Number: Q95044
CREATED: 31-JAN-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

It is possible to attach Clipper .DBF files to a Microsoft Access [database](#). To do so, attach it as a dBASE IV file. Note, however, the [index](#) may or may not be updateable.

Clipper provides an alternative indexing [method](#) (.NTX) to the dBASE standard (.NDX). The Clipper index format isn't recognized by Microsoft Access, so Microsoft Access cannot [update](#) it correctly.

However, if the Clipper application was developed using the dBASE .NDX format, Microsoft Access updates the index correctly.

[References](#)

INF: Set Focus to a Subform Control Using GoToControl Action

Article Number: Q95014
CREATED: 28-JAN-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can create a [macro](#) to move the focus to a [control](#) on a [subform](#) by using the GoToControl macro action to move first to the subform, which is a type of a control. Use the GoToControl action again to move to a particular control on the subform.

MORE INFORMATION

The macro action GoToControl does not allow you to enter the full [syntax](#) for the ControlName argument as:

```
Forms![orders]![orders subform].form![Product ID]
```

If you enter this syntax, you will receive the following error:

```
There is no control named 'Forms![orders]![orders  
subform].Form!Product id'.
```

To get around this, you need to set up a macro that first sets the focus to the Subform control, followed by a second action that sets the focus to a specific control.

In the sample [database](#) Northwind Traders (NWIND.MDB), the Orders and Orders Subform forms can be used as an example where you can have a macro set the focus to the Product Id on the Orders Subform.

Create a new macro with two separate GoToControl actions, then attach the macro to a button on the Orders form.

```
Actions  
-----  
GoToControl  
    Control Name: Orders Subform  
  
GoToControl  
    Control Name: Product Id
```

When the button is clicked, the focus will move to the Product Id control.

[References](#)

PRB: SQL Data Sources Dialog Box Lists No Servers

Article Number: Q95013
CREATED: 28-JAN-1993
MODIFIED: 12-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you are trying to attach, import from, or export to a Microsoft [SQL Server table](#), the SQL Data Sources [dialog box](#) is displayed, but no data sources (SQL Server machines) are listed.

CAUSE

The following causes are most likely:

- You did not use the Microsoft Open Database Connectivity ([ODBC](#)) Administration application to set up connections to SQL Server machines.
- The [ODBC Data Sources] section of your ODBC.INI file is missing or blank, or you have multiple copies of the ODBC.INI file and Microsoft Access is using a file in which the [ODBC Data Sources] section is missing.

RESOLUTION

Verify that you have a single copy of the ODBC.INI file on your computer, located in your Windows directory. Also, make sure there is an [ODBC Data Sources] section in your ODBC.INI file containing correct entries for the SQL Server machines you want to use.

MORE INFORMATION

To correct the problem, rename your ODBC.INI file and use the ODBC Administration application to complete the following tasks:

- Connect to the SQL Server machines you want to use.
- Create a new ODBC.INI file.

The contents of your new ODBC.INI file should resemble the following example:

```
[ODBC Data Sources]
MYSERVER=SQL Server
YOURSRV=SQL Server

[MYSERVER]
Driver=C:\WINDOWS\SYSTEM\SQLSRVR.DLL
Description=SQL Server on server MYSERVER
```

OemToAnsi=No
Network=dbnmp3
Address=\\MYSERVER\PIPE\SQL\QUERY

[YOURSRVR]
Driver=C:\WINDOWS\SYSTEM\SQLSRVR.DLL
Description=SQL Server on server YOURSRVR
OemToAnsi=No
Network=dbnmp3
Address=\\YOURSRVR\PIPE\SQL\QUERY

References:

"Microsoft Access User's Guide" version 1.0, pages 660-662

[References](#)

PRB: Error: Unable to Connect to Data Source

Article Number: Q95012
CREATED: 28-JAN-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.00

SYMPTOMS

You receive this error message:

Unable to connect to data source

CAUSE

The [SQL](#) server you are trying to access does not have a correct entry in the [SQLSERVER] section of the WIN.INI or the SQL server is down, overloaded, or otherwise unavailable.

RESOLUTION

To correct this situation, verify that the SQL server is up and available and that the entry in the [SQLSERVER] section of the WIN.INI file looks like this:

```
[SQLSERVER]
MY_SERVER=DBNMP3,\\MY_SERVER\PIPE\SQL\QUERY
```

If there are any spaces in this entry or if the server name in the \\MY_SERVER\PIPE\SQL\QUERY part of the statement is misspelled, you receive this error.

MORE INFORMATION

It is possible that there are other situations which may indirectly cause this error message. Here are several other things you can check to troubleshoot this problem:

1. Use File Manager to search for multiple, corrupted, or outdated copies of these files:

[ODBC](#).DLL - 44736 bytes, dated 10-16-92 or later
DBNMP3.DLL - 8241 bytes, dated 9-15-92 or later
SQLSRVR.DLL - 135792 bytes, dated 10-16-92 or later
NETAPI.DLL - depends on the [network](#) being used
COMMDLG.DLL - 89248 bytes, dated 10-25-92 or later

NOTE: The sizes and dates referenced are correct if the files were installed by Microsoft Access version 1.0.

2. Make sure the files listed above are in appropriate locations. All files

except NETAPI.DLL are usually installed in the Windows System directory. The NETAPI.DLL file is a network specific driver, so it's usually located in the network's program directory, which should be in the path.

3. Use another application, such as Q+E, SAF, or PowerBuilder, on the same computer to attempt to attach to the same data source. If you can't access the data source from another application, there is probably a problem with the network (NETAPI.DLL) or with the server, not with Microsoft Access or ODBC.
4. Attempt to attach to another data source, if available, such as another SQL server. If you can attach to another server, then the original server may be down or you may need to reconfigure the entry for that server using the ODBC Administration Utility.
5. Attempt to attach to the same data source from another computer. If you can't access the SQL server from another computer, there may be a network wide problem or the server may be down or not communicating.
6. Try increasing the ODBC timeout settings in the ODBC section of the MSACCESS.INI file.
7. Check with the [system administrator](#) to see if the SQL server is up, available for transactions, and operating correctly.
8. Reinstall ODBC after performing a clean boot and renaming the ODBC.DLL, DBNMP3.DLL, and SQLSRVR.DLL files.

References:

"Microsoft Access User's Guide," version 1.0, Appendix D, "Setting Up Microsoft Access on a Network," pages 660-662.

[References](#)

PRB: Troubleshooting 'ODBC - couldn't find ODBC.DLL' Error

Article Number: Q95011
CREATED: 28-JAN-1993
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

You receive the following error message:

ODBC - couldn't find ODBC.DLL

CAUSES

Here are three possible causes:

- The ODBC.DLL file you are using may be old.
- You may have duplicate ODBC.DLL files on your system, and you are using the wrong one.
- The ODBC.DLL file you are using may be damaged or corrupted.

RESOLUTION

The ODBC.DLL file installed by Microsoft Access version 1.0 is dated 10-16-92, has a size of 44736 bytes, and should be installed in the \WINDOWS\SYSTEM directory.

MORE INFORMATION

There are other situations that may indirectly cause this error message. To troubleshoot for these possibilities:

1. Use File Manager to search for multiple, corrupted, or outdated copies of these files:

File	Correct Dates and Sizes
DBNMP3.DLL	dated 9-15-92(8241 bytes) or later
SQLSRVR.DLL	dated 10-16-92(135792 bytes) or later
NETAPI.DLL	depends on the <u>network</u> being used
COMMDLG.DLL	dated 10-25-92(89248 bytes) or later

NOTE: The sizes and dates referenced are correct if the files are installed by Microsoft Access version 1.0.

2. Make sure the files listed in step 1 are in their appropriate locations. All files except NETAPI.DLL are usually installed in the \WINDOWS\SYSTEM directory. The NETAPI.DLL file is a network specific driver; it is usually located in the network's program directory, which should be in

the path.

3. Use another application, such as Q+E, SAF, or PowerBuilder, on the same machine to attempt to attach to the same data source. If you cannot access the data source from another application, you probably have a network problem with the NETAPI.DLL file or with the server, not with Microsoft Access or ODBC.
4. Attempt to attach another data source, if available, such as another [SQL](#) server. If you can attach to another server, the original server may be down, or you may need to reconfigure the entry for that server using the ODBC Administration Utility.
5. Attempt to attach the same data source from a different machine. If you cannot gain access to the SQL server from a different machine, you may have a network-wide problem, or the SQL server may be down or not communicating.
6. Try increasing your ODBC timeout settings in the ODBC section of the MSACCESS.INI file.
7. Check with the [system administrator](#) to see if the SQL server is up, available for transactions, and operating properly.
8. Reinstall ODBC after performing a clean boot and renaming the ODBC.DLL, DBNMP3.DLL, and SQLSRVR.DLL files.

References:

"Microsoft Access User's Guide," version 1.0, Appendix D, "Setting up Microsoft Access on a Network," pages 660-662.

[References](#)

PRB: Can't Scroll Query Grid with ARROW Key or Mouse

Article Number: Q95010
CREATED: 28-JAN-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you design a [query](#) and the [Query window](#) is sized so that the right side of the query grid is cut off (that is, only part of the right-most [field](#) is displayed), you cannot use the RIGHT ARROW key or drag with the mouse to scroll that part of the query grid.

STATUS

This behavior is by design.

RESOLUTION

Resize the Query window so that all of it is displayed.

MORE INFORMATION

Steps to Reproduce Behavior

1. In the sample Northwind Traders [database](#) (NWIND.MDB), open the Category List query in [Design view](#).
2. Resize the Query window so that the right-most [column](#) in the [QBE](#) grid with a field name in it is truncated on the screen.
3. Click in the start of the field [cell](#) and press the RIGHT ARROW key. The cursor leaves the screen but the right-most portion of the grid does not appear. The same thing happen if you try to drag with the mouse.

[References](#)

PRB: Trailing Spaces Automatically Truncated During Data Entry

Article Number: Q95009
CREATED: 28-JAN-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Form controls receiving input with a trailing space will ignore the trailing space(s), so the data entered may not be valid as a means of comparison. Also, variables in [Access Basic](#) that have been assigned a value containing a space character at the end of the text [string](#) will automatically be ignored when used in a conditional statement.

CAUSE

Microsoft Access automatically truncates (or strips) trailing spaces on data you enter into [form](#) controls to prevent the creation of a [record](#) that cannot be referenced because of accidental space character(s) being appended to the end of the text.

RESOLUTION

Do not use text strings that contain trailing spaces. If trailing spaces are necessary, insert your data in [Datasheet view](#) of the form.

STATUS

This behavior is by design.

MORE INFORMATION

Although both "a" or "a " are acceptable input in the example below, the data stored in the underlying [table](#) will always contain the value "a". The space following the "a" is truncated.

Steps to Reproduce Behavior

In a form:

1. Open a [database](#).
2. Create a form with a single text [field](#).
3. Enter the following for the field's Validation Rule property:

= "a "

4. View the form in [Form view](#).
5. Enter "a" in the field (with no space at the end).

In Access Basic:

1. Open a database.
2. Open a new [module](#) and choose [Immediate window](#) from the View menu.
3. Enter this [syntax](#) in the Immediate window (press ENTER after each statement):

```
? "a" = "a "
```

-or-

```
? chr(97) = chr(97) + chr(32)
```

4. Either statement will return a non-zero number indicating the statement is true, which means the values are equal.

[References](#)

PRB: Troubleshooting Out of Memory Errors in Queries

Article Number: Q95008
CREATED: 28-JAN-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When executing a [query](#), you may receive the following error message even though there are sufficient total memory and system resources:

Out of Memory

CAUSE

Microsoft Access allocates specific segments of memory for compiling a query. If your query requires more memory than is stored in the segment, you will get an Out of Memory message.

RESOLUTION

The following items help to avoid an Out of Memory message:

- Reduce the number of columns in the [table](#) design.
- Reduce the size of the [column](#) names in the table design.
- Reduce the number of columns selected by a query.
- Do not create a query based on another query; run only a single query at a time.

[References](#)

INF: How to Reference Controls or Fields on a Subform

Article Number: Q95007
CREATED: 28-JAN-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access enables a developer to create subform controls that display embedded forms on a form or report or display embedded reports on a report. You can reference a control or field that resides on a subform by using the Form property of the parent form. This article describes several techniques you can use to reference a control or field that resides on a subform or nested subform.

MORE INFORMATION

In Microsoft Access, a parent form can have many subforms, but a top-level parent form is limited to three levels of nesting for nested subforms.

To reference a control or field on the subform, use the Form property. For more information on the Form property, search for "Form," then "Form, Report Properties" using the Microsoft Access Help menu.

The following step-by-step example shows by example how to reference a field or control on a subform or nested subform:

1. Create three tables giving them each two fields as shown below:

Table: Main Table

ID: Counter
Field1: Text

Table: Sub Form Table

ID: Counter
Field2: Text

Table: Nested SubForm Table

ID: Counter
Field3: Text

2. Create three forms based on the three tables and set the form properties as shown:

Form: Nested Sub

ControlSource: Nested Subform Table
Field: ID

Field: Field3

Form: Sub

ControlSource: Sub Form Table

Field: ID

Field: Field2

Control: Nested Sub

Form: Main

ControlSource: Main Table

Field: ID

Field: Field1

Control: Sub

3. Create a new module, and add the following code to it:

```
Function DoIt ()
    Dim MyForm As Form
    Dim MySubForm As Form
    Dim MyControl As Control

    '=====
    'SUBFORM - The next four examples show how to set the value of
    'Field2 on the Sub subform. This example shows four ways to
    'reference the control:
    '
    ' 1. Straight reference using the Forms object.
    ' 2. Setting the Forms object to a form variable.
    ' 3. Setting the subform to a form variable.
    ' 4. Setting the field on the subform to a control variable.
    '
    'NOTE: Each example below is based on the previous example.
    'Some require that the object on the right side of the
    'equation be already initialized. Although variables are
    'used, you can substitute this with the proper Forms object.
    'The example of nested forms uses the Forms object in all the
    'examples.
    '=====

    'straight reference
    '-----
    Forms![Main]![Sub].Form!Field2 = "Hello"

    'referenced using form property
    '-----
    Set MyForm = Forms![Main]
    MyForm.[Sub].Form.Field2 = "Hello"

    'setting subform to a form object
    '-----
    Set MySubForm = MyForm.[Sub].Form
    MySubForm.Field2 = "Hello"

    'reference item on subform through control object
    '-----
```

```

Set MyControl = MySubForm.Field2
MyControl = "Hello"

'=====
'NESTED SUBFORM - The next four examples show the same ideas
'as shown above but for nested subforms.
'=====

'straight reference
'-----
Forms![Main]![Sub].Form![Nested Sub].Form!Field3 = "Hello"

'referenced using form property
'-----
Set MyForm = Forms![Main]![Sub].Form
MyForm.[Nested Sub].Form.Field3 = "Hello"

'setting subform to a form object
'-----
Set MySubForm = Forms![Main]![Sub].Form.[Nested Sub].Form
MySubForm.Field3 = "Hello"

'reference item on subform through control object
'-----
Set MyControl =
    Forms![Main]![Sub].Form![Nested Sub].Form!Field3
MyControl = "Hello"

End Function

```

[References](#)

PRB: Query Toolbar and Menus Are Not Updated Concurrently

Article Number: Q95006
CREATED: 28-JAN-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you are in Design view in the Query window, the Datasheet View and Run buttons on the toolbar remain unavailable until tables or queries are added to the query. The View Datasheet and Query Run menu commands are made available as soon as any cell on the query grid is modified (even if a table or query has not been added.)

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. In the Database window, choose the Query button, and then choose the New button.
2. In the Add Table dialog box, choose the Close button without selecting a table or query.
3. Notice that the toolbar and menu options mentioned above are unavailable.
4. Select the first cell in the grid, press the SPACEBAR, and then press ENTER.
5. The menu options are now available; however, if you try to use them at this point, you receive the following error message:

Query input must contain at least one table or query.

The toolbar remains unavailable until you add a table or query.

References

PRACC9303: Can't Scroll Query Grid Using Arrow Keys or Mouse

Article Number: Q95005
CREATED: 28-JAN-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you design a [query](#) and the query window is sized so that the right side of the query grid is cut off (shows only part of the right-most [field](#)), you can't use the right arrow or drag with the mouse to scroll the query grid.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Behavior

1. While working with a query in [Design view](#), create an entry by dragging a field to the query grid.
2. Resize the [Query window](#) so that the right-most field appears truncated.
3. Click in the field and press the right arrow key. The cursor leaves the screen but the grid doesn't scroll. The same thing happens if you try to drag and scroll in the field.

[References](#)

PRB: Referential Integrity Error Message with Shared Databases

Article Number: Q95004
CREATED: 28-JAN-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you share databases in Microsoft Access and try to update a record that another user has already updated, your update fails and Microsoft Access generates the following message:

Data has changed; Operation stopped.

When you choose OK, Microsoft Access generates the following write-conflict error message:

This record has been changed by another user since you started editing it. If you save the record, you will overwrite the changes the other user made.

You can choose one of three option buttons:

- Save Record
- Copy To Clipboard
- Drop Changes

Copying your changes to the Clipboard allows you to look at the values the other user has entered and paste your own changes back into the table. However, when you choose Paste, your attempt fails and Microsoft Access generates the following referential integrity error message:

Can't delete or change record. Since related records exist in table "<Table Name>", referential integrity rules would be violated.

CAUSE

Your table is on the "one" side of a one-to-many relationship and the Enforce Referential Integrity check box is selected. These symptoms occur regardless of the field being edited.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open two instances of Microsoft Access. In each instance, open the sample database NWIND.MDB. Make sure that the database not opened with exclusive access.
2. In each instance, open the Categories table.
3. In the first instance, change the Category Name field in the first record. Do not commit the change.
4. In the second instance, change the Category Name field in the first record. Commit the change by moving the indicator to a field in another record.
5. Restore the first instance and try to commit the change. Microsoft Access generates the first of the two error messages listed above. When the second error message appears, choose Copy To Clipboard.
6. Select the first record from the first instance of Microsoft Access. From the Edit menu, choose Paste.
7. Try to close the table. Microsoft Access generates the referential integrity error message listed above.

[References](#)

PRB: Apply Default Doesn't Change Height or Width of Labels

Article Number: Q95003
CREATED: 28-JAN-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

An existing [label control](#)'s Height and Width property settings are not affected when you select the control in the Form or [Report window](#)'s [Design view](#) and choose Apply Default from the Layout menu. Instead, the size of the label adjusts to fit the text in that control.

STATUS

This behavior is by design. The default height and width only affects the size of a box created for a new label. Labels that already contain text are not affected.

MORE INFORMATION

When creating a new [report](#) or [form](#) or modifying an existing one, you can change the default properties for any type of control. Default properties are the ones that determine the general appearance and behavior of a control.

To display the default properties for a particular control type in a form or report, switch to Design view and display the [toolbox](#) and property sheet. To display the toolbox, choose Toolbox from the View menu. To display the [property sheet](#), click the Properties button on the [toolbar](#), or choose Properties from the View menu. Select any tool in the toolbox to see the default properties for that type of control.

Once you change the default properties for a particular control type, you can choose Apply Default from the Layout menu to apply the new defaults to existing controls of that type on a form or report. For most control types, all properties listed in the default property box will be applied to the selected control. However, for Label controls, the Width and Height settings are not reset to match the defaults. Instead, the size of the label adjusts to fit the existing text in that control.

Steps to Reproduce Behavior:

1. Create a new form.
2. Add a label to the form that says "Label 1."
3. Display the property sheet and the toolbox.
4. Select the label control on the toolbox.

5. Enter the following properties for the default label:

Width: 1 in
Height: 1 in
BackColor: 255

6. Add another label to the form that says "Label 2".

7. Select the label that you created in step 2 and choose Apply Default from the Layout menu.

The BackColor of Label 1 is set to Red, but the height and width do not change.

8. Select the label that you created in step 6 and choose Apply Default from the Layout menu.

The height and width of Label 2 shrinks to fit the number of characters that you typed in.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 12, "Using Special Design Effects on Forms and Reports," page 312, and Chapter 18, "Designing Reports," page 430.

For more information, search for "default properties" then "Changing Default Property Settings for Controls" using the Microsoft Access Help menu.

For more information, search for "label properties" then "Label Control Properties" using the Microsoft Access Help menu.

[References](#)

INF: Creating Custom Navigation (VCR) Buttons on a Form

Article Number: Q94931
CREATED: 25-JAN-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

You can use the Go To command on the Records menu, the Up and Down Arrow keys, Page Up and Page Down, or the [vertical scroll bar](#) to move between records in a [datasheet](#) or [form](#). However, the most efficient way to move between records in large databases is with the [navigation buttons](#) in the [horizontal scroll bar](#) lower-left corner of the window. These navigation buttons are also known as VCR buttons.

You can replace the VCR buttons with custom navigation buttons. This article describes how to add command buttons to the form that go to the first, last, next, and previous actions.

This article assumes that you are familiar with [Access Basic](#) and how to create and use Access Basic procedures.

MORE INFORMATION

Use the following steps to create custom first, last, next, and previous buttons on a form to replace the VCR buttons:

1. Create four command buttons. Place them on the form in an appropriate place. Make the following changes to the [command button](#) properties:

```
Object: command button
-----
ControlName: First
Caption: First
OnPush: =GotoRecord("First")
```

```
Object: command button
-----
ControlName: Last
Caption: Last
OnPush: =GotoRecord("Last")
```

```
Object: command button
-----
ControlName: Next
Caption: Next
OnPush: =GotoRecord("Next")
```

```
Object: command button
-----
ControlName: Previous
```

Caption: Previous
OnPush: =GotoRecord("Previous")

3. Open a new [module](#) or a previously created module and enter this code:

Note: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declarations section of the module.  
'*****
```

Option Explicit

```
'=====
' Create the following GotoRecord() function in the Module
'=====
Function GotoRecord (ByVal Direction As String) As Variant
    Dim Choice, HoldChoice As Integer
    Dim MyForm As Form

    Direction = UCase(Direction)
    Select Case Direction
        Case "FIRST":
            Choice = 1
            Forms(Screen.Activeform.Formname)!Next.Enabled = -1
            Forms(Screen.Activeform.Formname)!Previous.Enabled = -1
        Case "LAST":
            Choice = 2
            Forms(Screen.Activeform.Formname)!Next.Enabled = -1
            Forms(Screen.Activeform.Formname)!Previous.Enabled = -1
        Case "PREVIOUS":
            Choice = 3
            Forms(Screen.Activeform.Formname)!Next.Enabled = -1
        Case "NEXT":
            Choice = 4
            Forms(Screen.Activeform.Formname)!Previous.Enabled = -1
        Case Else
    End Select

    If Choice <> 0 Then
        HoldChoice = Choice
        Choice = Choose(Choice, A_FIRST, A_LAST, A_PREVIOUS, A_NEXT, _
            A_NEWREC)
        On Error GoTo ErrorTrap
        DoCmd GoToRecord A_FORM, Screen.Activeform.Formname, Choice
        Exit Function
    End If

ErrorTrap:
    If HoldChoice = 3 Then
        DoCmd GoToControl "Next"
        Forms(Screen.Activeform.Formname)!Previous.Enabled = 0
    ElseIf HoldChoice = 4 Then
        DoCmd GoToControl "Previous"
        Forms(Screen.Activeform.Formname)!Next.Enabled = 0
    End If
End Function
```

```
End If  
Resume Next
```

```
End Function
```

4. Save the function in the module.

Now you can port the four buttons and the function to any form by using the `Screen.ActiveForm.Formname` property, which returns the name of the active form. This makes the `GotoRecord()` function universal. For the function to work correctly, you need to ensure that the control names of the buttons are identical to what is specified in this article.

References:

"Microsoft Access Introduction to Programming," version 1.0.

[References](#)

INF: Exporting Queries to ASCII

Article Number: Q94929
CREATED: 25-JAN-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Microsoft Access does not have a documented feature to export the results of a [query](#) to ASCII text files. However, this feature is available through the use of the TransferText [macro](#) action.

More Information:

The TransferText macro action does not specify the use of query names in the TableName parameter of the action. The parameter does not use a [combo box](#) for [table](#) name choices, so the user is not limited to using table names. By using the name of a query in the TableName parameter of a TransferText action, the results of a query can be exported to text files. This feature is undocumented. For specific instructions on using the TransferText macro action, search for TransferText using the Help menu.

[References](#)

PRB: 'Not Enough Memory to Run SETUP.EXE' Error

Article Number: Q94928
CREATED: 25-JAN-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

An attempt to run the Microsoft Access Setup program fails and Setup displays one of the following out-of-memory error messages:

Insufficient Memory

or-

Not enough memory to run SETUP.EXE

When this error occurs, sufficient space is available on the disk you are installing Microsoft Access to and your AUTOEXEC.BAT file contains a valid SET TEMP statement that specifies a directory that has free disk space.

CAUSE

One cause of this error relates to an incorrect CMOS setting for the floppy disk drive from which Microsoft Access is installed.

RESOLUTION

Modify the CMOS setting to correctly configure the disk drive. If you need assistance to change the settings appropriately, contact your dealer or hardware manufacturer.

MORE INFORMATION

The error messages above can occur when you install Microsoft Access from a 1.2 MB disk drive and the CMOS setting for that drive indicates that it is a 360 KB disk drive.

[References](#)

INF: How to Convert Twips to Pixels

Article Number: Q94927
CREATED: 25-JAN-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Because Microsoft Access stores dimension/location properties as twips, it may be necessary to convert to pixels in certain cases, such as when calling a Windows API [function](#). This article describes how to do this.

This article assumes that you are familiar with [Access Basic](#) and how to create and use Access Basic procedures.

MORE INFORMATION

The ConvertTwipsToPixels() function below can be used to convert twips to pixels. Note that pixels are not always square (the height and width are not the same), so it is necessary to pass in the desired "direction" to use (horizontal or vertical).

To add this function to your application, do the following:

1. Create a new [module](#) and enter the following Windows API declarations and constants into the [Declarations section](#):

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Option Explicit
Declare Function GetDC% Lib "User" (ByVal hw%)
Declare Function ReleaseDC% Lib "User" (ByVal hw%, ByVal hDC%)
Declare Function GetDeviceCaps% Lib "GDI" (ByVal hDC%, _
                                           ByVal iCapability%)
```

```
Const WU_LOGPIXELSX = 88
Const WU_LOGPIXELSY = 90
```

2. Add the following function:

```
Function ConvertTwipsToPixels (nTwips%, nDirection%) As Integer
    Dim hDC%                'Handle to device
    Dim nPixelsPerInch%
    Const nTwipsPerInch = 1440
    hDC% = GetDC(0)
    If (nDirection% = 0) Then    'Horizontal
        nPixelsPerInch% = GetDeviceCaps(hDC%, WU_LOGPIXELSX)
    Else                        'Vertical
        nPixelsPerInch% = GetDeviceCaps(hDC%, WU_LOGPIXELSY)
    End If
    ConvertTwipsToPixels = nTwips% / nPixelsPerInch%
End Function
```

```
End If
hDC% = ReleaseDC(0, hDC%)
ConvertTwipsToPixels = (nTwips% / nTwipsPerInch%) * nPixelsPerInch
End Function
```

3. To call this function, pass the number of twips you want to convert, and another parameter indicating whether the horizontal or vertical measurement (0 for horizontal, non-zero for vertical). The following is a sample call:

```
OldTwips = 2377
NewPixels = ConvertTwipsToPixels(OldTwips, 0)
```

[References](#)

PRB: Cannot Process Close Action in Report's OnOpen Property

Article Number: Q94926
CREATED: 25-JAN-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You receive the following error message when you attempt to process a Close Action in a [macro](#) that is assigned to the OnOpen property of a [report](#):

Can't run this action while processing a report [event](#).

RESOLUTION

Do not assign a Close action in a macro that is assigned to the OnOpen property of a report.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new macro called CloseForm:

Macro Name	Action
-----	-----
CloseForm	Beep
	Close

CloseForm Actions

Close

Object Type: Form

Object Name: Form1

2. Create another new macro called CloseReport:

Macro Name	Action
-----	-----
CloseReport	Beep
	Close

CloseReport Actions

Close

Object Type: Report
Object Name: Report1

3. Create a new form, and assign the CloseForm macro to the form's OnOpen property. Save this form as Form1.
4. Switch to Form view.
5. Create a new report, and assign the CloseReport macro to the report's OnOpen property. Save this report as Report1.
6. Switch to Print Preview.
7. The computer will beep, and the following error message will be displayed:

Can't run this action while processing a report event.

Notice that you received this error message with the report, but not when using the Close action with the OnOpen property of a form.

[References](#)

INF: Creating Result Equivalent to SQL UNION Operator Result

Article Number: Q94830
CREATED: 19-JAN-1993
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The [SQL](#) UNION operator is not supported in Microsoft Access. However, you can obtain an equivalent result (a UNION product) by using the following two-step process to create two queries, a [make-table query](#) and an [append query](#).

MORE INFORMATION

The product of a UNION includes all records from two related tables -- those that match on the related key [field](#) and those that do not match on the related key field.

For example, say you have two tables, Table A and Table B. Table A contains the following fields and records:

Key1	Field A
1	Seattle
2	Portland
4	Chicago

Table B contains the following fields and records:

Key2	Field B
1	Washington
3	Oregon
5	Illinois

The UNION product of the two tables would look like this:

Key	Field A	Field B
1	Seattle	Washington
2	Portland	
3		Oregon
4	Chicago	
5		Illinois

Use the following two-step process to obtain a UNION product in Microsoft Access:

Step One: Collect the Records that Have a Matching Key Value

1. Create a query, and add the two tables (Table A and Table B) to the query grid.
2. Join the two tables by using the mouse to connect Key1 in Table A to Key2 in Table B.
3. Double-click the join line between the two tables to bring up the Join Properties dialog box. Select the second option, "Include ALL records from 'Table A' and only those records from 'Table B' where the joined fields are equal."
4. Add three fields to the query grid: Key1 from Table A, Field A from Table A, and Field B from Table B. You can use the mouse to drag these fields onto the query grid.
5. Choose Make Table... from the Query menu. The Query Properties dialog box appears. The default selection is Current Database under Make New Table. Enter an appropriate table name (for example, UNION Result) for the result table, and choose the OK button.
6. Run the query to create the new table (UNION Result). It will contain the following fields and records:

Key1	Field A	Field B
1	Seattle	Washington
2	Portland	
4	Chicago	

Step Two: Add the Rest of the Records to the Result

To complete the UNION, you need to add the records in Table B that do not have a matching key field value in Table A.

1. Create another new query. Add two tables (Table B and UNION Result) to the query grid.
2. Join the two tables by using the mouse to connect Key2 in Table B to Key1 in the UNION Result table.
3. Double-click the join line between the two tables to bring up the Join Properties dialog box. Select the second option, "Include ALL records from 'Table B' and only those records from 'UNION Result' where the joined fields are equal."
4. Add three fields to the query grid: Key2 from Table B, Field B from Table B, and Key1 from UNION Result. You can use the mouse to drag the fields onto the query grid. You do not need Field A from UNION Result because UNION Result already contains all the rows from Table A.
5. Choose Append... from the Query menu. The Query Properties dialog box appears. Ensure that UNION Result is in the Table Name field under Append To. Then choose the OK button.
6. Verify that the query grid now has a line labeled Append To. Choose

Table Names from the View menu to show table names on the query grid.
Now the query grid should look something like this:

Field:	Key2	Field B	Key1
Table:	Table B	Table B	UNION Result
Sort:			
Append To:	Key1	Field B	
Criteria:			Is <u>Null</u>

Note that the Key1 field from the UNION Result table is not appended to anything and it has an Is Null restriction on it. This ensures that no records from UNION Result are duplicated and appended to it when you run the append query.

7. Run the append query. The UNION Result table looks like this:

Key1	Field A	Field B
1	Seattle	Washington
2	Portland	
3		Oregon
4	Chicago	
5		Illinois

As you can see, record 1 has a match on the joined fields, but all the other records are represented only once. This is the correct UNION product. If the Key1 values in UNION Result are not in ascending order, it is because there is no primary key on that column.

[References](#)

PRB: Btrieve Files Created with Btrieve 6.0 NLM Not Supported

Article Number: Q94829
CREATED: 19-JAN-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you attempt to attach to a Btrieve FILE.DDF file, the following error message is displayed:

Disk or Network error

RESOLUTION

This error message appears when you attempt to attach to a Btrieve file created with the Btrieve 6.0 NLM version. Microsoft Access does not support Btrieve files created with this networked version of Btrieve. Microsoft Access only supports Btrieve files created with Btrieve 5.x.

[References](#)

PRB: Empty Table List When Attaching Btrieve Table

Article Number: Q94828
CREATED: 19-JAN-1993
MODIFIED: 02-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you attach a Btrieve [table](#), you select the Xtrieve dictionary file (FILE.DDF) in the Select File box. When you choose OK, Microsoft Access opens the dictionary file and displays a list of tables you can import or attach. However, the list is empty.

CAUSE

You do not have the FIELD.DDF file that Microsoft Access requires to attach a Btrieve table.

RESOLUTION

Microsoft Access requires the following three .DDF files in order to attach a Btrieve table: FILE.DDF, INDEX.DDF, and FIELD.DDF. Make sure that you have all three .DDF files before you try to attach a Btrieve table.

MORE INFORMATION

When Microsoft Access cannot find the FIELD.DDF file, it creates a 6144-byte dummy FIELD.DDF file in the same directory in which FILE.DDF appears. If you find this dummy file, you know that you are missing the required FIELD.DDF file. Use Xtrieve or another other third-party utility to create the required .DDF files.

Btrieve and Xtrieve are manufactured by Novell, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

INF: Definitions for Settings in WIN.INI Btrieve Section

Article Number: Q94827
CREATED: 19-JAN-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access uses the Btrieve section of the WIN.INI file when it uses Btrieve files. After you install Microsoft Access, the WIN.INI file contains the following default settings in the [btrieve] section:

```
[btrieve]
options=/m:64 /p:4096 /b:16 /f:20 /l:40 /n:12 /t:C:\ACCESS\BTTRIEVE.TRN
```

MORE INFORMATION

Although these are the recommended option settings, you can change these options in your WIN.INI file using the following [table](#) as a guide. If the option you want to set is not on the line, add the option and desired setting at the end.

Option	Description	Required Value
-----	-----	-----
/m	Memory Size	At least 38
/p	Page Size	4096
/b	Pre-Image Buffer Size	16
/f	Open Files	At least 4
/l	Multiple Locks	At least double the setting for the Open Files option (/f)
/n	Files in a Transaction	At least 4
/t	Transaction Filename	Valid path to a .TRN file; in a multiuser environment, the network path to the .TRN file is shared by all users
/u	Compression Buffer Size	If your Btrieve data files are compressed, at least the length (in kilobytes) of the longest record in your data files
/i	Pre-image File Drive	
/c	Index Compaction	
/a	Activate Logging	
/s	Discard Unneeded Segments	

References:

For more information on these options, see Novell's manual "Btrieve for Windows Installation and Operation."

[References](#)

PRACC9301: Wrong RowSourceType for Combo Box Causes GP Fault

Article Number: Q94826
CREATED: 19-JAN-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

If you set the RowSourceType for a [combo box](#) or [list box](#) to an invalid value, when you switch from Form [Design view](#) to Form Browse view, a general protection (GP) fault may occur.

CAUSE

Valid values for the RowSourceType property are Table/Query, Value List, and Field List. If RowSourceType is set to Table/Query, an [SQL](#) SELECT statement is a valid value for the RowSource property.

If a SQL statement is incorrectly entered into RowSourceType, a GP fault may occur. This problem may also occur if the RowSourceType information includes random characters.

RESOLUTION

Be sure that RowSource type is set to a valid value: Table/Query, Value List, or Field List.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

Steps to Reproduce Problem

1. In the sample [database](#) NWIND.MDB, create a new, unbound blank [form](#), and place a combo box on the form.
2. If RowSourceType is set to

Select [employee id] from employees;

when you switch from Form Design view to Form Browse view, a dialog box with the following text is displayed:

'select [employee id] from employees;' isn't a valid setting for RowSource type property.

Choosing OK at this point would switch you to Form Browse view. (If you are using a list box, #Error is displayed in the list box.)

3. If the WHERE clause [employee id] = 7 is added to the SQL statement in RowSourceType, the result is the same.

If the WHERE clause is changed to [employee id] = forms!form10!xx, a GP fault occurs, and the following message is displayed:

```
Application Error  MSACCESS caused a General Protection  
Fault in module MSACCESS.EXE at 009A:09F6.
```

Note: xx, as referred to in the WHERE clause, is not a control on the form.

References:

For more information, search for "RowSource," then "Filling a List Box or Combo Box Using an SQL Statement" using the Microsoft Access Help menu.

"Microsoft Access User's Guide," version 1.0, Chapter 9, "Designing Forms," pages 242-244

[References](#)

INF: Error Messages May Be Caused by International Setting

Article Number: Q94825
CREATED: 19-JAN-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

An attempt to enter an expression that requires a list separator into the QBE grid field or criteria cells, or as the ControlSource for a text box in a Form or Report fails and Microsoft Access generates the following message:

Syntax error.

An attempt to enter an expression that requires a list separator into an Action Argument for a Macro action fails and Microsoft Access generates the following message:

Can't parse expression: 'Expression that was entered.'

CAUSE

These errors can be caused by an incorrect setting for the "slist" parameter in the [intl] section of your WIN.INI file. This setting specifies the character used to separate items in a list. In United States English, the most common list separator is a comma (,).

RESOLUTION

There are three methods to address this situation, as follows:

- Use the Microsoft Windows Control Panel to modify the list separator in your WIN.INI file. Choose International from the Settings menu. Specify the correct separator in the List Separator field.
- Use a text editor to modify the list separator in your WIN.INI file.
- Modify the expression to use the list character specified in your WIN.INI file.

MORE INFORMATION

When this error occurs, the WIN.INI file might appear as follows:

```
[intl]
slist=;
```

The default United States English settings are as follows:

```
[intl]  
slist=,
```

[References](#)

INF: How to Install the Database Analyzer on the Help Menu

Article Number: Q94822
CREATED: 19-JAN-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

To add the Database Analyzer to your [menu bar](#), add the following lines to the MSACCESS.INI file in the Windows directory:

```
[Menu Add-ins]
&Database Analyzer==StartAnalyzer()
```

```
[Libraries]
Analyzer.MDA=
```

Restart Microsoft Access. Now, Database Analyzer appears as a choice on the Help menu. When you choose Database Analyzer from the Help menu, Database Analyzer automatically starts. Now, it will appear as a menu option when you open any [database](#).

References:

For information on adding other items to the Help Menu, [query](#) on the following words in the Microsoft Knowledge Base:

msaccess.ini and menu

For additional information on the Database Analyzer, review the PSSKB.TXT text file in the Access directory. The answer to the last question (23) in the PSSKB.TXT file describes the Database Analyzer utility.

[References](#)

INF: Use Append Query to Set Initial Value of Counter Field

Article Number: Q94821
CREATED: 19-JAN-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can change the starting value of a Counter field to a number other than 1 by using an append query.

MORE INFORMATION

By design, Microsoft Access always numbers Counter fields beginning with the number 1. You cannot edit a Counter field or change its starting value.

However, you can force Microsoft Access to number a Counter field with a number you choose by doing the following:

1. Create a new table containing a single Number field. Set FieldSize to Long Integer.
2. Give the new Number field a name identical to the name of the Counter field in your original table.
3. Use an append query to join this new record to your original table. This action forces Microsoft Access to number any new Counter fields with your number + 1.

For the expanded procedure, see the "How to Renumber the Counter Field" section below.

Your original table has the following properties:

```
Table: Receiving
-----
FieldName: ID Number
Data Type: Counter

FieldName: Name
Date Type: Text
```

How to Renumber the Counter Field

1. Create a new table with the following structure and save it as Sending:

```
Table: Sending
-----
```

FieldName: ID Number
Data Type: Number
FieldSize: Long Integer

2. Open the Sending table in [Datasheet view](#). Add a new record by inserting a value in the Number field. The value you enter must be 1 less than the starting value you want for the Counter field in your original table. For example, if you want the Counter value in your original table to start at 100, enter 99 in the Number field.
3. Create a new append query to append the single record from the new table (Sending) to your original table (Receiving). To do this, choose Query, then choose New and select the Sending table. Save the query as AppendCounter.
4. From the Query menu, choose Append. Select Receiving from the Append To Table Name [combo box](#), then choose OK.

At this point, you have instructed Microsoft Access to append (or join) a field (or fields) from the Sending table to the Receiving table. Now you need to tell Microsoft Access which field or fields to append.

5. Append the ID Number field from the Sending table to the Receiving table either by selecting ID Number from the Field combo box in the query grid or by dragging the ID Number field from the Sending table to the query grid.

Because Microsoft Access recognizes that the two fields in Sending and Receiving have identical names, it automatically places ID Number in the Append To field of the query grid, as follows:

```
Query: AppendCounter
-----
FieldName: ID Number
AppendTo: ID Number
```

6. Run the query by choosing the exclamation point (!) button on the [toolbar](#). The following message is displayed:

1 Row(s) will be appended.

Choose OK to append the record to the Receiving table.

7. Delete the Sending table, then delete the newly appended [row](#) from the Receiving table.

You are now ready to add a new record to your original table. The Counter will start at the number you have chosen.

NOTE: Do not compact the [database](#) before adding a new record to the original table. If you do, Microsoft Access will reset the Counter value to the number 1.

REFERENCES

=====

For more information on append action queries and Counter fields, search on the following words here in the Microsoft Knowledge Base:

counter and append

For more information on customizing Counter fields, search on the following words here in the Microsoft Knowledge Base:

counter and custom

"Microsoft Access User's Guide," version 1.0, page 174

[References](#)

PRB: GraphWizard Does Not Run or 'Syntax Error' Is Displayed

Article Number: Q94601
CREATED: 11-JAN-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you choose the Graph icon from the [toolbox](#), the [GraphWizard](#) does not run, or the following error message is displayed:

Syntax Error

CAUSE

This error is caused by an incorrect List Separator entry in the [International] section of your WIN.INI file.

RESOLUTION

In Control Panel, select the International icon. In the List Separator box, enter the default value, a comma (,). Microsoft Graph should now run correctly. Note that different List Separator entries will cause different results.

MORE INFORMATION

Steps to Reproduce Behavior

1. In Control Panel, select the International icon.
2. Change the List Separator.

Different settings will cause different results. For example, a minus sign (-) will not open Graph or give an error message. A space will generate a "Syntax Error" message.

3. Create a new [form](#).
4. Choose the [graph](#) icon from the toolbox.
5. Select an area on your form and click the mouse to insert the graph. The border will briefly appear, but then disappear without opening the GraphWizard, or the "Syntax Error" message will be displayed.

[References](#)

PRB: 'Invalid Database Object' Occurs Inexplicably

Article Number: Q94600
CREATED: 11-JAN-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you are trying to use a [database](#) object (Database, [QueryDef](#), Table, Dynaset, [Snapshot](#)) within [Access Basic](#), you may receive the following error:

Invalid Database Object.

CAUSE

The problem occurs if you are within a [transaction](#), you call another [function](#) or procedure which dimensions the database, and then leave the function or procedure which implicitly close the object that was opened within the transaction.

In this case Microsoft Access will try to close it and since it was opened within a transaction, Microsoft Access will roll back ALL levels of nested transactions.

If any other [database objects](#) were opened within the transaction, rolling back has the effect of closing them. This means that any references to an object opened within a transaction after another object was forced to be implicitly closed will result in 'Invalid database object'.

This problem will not occur if you always close database objects before they go out of [scope](#). The problem only occurs if you Microsoft Access implicitly close them.

RESOLUTION

- You should always try to open database objects outside of transactions.
- You should always explicitly close database objects.

MORE INFORMATION

Steps to Reproduce Behavior

The following code fragment will illustrate the problem.

```
Function One ()  
    Dim MyDB As Database, MyTable As Table
```

```

Set MyDB = CurrentDB()
BeginTrans
  Set MyTable = MyDB.OpenTable("Table1")
  X = Two()
  Debug.Print MyTable.RecordCount "Invalid database object."
  MyTable.Close
CommitTrans
MyDB.Close
End Function

Function Two ()
  Dim MyDB2 As Database
  Set MyDB2 = CurrentDB()
End Function

```

MyDB loses scope, implicitly does roll back and close of MyDB with effect of also closing MyDB in the One function.

To circumvent the problem, change the procedure Two to:

```

Function Two()
  Dim MyDB2 as Database
  Set MyDB2 = CurrentDB()
  MyDB2.Close
End Sub

```

[References](#)

INF: Placing the User Login Name on a Form or in the Title Bar

Article Number: Q94599
CREATED: 11-JAN-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article explains how to use the User() [function](#) to display the current user name in a [control](#) on a Form or in the [title bar](#) of the Microsoft Access window.

More Information:

In Microsoft Access, you can implement [security](#) to control user access to the different objects in your [database](#). Then you can use the User() function to display the current user.

How to Display User Name in a Control on a Form

Use the following procedure to display a user name in a control on a [form](#):

1. Create a form and place an unbound [text box](#) control on the form.
2. Enter =User() in the text box ControlSource property.
3. Change to [Form view](#) to see the current user name displayed in the control.

Display user name in Microsoft Access Window Title Bar

Use the following procedure to display a user name in the title bar of the Microsoft Access window:

1. Create a new [module](#).
2. Enter the following [Access Basic](#) code:

Note: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'=====
'Declarations section of the module
'=====
Option Explicit

Declare Function SendMessage& Lib "User" ( _
    ByVal hwnd%, ByVal message%, ByVal wParam%, lParam As Any)
Declare Function GetActiveWindow% Lib "User" ()
```

```

Const WM_SETTEXT = &HC

DIM stTitle$
DIM x%

'=====
'The following function places the user name in the title bar.
'=====

Function SetDatabase windowWindowTitle ()
    stTitle$ = "User = " + User()
    x% = SendMessage(GetActiveWindow(), WM_SETTEXT, 0, ByVal stTitle$)
End Function

'=====
'End of code section
'=====

```

4. Save the module.
5. Create a new macro (ShowUser). Give it the RunCode action with the following function name as the action argument:


```

                SetDatabase windowWindowTitle()
            
```

Save the macro and name it ShowUser.
6. From the Run menu, choose Compile All.
7. Restart Microsoft Access and run the ShowUser macro. You should see the window title change from Microsoft Access to User = username

References:

For more information, search for "User," then "User Function" using the Microsoft Access Help menu.

[References](#)

PRACC9301: Incorrect Action Query Message Box Text

Article Number: Q94598
CREATED: 11-JAN-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

An error message generated by an [action query](#) (that is, a Delete, Make Table, Update, or Append [query](#)) may contain the following incorrect message box text:

Errors were encountered: The contents of fields in <number> [record](#)(s) were deleted, <number> record(s) were lost due to key violations and <number> record(s) were locked and couldn't be modified. Proceed anyway?

CAUSE

This problem is caused by the way Microsoft Access handles errors for action queries. Microsoft Access uses the same error message for every action query instead of using a separate error message for each type of action query.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

The portion of the above error message that reports key violations:

<number> record(s) were lost due to key violations

becomes active whenever [referential integrity](#) is violated during a query. However, this specific information only applies to Append and Make Table queries. If an Update or Delete query violates referential integrity, this portion of the error message erroneously gives the impression that records have been lost, when in fact they simply were not updated or deleted.

Steps to Reproduce Problem

-
1. Create the following two tables:

Table: Table1

FieldName: ID [PrimaryKey]
 DataType: Text
FieldName: Name
 DataType: Text

Table: Table2

FieldName: ID
 DataType: Text
FieldName: Hobby
 DataType: Text

2. Create a one-to-many relationship between Table1 and Table2, with ID as the matching field. Enforce referential integrity rules.
3. Add two records to each table.
4. Create the following query based on Table1:

Query: Query1

Field: ID
Field: Name

5. Change Query1 from a Select query to a Delete query by choosing Delete from the Query menu.
6. Run Query1 by choosing Run from the Query menu.
7. Choose the OK button to delete the records.

The following error message is displayed:

Errors were encountered: The contents of fields in 0 record(s) were deleted, 2 record(s) were lost due to key violations and 0 record(s) were locked and couldn't be modified. Proceed anyway?

This message correctly identifies that there were key violations (because of referential integrity), but it incorrectly indicates the loss of two records. In actuality, the records were not modified and therefore not lost.

[References](#)

INF: System Objects Retain System Object Properties

Article Number: Q94597
CREATED: 11-JAN-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

A system object always retains the original system object attributes. Operations performed on system objects such as Import, Rename, and Copy all preserve the flags in MSysObjects.

MORE INFORMATION

Operations such as Import, Rename, Copy, and Paste do not change the flags in MSysObjects. In order to get a copy of a system table without retaining the system settings, you should use a Make Table query.

Steps to Reproduce Behavior

1. From the View menu, choose Options. Set the argument for Show System Objects to Yes.
2. Import the MSysAccounts table from the SYSTEM.MDA database.
3. Rename the imported table to Tablefoo.
4. From the View menu, choose Options. Set the argument for Show System Objects to No.

The table Tablefoo disappears from the database container list because it has the attributes of a system object.

[References](#)

PRB: Error Message When Attaching Btrieve Table

Article Number: Q94420
CREATED: 06-JAN-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

When you try to attach to a Btrieve table in Microsoft Access, you receive the following error message:

<tablename> is corrupted or isn't a Microsoft Access database.

CAUSE

When Microsoft Access attaches a Btrieve table, it compares information in the Xtrieve dictionary file (FILE.DDF) with information in the TBLNAME.DAT file. The most common reasons for the above error message are differences in index information.

RESOLUTION

Redefine the Btrieve index so that it is exactly the same length as the field it indexes.

STATUS

This problem does not occur in Microsoft Access version 1.1.

MORE INFORMATION

In the Btrieve file format, the length of the index may not match the length of the field. This is a valid condition. The Xtrieve dictionary file (FILE.DDF) ignores these indexes, but Microsoft Access interprets them as invalid and returns the above error message.

You can correct this problem by turning the invalid indexes into valid indexes. Use one of the following options to ensure that the length of the index and the length of the field are identical:

- Ask the vendor who created the Btrieve files to change the length of the index.
- Use Btrieve or Xtrieve to redefine the index.
- Use any third-party utility available to modify the Btrieve index.

The following four examples all require changes to the Btrieve index:

Example 1

The TBLNAME.DAT file is defined with the following three columns:

Column name	Field Length	Index Length
-----	-----	-----
Firstname	10	<= Combined length of
Lastname	10	<= 20 characters
Company	25	

Index1 is defined with a length of 20 characters, the combined length of the Firstname and Lastname fields.

Resolution: Redefine the Btrieve index into one multiple-segment index, giving each segment a defined length of 10 characters.

Example 2

The TBLNAME.DAT file is defined with the following three columns:

Column name	Field Length	Index Length
-----	-----	-----
Firstname	10	4 characters
Lastname	10	
Company	25	

Index2 is defined with a length of four characters, shorter than the defined length of the Firstname field.

Resolution: Redefine the Btrieve index on Firstname. Give it a 10-character length, matching the size of the Firstname field.

Example 3

The TBLNAME.DAT file is defined with the following three columns:

Column name	Field Length	Index Length
-----	-----	-----
Firstname	10	12 characters
Lastname	10	
Company	25	

Index3 is defined with a length of 12 characters, slightly longer than the defined length of the Firstname field.

Resolution: Redefine the Btrieve index on Firstname. Give it a 10-character length, matching the size of the Firstname field.

Example 4

The TBLNAME.DAT file is defined with the following two columns:

Column name	Length
-----	-----

Firstname - 10
Lastname - 10
Company - 25

Index4 is defined on a numeric byte range index. There is no index defined on any field in the table.

Resolution: Remove the numeric byte range index completely, then redefine the index on an existing field. Be sure that the length of the index matches the length of the indexed field.

Btrieve and Xtrieve are manufactured by Novell, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

PRB: Exclusively Locked Error When Attaching to Btrieve Table

Article Number: Q94419
CREATED: 06-JAN-1993
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

When attaching a Btrieve [table](#), you get one of the following error messages when you choose FILE.DDF:

Table <tablename> is exclusively locked.

or

Couldn't open file <filename>.

CAUSE

When attaching to a table, Microsoft Access needs to open the target Btrieve table exclusively for a moment. If any other user has the target table open at the same time Microsoft Access is attempting to attach to it, you will see one of these error messages.

RESOLUTION

Find an available time when no other user is using the target Btrieve table. Once Microsoft Access attaches to the Btrieve table successfully, you will no longer receive the error messages.

STATUS

This problem does not occur in Microsoft Access version 1.1.

[References](#)

INF: Printing Microsoft Access Reports on HP Laserjet

Article Number: Q94418
CREATED: 06-JAN-1993
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you print to an Hewlett-Packard Laserjet II, set the option Print TrueType As Graphics in Print Setup. This is necessary because mixing TrueType (TT) fonts and graphical objects on the HP Laserjet printer can cause inconsistent results (that is, not true WYSIWYG). To set the option, choose Print Setup from the File menu. Choose the Options button. Then choose the box in the bottom left hand corner, Print True Type as Graphics.

MORE INFORMATION

Before setting the Print Option, you might see the following problems when printing Microsoft Access reports on a Hewlett-Packard Laserjet II:

1. Microsoft Access displays white text on a black line in the [report preview](#), but it prints a solid black line.
2. Microsoft Access begins to print reports containing lines (graphics) with the text aligned correctly, but subsequently prints the line and then prints the text shifted to the far right.

For additional information, [query](#) on the following words in the Microsoft Knowledge Base:

truetype and graphics and laserjet

[References](#)

PRB: Unable to Import Portion of Fixed-Width Text Files

Article Number: Q94417
CREATED: 06-JAN-1993
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

You cannot import any portion of a fixed-width text file.

CAUSE

By design, each record in a fixed-width text file must be separated from the next by a carriage return and line feed and all records must contain the same number of characters.

RESOLUTION

Modify the text file to ensure that all records meet these requirements:

1. Each record is on a different line.
2. Each record contains the same number of characters.

MORE INFORMATION

Microsoft Access cannot import fixed-width text records that are not delimited by a carriage return or line feed.

Mainframe users may be familiar with long strings of data where the first x number of characters are the first record, the next x number of characters are the second record, and so on. In order to import these records into Microsoft Access, you must ensure that each record is on its own line with a carriage return and line feed at the end of each record.

For example, consider the following string of characters:

```
lastnamefirstnameagelastnamefirstnameagelastnamefirstnameage
```

You cannot import these records into Microsoft Access without modification. If you try to import the above string as a fixed-width text file, you will either create an empty table or you will be unsuccessful and need to repair your database.

To modify the file, use Access Basic and follow these steps:

1. Read in the file in Binary mode with a fixed length string that is the same length as the record.
2. Write each string to a new file that is open for sequential output.

3. Repeat steps 1 and 2 until you reach the end of file.

Microsoft Access can import the following records:

```
lastnamefirstnameage  
lastnamefirstnameage  
lastnamefirstnameage
```

To import a file containing records like these, follow these steps:

1. Choose Imp/Exp Setup from the File menu.
2. Change the Text Delimiter from " to {none}.
3. Input the following settings for Field Information:

Field Name	Data Type	Start	Width
Last	Text	1	8
First	Text	9	9
Age	Number	17	3

4. Choose OK, and save this specification as TEST.
5. From the File menu, choose Import.
6. Select Text (Fixed-Width). Then choose OK.
7. Select the text file you want to import. Then choose Import.
8. In the Import Text Options dialog box, select the TEST specification. Then choose OK.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 4, "Importing, Exporting, and Attaching," pages 75-81.

[References](#)

INF: Adding a Dynamic Counter to a Query to Count Records

Article Number: Q94397
CREATED: 06-JAN-1993
MODIFIED: 14-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can identify the position of a record by using a dynamic counter value in a query field. Microsoft Access has no intrinsic function to do this, but you can do it by using the user-defined function.

MORE INFORMATION

There are two steps to the process:

1. Create a user-defined function named QCntr() to increment the counter for each new record. Make the function look something like this:

```
Declarations:
Global Cntr
Function QCntr(x) As Long
    Cntr = Cntr + 1
    QCntr = Cntr
End Function
```

2. Create the following user-defined function named SetToZero to reset the Cntr value each time the query is run:

```
Function SetToZero()
    Cntr = 0
End Function
```

3. Call the function in the query field by putting QCntr() in an expression. Use the following as a guide:

```
Field: Expr: QCntr([Field])
```

Here [Field] is any field from the underlying table or query. [Field] is passed to the function to force evaluation of the function for every record in the query's dynaset. It doesn't matter which field you use as long as it has a valid field name.

4. Run the query from a macro with the following actions:

```
Macro      Action
-----
[Macroname] Run Code
              Open Query
```

Macro Action Arguments

Run Code

Function Name: SetToZero()

Open Query

Query Name: [Query Name]

View: Datasheet

Data Mode: Edit

Running the query from this macro resets the function QCntr to zero in each iteration to provide accurate results.

REFERENCES

=====

For more information on how to make this function execute more than once for each record, depending on what you do with the dynaset after the query is run, query on the following words here in the Microsoft Knowledge Base:

custom and function and executes

[References](#)

INF: How to Show Field in Datasheet View But Not Form View

Article Number: Q94396
CREATED: 06-JAN-1993
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There is no inherent command to display a field in Datasheet view that is invisible in Form view. By changing the color characteristics and the size, you can create a control that will literally be invisible in Form view yet visible in Datasheet view.

MORE INFORMATION

To create a control that is visible in Datasheet view and invisible in Form view, you need to set the following properties of the control:

```
BorderStyle: Clear  
Width: 0  
Height: 0
```

You will also have to remove the caption box if one exists with the control.

NOTE: The drawback of this method is that although you cannot see the control, the control still has a tab order. Because of this, a user can have the focus set to the control immediately before the invisible control and press TAB. The focus will move to the control, yet the user will not see anything, making it look as though the cursor has disappeared. Pressing TAB again or setting focus to another control with the mouse will cause the cursor to reappear.

[References](#)

PRB: Trailing CR/LFs Stripped from Text Box when Retrieved

Article Number: Q94395
CREATED: 06-JAN-1993
MODIFIED: 23-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

Trailing carriage returns/linefeeds (CR/LFs) are stripped out when retrieving data from a [text box](#). Although it is possible to stuff CR/LFs into a text box, when attempting to retrieve the data, the trailing CR/LFs are truncated.

STATUS

We are considering modifying this behavior in a future release of Microsoft Access.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the Customers [form](#) in the sample [database](#) Northwind Traders (NWIND.MDB).
2. Open the [Immediate window](#) and position it on the screen so that it and the Customers form are both in view.
3. Type the following command in the Immediate window:

```
Forms!Customers.[Contact Name] = "Chris" & Chr$(13) & Chr$(10)
```

At this point, the Contact Name [control](#) on the form should be updated and should contain the text "Chris" with a blank line below it. This shows that the CR/LF is in the text box.

4. Type the following command in the Immediate window.

```
MyValue = Forms!Customers.[Contact Name]
```

The [variable](#) MyValue will contain the text "Chris", but there will be no CR/LF at the end of the text. This can be tested by executing the following command from the Immediate window:

```
Print MyValue & "*"
```

This will return the value "Chris*", which shows that there is no CR/LF at the end of MyValue.

[References](#)

PRB: Control Menu Move Command Requires Keyboard Input

Article Number: Q94394
CREATED: 06-JAN-1993
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

After choosing the Move command from the Control menu of any non-standard window, you are unable to move the window's location by using the mouse.

CAUSE

There are two non-standard Microsoft Access windows: the Palette and Toolbox windows. Like other windows, you can move the window location by dragging the title bar of the window to the new location. But unlike other windows, dragging the title bar will have no effect if you choose the Move command from the window's Control menu.

RESOLUTION

After choosing the Move command from the Control menu of any non-standard window, you must use the ARROW keys on the keyboard to complete moving the window.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the sample database NWIND.MDB.
2. Using the mouse, drag the title bar of the Database window to a new location on your screen.
3. Release the mouse button. You will have successfully moved the Database window.
4. Press ALT+minus sign (-) to open the Database window's Control menu.
5. Choose Move.
6. Repeat steps 2 and 3 above.
7. In order to move this window, you need to use the ARROW keys on your keyboard.
8. Open the Categories form in Design view. Repeat steps 2-7

above on either the Toolbox window or the Palette window (available from the form's View menu).

Additional References:

Microsoft Access "Getting Started," version 1.0, page 78

[References](#)

PRB: 'Welcome to Microsoft Access' Screen Disables Status Bar

Article Number: Q94393
CREATED: 06-JAN-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you start Microsoft Access with the "Welcome to Microsoft Access" screen, the status bar will not show whether the CAPS LOCK and NUM LOCK keys are on or off.

RESOLUTION

If you turn off the "Welcome to Microsoft Access" screen, the status bar functions normally. Another workaround is to close the screen, then switch the NUM LOCK and CAP LOCK keys off then on. This will reinitialize the status bar.

MORE INFORMATION

Steps to Reproduce Behavior

1. Turn on the CAPS LOCK and NUM LOCK keys.
2. Start Microsoft Access with the "Welcome to Microsoft Access" screen enabled.

Notice that the status bar doesn't show that the CAPS LOCK and NUM LOCK keys are on.

[References](#)

PRB: Border Properties Ignored with SpecialEffect

Article Number: Q94392
CREATED: 06-JAN-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The following properties are ignored when you set the SpecialEffects control to Sunken or Raised:

BorderColor
BorderWidth
BorderStyle

STATUS

This is behavior by design.

[References](#)

PRB: Cannot Use 'y' 'n' 't' or 'f' in Yes/No Field

Article Number: Q94391
CREATED: 06-JAN-1993
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

You receive the following error message when you type the letter 'y' or 'n' into you field:

The value you entered isn't appropriate for this field.

CAUSE

You are not typing the entire word into a Yes/No Boolean field, a True/False field, or an On/Off field.

RESOLUTION

Type the entire word 'yes' or 'no' instead of the abbreviation 'y' or 'n' into a field defined with a Yes/No data type. The capitalization does not make a difference.

[References](#)

PRB: Press ENTER when Database Window is Minimized Opens a Doc

Article Number: Q94390
CREATED: 06-JAN-1993
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If no documents are open and you have the Database window minimized when you press ENTER, the object highlighted in the Database window will open. However, if there is an open object (minimized or not), nothing happens when you press ENTER on the Database window icon.

RESOLUTION

To restore the Database window, do one of the following:

- Double-click on the icon with the mouse.
- Press ALT+minus sign (-) to access the Control menu, then press ENTER.

[References](#)

PRACC9301: Selecting Options in Find... Dialog Box under Edit

Article Number: Q94222
CREATED: 30-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

There are three known problems with the Find... [dialog box](#) under the Edit menu in Microsoft Access:

1. The check boxes for Match Case and Search Fields as Formatted work correctly when selected with a mouse, but mistakenly allow both boxes to be selected when using the plus (+) and minus (-) keys to select or clear the check boxes.
2. There is a slight painting problem when you select the All Fields [option button](#) under Search In. The default value is false for the Match Case and Search Fields as Formatted check boxes, but after you select the All Fields option button, Microsoft Access dims the Search Fields as Formatted [check box](#) caption and selects it.
3. After experiencing the painting problem and seeing the dimmed X in the Search Fields as Formatted check box, if you again select Current Field under Search In, Microsoft Access incorrectly selects the Search Fields as Formatted check box. This occurs whether you use the mouse or the keyboard to select Current Field.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

[References](#)

INF: Access Table Structure Is Variable Length

Article Number: Q94221
CREATED: 30-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access will set the field length of a text field to the length of the entered text if the text entered is shorter than the maximum field size.

MORE INFORMATION

When you set Field Size in the Field Properties of a table in design mode, you are setting the maximum length for the field. If text entered into that field is shorter than the specified maximum length, then the length of that field equals the length of the text that has been entered. This means whether you design a text column as 20 characters or 255 characters, it will only take up the amount needed to store the field. The difference between the two is that one is limited to 20 characters on entry while the other will allow up to 255 characters.

References:

Microsoft Access "User's Guide," version 1.0, chapter 3, page 38

[References](#)

INF: Back Button Disabled After Selecting AccessWizard Type

Article Number: Q94220
CREATED: 30-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

When you use either the FormWizard or ReportWizard in Microsoft Access after you select the type of AccessWizard to use (Single Column, Tabular, Graph, and so on), and then choose OK to continue, the Back button is disabled. This means that you cannot go back to reselect a different type of Wizard. The only thing you can do is to choose Cancel and start the AccessWizard process over again.

[References](#)

PRB: GP Fault Using an Out of Date ODBC Driver

Article Number: Q94219
CREATED: 30-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Microsoft Access displays the following error message when using an out of date [ODBC](#).DLL driver:

General Protection Fault (GPF) in ODBC.DLL

CAUSE

When Microsoft Access initiates an ODBC connection, ODBC does not check the version of the ODBC driver. The ODBC setup program will only replace an older version of the ODBC.DLL with a newer version if the file resides in the \WINDOWS\SYSTEM subdirectory. If the file is found elsewhere, ODBC setup places a copy of ODBC.DLL in the \WINDOWS\SYSTEM subdirectory. This can cause a problem if the directory that contains the older .dll is in the path, and is found first.

RESOLUTION

Rename or delete the old ODBC.DLL and run the ODBC setup program from the setup disks that come with Microsoft Access. This will install a correct version of ODBC.DLL.

[References](#)

PRB: Unable to Paste Information into Property Sheet

Article Number: Q94218
CREATED: 30-DEC-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You are unable to paste information into a property sheet using the Edit menu's Paste command. Instead, you keep creating a new label control on your form or report.

CAUSE

You are using menu commands that do not apply to pop-up windows.

RESOLUTION

Use equivalent keyboard actions rather than selecting commands from the Microsoft Access menu bar. In this case, use CTRL+V or SHIFT+INSERT to paste information into a pop-up window.

STATUS

This behavior is by design.

MORE INFORMATION

Property sheets are actually pop-up windows. When you use menu commands, Microsoft Access returns focus (from the pop-up window) to the parent window. When you use equivalent keyboard actions, Microsoft Access retains focus on the current edit control (in this case the property sheet).

Therefore, selecting the Paste command from the Edit menu will paste a new label in the "parent" form or Report window.

Steps to Reproduce Behavior:

1. In the sample database NWIND.MDB, create a form based on the Orders table.
2. In this form, create a combo box bound to the Customer ID.
3. Go to the Customer List query, and copy the SQL statement to the Clipboard.
4. Return to the form and select the RowSource property for the combo box.

5. Try to paste the SQL code into the RowSource property.

If you use CTRL+V or SHIFT+INSERT, Microsoft Access pastes the information from the Clipboard into the property sheet. If you choose Paste from the Edit menu, Microsoft Access creates a new label on the form.

References:

For more information, search for keyword "pop-up forms" then "keyboard: editing keys" using the Microsoft Access Help menu.

[References](#)

INF: Importing DataEase Data Files

Article Number: Q94194
CREATED: 29-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not import DataEase data files directly, but you can export your DataEase data to a formatted disk file that Microsoft Access can import. The information in this article applies to DataEase versions 4.2, 4.5 and 4.53 a character based relational [database](#) system.

To export DataEase data in DataEase, create a DQL [report](#), specify the Format as export, and set the Print Style to print to a disk file. Choose one of the following export formats to create a file Microsoft Access can import:

Lotus (creates a .WKS file)
Variable Length (creates a delimited ASCII text file)
Fixed Length (creates a fixed-width ASCII text file)

MORE INFORMATION

Export the DataEase data in DataEase. Then import that file in Microsoft Access.

Export Using DataEase

To create a formatted disk file in DataEase 4.x, create a DQL report using the following steps. For more detailed information about creating reports, please refer to the DataEase documentation.

1. Define the report specifying the fields you want to include in the output.
2. Specify the report Format as Export by choosing either Define Format or Modify Format under either DQL Advanced Processing or Query by Example - Quick Reports.

Select 7. Export under Define Format or Modify Format. You should see the following list of the export formats DataEase supports.

- * 1. Lotus (wks)
- 2. GrafTalk
- 3. MultiMate
- 4. WordPerfect
- 5. Mail-Merge
- * 6. Variable Length
- * 7. Fixed Length

8. DIF

Choose 1, 2, or 6. Depending on your choice, DataEase may ask you if you want to include field names in the exported file.

3. Specify the Print Style for the report. Enter the Report Destination as option #3 Disk to print to a disk file. Then enter a valid file name in the Filename field.

Import Using Microsoft Access

1. Open the database.
2. From the File menu, choose Import.
3. In the Data Source box, select the import format that matches the export format you chose in DataEase.

Choosing Export/Import Format

Here are some things to consider when choosing an export/import format.

Microsoft Access can import Lotus (.WKS) files and create a table with field names if you included the field names when exporting in DataEase. During the import process, Microsoft Access asks you if the first row contains field names. Most DataEase data types should convert to appropriate Microsoft Access field types.

If you choose the Variable Length format when exporting, DataEase will ask you to enter the field separator. Enter a comma (,). Microsoft Access recognizes the comma as the default field separator.

If you choose the Fixed Length format when exporting in DataEase, you will need to create an Import Specification file in Microsoft Access prior to importing the file.

For additional information registered DataEase users can contact DataEase International HotLine Support at 1-203-374-2825.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 4, "Importing, Exporting, and Attaching," pages 72-81

[References](#)

INF: Attaching to FoxPro Data in Microsoft Access

Article Number: Q94193
CREATED: 29-DEC-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

According to the marketing literature, Microsoft Access cannot attach to FoxPro data, but it is able to import FoxPro data. There is an exception to this rule, which allows a user to attach to, rather than import, FoxPro data.

MORE INFORMATION

FoxPro uses a DBF file structure, similar to the structure that dBASE uses. However, Microsoft Access cannot attach to most FoxPro files because the memo and index structures are different. Consequently, you can attach a FoxPro table as a dBASE IV table if it meets the following two conditions:

- The FoxPro table does not contain any memo fields
- The FoxPro table is attached without any FoxPro index files

If either of these conditions is not met, then you must import the data from the FoxPro table instead of attaching to it.

[References](#)

PRB: Incorrect Characters in Crosstab Query Column Headings

Article Number: Q94192
CREATED: 29-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If the data in a [field](#) used as a [column](#) heading in a [crosstab query](#) contains a decimal, an exclamation mark, or a square bracket character, this character will be changed to an underscore character in the column heading. This does not happen if the same data is used as a [row](#) heading.

CAUSE

Queries can be used as input to other queries, forms, or reports, and therefore column headings must be formatted as field names. Because field names cannot contain decimals (.), exclamation marks (!), or square bracket characters ([or]), these characters are converted to underscore characters in the column headings of the crosstab [query](#).

STATUS

This behavior is by design.

References:

For more information on names, search for "naming conventions" then "Standard Naming Conventions" using the Microsoft Access Help menu.

For more information on crosstabs, search for "crosstab queries" then "Summarizing Data Using a Crosstab Query." using the Microsoft Access Help menu.

[References](#)

PRB: Query Error Caused by Colon in Field Name

Article Number: Q94191
CREATED: 29-DEC-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A [query](#) results in one of the following error messages:

Query must have at least one output [field](#).

-or-

Can't set [criteria](#) unless you've specified a field.

CAUSE

The query criteria includes a field name that ends in a colon (for example, `FieldName:)`).

RESOLUTION

When a field name includes a colon, you must enclose the field name in brackets (for example, `[FieldName:]`).

STATUS

This behavior is by design.

MORE INFORMATION

In a query's [Design view](#), you can rename a field to be more descriptive. Place the insertion point in front of the first letter of the field name in the [QBE](#) grid. Then type the new name followed by a colon. Do not delete the underlying field name. Now, in [Datasheet view](#), you will see the descriptive [label](#) instead of the underlying field name.

Using this technique to change a field name in a query's Design view automatically changes the heading in the query's Datasheet view, and it changes the field name in a [form](#) or [report](#) based on the query. But the underlying field name in the [table](#) does not change.

The error messages shown in SYMPTOMS occur because without brackets around the field name, Microsoft Access interprets the colon in the field name as a request to add a label (rename) that field.

Steps to Reproduce Behavior

1. Create a new table named Test with one field named FullName:.

2. Save the table as Table1.
3. Create a query based on Table1.
4. Drag the field FullName: to the query grid.
5. Run or view the results of the query.
6. You will receive this error:

Query must have at least one output field.

7. Set the criteria for FullName: to a* or any other criteria.
8. You will receive this error:

Can't set criteria unless you've specified a field.

You can prevent the error messages in steps six and eight by enclosing the field name and colon in brackets (for example, [FullName:]).

References:

For more information, search for "Renaming a Field in a Query" using the Microsoft Access Help menu.

Microsoft Access "User's Guide," version 1.0, Chapter 5, Query Basics, page 96.

[References](#)

PRB: Arrow Keys Inconsistent Between Forms and Datasheets

Article Number: Q94190
CREATED: 29-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you set ARROW key behavior to Next Field, pressing an ARROW key at the end of a [field](#) in [Datasheet view](#) will move you to the next field. However, in [Form view](#), pressing an ARROW key at the end of a [control](#) has no effect (that is, it will not move the focus to the next control).

CAUSE

This behavior is by design.

MORE INFORMATION

You can set certain keyboard attributes, such as ARROW key behavior (from the View menu choose Options, then select Keyboard). You can set the ARROW key behavior to either Next Field or Next Character. When set to Next Field, the ARROW key behavior will vary between Datasheet view and Form view. The difference in behavior depends on the mode of the Microsoft Access text editor.

If the editor is in "enter" mode (the entire field or control is selected), pressing an ARROW key in either Datasheet view or Form view moves you to the next field or control and remains in enter mode. When you reach the last field or control, you will move to the next [record](#).

If the editor is in "edit" mode (text has been entered), pressing an ARROW key at the end of a field in Datasheet view will move you to the next field and return to enter mode. However, in Form view, pressing an ARROW key at the end of a control has no effect. This behavior is by design, and is meant to ensure that pressing an ARROW key on the last control in edit mode does not move to the next record.

References:

For more information on text editor, [query](#) in the Microsoft Knowledge Base on the following key words:

key and behavior and edit and box

[References](#)

PRB: Can't Create Two Relationships Between Same Tables

Article Number: Q94189
CREATED: 29-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you try to create two relationships between the same two tables, Microsoft Access will automatically delete the first [relationship](#).

STATUS

This behavior is by design.

MORE INFORMATION

Steps To Reproduce Behavior

-
1. Create two tables that share a common [field](#):

Table: Table1

Field Name: Test [Primary key]
Date Type: Text

Table: Table2

Field Name: Test
Date Type: Text

2. From the Edit Menu choose Relationships and create a relationship between the Table1 and Table2 on the Test field with the Enforce Referential Integrity option checked. Choose Add.

3. Create another relationship between the two tables. Choose Add.

Creating the second relationship deletes the first relationship.

[References](#)

PRB: Form Objects Not Placed on Grid by Wizard

Article Number: Q94176
CREATED: 29-DEC-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When using FormWizards, objects do not appear to be aligned to the grid even though Snap To Grid is on.

CAUSE

FormWizards are designed to place objects on the [form](#) and position them in the appropriate manner. Snap To Grid is a feature to help the user place controls on forms. Since the user is not positioning the objects manually, there is no need for FormWizards to use the Snap To Grid feature.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open NWIND.MDB.
2. Use FormWizards to create a new Single Column form based on the [Employee table](#).
3. Place all fields on the form.
4. Pick Standard style.
5. Open in design mode.

Results: The form objects are NOT aligned to the grid, even though the Snap To Grid option is checked.

[References](#)

PRB: Date/Time Formats Not Used In List or Combo Box Controls

Article Number: Q94175
CREATED: 29-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A list or combo box based on a Date/Time field does not display the date or time format assigned to the underlying field. This is particularly noticeable in list or combo box control on a form.

CAUSE

Microsoft Access does not yet provide a Format property for a list or combo box even though the underlying table field has a defined Format property.

RESOLUTION

Use the Format() function in an expression in a query to format the data. Then base the list or combo box on the query field instead of the underlying table field. In other words, in Query Design view, assign an expression to a field in the QBE (Query by Example) grid. Use the Format() function in the expression to format the Date/Time field.

For example, define a query named QueryTest. Then instead of entering the name of the underlying Time/Date field ([Event Date]) as the field on the QBE grid, enter this expression:

```
Event Day: Format([event date],"mmm d", "yyyy")
```

Base the list or combo box on this query field to display [Event Date] with the desired format.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new table containing these two fields:

Field Name	Data Type
Event ID	Counter
Event Date	Date/Time

2. Assign a Format property to [Event Date] as follows:

Event Date	Field Properties
Format	mmmm d, yyyy

3. Save the table naming it Events. Switch to [Datasheet view](#). In the first [record](#) enter 8/11/92 in the Event Date field. Add a few more records with any valid dates.

At this point the first record's date displays correctly as August 11, 1992.

4. Create a new form, and base it on the Events table. Add a [list box](#) control to the form. Set the properties for the list box as follows:

```
Object:Listbox
-----
Control Name:      Event ID
Control Source:    Event ID
Row Source Type:   Table/Query
Row Source:        Events
Column Count:      2
Column Heads:      Yes
Column Widths:     0;1
Bound Column:      1
```

5. Switch to [Form view](#). As you can see, the date in the list box control is 8/11/92 not August 11, 1992. Microsoft Access ignores the Format property set on the underlying field. The same thing happens with time formats.

[References](#)

PRB: 'FastForward' Button Is Unavailable in Graph FormWizard

Article Number: Q94174
CREATED: 29-DEC-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

When you use the Graph FormWizard to create a [form](#), the "Fast Forward" button (>>|) is unavailable if you use the button labeled "Back" to move back one or more screens.

STATUS

This problem does not occur in Microsoft Access version 1.1.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new form using the Graph FormWizard.
2. Proceed to the last Graph FormWizard screen. (This screen gives you the option to open or design your form.)
3. Choose the button labeled "Back" to go back any number of screens.

The "Fast Forward" button is now unavailable. This button should be available to move you to the last screen of the Graph FormWizard.

[References](#)

INF: How to Retrieve Information from the Clipboard

Article Number: Q94162
CREATED: 29-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There is no inherent command within Microsoft Access to retrieve information from the clipboard. To retrieve information from the clipboard, you need to define an [Access Basic function](#) that calls several Windows API functions. This article defines a function that retrieves text from the clipboard.

MORE INFORMATION

Add the code defined below to an Access Basic [module](#). Make sure that the declare functions are all listed on one line and that the MsgBox procedure is also defined on one line. Since you may have some of the Windows API functions defined in an underlying library, some of them might not be needed. If this is the case, remove or comment the declarations from your code.

To call the function you can make a [string](#) assignment to the returned value, such as:

```
ReturnString$ = ClipBoard_GetData ()
```

Sample Code

NOTE: In the following sample code, an underscore `_` is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

Option Explicit

```
Declare Function OpenClipboard% Lib "User" (ByVal hwnd%)  
Declare Function GetClipboardData% Lib "User" (ByVal wFormat%)  
Declare Function GlobalAlloc% Lib "Kernel" (_  
    ByVal wFlags%, ByVal dwBytes&)  
Declare Function GlobalLock& Lib "Kernel" (ByVal hMem%)  
Declare Function lstrcpy& Lib "Kernel" (_  
    ByVal lpString1 As Any, ByVal lpString2 As Any)  
Declare Function GlobalUnlock% Lib "Kernel" (ByVal hMem%)  
Declare Function CloseClipboard% Lib "User" ()  
Declare Function GlobalSize& Lib "Kernel" (ByVal hMem%)
```

```
Global Const GHND = &H42  
Global Const CF_TEXT = 1  
Global Const MAXSIZE = 4096
```

```

Function ClipBoard_GetData()
    Dim hClipMemory%
    Dim lpClipMemory&
    Dim MyString$
    Dim Junk&
    Dim X%

    If OpenClipboard(0&) = 0 Then
        MsgBox "Could not open the clipboard. Another _
        application could have it open"
        Exit Function
    End If

    '-----
    ' Obtain the handle to the global memory
    ' block that is referencing the text.
    '-----
    hClipMemory% = GetClipboardData(CF_TEXT)
    If IsNull(hClipMemory%) Then
        MsgBox "Could not allocate memory"
        GoTo OutOfHere
    End If

    '-----
    ' Lock clipboard memory so we can reference
    ' the actual data string
    '-----
    lpClipMemory& = GlobalLock(hClipMemory)

    If Not IsNull(lpClipMemory&) Then
        MyString$ = Space$(MAXSIZE)
        Junk& = lstrcpy(MyString$, lpClipMemory)
        X% = GlobalUnlock(hClipMemory)

        'peel off the null terminating character
        MyString$ = Mid(MyString$, 1, InStr(MyString$, Chr$(0)) - 1)
    Else
        MsgBox "Could not lock memory to copy string from."
    End If

    OutOfHere:

    X% = CloseClipboard()
    ClipBoard_GetData = MyString$

End Function

```

References:

"Programming Windows by Charles Petzold", Second Edition, pg 780-781.

[References](#)

INF: Microsoft Access 1.0 Disk Contents (3.5 inch, 1.44 MB)

Article Number: Q94114
CREATED: 28-DEC-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SUMMARY

The following lists give the directory contents for each disk shipped with Microsoft Access version 1.0 (3.5 inch, 1.44 MB).

MORE INFORMATION

Disk 1

Volume Label: DISK1

1	WMF	4470	10-25-92	12:00a
1E	WMF	4022	10-25-92	12:00a
BTRVISAM	DL\$	57918	10-25-92	12:00a
COMMDLG	DL\$	46406	10-09-92	10:02a
COMMDLG	DL_	89248	10-25-92	12:00a
CTL3D	DL\$	7524	08-16-92	7:35p
DBSISAM	DL\$	135919	10-25-92	12:00a
DDEML	DL\$	20305	03-10-92	3:10a
DETCMD	DL_	24128	10-25-92	12:00a
MSACCESS	IN\$	519	10-25-92	12:00a
MSACCESS	RE\$	552	10-25-92	12:00a
MSAIN100	DL\$	71625	10-25-92	12:00a
MSAJU100	DL\$	38945	10-25-92	12:00a
OLECLI	DL\$	41083	10-09-92	10:02a
OLESVR	DL\$	13129	10-09-92	10:02a
ORDENTRY	TX\$	2281	10-25-92	12:00a
ORDENTRY	HL\$	14224	10-25-92	12:00a
ORDENTRY	MD\$	200396	10-25-92	12:00a
PIM	HL\$	8119	10-25-92	12:00a
SETUP	EXE	21472	10-25-92	12:00a
SETUP	INI	149	11-25-92	10:51a
SHARE	EX\$	8112	10-09-92	10:02a
SHELL	DL\$	23296	10-09-92	10:02a
SMALLB	FO\$	9687	03-06-92	7:16p
SMALLE	FO\$	11236	03-06-92	7:16p
SMALLF	FO\$	9376	03-06-92	7:16p
STFSETUP	EX_	477376	10-25-92	12:00a
STFSETUP	IN_	56427	10-25-92	12:00a
VER	DL\$	5653	10-09-92	10:02a
VER	DL_	9008	10-25-92	12:00a
WORKGRP	IN_	1393	10-25-92	12:00a

31 file(s) 1413998 bytes

Disk 2

Volume Label: DISK2

2	WMF	8022	10-25-92	12:00a
2E	WMF	6870	10-25-92	12:00a
MSACCESS	HL\$	1189376	10-25-92	12:00a
PDXISAM	DL\$	106830	10-25-92	12:00a
REGEDIT	EX\$	18176	10-09-92	10:02a
WBTRCALL	DL\$	30679	10-25-92	12:00a
		6 file(s)	1359953 bytes	

Disk 3

Volume Label: DISK3

3	WMF	8950	10-25-92	12:00a
3E	WMF	8470	10-25-92	12:00a
MSABC100	DL\$	187802	10-25-92	12:00a
MSAES100	DL\$	16899	10-25-92	12:00a
MSAFIN	DL\$	18553	10-25-92	12:00a
WIZARD	MD\$	174297	10-25-92	12:00a
MSACCESS	EX\$	790199	10-25-92	12:00a
		7 file(s)	1205170 bytes	

Disk 4

Volume Label: DISK4

4	WMF	8246	10-25-92	12:00a
4E	WMF	7766	10-25-92	12:00a
ANALYZER	MD\$	49740	10-25-92	12:00a
MSAJT100	DL\$	338672	10-25-92	12:00a
NEWCUST	DB\$	1212	10-25-92	12:00a
NWIND	MD\$	555226	10-25-92	12:00a
PIM	MD\$	109267	10-25-92	12:00a
SYSTEM	MD\$	98863	10-25-92	12:00a
UTILITY	MD\$	47055	10-25-92	12:00a
WINHELP	EX\$	147911	10-09-92	10:02a
WINHELP	HL\$	28161	10-09-92	10:02a
		11 file(s)	1392119 bytes	

Disk 5

Volume Label: DISK5

5	WMF	4662	10-25-92	12:00a
5E	WMF	4150	10-25-92	12:00a
CBTLIB4	DL\$	11179	10-25-92	12:00a
CUECARD	EX\$	60269	10-25-92	12:00a
CUECARDS	DL\$	2047	10-25-92	12:00a
CUECARDS	LE\$	887544	10-25-92	12:00a
GRAPH	EX\$	364877	10-09-92	10:02a

MSGGRAPH HL\$ 110291 10-09-92 10:02a
8 file(s) 1445019 bytes

Disk 6

Volume Label: DISK6

6	WMF	5846	10-25-92	12:00a
6E	WMF	5366	10-25-92	12:00a
MSACCESS2	HL\$	1133393	10-25-92	12:00a
PSSKB	TX\$	7951	10-25-92	12:00a
README	TX\$	9465	10-25-92	12:00a
ERRATA	TX\$	9805	10-25-92	12:00a

6 file(s) 1171826 bytes

ODBC Installation Diskette

Volume Label: ODBC

DBNMP3	<u>DLL</u>	8241	10-22-92	3:06p
INSTCAT	<u>SQL</u>	93796	10-25-92	12:00a
MSCOMSTF	DLL	74528	10-25-92	12:00a
MSDETSTF	DLL	24544	10-25-92	12:00a
MSINSSTF	DLL	65440	10-25-92	12:00a
MSSHLSTF	DLL	14928	10-25-92	12:00a
MSUILSTF	DLL	6144	10-25-92	12:00a
ODBC	DLL	44736	10-25-92	12:00a
ODBC	INF	1226	10-25-92	12:00a
ODBCADM	EXE	9136	10-25-92	12:00a
ODBCADM	HLP	4942	10-25-92	12:00a
ODBCINST	DLL	6016	10-25-92	12:00a
SETUP	EXE	54464	10-25-92	12:00a
SETUPI	DLL	44032	10-25-92	12:00a
SQLSETUP	DLL	11024	10-25-92	12:00a
SQLSRVR	DLL	135792	10-25-92	12:00a
VER	DLL	9008	10-22-92	3:06p
LANMAN21	<DIR>		10-25-92	12:00a

18 file(s) 607997 bytes

Windows For Workgroups Driver Diskette

Volume Label: WFWDRV

WFWDRV	EXE	23581	11-02-92	3:11a
WFWNET	DR_	165600	11-02-92	3:11a

2 file(s) 189181 bytes

[References](#)

INF: Microsoft Access 1.0 Disks 5-9 (5.25 inch, 1.2 MB)

Article Number: Q94113
CREATED: 28-DEC-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

Summary:

The following lists give the directory contents for disks five through nine including the [ODBC](#) Setup disk and the Windows for Workgroups Driver disk shipped with Microsoft Access version 1.0 (5.25 inch, 1.2 MB).

More Information:

Disk 5

Volume Label: DISK5

5	WMF	4662	10-25-92	12:00a
5E	WMF	4150	10-25-92	12:00a
CBTLIB4	DL\$	11179	10-25-92	12:00a
CUECARD	EX\$	60269	10-25-92	12:00a
CUECARDS	DL\$	2047	10-25-92	12:00a
CUECARDS	LE\$	887544	10-25-92	12:00a
6 file(s)		969851	bytes	

Disk 6

Volume Label: DISK6

6	WMF	3702	10-25-92	12:00a
6E	WMF	3222	10-25-92	12:00a
BTRVISAM	DL\$	57918	10-25-92	12:00a
DBSISAM	DL\$	135919	10-25-92	12:00a
GRAPH	EX\$	364877	10-09-92	10:02a
MSAJT100	DL\$	338672	10-25-92	12:00a
MSAJU100	DL\$	38945	10-25-92	12:00a
MSGRAPH	HL\$	110291	10-09-92	10:02a
PDXISAM	DL\$	106830	10-25-92	12:00a
REGEDIT	EX\$	18176	10-09-92	10:02a
WBTRCALL	DL\$	30679	10-25-92	12:00a
11 file(s)		1209231	bytes	

Disk 7

Volume Label: DISK7

7	WMF	5846	10-25-92	12:00a
7E	WMF	5366	10-25-92	12:00a
MSACCES2	HL\$	1133393	10-25-92	12:00a
PSSKB	TX\$	7951	10-25-92	12:00a

```
README    TX$      9465 10-25-92 12:00a
ERRATA    TX$      9805 10-25-92 12:00a
          6 file(s)    1171826 bytes
```

ODBC Setup Disk

Volume Label: ODBC

```
DBNMP3    DLL      8241 10-22-92  3:06p
INSTCAT   SQL     93796 10-25-92 12:00a
MSCOMSTF  DLL       74528 10-25-92 12:00a
MSDETSTF  DLL       24544 10-25-92 12:00a
MSINSSTF  DLL       65440 10-25-92 12:00a
MSSHLSTF  DLL       14928 10-25-92 12:00a
MSUILSTF  DLL        6144 10-25-92 12:00a
ODBC      DLL     44736 10-25-92 12:00a
ODBC      INF     1226 10-25-92 12:00a
ODBCADM   EXE     9136 10-25-92 12:00a
ODBCADM   HLP     4942 10-25-92 12:00a
ODBCINST  DLL     6016 10-25-92 12:00a
SETUP     EXE    54464 10-25-92 12:00a
SETUPI    DLL     44032 10-25-92 12:00a
SQLSETUP  DLL     11024 10-25-92 12:00a
SQLSRVR   DLL    135792 10-25-92 12:00a
VER       DLL     9008 10-22-92  3:06p
LANMAN21  <DIR>      10-25-92 12:00a
          18 file(s)    607997 bytes
```

Windows for Workgroups Driver Disk

Volume Label: WFWDRV

```
WFWDRV    EXE     23581 11-02-92  3:11a
WFWNET    DR_     165600 11-02-92  3:11a
          2 file(s)    189181 bytes
```

[References](#)

INF: Microsoft Access 1.0 Disks 1-4 Contents (5.25 Inch 1.2MB)

Article Number: Q94112
CREATED: 28-DEC-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

The following lists the directory contents of disks 1-4 shipped with Microsoft Access version 1.0 [5.25 inch, 1.2 megabytes (MB)].

Disk 1

Volume Label: DISK1

1	WMF	4470	10-25-92	12:00a
1E	WMF	4022	10-25-92	12:00a
COMMDLG	DL\$	46406	10-09-92	10:02a
COMMDLG	DL_	89248	10-25-92	12:00a
CTL3D	DL\$	7524	08-16-92	7:35p
DDEML	DL\$	20305	03-10-92	3:10a
DETCMD	DL_	24128	10-25-92	12:00a
MSACCESS	IN\$	519	10-25-92	12:00a
MSACCESS	RE\$	552	10-25-92	12:00a
MSAIN100	DL\$	71625	10-25-92	12:00a
OLECLI	DL\$	41083	10-09-92	10:02a
OLESVR	DL\$	13129	10-09-92	10:02a
ORDENTRY	TX\$	2281	10-25-92	12:00a
ORDENTRY	HL\$	14224	10-25-92	12:00a
ORDENTRY	MD\$	200396	10-25-92	12:00a
PIM	HL\$	8119	10-25-92	12:00a
SETUP	EXE	21472	10-25-92	12:00a
SETUP	INI	149	10-25-92	12:00a
SHARE	EX\$	8112	10-09-92	10:02a
SHELL	DL\$	23296	10-09-92	10:02a
SMALLB	FO\$	9687	03-06-92	7:16p
SMALLE	FO\$	11236	03-06-92	7:16p
SMALLF	FO\$	9376	03-06-92	7:16p
STFSETUP	EX_	477376	10-25-92	12:00a
STFSETUP	IN_	56600	10-25-92	12:00a
VER	DL_	9008	10-25-92	12:00a
VER	DL\$	5653	10-09-92	10:02a
WORKGRP	IN_	1393	10-25-92	12:00a
		28 file(s)	1181389 bytes	

Disk 2

Volume Label: DISK2

2	WMF	8022	10-25-92	12:00a
2E	WMF	6870	10-25-92	12:00a

MSACCESS HL\$ 1189376 10-25-92 12:00a
3 file(s) 1204268 bytes

Disk 3

Volume Label: DISK3

3	WMF	8950	10-25-92	12:00a
3E	WMF	8470	10-25-92	12:00a
MSABC100	DL\$	187802	10-25-92	12:00a
MSAES100	DL\$	16899	10-25-92	12:00a
MSAFIN	DL\$	18553	10-25-92	12:00a
WIZARD	MD\$	174297	10-25-92	12:00a
MSACCESS	EX\$	790199	10-25-92	12:00a

7 file(s) 1205170 bytes

Disk 4

Volume Label: DISK4

4	WMF	8246	10-25-92	12:00a
4E	WMF	7766	10-25-92	12:00a
ANALYZER	MD\$	49740	10-25-92	12:00a
NEWCUST	DB\$	1212	10-25-92	12:00a
NWIND	MD\$	555226	10-25-92	12:00a
PIM	MD\$	109267	10-25-92	12:00a
SYSTEM	MD\$	98863	10-25-92	12:00a
UTILITY	MD\$	47055	10-25-92	12:00a
WINHELP	EX\$	147911	10-09-92	10:02a
WINHELP	HL\$	28161	10-09-92	10:02a

10 file(s) 1053447 bytes

References:

[References](#)

PRB: Hewlett-Packard Dashboard GP Faults in Microsoft Access

Article Number: Q94111
CREATED: 28-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You get a General Protection (GP) Fault when using Microsoft Access with the Hewlett-Packard Dashboard 1.0.

RESOLUTION

Hewlett-Packard Company has released a patch that fixes this problem. Download the patch from the CompuServe HPSYS forum. The file you want is DB101U.EXE in Library 11.

MORE INFORMATION

To obtain the file, sign on to CompuServe and follow this procedure:

1. Type GO HPSYS at the ! prompt in CompuServe, and press ENTER. CompuServe will give you general information about the forum and ask you to [join](#).
2. Press ENTER until you reach the Forum! prompt.
3. Type LIB, and press ENTER. When CompuServe prompts you for a library number, type 11, and press ENTER.
4. Type DOWN DB101U.EXE, and press ENTER. CompuServe will ask you for a local file name. Enter DB101U.EXE along with the path indicating where you want to store the file, and press ENTER. For example, you might enter this:.

```
C:\WINDOWS\DB101U.EXE
```

At this point, CompuServe will begin to download the file.

5. When the download process is complete, you are ready to run the patch. When you run it, it will expand into more than one file, so create a subdirectory and copy DB101U.EXE into it. Then change to that subdirectory and enter DB101U at the MS-DOS [command prompt](#).

DB101U.EXE will automatically extract and expand the patch and a readme file. Hewlett-Packard Company recommends that you follow the instructions in that readme file.

Dashboard is a trademark of Hewlett-Packard Company. Dashboard is manufactured by a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

References

PRB: Full Path Causes Endless Loop in StartApp() Example

Article Number: Q94110
CREATED: 28-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

Passing the full path of a [DDE](#) server to the StartApp() [function](#) will cause an endless loop. The function will continue to run, starting up new copies of the specified DDE server, until the user stops the code or the workstation runs out of memory and the code fails.

The function works correctly if the name of the DDE server without the path is used.

MORE INFORMATION

The sample [database](#) NWIND.MDB contains a [module](#) called Introduction to Programming. Many of the [Access Basic](#) codes used as examples in the "Microsoft Access Basic: An Introduction to Programming" manual, version 1.0 are included in this module. The sample function StartApp() from page 142 contains a routine that can be used to initiate a DDE channel.

If the DDEInitiate() in the StartApp() function fails, the error handling routine StartApp: attempts to start the server. The error routine then passes [control](#) back to the main body of code, where the DDEInitiate is tried again.

If the full path to the DDE server is used when the StartApp() function is called, the DDEInitiate() function will fail, but the Shell() function in the error routine will be successful.

The following is an example of using StartApp() that will cause an endless loop:

```
=StartApp("c:\winword\winword.exe","system")
```

The following is an example that will work correctly:

```
=StartApp("winword","system")
```

[References](#)

PRB: Trailing CR/LFs Stripped From Text Box When Retrieved

Article Number: Q94109
CREATED: 28-DEC-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Trailing carriage returns and line feeds (CR/LFs) are deleted when you retrieve data from a [text box](#). Although you can insert CR/LFs into a text box, when you attempt to retrieve the data, they are deleted.

STATUS

This is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the Customers Form in the sample Northwind Traders [database](#) (NWIND.MDB).
2. Open the [Immediate window](#), and position it on the screen so that it and the Customers [form](#) are both in view.
3. Type the following command in the Immediate window:

```
Forms!Customers.[Contact Name] = "Chris" & chr(13) & chr(10)
```

At this point, the Contact Name [control](#) on the form should be updated and should contain the text 'Chris' with a blank line below it. This shows that the CR/LF is in the text box.

4. Type the following command in the Immediate window.

```
MyValue = Forms!Customers.[Contact Name]
```

The [variable](#) MyValue contains the text 'Chris', but there is no CR/LF at the end of the text. This can be tested by executing the following command from the Immediate window:

```
Print MyValue & "*" 
```

This returns the value 'Chris*' which shows that there is no CR/LF at the end of MyValue.

[References](#)

INF: MS Access Does Not Support Parameter Queries as DDE Topic

Article Number: Q94108
CREATED: 28-DEC-1992
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not support parameter queries through dynamic data exchange (DDE). To perform a parameter-driven [query](#), Microsoft Access must receive parameters in a manner that DDE does not support.

MORE INFORMATION

An attempt to initiate a DDE channel with a [parameter query](#) fails. The resulting error message depends on the DDE client application involved. When the DDE client is Microsoft Visual Basic, it displays the following message:

No foreign application responded to DDE Initiate

Microsoft Excel displays the following error message:

Remote Data not Accessible. Start "MSACCESS.EXE"?

One [method](#) to work around this situation links a Microsoft Access [form](#) to the DDE client. This [link](#) transfers the parameter to Microsoft Access. You then can modify the query to retrieve the parameter from the form.

To implement this method in Microsoft Excel, perform the following five steps and their respective subprocedures:

1. Create the DDE Link

- Start Microsoft Excel and open a new Macro sheet.
- In Cell A1, type "DAIR" (without the quotation marks).
- Save the [macro](#) sheet as DDETEST.XLM.
- Select Cell A1 and choose Copy from the Edit menu.
- Start Microsoft Access and open the sample [database](#) NWIND.MDB.
- Create a new, unbound form.
- From the Edit menu, choose Paste Special.
- In the Data Type [list box](#), select Text, then choose Paste Link.

A Text box that contains the following formula is created:

```
=DDE("Excel","C:\ACCESS\DDETEST.XLM","R1C1")
```

- i. If the Properties box is not available, choose Properties from the View menu.
- j. Select the text box created in Step g. Change the Control Name property to "DDE Parameter" (without the quotation marks).
- k. Save the form as [DDE Link Form].

2. Modify the Query

- a. Open the [Products By Category (Parameter)] query in Design view.
- b. From the Query menu, choose Parameters. Delete the [Enter a Category ID] criteria from the query.
- c. In the [Category ID] field in the query grid, replace the existing query with the following:

```
=Forms![DDE Link Form]![DDE Parameter]
```

- d. Test the query. Verify that the DDETEST.XLM macro sheet is open in Microsoft Excel and that the [DDE Link Form] in Microsoft Access is open in Form view. Run the query and verify that the products displayed are in the "DAIR" category.

3. Create a Macro to Open the Form

- a. Create a new macro in Microsoft Access.
- b. Add the following action to the macro:

```
OpenForm
  Form Name: DDE Link Form
  View: Form
```

- c. Save the macro as Open Form.

4. Create a Macro to Run the Query

- a. Create a new macro in Microsoft Access.
- b. Add the following action to the macro:

```
OpenQuery
  Query Name: Products by Categories (Parameter)
  View: Datasheet
```

- c. Save this macro as Run Query.

5. Create the DDE Macro

- a. In the DDETEXT.XLM macro sheet in Microsoft Excel, enter the following macro:

Cell	Command
A1	DAIR
A2	
A3	chan=INITIATE("MSAccess","System")
A4	=EXECUTE(chan,"Open Form")
A5	=APP.ACTIVATE("Microsoft Access")
A6	=SEND.KEYS("{F9}")
A7	=TERMINATE(chan)
A8	=ON.TIME(0.010,Run_Query)
A9	
A10	chan=INITIATE("MSAccess","System")
A11	=EXECUTE(chan,"Run Query")
A12	=TERMINATE(chan)
A13	=RETURN()

- b. Select Cell A10. Choose Define Name from the Formula menu. In the Define Name dialog box, type "Run_Query" in the Name field. Choose Command in the Macro group box, then choose OK. This step assigns the name Run_Query to the macro that begins in Cell A10.
- c. In Microsoft Access, close all forms, macros, and queries.
- d. To run the macro, make DDETEXT.XLM the active macro sheet and select Cell A3. Then choose Run from the Macro menu and choose OK.

NOTE: The Microsoft Excel macro runs much more quickly if you set the DDE Timeout option in Microsoft Access to 1 second.

Questions & Answers

Q. Why does the Microsoft Excel macro use SendKeys?

A. Microsoft Excel instructs Microsoft Access to open the form [DDE Link Form]. When it opens this form, Microsoft Access attempts to update the DDE link in the [DDE Parameter] control. However, Microsoft Excel does not process any requests until it receives notification that Microsoft Access has successfully completed the macro.

In turn, Microsoft Access does not complete executing the macro until it successfully updates the link to Microsoft Excel. Eventually, Microsoft Access times out and displays the #Error message. The SendKeys command sends the F9 key sequence to Microsoft Access to refresh the form.

Q. How can I send this information to Microsoft Excel?

A. To transfer the results of a query to Microsoft Excel, change the

query to a [make-table query](#). Then use the DDERequest() [function](#) to retrieve the data in the table from Microsoft Excel.

REFERENCES

=====

For information about the DDE server topics that Microsoft Access supports, please refer to the README.TXT file for Microsoft Access version 1.0. This file is located in your program directory.

For more information on parameter queries, please refer to the "Microsoft Access User's Guide," version 1.0, Chapter 7, "Designing Action Queries and Parameter Queries," pages 178-181.

For more information on creating DDE links in forms and reports, please refer to the "Microsoft Access User's Guide," version 1.0, Chapter 13, "Using Pictures, Graphs, and Other Objects," pages 337-339.

[References](#)

INF: Importing Fixed-Width Text Files

Article Number: Q94107
CREATED: 28-DEC-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

This article describes the [fixed-width text file](#) format and explains how to import this type of file.

MORE INFORMATION

A fixed-width text file in Microsoft Access is a file that contains records of a fixed, consistent width. The data fields in each [record](#) begin at fixed offsets from the beginning of the record.

You cannot import files that have [variable](#) length records that do not contain [field](#) separators (for example, commas or tabs). These files are not fixed-width text files. To import this kind of file, you need to first process the file using an editor to insert field separators or to turn the file into a true fixed-width text file by, for example, adding spaces to each record to make all records the same width (number of characters).

Microsoft Access imports fixed-width text files by casting a record template into the file at calculable offsets. It uses the offsets to determine the logical beginning of each new record. Microsoft Access determines the logical record width by scanning the first data record of the file to locate the first carriage return and line feed (CR/LF). It assumes that the first CR/LF marks the end of the first record (see point three later in this article).

The data fields defined in the template come from the information you supplied when you set up the import/export specification. Microsoft Access uses the template to extract the individual field values. Then it enters the field values into [table](#) and advances the template to the next record.

To import fixed-width text files successfully, you must meet the following conditions:

1. The file must be truly fixed-width. That is all records have the same width with all fields in the records beginning at consistent offsets.
2. The import/export specification you set up must match the physical file. Microsoft Access uses the specification to create the template of fields after inspecting the physical file to determine the overall record size. If the field template differs from the actual record size, you may get bad results.

Ensure that the greatest (Offset + Width - 1) in your import/export

specification is less than or equal to the physical record width excluding the CR/LF at the end of the record. In other words, an import/export specification that defines two fields -- ID beginning at offset 1 for a width and of 8 and DATA beginning at offset 11 for a width of 50 -- then your physical file must contain records of no less than 60 bytes (11 + 50 - 1) not counting the CR/LF. Microsoft Access will ignore records that have extra data beyond the last byte defined in the import/export specification.

3. The first record in the file must not contain embedded CR/LF characters. Otherwise, the Microsoft Access record size calculation will be wrong.

References:

For more information, search for "Fixed-width text files" using the Microsoft Access Help menu. Also, see the "Microsoft Access User's Guide," version 1.0, Chapter 4, "Importing, Exporting, and Attaching," pages 72-85.

[References](#)

PRB: Access Adds Date Delimiters When Importing Text

Article Number: Q94106
CREATED: 28-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When importing a text file that contains date values that do not contain a date delimiter such as a '/' into Microsoft Access version 1.0, the '/' date delimiter will be added to the date field in the Table. An example date would be 112192. This date value would be loaded into Microsoft Access as 11/21/92.

CAUSE

Microsoft Access will automatically separate the month, day, and year with a '/' date delimiter when importing a text file, regardless of whether the text file is formatted with a date delimiter. This only occurs when you import the file to an existing table only. If the import created the table then the 112192 date value would be loaded as 111292 with a Number datatype.

STATUS

his behavior is by design.

[References](#)

PRB: Append Query Results in Duplicate Counter Numbers

Article Number: Q94105
CREATED: 28-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A table that you append data to using an Append Query now has duplicated numbers in a field defined as a Counter data type. The expected result was appended data with incremented numbers in the Counter field.

CAUSE

Your Append Query is explicitly referencing the field with the Counter data type. You are, in essence, overriding the automatic counter function that you specified in your table design. This may result in duplicate numbers.

RESOLUTION

Do not include a field with a Counter data type in the Append To row of the QBE grid in your Append Query. By omitting this field in your Append Query, Microsoft Access will automatically generate incremental numbers in the appended table's Counter field.

STATUS

This behavior is by design.

MORE INFORMATION

You will only get duplicates if there is no Primary Key or Index (No Duplicates) defined on the Counter field in the table that you are appending to. By definition, primary keys are unique.

Including fields with Counter data types in your Append Query might also result in non-contiguous numbers in the table that you are appending to. For example, if the table you are appending to has values of 1 to 20 in the Counter field, and the table you are appending from has values of 20, 34, and 55, you will end up with one duplicate value. The next time you add a new row to the appended to table, the Counter field will be assigned a value of 56.

[References](#)

PRB: Err Msg: 'Couldn't Update: Currently Locked by User...'

Article Number: Q94080
CREATED: 23-DEC-1992
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The following message appears either when two users simultaneously try to update a similar object in a shared database or on a single-user system if the system runs out of locks:

```
    Couldn't update; currently locked by user '<user name>' on machine  
    '<machine name>'
```

CAUSE

On a multiuser system, a conflict exists with the SYSTEM.MDA file. By design, Microsoft Access cannot simultaneously update the MsysObjects table in the SYSTEM.MDA file for multiple users.

On a single-user system, there are not enough locks specified in the AUTOEXEC.BAT file.

RESOLUTION

On a multiuser system, try saving the object again. If no one is updating a similar object at the same time, the Save command should work. This error can occur in Access Basic code also, so you will need to handle error trapping for this.

On a Single-User System

1. Exit Windows.
2. Change to the Access directory and delete all *.LDB files.
3. Change to the Windows directory and the Temp subdirectory. Delete all ~*.TMP files.
4. Edit the AUTOEXEC.BAT file. Look for the line, C:\DOS\SHARE.EXE. Change this line to C:\DOS\SHARE.EXE /L:500. You will not find this line if you are running VSHARE.EXE, OS/2, or you are currently running on a network.
5. Restart the computer.

MORE INFORMATION

The SYSTEM.MDA file conflict is likely to occur only on systems with

many (100+) users.

Steps to Reproduce Behavior

1. User A and user B open the same database on a shared drive.
2. Both users create a new object of similar type (such as a table, form, report, query, or macro.)
3. Both users choose Save from the File menu and enter a unique filename.
4. Simultaneously, user A and user B press ENTER to update the object.
5. Microsoft Access presents the error message listed above. Choose OK to dismiss the message and try again.

[References](#)

INF: Northwind 'Sales Totals by Amount' Report Is Incorrect

Article Number: Q94043
CREATED: 22-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The [report](#) named "Sales Totals by Amount" included in the sample [database](#) Northwind Traders (NWIND.MDB) does not work correctly. The report uses the Conditional Page Break [macro](#) to print 10 lines on each page of the report. This works properly for only the first page because the counter on the report is not being reset. To enable this to work correctly, you must change the Conditional Page Break macro. In the macro named Show Break within the macro Conditional Page Break, you must change the Condition to read as follows:

```
[COUNTER] MOD 10 = 0
```

This will now perform a [page break](#) after every 10 lines of the report for every page.

[References](#)

INF: Function Deletes Table Without Using DoMenuItem

Article Number: Q94042
CREATED: 22-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can create and use the DeleteTable() [function](#) to delete a [table](#) without using the DoMenuItem() or SendKeys() functions.

MORE INFORMATION

Microsoft Access stores all references to objects (tables, queries, forms, reports, macros, and modules) in a system table called MSysObjects. You can delete an object from the MSysObjects table if you have permission to do so.

Use the following steps to assign yourself permission to delete objects from the MSysObjects table:

1. From the Security menu, choose Permissions.
2. Choose modify rights for the table by selecting or clearing the appropriate check boxes.

The DeleteTable() function accepts the name of a table as its parameter. It finds the [record](#) in MSysObjects that corresponds to that table name and deletes that record. The function also deletes records from other system tables, including MSysACEs, MSysIndexes, and MSysColumns, that relate to the deleted table. The function returns a value of True if it successfully deletes the table you specified; it returns a value of False if it cannot find a table with the name you specified.

NOTE: Modifying data in any system table can have serious ramifications. Do not attempt to do this unless you are thoroughly familiar with the [database](#) and know exactly where, when, and how the table you want to delete is used elsewhere in the database. It is best to use the DeleteTable() function on temporary tables only.

The [Access Basic](#) code for the DeleteTable() function is as follows:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Option Explicit
Function DeleteTable (TableName as String)

    Dim SQL As String, Msg As String
```

```

If DCount("[Name]", "MSysObjects", "[Name] = '" &
    & TableName & "'") = 0 Then
    DeleteTable = FALSE
    Exit Function
End If

On Error GoTo DeleteErrorProc

' Suppress the "Do you want to delete these records?" message.

    DoCmd SetWarnings False

' SQL statement to delete appropriate entries from the MSysColumns
' table follows:

    SQL = "DELETE DISTINCTROW MSysObjects.Name, MSysColumns.*"
    SQL = SQL & " FROM MsysObjects, MSysColumns,"
    SQL = SQL & " MSysColumns INNER JOIN MSysObjects ON"
    SQL = SQL & " MSysColumns.ObjectID = MSysObjects.Id"
    SQL = SQL & " WHERE ((MSysObjects.Name = '" & TableName & "'));"

    DoCmd RunSQL SQL

' SQL Statement to delete appropriate entries from MSysACEs table
' follows:

    SQL = "DELETE DISTINCTROW MSysObjects.Name, MSysACEs.*"
    SQL = SQL & " FROM MSysObjects, MSysACEs,"
    SQL = SQL & " MSysObjects INNER JOIN MSysACEs ON"
    SQL = SQL & " MSysObjects.Id = MSysACEs.ObjectID"
    SQL = SQL & " WHERE ((MSysObjects.Name= '" & TableName & "'));"

    DoCmd RunSQL SQL

' SQL statement to delete references to indexes in MSysIndexes follows:

    SQL = "DELETE DISTINCTROW MSysObjects.Name, MSysIndexes.*"
    SQL = SQL & " FROM MSysObjects, MSysIndexes,"
    SQL = SQL & " MSysObjects INNER JOIN MSysIndexes ON"
    SQL = SQL & " MSysObjects.Id = MSysIndexes.ObjectID"
    SQL = SQL & " WHERE ((MSysObjects.Name= '" & TableName & "'));"

    DoCmd RunSQL SQL

' SQL statement to delete references to table object from MSysObjects
' follows:

    SQL = "DELETE DISTINCTROW MSysObjects.Name "
    SQL = SQL & " FROM MSysObjects"
    SQL = SQL & " WHERE ((MSysObjects.Name = '" & TableName & "'));"

    DoCmd RunSQL SQL

' If you decide to update the Database window, include the following
' command:

```

```

SendKeys "{F11} %VQ%VT"

' Test to make sure the object was deleted, then exit.

If DCount("[Name]", "MSysObjects", "[Name] = " _
    & TableName & "") = 0 Then
    DeleteTable = True
    Exit Function
EndIf

' If error occurs, display the error message and terminate with error.
DeleteErrorProc:
    Msg = "An unexpected error has occurred." & Chr(13) & Chr(10)
    Msg = Msg & "Error: " & Error$
    MsgBox Msg
    DeleteTable = False

End Function

```

References:

"Microsoft Access User's Guide," version 1.0, chapter 25

[References](#)

PRB: Grid Lines Not Visible In Table Datasheet View

Article Number: Q94041
CREATED: 22-DEC-1992
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Grid lines between records and fields are not visible when you open tables.

CAUSE

The color palette has been changed and now the colors for the background and the grid lines are the same or very close in color.

RESOLUTION

Change the Windows Background setting to the default color palette or to a new color in the color section of the Microsoft Windows Control Panel and the grid lines will reappear when you open a [table](#) in Access.

[References](#)

INF: Using Sendkeys to Change an OLE Link

Article Number: Q94040
CREATED: 22-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

To change the source of a linked [OLE](#) (Object Linking and Embedding) object, you must choose Object, Change Link from the Edit menu.

The text below describes the three steps required to create a [macro](#) that automatically changes the [link](#) based on a filename stored in a [table](#).

MORE INFORMATION

Step 1: Create the Sample Table

Perform the following three steps.

1. Create a table with the following fields:

```
Table: Table1
=====
  FieldName: Key
      DataType: Counter
  FieldName: File Name
      DataType: Text
  FieldName: Linked Object
      DataType: OLE Object
```

2. Save the table as Linked References.

3. In Browse mode, add the following records to the table. The bitmaps used in this example are stored in the Microsoft Windows directory (by default, C:\WINDOWS). To embed the objects, open each [bitmap](#) in Microsoft Windows Paintbrush, choose Copy from the Edit menu, switch to Microsoft Access, place the cursor in the Linked Object [field](#), and choose Paste Link from the Edit menu.

Key	Filename	Linked Object
---	-----	-----
(counter)	ARCADE.BMP	Paste Link ARCADE.BMP
(counter)	ARCHES.BMP	Paste Link ARCHES.BMP
(counter)	ARGYLE.BMP	Paste Link ARGYLE.BMP

Step 2: Create the Form

Perform the following three steps.

1. Create a new form based on the table (Linked References).
2. Drag all three fields from the field list to the form.
3. Save the form as "OLE Test" (without the quotation marks).

Step 3: Create the Macro

Perform the following four steps.

1. Create a new macro and add the following actions:

Action	Arguments
-----	-----
GoToControl	Linked Object
SendKeys	%eol
SendKeys	=Forms![OLE Test]![File Name]
SendKeys	~

NOTE: There are three SendKeys actions to make troubleshooting the macro simple.

2. Save the Macro as "Update OLE Link" (without the quotation marks).
3. Open the form [OLE Test] and select the control [File Name].
4. Set the property After Update to the macro [Update OLE Link].

To test the form, switch to Browse mode. Change the field [File Name] in the first record to LEAVES.BMP. The OLE object should update to display the bitmap LEAVES.BMP, which is also in your Windows directory.

References:

For more information on embedding OLE objects, see Chapter 13, "Using Pictures, Graphs, and Other Objects," of the "Microsoft Access User's Guide" for version 1.0.

[References](#)

PRACC9301: Memo Field with Format Property Truncates Data

Article Number: Q94039
CREATED: 22-DEC-1992
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you apply formatting to a Memo field by using the Format field property, Microsoft Access truncates the data at 255 characters. The field retains its Memo data type, but data after the first 255 characters disappears. This happens to existing data in the Memo field as well as any new data you enter.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

[References](#)

INF: Attach R:BASE Table by Exporting dBASE Format in R:BASE

Article Number: Q94037
CREATED: 22-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not ship with an ISAM driver that supports attaching R:BASE tables to Microsoft Access databases. However, R:BASE can export tables into several different formats that Microsoft Access can import.

To retain column names and data type definitions, export R:BASE tables in R:BASE using the dBASE format.

MORE INFORMATION

R:BASE can export the following data types:

- ASCII delimited
- ASCII fixed width
- DIF (VisiCalc or other program)
- SYLK (Multiplan or other program)
- dBASE II, dBASE III, or dBASE III Plus (DBF)
- Lotus 1-2-3 (version 1.0 and 2.0) or Symphony (WKS, WK1, or WRK)

The products included here are manufactured by vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

PRB: Cannot Perform OpenTable on an Attached Table

Article Number: Q94036
CREATED: 22-DEC-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you try to use an OpenTable [method](#) on an [attached table](#), you will receive the following error message:

Can't perform operation; it is illegal.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Attach any valid [table](#). From the File menu, choose Attach.
2. Create the following [module](#):

```
'*****  
'Declarations section of the module  
'*****
```

Option Explicit

```
'=====
```

```
'The following will set the database to the current database and  
'attempt to open the attached SQL table. Name equals the attached  
'tables name.  
'=====
```

```
Sub Test  
    Dim db as Database  
    Dim t as Table  
    Set db = CurrentDB()  
    Set t = db.OpenTable("dbo_name")  
End Sub
```

3. Compile the module, then run the module. You will receive the following message:

Can't perform operation; it is illegal.

[References](#)

PRB: Can't Check to See If Database Was Opened Exclusively

Article Number: Q94035
CREATED: 22-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In a multiuser environment, you cannot use the OpenDatabase command to find out if the current database was opened exclusively. If you attempt to open the current database with the OpenDatabase command in Access Basic and the current database was originally opened exclusively, the database will open and no error will be returned.

CAUSE

Microsoft Access ignores the option flags for all opens except the first for a given window's task, so a system administrator cannot use this to test if a database has been opened exclusively. The technique doesn't work because Microsoft Access does not check the option flags.

STATUS

This behavior is by design.

MORE INFORMATION

The following Access Basic subroutine successfully opens the current database, even if the current database has been opened exclusively.

Steps to Reproduce Behavior

1. Open a database called DB1.MDB.
2. Enter the Access Basic sub shown below in a new module:

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****  
'Declarations section of the module  
'*****
```

Option Compare Database

```
'=====
```

```
' The following sub is named Test().
```

```
'=====
```

```
Sub Test ()
  On Error GoTo ErrorHandler

  Dim db As Database

  Set db = OpenDatabase("db1.mdb", False)
  db.Close
  Exit Sub

ErrorHandler:
  If Err = errCantOpenDbShared Then
    MsgBox "You opened the database exclusively. Please_
      re-open it non-exclusively!"
    DoCmd SelectObject A_TABLE, "", True
    DoCmd Close
  Else
    Resume Next
  End If
End Sub
```

3. Run the subroutine from the Immediate Window by typing Test.

As a result, you will see that you do not receive an error message.

[References](#)

PRB: Query by Form Returns No Rows When All Rows Expected

Article Number: Q94027
CREATED: 22-DEC-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

While specifying criteria for a query in a previously designed form, you leave one of the fields empty because you want all the records, not just the ones that match a certain value. But instead of getting all the records, as you expected, you get none at all.

CAUSE

Queries that contain empty criteria fields may return unexpected results because an empty criteria field results in the following condition in the WHERE clause:

Like Null

This condition is always false because any operation that includes a null value returns a null result.

Queries can reference form fields through implicit parameters -- making a query by form. For example, an application developer might put the following criteria on the CustomerName field:

Like Forms!CriteriaDialog!CustomerName

Here CriteriaDialog is a form that holds the criteria fields. Someone using the application might omit a customer name entry in the CriteriaDialog form in the hopes of seeing all the rows. However, the query actually returns no rows because of the null reference in the resulting WHERE clause.

RESOLUTION

Application developers can work around this potential problem by adding the following function to a module:

```
Function CNulls (v As Variant, subs As Variant) As Variant
    If (IsNull(v)) Then
        CNulls = subs
    Else
        CNulls = v
    End If
End Function
```

The CNulls function converts null values to a given value. Essentially, if the first argument to the function is null, the second

argument is returned. Otherwise, the first argument is returned unchanged.

After you add the function to a module, change the criteria to read like this:

```
Like CNulls( Forms!CriteriaDialog!CustomerName, "*" )
```

Now, if the user does not supply a customer name, the CNulls function will return the asterisk, and the condition will return all rows that contain data.

[References](#)

PRACC9212: Datasheet Form Saved as Subreport Causes GP Fault

Article Number: Q94026
CREATED: 22-DEC-1992
MODIFIED: 28-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

A report containing a subreport created from a form causes a general protection (GP) fault in Microsoft Windows if the Default View property of the original form is set to Datasheet.

The following error message appears immediately after you choose Print Preview or Print from the File menu:

MSACCESS caused a General Protection Fault in module MSACCESS.EXE
at 006A:0A26

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

Steps to Reproduce Problem

1. Open the Categories subform in Design view.
2. From the File menu, choose Save As Report.
3. Create a new report based on the Categories table.

NOTE: The GP Fault only occurs if the main report is bound.

4. Drag the Categories subform report that you created in step 2 to the main report that you created in step 3.

Result: The error message in "Symptoms" above appears.

Workaround

You can use a form as a subreport by embedding it directly in the main report. You do not need to choose the Save As Report menu option. A form with its Default View property set to Datasheet will not cause a GP Fault when it is embedded in a report.

1. Create a new report.

2. Drag the Categories subform from the [Database window](#) to the main report.

-OR-

1. Open the original form in Design view.
2. Change the Default View property to either Single Form or Continuous Forms.
3. From the File menu, choose Save As Report.
4. Embed the new form in your report.

NOTE: The Main Form/Subform FormWizard creates the subform with the Default View property already set to [Datasheet view](#).

[References](#)

INF: Installing Microsoft Access on a Diskless Workstation

Article Number: Q94010
CREATED: 22-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The third paragraph on page 658 of the "Microsoft Access User's guide," version 1.0, states that a workstation that runs Microsoft Access from the [network](#) requires 1 megabyte. This means that a workstation that has a hard disk requires 1 megabyte of hard disk space, and a diskless workstation requires 1 megabyte of space in the user's directory on the network -- not on the local workstation.

MORE INFORMATION

You can install and run Microsoft Access from a diskless workstation as long as your workstation meets the basic requirements for running Microsoft Access.

References:

"Microsoft Access User's Guide," version 1.0, Appendix D, "Setting up Microsoft Access on a Network," page 658.

[References](#)

INF: How to Use DDE to Fill a List Box

Article Number: Q94003
CREATED: 22-DEC-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article contains an example of an [Access Basic](#) list [function](#) that uses dynamic data exchange ([DDE](#)) to populate a [list box](#).

The example assumes that you are familiar with DDE, how to create modules in Access Basic, and how to create list boxes on forms.

MORE INFORMATION

The function below will use DDE to retrieve a list of countries from Microsoft Excel. The list will be used to populate a list box bound to the Country [field](#) of the Suppliers [table](#) in the sample [database](#) NWIND.MDB.

1. In a new worksheet in Excel, enter the following in the first [column](#):

```
A1  Australia
A2  China
A3  Scotland
```

2. Save the worksheet as COUNTRY.XLS.
3. Open NWIND.MDB in Microsoft Access and open a new [module](#).
4. Place the following in the [Declarations section](#):

```
Option Explicit
Dim Countries(10) As String
Dim CountryData As String
```

5. Create the following subroutine:

```
Sub DDEFillCountries ()
    Dim EndString%, i%
    Dim Chan

    EndString% = 0
    i% = 0
    Chan = DDEInitiate("Excel", "COUNTRY.XLS")
    CountryData$ = DDERequest(Chan, "R1C1:R10C1")
    DDETerminate Chan
    EndString% = InStr(1, CountryData, Chr(10))

    Do Until Len(CountryData$) = 0
```

```

        Countries(i%) = Left(CountryData, EndString% - 1)
        EndString% = InStr(1, CountryData, Chr(10))
        CountryData$ = Right(CountryData, _
            Len(CountryData) - EndString% - 1)
        i% = i% + 1
    Loop

End Sub

```

6. Create the following function:

```

Function DDEFillList (fld As Control, id, row, col, code)
    Static MonthNum

    Select Case code
        Case 0
            Call DDEFillCountries
            DDEFillList = True
        Case 1
            DDEFillList = id
        Case 3
            DDEFillList = 12
        Case 4
            DDEFillList = 1
        Case 5
            DDEFillList = -1
        Case 6
            DDEFillList = Countries(row)
    End Select
End Function

```

7. Save the module as DDE Fill List Box.

8. Create a new form based on the Suppliers table.

9. Add a list box bound to the Country field and set the following properties:

```
RowSourceType: DDEFillList
```

10. Verify that Excel is running and the worksheet COUNTRY.XLS is open. Switch to browse mode. The list box should now be populated with the countries that you entered into Excel.

NOTE: This article demonstrates a simple example and is not a complete and robust function. If you intend to use this with a form, there are several things that you may want to add to your code:

- Error checking for the DDE portion to handle cases where Excel is not running or is busy.
- The example assumes that there are ten countries listed in Excel.
- This list function does not return the number of rows in the data to Microsoft Access.
- The form may need to be refreshed to show updated data.

References:

Microsoft Access "User's Guide," version 1.0, chapter 9

"Microsoft Access Basic: An Introduction to Programming," version 1.0,
page 82

[References](#)

INF: How to Open UTILITY.MDA

Article Number: Q93929
CREATED: 21-DEC-1992
MODIFIED: 14-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

If you try to open the UTILITY.MDA file, you receive the following error message:

The database "UTILITY.MDA" is already open as a library database.

To open the UTILITY.MDA file, you must create a copy of UTILITY.MDA, rename the copy, and then open it in Microsoft Access.

More Information:

Use the following steps:

1. Copy UTILITY.MDA to UTILITY.SAV.
2. In Microsoft Access, open UTILITY.SAV by choosing Open from the File menu.
3. Choose OK when the following error messages are displayed:

Tried to load module with duplicate procedure definition:
BuilderZoom

Tried to load module with duplicate procedure definition: DDB

Now that you have edited the copy of UTILITY.MDA, you can rename the original UTILITY.MDA file and substitute the edited copy of the UTILITY.MDA file for the one that Microsoft Access uses when the application is opened.

UTILITY.MDA is a collection of tables, forms, and modules used by Microsoft Access. You should always have a backup copy of UTILITY.MDA; this is a vital database to Microsoft Access and any changes in it may result in unexpected behavior.

[References](#)

PRB: Exporting Table with OLE Object or Memo Field to Paradox

Article Number: Q93928
CREATED: 21-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you export a table from Microsoft Access that contains a field with an OLE Object or Memo data type in Paradox 3.x format, the resulting Paradox table contains one entry that says m_i_c_r_o_s_o_f_t. You can see this result by importing the Paradox table you just exported and then opening the table.

MORE INFORMATION

Steps to Reproduce Behavior

-
1. In Microsoft Access, copy any existing table to a new name.
 2. In Design view, add an OLE Object field or Memo field to the copied table.
 3. After saving and closing the table, export it choosing Paradox 3.x as the data destination and TESTFLD.DB as the name. You will get this error: "Invalid Data Type." However, the TESTFLD.DB file is created.
 4. Import TESTFLD.DB using Paradox 3.x as the Data Source.
 5. Open the resulting table. You will see a table consisting of one field with the field name m_i_c_r_o_s_o_f_t and the data type Number, but there is no data in the table.

Paradox is manufactured by Borland International, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

References

INF: Repeat Group Name at the Top of New Column or Page

Article Number: Q93927
CREATED: 21-DEC-1992
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to create a [report](#) that prints "continued" or "Group Name continued" at the top of each new [column](#) or page when grouped information flows from one column or page to another. At the end of this article, you'll find sources for additional information on multiple-page and multiple-column report formats.

MORE INFORMATION

Microsoft Access prints a single [group header](#) at the beginning of a group to identify the beginning of the group. It does not repeat the [header](#) at the top of subsequent columns or pages. However, you can use the procedure described below to force a report to repeat the group name if the detail records in a group span several columns or pages. This article uses the report, List of Products by Category from the sample [database](#) NWIND.MDB.

Three-step Process

The report uses macros to identify the group. It uses the Top property to determine whether a detail section is being formatted at the top of a column or page. If it is, the report shows the contents of the text box that contains the group name and "continued," and it hides the other controls in the detail section. It is a three-step process.

Step One: Create a New Macro Group Containing Three Macros

You need a [macro group](#) (Continued) containing three macros (Page Reset, Set Top1, and Print) to check for certain conditions and perform specific actions based on those conditions. By setting conditions, you give the macro the branching characteristics of a program without using a programming language.

The Page Reset macro sets the value of the Top1 [text box](#) to the value of the Top property if the current value of Top1 is 0 when Microsoft Access formats the group header.

The Set Top1 macro sets the value of the Top1 text box to 0 when Microsoft Access formats the [page header](#).

The Print macro sets the value of the Top1 text box to the value of the Top property if the value of Top1 is 0 when Microsoft Access formats a detail section. Or if the Top property is equal to the value of the Top1 text box, it sets the Visible property of the Continued

text box to Yes and the Visible property for each of the controls in the detail section to No. In the List of Products by Category report, there are four controls to hide in the detail section. It also sets the NextRecord property to False. Otherwise, it hides the message and prints the detail record.

To create the macro group:

1. Create a new macro.
2. From the View menu, choose Macro Names or choose the Macro Names button on the tool bar.
3. From the View menu, choose Conditions or choose the Conditions button on the tool bar.

Microsoft Access displays the complete grid in the Macro window including the Macro Name, Condition, and Action columns.

4. Enter the macro names, conditions, actions, and action arguments using the following two tables as a guide. Use the first table to fill the grid and use the second to add action arguments to each action. Do not enter the reference letters: [a], [b], [c], and so on. These reference letters are there to help you associate the actions shown in the first table with the action arguments shown in the second table.

Macro Name	Condition	Action	
Page Reset		SetValue	[a]
Set Top1	[Top1]=0	SetValue	[b]
Print	[Top1]=0	SetValue	[c]
	Top=[Top1]	SetValue	[d]
	...	SetValue	[e] -- repeat for
	...	SetValue	[e] -- each <u>control</u>
	...	SetValue	[e] -- in the
	...	SetValue	[e] -- detail section
	...	SetValue	[f]
	...	StopMacro	[g]
		SetValue	[h]
		SetValue	[i] -- repeat for
		SetValue	[i] -- each control
		SetValue	[i] -- in the
		SetValue	[i] -- detail section

The ellipses in the Condition column are required for lines e, f, and g; enter them exactly as shown. Microsoft Access evaluates macro conditions as true or false. If the expression is true, Microsoft Access performs the action. If it is false, Microsoft Access ignores the action. The ellipses in the Condition column force Microsoft Access to perform the action on that line if the condition on the preceding line is true.

Here is the second table. It gives the action arguments for each action:

Action	Action Arguments
[a] SetValue	Item: [Top1] Expression: 0

```

[b] SetValue Item: [Top1]
      Expression: Top
[c] SetValue Item: [Top1]
      Expression: Top
[d] SetValue Item: [Continued].Visible
      Expression: Yes
[e] SetValue Item: [Product Name].Visible
      Expression: No
[e] SetValue Item: [Product Id].Visible
      Expression: No
[e] SetValue Item: [Quantity Per Unit].Visible
      Expression: No
[e] SetValue Item: [Units In Stock].Visible
      Expression: No
[f] SetValue Item: NextRecord
      Expression: False
[g] StopMacro
[h] SetValue Item: [Continued].Visible
      Expression: No
[i] SetValue Item: [Product Name].Visible
      Expression: Yes
[i] SetValue Item: [Product Id].Visible
      Expression: Yes
[i] SetValue Item: [Quantity Per Unit].Visible
      Expression: Yes
[i] SetValue Item: [Units In Stock].Visible
      Expression: Yes

```

- The SetValue actions in lines e and i are used to hide the controls in the detail section of your report. The List of Products by Category report has 4 controls in the detail section: Product Name, Product Id, Quantity Per Unit, and Units In Stock. You should repeat lines e and i for the number of controls in your detail section. For example, if you have only one control in the detail section, you need only one e and one i.
- Save the macro group and specify Continued as the name. This makes Continued the name of the macro group. It now appears in the list of macros and macro groups in the [Database window](#). Use the following [syntax](#) to specify each macro in the macro group when you attach a macro to a property on the report:

```
macrogroupname.macroname
```

For example, the following specifies the Print macro, continued.print

Step Two: Attach the Macros to the Report

- Open the desired report in [Design view](#).
- Set the following properties for the specified sections by selecting each section with the mouse and viewing its properties:

Section Name	Property	Expression
Page Header	OnFormat	Continued.Page Reset

Category Name Header	OnFormat	Continued.Set Top1
Detail	OnFormat	Continued.Print
	CanShrink	Yes
Category Name Footer	OnFormat	Continued.Set Top1

This sets the Top1 control to the current value of the Top expression to keep track of the current [twip](#) value (position) of the top.

Step Three: Add Controls to the Report and Set Properties

1. Open the report in Design view.
2. Create an unbound text box control named Top1.
3. Place the Top1 text box control in the Category Name header section.
4. Assign the following property values to the Top1 text box control:

Property	Expression
ControlName:	Top1
Visible:	No

5. In the detail section, create a text box bound to an expression. Locate it near the top in the same vertical location as that of the group header [label](#). Assign the following property values to the text box:

Property	Expression
ControlName:	Continued
ControlSource:	=[Category Name] & " continued"
Visible:	No
CanShrink:	Yes

NOTE: Replace Category Name with the name of the [field](#) being grouped.

6. Save and run the report.

The approach described here works well for both single-column and multiple-column reports, but there is a simpler [method](#) you can use for a single-column, multiple-page report.

References:

For more information on the simpler method, [query](#) on Continued and Groups here in the Microsoft Knowledge Base.

For more information on how to make the page header span the full width of a multiple-column report, [query](#) on Span and Width here in the Microsoft Knowledge Base.

For more information, search for "multiple-column reports," then "Creating Snaking Columns on a Report" using the Microsoft Access Help menu.

Also, see the "Microsoft Access User's Guide," version 1.0, Chapter 19,

"Sorting and Grouping Data."

[References](#)

INF: Copying and Pasting Text in a Property Sheet

Article Number: Q93926
CREATED: 21-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You cannot use the Edit menu's Copy or Paste command to edit text in a property sheet.

Instead, use the equivalent keystroke combinations for the Copy and Paste commands. CTRL+C copies information to the clipboard, while CTRL+V pastes information from the clipboard.

[References](#)

PRB: Microsoft Access Can't Recognize SQL Server Double Index

Article Number: Q93925
CREATED: 21-DEC-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

Microsoft Access does not recognize a [SQL Server table index](#) if the index has a `DOUBLE` [data type](#).

STATUS

This problem does not occur in Microsoft Access version 1.1.

MORE INFORMATION

Microsoft Access requires a unique index on any SQL Server table for updates (editing), and it cannot use the data type `DOUBLE` as a unique index. After reading the `DOUBLE` index value on the server, Microsoft Access converts the value from binary to decimal and back to binary, causing rounding errors, before storing it in a Microsoft Access virtual table (VT). The same conversion process occurs when Microsoft Access sends the value back to the server for an [update](#). Microsoft Access uses the index value to find the [row](#) in the table where the update is needed. Because of the rounding errors, Microsoft Access cannot find the correct row to make the update, so the update fails. Because updates cannot be made on this type of index, even though it may be unique, Microsoft Access ignores the index.

[References](#)

PRACC9301: HPPCL5A Printer in DEBUG Windows Causes GP Fault

Article Number: Q93728
CREATED: 20-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Printing on a HPPCL5A printer while using the DEBUG version of Windows 3.1 causes a general protection (GP) fault and generates the following error message:

```
err ACCESS->HPPCL5A GLOBALSIZE+C: Invalid global handle: 0x000
```

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Use DEBUG Windows.
2. Open a database.
3. Create a new report.
4. Place two labels that overlap on the report.
5. Set the printer to the HPPLC5A printer. Then choose Print from the File menu.

[References](#)

PRB: Cannot Import YYYYMMDD Dates in Fixed-Width Text Files

Article Number: Q93727
CREATED: 20-DEC-1992
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Microsoft Access will not import fixed-width text files if there is a date field in the text file formatted as YYYYMMDD.

CAUSE

Microsoft Access does not recognize the YYYYMMDD date format in text files, and will generate an error log when trying to import a text file that includes a date in this format. The error log will show a "Type Conversion Failure." The file will not import.

RESOLUTION

One workaround is to format the date as MMDDYY.

Another workaround is to import the YYYYMMDD field as a six-character text field. Next, create a new field in the table called, for example, NewDate, with a data type of Date/Time. Then run an update query to derive the NewDate field from the OldDate field as follows:

Note: In the following sample code, an underscore(_) is used as a line-continuation character. Remove the underscore when re-creating this code in Microsoft Access.

Query: ReformatDate

Field Name: NewDate

Update to: Mid([OldDate],3,2) & "/" & Right([OldDate],2) & _
 "/" & Left([OldDate],2)

STATUS

This behavior is by design.

MORE INFORMATION

When importing a fixed-width text file with a date field formatted as YYYYMMDD, Microsoft Access generates an Import Error log table, but does not import the data. If you try to import the text file as delimited the data is imported, but Microsoft Access changes the format from date to number or text, depending upon the date delimiter.

[References](#)

PRB: Overlapped Printing an Access Query Using TTY.DRV

Article Number: Q93726
CREATED: 20-DEC-1992
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you print a [query](#) using the Windows Generic Text Only printer driver (TTY.DRV), the print overlaps on every sixth or seventh line.

RESOLUTION

The TTY.DRV driver can print only six lines per inch. You must set the [row](#) height to 12 point to make the line spacing work correctly with this printer driver. To change the row height, choose Row Height from the Layout menu and type "12" (without the quotation marks). This allows the TTY.DRV to print your query without overlapping text.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open a query.
2. From the File menu, choose Print Setup.
3. Select the Specific Printer [option button](#), and select Generic Text Only from the list of printers. (This driver must be installed on your machine for this to option to appear.) Choose the OK button.
4. From the Query menu, choose Run.
5. From the File menu, choose Print, then choose the OK button.

Approximately every sixth line of print is overlapped.

[References](#)

PRACC9301: Non-TrueType Fonts with 8514.DRV Causes GP Fault

Article Number: Q93725
CREATED: 20-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Using fonts other than TrueType fonts with the Windows 8514.DRV video driver may cause a general protection (GP) fault or other problems.

RESOLUTION

Use TrueType fonts with the Windows 8514 driver.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

Scenario 1

1. Design a new form. Make it about eight inches wide.
2. Drop in a label.
3. Set the font to MS Sans Serif 8pt if it isn't already.
4. Type a long text line that spans about five inches.
5. Size the box to fit the entire line of text on one line.
6. Type 127 for the point size.
7. Choose Size to fit from the Layout menu.
8. Note the interesting overlapping text at the beginning.
9. Set bold on.
10. Set italics on -- this takes a while.

Notice the interesting racing lines. This only happens when you use fonts other than TrueType fonts. For example, if you use Arial, everything looks fine.

Scenario 2

1. Design a new form and make it 22 inches wide.
2. Drop in a label.
3. Set the font to Arial 8pt Bold if it isn't already.
4. Type a long text line that spans about 18 inches.
5. Size the box to fit the entire text line on one line.
6. From the toolbar, select MS Sans-Serif for the font.

What happens varies, but in the end you always receive a message saying Microsoft Access caused a GP Fault in 8514.DRV.

References

PRB: 'Generic/Text Only' Driver Print Quality List in Error

Article Number: Q93724
CREATED: 20-DEC-1992
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you select "Generic/Text Only" as your printer, the Print Quality list in the Print [dialog box](#) contains only four options: High, Medium, Low, and Draft.

CAUSE

This behavior occurs because each printer driver, regardless of its capabilities, has all four options built in. In some cases, as in the "Generic/Text Only" driver, all four options perform the same [function](#). In other drivers, the Print Quality list shows only valid options.

STATUS

This behavior is by design.

[References](#)

PRB: Hourglass May Not Change After Illegal Function Call

Article Number: Q93723
CREATED: 20-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A user can change the mouse pointer from an arrow to an hourglass cursor with a user-defined [function](#) in [Access Basic](#). If the user encounters an illegal function, or halts the code after setting the pointer to an hourglass, the hourglass cursor does not change back to the arrow pointer. However, you can still open and close items by clicking on them.

RESOLUTION

You can type "DoCmd Hourglass False" in the [Immediate window](#) to get the pointer back while in design of the [module](#).

STATUS

This behavior is by design. Access Basic will never reset Echo, Warnings, or Hourglass when run from Access Basic. These are only reset when the last script runs.

MORE INFORMATION

Because an illegal function call stopped the execution of the function before the command to reset the mouse pointer from an hourglass to an arrow, the mouse pointer remains in the hourglass state.

References:

"Microsoft Access Basic: An Introduction to Programming," chapters 1-5

"Microsoft Access Language Reference," version 1.0, Part 1

[References](#)

PRB: Microsoft Access Cannot Paste Entire Column in Query Grid

Article Number: Q93722
CREATED: 20-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you use the Query Grid, an attempt to cut and paste (or to copy and paste) an entire column places incorrect results into the target column.

RESOLUTION

Before you paste information from the clipboard, select an object that has the same size as the information in the clipboard. In this example, select an entire column when you paste the information into the query grid.

STATUS

This behavior is by design.

MORE INFORMATION

Perform the following four steps to demonstrate this behavior.

Steps to Reproduce Behavior

-
1. Create a query.
 2. Select an entire column in the Query Grid (click the mouse once at the top of a column when the mouse cursor changes to a down arrow).
 3. From the Edit menu, choose either Cut or Copy.
 4. Select a cell (to paste the copied column into) and choose Paste from the Edit menu.

The pasted values are not correct. The initial value displayed will probably be a "1", which indicates that the Show box was checked in the column you copied. You can scroll through values in the cell by using the up and down arrow keys. If you attempt to move off the cell by clicking elsewhere on the grid, you will receive a syntax error message.

This behavior occurs pasting into any object in Microsoft Access.

References:

For more information on pasting information in the query grid, search the

Knowledge Base here for
queries and paste

[References](#)

PRB: Column in QBE Grid Keeps Focus After Clicking Field List

Article Number: Q93721
CREATED: 20-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

After you select a column in the QBE grid, the column doesn't lose focus when you click a field list in the upper-portion of the Query window.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new query and add a table to it.
2. Drag a few columns to the query grid.
3. Select a column in the query grid by clicking the column selector above the field box.
4. Click the field list for the table.

The query grid column does not lose the focus.

Pressing the DEL key at this point removes the table and all the information in the query grid.

[References](#)

PRB: DDE Statements Invalid in List Box or Combo Box Control

Article Number: Q93720
CREATED: 20-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you place a dynamic data exchange (DDE) function as the ControlSource of a list box or combo box, you will receive a #Error in the list box when viewed in Browse mode. However, if you view the form in Datasheet mode, the first object in the DDE data will show up.

Placing a DDE function in the RowSource property causes an error message in both Browse and Datasheet modes of the form.

RESOLUTION

Using Access Basic, you can create a list function that returns to a list box or combo box. You may use any Access Basic routine to obtain the information used in the list, including DDE commands. The list function is documented on pages 83 through 85 of the "Microsoft Access Introduction to Programming" manual. Coding dynamic data exchange is documented in chapter 9 of the same manual.

STATUS

This behavior is by design.

MORE INFORMATION

The ControlSource property of a list box or combo box may be blank if the control is unbound, or may contain the name of the field that the control is bound to; it may not contain a function.

The RowSource property of a list box or combo box may contain a list of values, the name of a table or query, or an SQL string. This property provides the list of values the user may select from.

Steps to Reproduce Behavior

1. Open an Excel spreadsheet and enter some text in Row 1, Column 1. Save the spreadsheet as TEST.XLS.
2. Create a new form in Microsoft Access and place a list box on the form.

Object: Listbox

ControlSource: =DDE("Excel","Test.xls","R1C1")

3. Make sure TEST.XLS in Excel is open.
4. View the form in Browse mode. Notice that there is a #Error in the list box.
5. View the form in Datasheet mode. Notice the data from Excel is now there.

[References](#)

PRB: Changes to Shared Module Are Kept Local Until MDB Closed

Article Number: Q93719
CREATED: 20-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Two or more workstations open a shared Database (MDB) that has at least one module. Changes are made to the module by workstation A. The module is then saved by workstation A. Workstation B opens the same module and does not see the changes made on workstation A.

STATUS

This behavior is by design.

MORE INFORMATION

The only way to update modules is to close and reopen the database. Microsoft Access loads all modules once when the database is opened. Changes made by one workstation will not affect any other workstation(s) until the database is closed and then reopened.

[References](#)

PRB: Unable to Change Permissions (Security)

Article Number: Q93718
CREATED: 20-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you try to open an object or change permissions for an account, you receive one of the following error messages:

You can't view this object's permissions.

-or-

No permission for <object name>.

CAUSE

Only the database creator, or a member of the Admins group, has full permissions to all database objects. This person is also the only user who can assign or unassign security permissions.

RESOLUTION

Contact your database administrator. If you are the database administrator, check to make sure you are logged in as a user who is a member of the Admins group. If you still have the original user Admin, make sure it is a member of the Admins group and try again.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 25,
"Administering a Database System," pages 609-629

For more information search for "Error Messages: Reference" using the Microsoft Access Help menu.

[References](#)

PRB: Microsoft Access Appears to Hang During Database Compact

Article Number: Q93717
CREATED: 20-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When Microsoft Access performs a database compact operation, the status bar indicates the percentage of the task that is complete. At times, the status bar may remain on one value for a very long time and it may appear that Microsoft Access has hung. However, in most cases when this occurs, Microsoft Access continues to work on the compact operation.

CAUSE

When Microsoft Access performs a database compact operation, it copies each object in the database into a new database. Because different objects have different sizes, some objects require more time to copy into the new database. The value displayed in the status bar indicates the percentage of the number of objects copied, not the percentage of the total size of all objects copied.

RESOLUTION

This behavior is by design.

MORE INFORMATION

For example, if your database contains a very large table, the displayed percentage could go up to 90% rather quickly and then remain static while Microsoft Access copies the table. While Microsoft Access copies the table, it may appear to be hung. The 90% value indicates that 90% of the objects have been copied; it does not indicate that 90% of the processing is complete.

References:

"Microsoft Access User's Guide", version 1.0, page 627.

[References](#)

INF: Access Cannot Attach or Import Paradox 4.0 Tables

Article Number: Q93699
CREATED: 17-DEC-1992
MODIFIED: 26-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Version 1.x of Microsoft Access supports attaching to and importing from Paradox 3.x tables. Microsoft Access does not support attaching to Paradox 4.0 tables. If you attempt to attach to a version 4.0 [table](#), you will be prompted for a password.

MORE INFORMATION

If you are prompted to enter a password when you attach to a Paradox table, and you know that there is no password for the table, it is possible the table is a Paradox 4.0 file.

NOTE: These directions apply to Paradox 4.0 only; these directions do not apply to Paradox for Windows.

To use data created in Paradox 4.0, use Paradox to restructure the table in version 3.5 format. Paradox provides the capability to create files in Compatible format; this will create files in version 3.5 format.

In Paradox, you can save a Paradox 4.0 table in version 3.5 format by following these steps:

1. Copy the Paradox 4.0 table.
2. In Paradox, run the Modify/Restructure menu command.
3. Enter the name of the table.
4. Choose FileFormat/Compatible from the menu.
5. Press F2 to save the file in the restructured format.

For more information on restructuring Paradox tables, see the Paradox 4.0 documentation.

Paradox is manufactured by Borland International, a vendor independent of Microsoft. Microsoft makes no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

INF: How To Get Microsoft Works Files into Access

Article Number: Q93698
CREATED: 17-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not directly import the Microsoft Works for Windows or Microsoft Works for MS-DOS [database](#) (.WDB) file format. Microsoft Access does, however, import any version of Microsoft Works spreadsheet (.WKS) files. This article explains how you can transfer your database and spreadsheet files from Microsoft Works into Microsoft Access tables.

MORE INFORMATION

You can import spreadsheet files (.WKS) directly into Microsoft Access. To import database files you should save them as dBase III or IV files, or convert them to Works spreadsheet files first.

Microsoft Works for Windows Database files

Database files created in Microsoft Works can be imported by Microsoft Access if they are converted to a dBase III or IV file format. The conversion retains [field](#) names and most data types, with these exceptions:

- Currency fields are converted to Number fields.
- Fields defined as formulas are converted to Number fields with values.

Conversion process:

1. Start Microsoft Works and open your Works database file.
2. From the File menu choose Save As.
3. Using the "Save File as Type" [combo box](#), change the format to dBase III or dBase IV.
4. Change the name of the file so it has a .DBF extension and choose OK.
5. Start Microsoft Access and open the .MDB file you want to import to.
6. From the File menu choose Import.

7. Select dBase III or dBase IV from the format list.
8. Select the name of your .DBF file from the list. If it does not appear in the list you may need to change the drive and directory to the correct location.
9. Select the Import button. The file is imported as a Microsoft Access table with the same name as the .DBF file.

Microsoft Works for MS-DOS Database files

Save your Microsoft Works 2.0 database files in the format "Text & Commas". Microsoft Access imports this file type as Text (delimited) files.

Save your Microsoft Works 3.0 database files in dBase III or dBase IV format following the method outlined above.

Microsoft Works spreadsheet files (either version)

Spreadsheet files, created in either the MS-DOS or Windows version of Works, can be imported by Microsoft Access with little effort. If the first row of the spreadsheet contains a name for each column this can be recognized as the name of each field in the new Microsoft Access table. If there is only data in the spreadsheet Microsoft Access creates default field names for the table. Most data types are retained, with these exceptions:

- Fields formatted as TRUE or FALSE are converted to Number fields. A FALSE value is converted to 0 and a TRUE value is converted to 1.
- Formula fields are converted to Number fields with values.

Conversion process:

1. Start Microsoft Access and open the .MDB file you want to import the Microsoft Works spreadsheet file to.
2. From the File menu, choose Import.
3. Select Lotus (WKS) from the format list.
4. Select the name of the .WKS file from the list. If it does not appear, you may need to change the drive and directory to the correct location.
5. If the first row of the spreadsheet contains the column names, select the option "First Row Contains Field Names".
6. Select the Import button. The file will be imported as a Microsoft Access table with the same name as the .WKS file.

Depending upon the type of data in your Microsoft Works files, you may choose either method to convert files to Microsoft Access tables.

For more information about sharing Microsoft Works data with other programs, see the "Microsoft Works User's Guide".

References:

"Microsoft Access User's Guide," version 1.0, pages 64-65, pages 72-75

[References](#)

INF: How to Dial a Phone Number in Microsoft Access

Article Number: Q93696
CREATED: 17-DEC-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There are no built-in commands or functions in Microsoft Access that allow you to dial a phone number through a serial port. However, it can be accomplished programmatically. This article discusses a [method](#) that uses the Microsoft Windows application program interface (API) [function](#) calls to dial a phone number in Microsoft Access. Using this information, you can dial a number by creating a new [form](#) and [module](#), or apply the code to an existing form and module.

This article assumes that you are familiar with [Access Basic](#) and Microsoft Windows APIs. In this article the use of error trapping and the use of the communication APIs has been minimized to keep the information as clear and concise as possible.

MORE INFORMATION

To dial a phone number in Microsoft Access, you utilize the Microsoft Windows APIs in the Dynamic Link Library (DLL) USER.EXE file. There are seventeen communications APIs that are available in the USER library; of these, three particular communication APIs are necessary to dial phone numbers: OpenComm(), CloseComm(), and WriteComm().

Listed below are instructions for creating a simple form and module that dial a phone number through a modem based on the contents of two controls on a form. A more elegant solution can be created by passing arguments to the function and using CASE statements to deal with the potential errors that can be returned by the APIs.

To run this example:

1. Create a form called fDialer.
2. Place two unbound Text Box controls on the form. One Text box will be for identifying the serial port the modem is attached to, and the other text Box will hold the phone number to dial.

```
Object: Text Box
-----
Control Name: CommPort

Object: Text Box
-----
Control Name: PhoneNumber
```

- Place a command button on the form and assign the OnPush property to a function you will create later named DialNumber().

```
Object: Command Button
-----
OnPush: =DialNumber()
```

- Create a new module, enter the code below, and save it as Comm DLL.

NOTE: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****
'Declarations section of the module
'*****

Option Compare Database
Option Explicit

Declare Function OpenComm% Lib "User" (ByVal lpComName$, _
    ByVal wInQueue%, ByVal wOutQueue%)

Declare Function CloseComm% Lib "User" (ByVal nCid%)

Declare Function WriteComm% Lib "User" (ByVal nCid%, ByVal lpBuf$, _
    ByVal nSize%)

Const ID_CANCEL = 2

'=====
' The following function is named DialNumber().
' The variables in this function are local to the function since
' they have been DIMed in the function.
'
' The following function will initialize the variables:
'
' - sCommPort$ to the value entered in the control on the form
' - sPhoneNumber$ to the value entered in the control on the form
' - sDial$ to the dial command and the phone number to dial
' - iWritePort% to the return value of the API WriteComm function
' - iOpenPort% to the return value of the API OpenComm function
' - iClosePort% to the return value of the API CloseComm function
' - iCharCount% to the length of sDial$ variable
' - iStart% to the number of seconds since midnight
'
' - Comm port is entered on the form fDialer!CommPort
'=====

Function DialNumber ()
    Dim sCommPort$
    Dim sPhoneNumber$
    Dim sDial$
    Dim iWritePort%
    Dim iOpenPort%
    Dim iClosePort%
    Dim iCharCount%
    Dim iStart#
    Dim Response%
```

```

sCommPort$ = Forms!fDialer!CommPort
sPhoneNumber$ = Forms!fDialer!PhoneNumber
sDial$ = "ATDT" & sPhoneNumber$ & Chr$(13)+ Chr$(10)
iCharCount% = Len(sDial)

iOpenPort% = OpenComm(sCommPort$, 1024, 128)

If (iOpenPort% < 0) Then
    MsgBox "Unable to Dial Number: Make sure no other devices_
        are using the Communications Port.", 16, "Error Dial_
        Failed"
End If

'=====
' - iWritePort% sends the phone number to the modem
'
' If this is successful, the start time is returned and a loop
' for 3 seconds is started to allow time for the modem to dial
' before displaying a message box.
' The user can cancel the operation and the modem hangs up.
' If the modem cannot dial, there may be a problem writing to the
' port.
' The comm port is closed at the end of the module to make it
' available to other applications.
'=====

iWritePort% = WriteComm(iOpenPort%, sDial, iCharCount%)

If (iWritePort% > 0) Then
    iStart# = Timer

    Do
        Loop Until Timer >= iStart# + 3

        Response% = MsgBox("Pick up the Phone and Press OK or Cancel_
            to Quit and Hang up.", 65, "Dialing_
            Successful")

        If Response% = ID_CANCEL Then
            'Reset the modem and take it off line.
            iWritePort% = WriteComm(iOpenPort%, Chr$(13), 1)
        End If

    Else 'Otherwise Writing to Port must have failed.

        MsgBox "Unable to Dial Number: Make sure no other devices_
            are using the Communications Port.", 16, "Error Dial_
            Failed"

    End If

    iClosePort% = CloseComm(iOpenPort%)

End Function

```

10. Save the Module.

11. Open the form `fDialer` and enter the serial port (example: `COM1` or `COM2`) in the `CommPort` control that your modem is attached to. Enter a phone number in the `PhoneNumber` control. Click on the `Dial` button. The modem should dial the phone number specified.

[References](#)

INF: Passing Values from a Form to a Parameter Query

Article Number: Q93695
CREATED: 17-DEC-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

When you run a [parameter query](#), Microsoft Access prompts you to enter each parameter in a separate [dialog box](#). This article explains how to create a [form](#) that will collect all the values for a parameter [query](#).

More Information:

The following step-by-step example uses a [table](#) (Orders) from the Northwind Traders sample [database](#) (NWIND.MDB) supplied with Microsoft Access. The example shows you two things: how to create a form that prompts for a date value and how to add a [command button](#) to the form to execute the query.

1. Open the sample NWIND.MDB database.
2. Create a query based on the Orders table. Add the following [field](#) names with the specified [criteria](#) and save the query as MyQuery.

Field	Criteria
Customer ID	
Order ID	
Order Date	> [Forms]![MyForm]![MyDate]

3. Create a new [macro](#) with the following [action argument](#), and save it as MyMacro.

Action	Action Argument	Value
OpenQuery	Query Name	MyQuery

4. Create an unbound, blank form by leaving the Select A Table/Query box empty and clicking the Blank Form button.
5. Place a [text box control](#) on the form and set the following properties:

```
Text Box
-----
ControlName: MyDate
Format: Short Date
```

6. Bring the [Database window](#) to the front by pressing F11.
7. Select the Macro button with the mouse, and drag MyMacro onto the

form. This will generate a command button with the OnPush property set to call MyMacro.

8. Click in the Form window to bring it to the front, and then save the form as MyForm.
9. Switch to Form view, enter a date such as 1/1/91, and then click the MyMacro button. This will execute the macro MyMacro that runs the query MyQuery using the date entered in the MyDate text box as the query parameter.

[References](#)

PRB: Outdated COMMDLG.DLL Error Message During Setup Program

Article Number: Q93694
CREATED: 17-DEC-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you try to open a [database](#), you receive the following error message:

Outdated "COMMDLG.DLL." Please reinstall Microsoft Access.

CAUSE

There are three possible causes of this problem:

1. More than one version of COMMDLG.DLL on your computer.
2. An outdated copy of the COMMDLG.DLL file.
3. An incompatible copy of COMMDLG.DLL.

RESOLUTION

The three solutions below correspond with the three causes listed above:

1. Rename the duplicate versions of COMMDLG.DLL.
2. Make sure the COMMDLG.DLL file in the WINDOWS system subdirectory is newer than the following:
 - 3/10/92 (for Microsoft Windows version 3.1)
 - 10/01/92 (for Windows for Workgroups)
 - 10/25/92 (for Microsoft Access)
3. If you have a third-party COMMDLG.DLL file, make sure that it is compatible with Microsoft Access.

The three sections under "More Information" describe these procedures.

MORE INFORMATION

Searching for Copies of COMMDLG.DLL File

To search for all occurrences of the COMMDLG.DLL file, follow these steps:

1. Exit Windows.

2. Search for any other copies of COMMDLG.DLL on the current drive by typing the following commands at the MS-DOS prompt:

```
cd\  
dir commdl原因.dll /s
```

Repeat these commands on all other drives and rename or delete any copies of the COMMDLG.DLL file that you find. When you finish, you should have a single copy of COMMDLG.DLL on your computer, located in the WINDOWS system subdirectory.

Installing a New COMMDLG.DLL File

If you delete or rename all duplicates of the COMMDLG.DLL file and continue to get the above error message in Microsoft Access, copy COMMDLG.DL_ from Disk 1 of Microsoft Access and EXPAND.EXE from Disk 3 of Windows version 3.1. Then use EXPAND.EXE to expand the COMMDLG.DL_ file as follows:

```
c:\<path>\EXPAND <path>\COMMDLG.DL_ c:\windows\system\COMMDLG.DLL
```

NOTE: Do not copy COMMDLG.DL\$ from the Microsoft Access disks. Microsoft Access Setup is required to decompress COMMDLG.DL\$.

Ensuring No Other Applications Are Open

When Microsoft Access Setup tries to update the COMMDLG.DLL file, you may receive an error message stating that Microsoft Access was unable to access the file. In most cases, another application is using the COMMDLG.DLL file.

To ensure that no other applications are open when you run Microsoft Access Setup, use the following steps:

1. Add a new program group in Windows Program Manager called Temporary.
2. Move all the icons from the Startup program group to the Temporary program group.
3. Edit the [Windows] section of the WIN.INI file as follows:

```
[Windows]  
RUN=  
LOAD=
```

4. Restart Windows and run Microsoft Access Setup.

If you continue to encounter problems, edit the AUTOEXEC.BAT file to look for conflicting terminate-and-stay-resident (TSR) programs.

[References](#)

PRACC9211: Unique Values Only Select Query Shows Duplicates

Article Number: Q93692
CREATED: 17-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

The result of a [query](#) includes duplicate values even though you selected the Unique Values Only [check box](#) (SELECT DISTINCT) in the Query Properties [dialog box](#).

CAUSE

Duplicate values occur when both of the following conditions are met:

1. The query input data set contains more than approximately 64K of data and the duplicate values are widely scattered in the input data.

For example, a SELECT DISTINCT query on a text [column](#) with an average length of 20 characters will not display duplicate values unless the query input data set included more than 3,300 values ($65536 / 20 = 3276.8$). Note that if the query is based on a [table](#) that has 10,000 rows, but the specified [criteria](#) retrieves only 20 rows, it is the set of 20 rows that determines the size of the query input data set. In other words, it is the [WHERE clause](#) that determines the size of the query input data set.

2. The values that have duplicates appear close (maybe in the first dozen values) to what would normally be the beginning of the query output. By default, a SELECT DISTINCT query displays data sorted implicitly. For example, the following displays the result sorted by Order ID:

```
SELECT DISTINCT [Order Details].[Order ID]
FROM [Order Details];
```

However, in a SELECT DISTINCT query that uses more than one column and sorts by a column other than the first one, the duplicates could appear anywhere in the output. To see if there are duplicates, you need to remove the sort.

NOTE: This problem also appears in a Crosstab query as duplicated columns with names like Field1. Use Fixed Column heading to prevent the problem.

RESOLUTION

In addition to SELECT DISTINCT queries, you can use GROUP BYs in a [totals query](#) to eliminate duplicates in the output. Both SELECT DISTINCT queries and GROUP BYs in a totals query are limited to

10 columns. Usually, GROUP BY is faster when the resulting data set is much smaller than the input data set, and SELECT DISTINCT is faster when there are only a few duplicates.

To ensure the correct results, use GROUP BYs in a totals query instead of using the Unique Values Only query property. For example, the following two select queries give identical results:

```
SELECT DISTINCT [Order Details].[Order ID]
FROM [Order Details];
```

```
SELECT DISTINCTROW [Order Details].[Order ID]
FROM [Order Details]
GROUP BY [Order Details].[Order ID];
```

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 6, "Designing Select Queries," pages 128-130 and pages 142-148.

For more information, search for "DISTINCT," then "ALL, DISTINCT, DISTINCTROW Predicates ([SQL](#))" using the Help menu.

[References](#)

INF: Additions to User's Guide Appendix A Specifications

Article Number: Q93691
CREATED: 17-DEC-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

This article includes additions to the "Microsoft Access User's Guide," version 1.0, Appendix A, "Microsoft Access Specifications," pages 633-635.

MORE INFORMATION

Tables

Attribute	Maximum

Number of Primary Key fields and Matching Fields when defining a Relationship between tables	5

Queries

Attribute	Maximum

Number of characters in a cell	255
Number of GROUP BY fields in a Totals query	10
Number of fields allowed when query property UniqueValuesOnly is selected (SELECT DISTINCT)	10
Number of ANDs in a WHERE clause (Criteria)	40
Number of ANDs in a HAVING clause	40

[References](#)

INF: How to Update a Table Based on Values in a Second Table

Article Number: Q93690
CREATED: 17-DEC-1992
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

To transfer data between two tables, create an additional field in one of the tables to store the common data, then create an update query to initiate the transfer.

MORE INFORMATION

This process can be illustrated using the sample database NWIND.MDB. The following example adds a Product Name field to the Order Details table, then copies the Product Name field from the Products table based on the Product ID field.

NOTE: Normally you would not duplicate the Product Name field, since duplicate data is not good database design.

1. Open the Order Details table in Design view.
2. Add a Product Name field with the default data type of Text.
3. Close the Order Details table and save your changes.
4. Create a new query based on the Order Details and Products tables.

NOTE: These tables are automatically joined on the Product ID field because a relationship has already been defined. If your tables are not joined, join them on the original linked field.

5. From the Query menu, choose Update.
6. From the View menu, choose Table Names. Drag the Product Name field from the Order Details table to the first cell in the Field row of the query grid.
7. In the Update To row, type:

```
[Products].[Product Name]
```

8. From the Query menu, choose Run. Close the query.

For example, transferring data between tables is helpful if you have a table containing both products and prices. If database users receive an ASCII file with Product IDs and new prices for some products, they can import the ASCII file and create an update query as described above to update the Price field in the existing Products table.

For more information about using an update query with two tables,
query on the following words here in the Microsoft Knowledge Base:

update and counter and query

References:

"Microsoft Access User's Guide," version 1.0, Chapter 7

[References](#)

INF: Export Btrieve Table Without Existing Xtrieve FILE.DDF

Article Number: Q93688
CREATED: 17-DEC-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

To export a Btrieve table by using the Microsoft Access user interface, you must ensure that an Xtrieve dictionary file (FILE.DDF) exists. If the FILE.DDF file does not exist, use the macro action Transfer Database to export the table.

MORE INFORMATION

Multiple Btrieve tables can exist in Xtrieve databases, just as Microsoft Access databases can have multiple tables. The FILE.DDF file describes the structure of the Xtrieve database.

Example

The following sample macro (TransBtrieve) transfers the NWIND Customers table to a Btrieve Table named Customers in an Xtrieve database named C:\ACCESS\BTV00000.DAT. If C:\ACCESS is a valid directory, the export occurs even if BTV00000.DAT does not exist.

Use the following as a guide to create the sample macro. Note that the database name can be any complete file specification.

Macro Name	Action
TransBtrieve	TransferDatabase

TransferDatabase Action

Transfer Type: Export
Database Type: Btrieve
Database Name: C:\ACCESS\BTV00000.DAT
Object Type: Table
Source: Customers
Destination: Customers
Structure Only: No

Microsoft Access creates the following four files if they do not exist:

FILE.DDF
INDEX.DDF
FIELD.DDF
BTV00000.DAT

Transferring more than one table adds data to the existing files, but does

not create more files -- as long as the database name is not changed.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 4, "Importing, Exporting, and Attaching," page 86.

[References](#)

INF: Btrieve 6.0 Data Type Does Not Work with Access

Article Number: Q93687
CREATED: 17-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access was not tested with Btrieve version 6.0. Therefore, new data types introduced with this version of Btrieve may not be compatible with Microsoft Access.

MORE INFORMATION

Btrieve 6.0 introduced a new [data type](#): Negative Sign Trailing Numeric Field. Microsoft Access does not explicitly translate this data type.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 4, page 66

[References](#)

PRB: 'Couldn't find object <tablename>' Error with Btrieve

Article Number: Q93685
CREATED: 17-DEC-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When attempting to import or attach Btrieve files, you receive the following error message:

 Couldn't find object <tablename>

CAUSE

Microsoft Access requires all three Xtrieve data dictionary files (FILE.DDF, INDEX.DDF, and FIELDS.DDF) in order to import or attach Btrieve files. The path statements in FILE.DDF may not be correct for the current environment.

RESOLUTION

Inspect the path statements in FILE.DDF. Either move the Btrieve data files to the locations specified in FILE.DDF, or Use Btrieve or a third-party tool to modify the FILE.DDF file so that the path statements to the data files are correct.

MORE INFORMATION

Microsoft Access requires all three Xtrieve data dictionary files (FILE.DDF, INDEX.DDF, and FIELDS.DDF) to be able to import or attach Btrieve files.

FILE.DDF contains the MS-DOS file name of each data [table](#) in the dictionary. These file names can be in any of the following valid MS-DOS formats:

- File Only (For example, MYDATA.DAT)
- Relative Path (For example, ..\DEV\TEST\MYDATA.DAT)
- Fully Qualified (For example, C:\TOOLS\DEV\TEST\MYDATA.DAT)
- Universal Naming Convention -- UNC (For example, (\\MYSERVER\USERS\JOE\TOOLS\DEV\TEST\MYDATA.DAT)

When importing or attaching a Btrieve file, Microsoft Access retrieves the MS-DOS file name from FILE.DDF and attempts to open the file. If the file does not exist in the location specified, you will see this error:

 Couldn't find object

In the case of the File Only and Relative Path formats, the ability to open the file depends on the accuracy of the current directory. For example, in the File Only format, MYDATA.DAT must exist in the current directory. In the Relative Path format, MYDATA.DAT must exist in the ..\DEV\TEST directory, which in turn must exist relative to the current directory.

Exactly which directory is the current directory is complicated by the fact that Windows-based applications all share the same environment and the same current directory. For example, if you change the current directory in Microsoft Excel by choosing Open from the File menu and changing directories to open a file, you also change the current directory for Microsoft Access.

Therefore, to fix the problem, you need to find out what path statements are included in FILE.DDF. You can do this by opening FILE.DDF in Microsoft Write, the word processor provided with Windows. When Microsoft Write asks if you want to convert this file to Microsoft Write format, specify No Conversion.

WARNING: do NOT save the file after you finish viewing it with Microsoft Write. If you do, you will corrupt FILE.DDF beyond repair.

After opening FILE.DDF in Microsoft Write, you will see several characters that look like a box. Scroll to the end of the file. There you will see the MS-DOS file names of the Btrieve data files. The extension is usually .DAT, but it does not have to be. Once you have noted the names and paths, close FILE.DDF taking care NOT to save the changes.

Now that you have the file names, here are two things you can do to fix the problem:

1. Move the files into the correct location(s). This is the easiest [method](#) and the one Microsoft recommends.
2. Use Btrieve or a third-party tool to modify the FILE.DDF file so that the paths are correct. This method is much more complicated. Microsoft recommends you do not attempt them unless you are a Btrieve expert.

[References](#)

INF: Microsoft Access Performance Enhancement Recommendations

Article Number: Q93684
CREATED: 17-DEC-1992
MODIFIED: 17-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you run Microsoft Access on a 4MB machine, you can gain performance improvements by altering the configuration of memory usage on the machine.

MORE INFORMATION

The following steps can be used to enhance the relative performance of Microsoft Access on computers with 4MB of RAM:

1. Do not use any of your RAM for a RAM disk.
2. Use a maximum of 512K for SmartDrive or other disk caches.
3. Set your [network](#) installation to use less than 200K of RAM, if possible.
4. Do not run several other applications that require large amounts of memory. Running even one other large application (such as a spreadsheet, word processor, or alternate desktop manager such as "Norton Desktop for Windows") can significantly degrade the performance of Microsoft Access.
5. Run Windows in standard mode.
6. Set the MaxBufferSize entry in MSACCESS.INI to a value less than 512. In low memory cases this will help, even though less memory will be used by Microsoft Access for buffering [database](#) I/O. (See the Microsoft Access README.TXT file for details on setting this value.)
7. Open databases as Exclusive and Read/Only, if possible.
8. If you do not want to use Access Wizards, you can disable them by removing the line "WIZARD.MDA=RO" from the [Libraries] section of the MSACCESS.INI file. This frees up over 300K of RAM, but takes away the ability to use Access Wizards.
9. Keep the number of Windows applets running to a minimum.
10. If you have a full-screen background [bitmap](#) (or "wallpaper") on your Windows desktop, replace it with a smaller bitmap, or no bitmap at all. For a standard VGA video display, this frees about 256K. For a 1024x768 256 color display, this can free about 3/4 of a megabyte. (Your actual memory savings may vary, depending on the display.)

11. Keep in mind that CD-ROM device drivers, sound board drivers, screen savers, MIDI drivers, multi-media support drivers, and other drivers take extra memory. The 4MB recommendation does not take this into account. If you need to have several drivers running under Windows, then more memory than 4 MB will be required for Microsoft Access to run efficiently.

The 4MB RAM recommendation was based on a 386/20 computer, MS-DOS 5.0, unmodified Windows 3.1, VGA display, mouse, and network workstation software. If your configuration is similar to, or better than this, you will get good performance. (This has been verified by benchmarks with Microsoft Access fully installed.) However, if your system's configuration is more sophisticated, you will require more memory for peak Microsoft Access performance.

[References](#)

PRB: Cannot Write README.TXT Error Msg During Installation

Article Number: Q93590
CREATED: 16-DEC-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

While installing Microsoft Access, you see the following error message when SETUP.EXE is processing Disk 6 or Disk 7:

Cannot Write C:\MSACCESS\README.TXT

CAUSE

This is a setup problem that occurs setup when Microsoft Access is writing the README.TXT file. There are a number of possible causes.

The MS-DOS APPEND command, located in the AUTOEXEC.BAT file, causes this error if there is a README.TXT file in any of the directories that are specified on the APPEND line.

When SETUP.EXE tries to install README.TXT, the APPEND command can cause MS-DOS to tell SETUP.EXE that there is a file with the same name in the installation directory. When SETUP.EXE is notified by MS-DOS that there is an existing file with the same name, SETUP tries to overwrite the existing file. MS-DOS returns an error code stating that the file is write protected.

Some customers have reported possible problems installing the README.TXT file when using certain terminate and stay resident programs (TSRs) such as Quicken's BillMind and Logitech's Click and Logimenu.

RESOLUTION

There are several possible workarounds. They are detailed in the More Information section below.

STATUS

This problem does not occur in Microsoft Access version 1.1

MORE INFORMATION

Before trying any of the following workarounds, ensure that you have a standard Windows configuration. Then exit from Windows and from all applications. Modify your AUTOEXEC.BAT file to remove any non-standard TSRs, and then restart your computer.

Workaround One

1. Use File Manager to search the entire hard drive for all README.TXT files.
2. Rename all README.TXT files to another file name (for example, READTHIS.TXT).
3. Install Microsoft Access from the original disks.

Workaround Two

1. Change to the directory of the path given in the error message.
2. Use a text editor such as the MS-DOS Edit program to create an empty README.TXT file.
3. Install Microsoft Access from the original disks.

Workaround Three

1. Make a duplicate copy of Disk 6 or Disk 7.
2. On the new disk, copy the README.TX\$ file to a new file named README.2
3. On the new disk, rename the README.TX\$ file to README.1. Then rename README.2 to README.TX\$.
4. Install Microsoft Access using the new disk you made.

Workaround Four

1. Make a duplicate copy of Disk 6 or Disk 7.
2. On the new disk, rename the README.TX\$ file to README.1.
3. On the new disk, use a text editor such as the MS-DOS Edit to create a new file and save it as README.TX\$.
4. Install Microsoft Access using the new disk you made.

[References](#)

INF: Using Data from Access in a Word for Windows Print Merge

Article Number: Q93427
CREATED: 10-DEC-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

Microsoft Word for Windows does not supply a converter for Microsoft Access [database](#) files. To use data from a Microsoft Access database in Microsoft Word for Windows, the data must be converted to a format that Microsoft Word for Windows recognizes. This can be done by exporting the data from Microsoft Access as text, or copying tables and queries from Microsoft Access and pasting them into a Microsoft Word for Windows document.

This article describes the steps to create a data file in Microsoft Word for Windows using each of the methods described above.

MORE INFORMATION

A data file is a comma- or tab-separated file that contains information that can be used in Microsoft Word for Windows for a print or mail merge.

A [field](#) name in Microsoft Word for Windows cannot contain more than 20 characters. A field name must begin with a letter and can contain only letters, numbers, and the underline character (). For example:

```
FirstName,LastName,Address,City,State,Postal_Code  
Randy,Johnson,123 West St.,Baker,OR,97445  
Jennifer,Smith,5 Circle Court,Yorkshire,WI,34507
```

NOTE: Do not include spaces after the commas. Spaces will be inserted as part of the mail merge.

Method One: Copying and Pasting

1. From the [Database window](#), select the [table](#) or [query](#) that contains the information you want.
2. Choose Copy from the Edit menu.
3. Activate Microsoft Word for Windows and open a new document.
4. From the Edit menu, select Paste.

Note: If there are spaces in your field name you will need to delete them once you have pasted the data into Microsoft Word for Windows.

5. Save this document as MyData.Doc.

Because this [method](#) does not place each field inside quotation marks, you may encounter errors if any of the fields contain tabs.

Method Two: Using The Transfer Text Macro Action

1. Open a new [macro](#).
2. Add the macro action TransferText with the following arguments:

Transfer Type: Export Delimited
Specification Name: <leave blank>
Table Name: <name of the table or query you wish
to export.>
File Name: <valid DOS file name.>
(Example: c:\winword\mydata.txt)
Has Field Name: Yes

This is a one time export. If the data in your database changes you will need to re-export it, overwriting the original file.

3. Perform the following steps in Microsoft Word for Windows:
 - a. Open the main document.
 - b. Choose Print Merge from the File menu.
 - c. Choose the Attach Data File button.
 - d. Select the exported text file that you created in step 2.

NOTE: If you did not specify a full path name for the text file, it will be in the same directory as your Microsoft Access database.

Now you can use the merge features of Microsoft Word for Windows: Insert Merge Fields, Edit Data Field, Check, and Print Merge. The check button will make a pass through the data document and check for errors, such as those caused by commas or tabs stored in the Microsoft Access fields.

References:

"Microsoft Word for Windows User's Guide," version 2.0, pages 620-622

[References](#)

PRB: 'Too Many Expressions in Distinct Output' Error

Article Number: Q93299
CREATED: 03-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

An attempt to create an [SQL](#) statement in the SQL [dialog box](#) or in [Access Basic](#) fails and Microsoft Access displays the following message:

Too many expressions in DISTINCT output.

CAUSE

The [query](#) contains a DISTINCT clause that lists more than ten expressions. The online help file documents the ten [field](#) limit as the cause of the error. However, help provides the only documentation of this limit. The other reference, Appendix B in the "Microsoft Access Language Reference," version 1.0, does not state a ten field limit for the DISTINCT clause.

RESOLUTION

There are two methods to deal with this error message:

- Modify the query to remove the [reserved word](#) DISTINCT.
- Reduce the number of fields in the DISTINCT clause to ten or fewer.

One possible [method](#) to work around this error involves concatenating some of the fields together into an [expression](#) when your query checks for DISTINCT rows. Each concatenation equals one DISTINCT expression. For example, substitute the code in example 1 below for the code in example 2 below.

Example 1

```
SELECT DISTINCT
    table1.lname & table1.fname,
    table1.address, ...;
```

Example 2

```
SELECT DISTINCT
    table1.lname,
    table1.fname,
```

table1.address,...;

[References](#)

INF: How to Change Read-Only Properties at Run Time

Article Number: Q93298
CREATED: 03-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not allow you to change some control properties, such as Color and Caption, at runtime. This article discusses how you can get around this limitation.

MORE INFORMATION

You can work around this limitation by creating two controls that are identical except for the property you want to change. You place these controls at the same location with their Visible property set to No. At the point where the property is to change, set the Visible property of the control with the desired property setting to True using a Macro SetValue action, and set the Visible property of the other control to False in the same manner.

The following example demonstrates this method. In the example, a form appears with one button and text box. The text box is Blue, and the button has the Caption "Red." When you click the button, the text box's Color will change to red and the button's caption will change to "Blue." Clicking the Blue button will toggle back to Red, and so on.

1. Create a new blank form.
2. Add two text box controls to the form and set their associated properties:

```
Text Box
-----
Control Name: Red
Fill Color: 255
Visible: No
```

```
Text Box
-----
Control Name: Blue
Fill Color: 16711680
Visible: Yes
```

3. Position the text box called "Blue" directly on top of the text box called "Red" and make sure they are exactly the same size so that one covers the other.
4. Add two command button controls to the form. Name one button "RedButton" and the other "BlueButton," and set their associated

properties:

```
Command Button
-----
Control Name: RedButton
Caption: Change to Red
Visible: Yes
On Push: ChangeColor.Red
```

```
Command Button
-----
Control Name: Blue Button
Caption: Change to Blue
Visible: No
On Push: ChangeColor.Blue
```

5. Position command button called "BlueButton" directly on top of the command button called "RedButton" and make sure they are exactly the same size so that one covers the other.
6. Add a text box to the form. Delete its label and set the text box's associated property:

```
Text Box
-----
Control Name: GiveMeFocus
Width: 0
```

The purpose of the zero width for the text box is to maintain focus while visible properties of other controls are changing.

7. Close and save the form as "ChangeColor"
8. Create a macro as shown below:

Macro Name	Action	Arguments
Blue	GoToControl	ControlName .. GiveMeFocus
	SetValue	Item [Blue]
		Expression ... [Red]
	SetValue	Item [Blue].Visible
		Expression ... True
	SetValue	Item [Red].Visible
		Expression ... False
Red	GoToControl	ControlName .. GiveMeFocus
	SetValue	Item [Red]
		Expression ... [Blue]
	SetValue	Item [Red].Visible
		Expression ... True
	SetValue	Item [Blue].Visible
		Expression ... False
	SetValue	Item [BlueButton].Visible
		Expression ... True

```
SetValue      Item ..... [RedButton].Visible  
              Expression ... False
```

9. Close and save the macro as "ChangeColor."

Open the ChangeColor form, type something into the text box, and click the button. Notice that each click of the button changes the caption of the button and the color of the text box without changing the value of the text box.

[References](#)

INF: How to Remove Attached Tables

Article Number: Q93297
CREATED: 03-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The TransferDatabase action is the [macro](#) equivalent of the Attach Tables command on File menu. However, there is no action to detach a [table](#). To remove an [attached table](#) from the user interface, select the table in the [Database window](#) and press the DELETE key. You can also create a macro to detach a table or you can call the action from a [Access Basic module](#) using the DoCmd statement.

MORE INFORMATION

To detach a table in Microsoft Access, create the following macro:

Macro Name	Action
RemoveTable	SelectObject DoMenuItem

RemoveTable Action

SelectObject
Object Type: Table
Object Name: <tablename>
In Database Window: Yes

DoMenuItem
Menu Bar: Database
Menu Name: Edit
Command: Delete

In Access Basic, you must use DoCmd to run the macro actions.

[References](#)

INF: View Procedures Sorts Modules Alphabetically

Article Number: Q93296
CREATED: 03-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you choose Procedures from the View menu after opening a [module](#), the Modules box displays the modules in alphabetic order. If the modules are numbered, they may appear out of order. You can rename the procedures so they appear properly sorted in the list.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create three modules (Module1, Module2, Module19).
2. Open Module1, and choose Procedures from the View menu.

The modules are listed Module1, Module19, and Module2, since they are sorted alphabetically.

To have the list sorted properly, rename Module1 and Module2 as Module01 and Module02. The modules will now be sorted as Module01, Module02, and Module19.

Note: The modules are sorted the same way in the [Database window](#).

[References](#)

PRB: 'Couldn't Open File <database name>'

Article Number: Q93295
CREATED: 03-DEC-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Attempting to open a [database](#) exclusively when another user already has it open results in the error message:

 Couldn't open file <database name>.

CAUSE

The help topic says in part "You must have ReadData permission for the file in order to view its data." The error message and help text are misleading because the problem is that the file cannot be opened exclusively.

To open the file successfully:

1. Press ESC or choose the OK button to exit the error message.
2. From the File menu, choose Open Database.
3. Select the database you want to open.
4. Clear the Exclusive [check box](#).
5. Choose OK.

STATUS

This behavior is by design.

[References](#)

INF: All Duplicate Macro Names in Macro Group Are Executed

Article Number: Q93294
CREATED: 03-DEC-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you have two macros in the same macro group with the same name, both macros will be executed. This problem will not occur if the macros are not consecutive in the macro group--that is, there is a different macro (Macro Name: "Between") between the two macros with the same name (Macro Name: "First One").

MORE INFORMATION

Steps to Reproduce Behavior

-
1. Create a new macro with two actions:

MacroName	Action	Argument
First One	MsgBox	"This is the message"
First One	MsgBox	"This is the message"

2. Save the macro as Macro1.
3. Run the macro using Macro1.FirstOne.

The MsgBox will appear twice.

[References](#)

PRACC9303: Error When Use SQL in Report RecordSource Property

Article Number: Q93293
CREATED: 03-DEC-1992
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You receive the following error message when you try to use a valid SQL statement as the RecordSource in a report:

This report is bound to a table or query that doesn't exist:
'<SQL statement>'.

RESOLUTION

Create a query using the desired SQL statement, and base your report on that query.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Open the NWIND.MDB database.
2. Open the Category List query in Design view.
3. To bring up the SQL statement for this query, choose SQL from the View menu.
4. Highlight the contents of the SQL window. Then use the keystroke combination CTRL+C to copy the SQL statement to the clipboard.
5. Choose Cancel to close the SQL dialog box. Then close the Category List query.
6. Open a new unbound form, and paste the SQL statement into the form's RecordSource property by using CTRL+V.
7. Switch to Form view.
8. Open a new unbound report, and paste the SQL statement into the report's RecordSource property by using CTRL+V.

9. Click the Print Preview button on the [toolbar](#).

You receive the error message on step 9, but not on step 7. You can use a valid SQL statement as the RecordSource in a form, but not in a report.

[References](#)

PRB: Close Dynaset Based on Form Multiple Times

Article Number: Q93292
CREATED: 03-DEC-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you create a [dynaset](#) that is based on a [form](#) through Access Basic, you can have multiple `dynaset.Close` statements without generating an error message.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

-
1. Open NWIND.MDB and create a new [module](#).

```
'*****  
'Declarations section of the module.  
'*****
```

Option Explicit

```
'=====
```

```
' The following function will create a dynaset based on
```

```
' the Add Products form in the NorthWind database. The
```

```
' function will then attempt to close the dynaset twice.
```

```
'=====
```

```
Function Base()  
    Dim MyDyna as Dynaset  
    Set MyDyna=Forms![Add Products].Dynaset  
    MyDyna.Close  
    MyDyna.Close  
End Function
```

2. Run the code. Notice that there are no error messages even though the code attempts to close the dynaset more than once.

[References](#)

PRB: Form Opened in Form View Is Completely Blank

Article Number: Q93261
CREATED: 03-DEC-1992
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You open a [form](#) in [Form view](#), and it is completely blank. There are no controls or text. Everything is blank.

CAUSE

Typically, this happens because the [dynaset](#) contains no records or privileges have been limited.

RESOLUTION

Controls located in form headers or footers are always visible -- even when privilege limits or empty dynasets prevent the display of controls located in the detail section. Therefore, when you design a form, place necessary controls such as command buttons and check boxes in the form [header](#) or [footer](#) section. Also, you should ensure that each dynaset contains at least one [record](#).

STATUS

This is by design.

MORE INFORMATION

The following six situations may cause a blank form to display:

1. The form is based on a [query](#) that contains a one-to-many [join](#), and the query returns no records.
2. The form is not updatable at all. It is a [read-only](#) form with the AllowUpdating property set to No Tables, or it is a static form that displays results by using a GROUP BY clause or a DISTINCT operator.
3. The form is based on a query that contains a one-to-one join, and one of the tables is in a read-only [database](#).
4. The form is based on a query that contains a many-to-one inner join, and the query does not contain all the joined columns of the many-side [table](#).
5. The form is based on a query that includes an updatable remote table but does not include all the columns that make up the [primary key](#).
6. The form is read-only and is based on an empty table.

Steps to Reproduce Situation 1

1. Change the Catalog query in the NWIND sample database placing IS NULL in the criteria cell for the field [Category Name].
2. Create a new form based on the modified query, and place some fields and a command button in the detail section.
3. Switch to Form view.

Result: The blank form is displayed.

Steps to Reproduce Situation 6

1. Create a new table. Do not enter any data.
2. Create a new form based on the table. Add a command button and make the form read-only.
3. Switch to form view.

Result: A blank form is displayed.

The form displays blank in all situations because the dynaset is empty. The form would display as expected if you were adding new records, but the setting for the form or the joins in the query prevent you from using the form to add new records.

To work around this issue, place necessary controls or command buttons in the form header and footer. Headers and footers display even when the detail section is blank.

[References](#)

PRACC9303: Error Msg: Data Has Changed, Operation Aborted

Article Number: Q93146
CREATED: 24-NOV-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

When a user (User1) attempts to commit a record in an attached Paradox table and that record has been changed by another user (User2), User1 sees this error message:

Data has changed; operation aborted.

After User1 chooses the OK button, a second error message appears:

This record has been changed by another user since you began editing it... Save Record, Copy to Clipboard, Drop Changes.

If User1 selects the Save Record option, the first error message is repeated. User1 is caught in a continuous loop, unable to commit the record.

RESOLUTION

To save the changes made to the current record, User1 must:

1. Select the Copy to Clipboard option.
2. Select the record User1 was attempting to change.
3. Choose Paste from the Edit menu.
4. Commit the record.

User1 can prevent User2 from changing a record while User1 is editing it by changing the multiuser option to Edited Records.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

These steps require either two users or two instances of Microsoft Access.

1. Both users attach to a Paradox Table.
2. Both users set their multiuser options to No Locks.
3. User1 starts to edit a field but does not commit the record.
4. User2 changes and commits the same record by choosing Save Record from the File menu or by moving the cursor to the next record.
5. User1 attempts to commit the changed record.

Error Message #1: Data has changed; operation stopped.

Error Message #2: This record has been changed by another user since you began editing it... Save Record, Copy to Clipboard, Drop Changes.

[References](#)

INF: How Access Handles Logins to Attached SQL Tables

Article Number: Q93145
CREATED: 24-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

How Microsoft Access handles a login to an attached [SQL table](#) depends on how the table was originally attached, and whether the login and password were saved with the table information. It may also depend on who attached the SQL table, what rights that person has to that table, what rights you have to that table, and where the tables are located (different servers, different databases, same [database](#)).

MORE INFORMATION

If the "Save login ID and password locally" option was enabled when a SQL table is attached, you will not be prompted for a login ID and password. Instead, you automatically receive the rights to each table that are appropriate for the login ID used when the table was attached.

If this option was not enabled, you are prompted for a login ID and password once for each unique data source, and you receive the rights to each table that are appropriate for the login ID that you use.

NOTE: A unique data source is composed of the unique combination of a SQL server and a database on that server.

So, if you attach multiple SQL tables and use different login IDs with different rights for each table, but do not enable the option to save the login and password information, the next time you access the tables, the login ID and password you supply for each data source will determine the authority you have to the tables from that data source.

References:

"Microsoft Access User's Guide," version 1.0, pages 60-62, 66-68

[References](#)

INF: Microsoft Access is Pen Compatible but not Pen Aware

Article Number: Q93144
CREATED: 24-NOV-1992
MODIFIED: 05-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access is pen compatible but not pen aware. This means you can use the pen as a substitute for the mouse.

MORE INFORMATION

The request to allow Microsoft Access access to Pen extensions has been noted as an enhancement request and will be considered for future releases.

[References](#)

INF: Online Help for Sigma Button Is Incorrect

Article Number: Q93143
CREATED: 24-NOV-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

In Microsoft Access Help, if you search for "Sigma Button," you are incorrectly pointed to information about the Reinitialize button. Instead, you should be pointed to information about the Totals button.

MORE INFORMATION

One of the tools available on the [toolbar](#) in the Query Design window is the Totals button. The symbol used to identify the Totals button is the Greek letter sigma.

The Reinitialize button is one of the tools available on the toolbar in the Module Design window.

[References](#)

PRB: Option Button Appears Fuzzy in Datasheet View

Article Number: Q93142
CREATED: 24-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you create a [form](#) and place an [option button](#) on the form, then switch to [Datasheet view](#), the outline of the option button will appear fuzzy. This happens when the focus is on any option button in Datasheet view.

RESOLUTION

Move the focus to a [field](#) that does not contain an option button. The option button will change to a normal appearance.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Problem

1. Create a new blank form.
2. From the [toolbox](#), select the Option Button and place it on the form.
3. Switch to Datasheet view.
4. Make sure the focus is on the button. Notice the outline of the button has a fuzzy appearance.

[References](#)

INF: How to Pass a Single Byte of Data to Windows API Calls

Article Number: Q93141
CREATED: 24-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not have a single-byte [data type](#). To pass a byte to an external [function](#) such as a Windows application programming interface (API) function or a dynamic-[link](#) library (DLL), you should declare the byte as an integer.

If you need to pass a single byte of information in a data type, such as the RGBQUAD structure, you should pass the data structure as a [string](#) type.

MORE INFORMATION

The Intel chip supports only a full word on the stack. You cannot put just 8 bits on the stack; you must use the full 16 bits. If you attempt to pass only 8 bits, you will end up passing 16 bits (your 8 bits of data plus an additional 8 bits that will be "padded" on). This is a limitation of the processor, not Microsoft Access.

To pass a single byte of data in a structure such as the RGBQUAD, use a fixed-length string of 1. For example, suppose you want to pass a single byte in the "C" RGBQUAD data structure, which looks like this:

```
typedef struct tagRGBQUAD{
    BYTE    rgbBlue;
    BYTE    rgbGreen;
    BYTE    rgbRed;
    BYTE    rgbReserve;
} RGBQUAD
```

You can redefine this structure with [Access Basic](#) as:

```
Type RGBQUAD
    rgbBlue As String * 1
    rgbGreen As String * 1
    rgbRed As String * 1
    rgbReserve As String * 1
End Type

Dim RGB As RGBQUAD

RGB.rgbBlue = Chr$(10)
RGB.rgbRed = Chr$(14)
RGB.rgbGreen = Chr$(15)
```

NOTE: In the above example, Blue=10, Red=14, and Green=15.

For more information on this topic, search for "Type Statement" using the Microsoft Access Help menu.

[References](#)

PRB: Cannot Rollback in Transaction without Closing Database

Article Number: Q93140
CREATED: 24-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A subroutine or a function contains a transaction process which calls a subroutine or function that dimensions and opens a database using the OpenDatabase method. When the subroutine or function exits, Microsoft Access implicitly closes the database (no Close method is called). When the subroutine or function attempts to execute the CommitTrans or Rollback statement, Microsoft Access displays the following message:

Commit or Rollback without BeginTrans.

CAUSE

When the called subroutine or function exits, Microsoft Access implicitly closes the database. Internally, the initial attempt to close the database fails because the database was opened in a transaction. Microsoft detects this error and performs an implicit transaction rollback, then it closes the database successfully.

RESOLUTION

To perform a Rollback in the called subroutine or function, you must first use call the Close statement to close the open database.

STATUS

This behavior is by design.

MORE INFORMATION

Perform the following three steps to demonstrate this behavior.

Steps to Reproduce Behavior

1. Create the following module.

```
'*****  
'Declarations section of the module.  
'*****  
  
Option Explicit  
  
'=====
```

'The following Sub procedure starts a transaction, calls another sub procedure, then attempts to rollback.

```
Sub Test1
    BeginTrans
        Call Test2
    Rollback
End Sub
```

'The following Sub procedure sets a Database variable to the current database.

```
Sub Test2
    Dim MyDB as Database
    Set MyDB = CurrentDB()
    'the database is implicitly closed on exit
End Sub
```

2. Compile the module.
3. In the Immediate window, enter Test1. Microsoft Access generates the following message:

Commit or Rollback without BeginTrans.

Perform the following four steps to demonstrate the resolution to this problem.

1. Perform Step 1 above.
2. Edit the second function to add MyDB.Close, as follows.

```
Sub Test2
    Dim MyDB as Database
    Set MyDB = CurrentDB()
    MyDB.Close
End Sub
```

3. Compile the module.
4. In the Immediate window enter Test1. The Rollback works correctly because the code explicitly closes the database opened in the transaction.

[References](#)

PRB: Inaccurate Message When Opening MDB Shared vs. Exclusive

Article Number: Q93139
CREATED: 24-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you open a [database](#) exclusively, then attempt to open the same database as shared in another instance of Microsoft Access, you will receive the following error:

'Database.mdb' This file is already in use. Use a new filename or close the file in use by another application.

When you open the database as shared, then attempt to open the same database exclusively in another instance of Microsoft Access, you will receive the error:

Couldn't open file '<path><filename>'.

CAUSE

The second error can be confusing because it does not state why the file couldn't be opened. You can get around this error by opening the database in the second instance of Microsoft Access as shared, or by closing the database in the first instance of Microsoft Access.

STATUS

The behavior is by design.

[References](#)

INF: Using Column Property of Combo Box to Update Text Box

Article Number: Q93138
CREATED: 24-NOV-1992
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

You can display multiple fields in a combo or [list box](#) on a [form](#) or [report](#) even when those fields come from a [table](#) that is not bound to the form or report. You can also [update](#) controls such as text boxes with new information based on what a user selects from a combo or list box.

More Information:

To accomplish these objectives, use one of these three techniques:

1. Use Row Fix-up in forms. For more information on Row Fix-up, [query](#) on the following words in the Microsoft Knowledge Base:

"Row" and "Fix-up"

2. Use multiple DLookup() functions in forms and reports. For more information on the DLookup() [function](#), query on the following words in the Microsoft Knowledge Base:

"DLookup" and "forms" and "reports"

3. Use the Column property of a multiple-[column combo box](#) to update a [text box control](#) with new information as focus moves from [row](#) to row in the combo box. The next section describes this technique in detail.

Using the Column Property of a Combo Box

By assigning the Column property of a multiple-column combo or list box to a text box, you can display one column from the current combo-box selection in the text box. Microsoft Access automatically updates the text box as focus changes from row to row in the combo box. Here is the procedure:

1. Create your form or report based on the desired table or query.
2. Create a combo or list box that retrieves information from more than one [field](#). For example, you might use the following multiple-column Select statement as the RowSource for a combo or list box to display information from several columns in the Categories table:

```
Select [Category Id], [Category Name], [Description] from  
Categories Order by [Category Name];
```

For this example, set the RowCount property to 3, and set the ColumnWidth property to an appropriate size for the combo or list box. Once you have the combo or list box sized correctly and defined to return multiple fields, you can use the Column property to display the current selection in a text box control. Choose one entry in the list box first or it will return a null.

The Column property uses a reference argument to refer to a specific column in the multiple-column combo or list box. Use Column(0) to refer to the first column, Column(1) to refer to the second column, and so on.

This example uses Column(1) to refer to [Category Name] -- the second column in the combo box.

3. To display the [Category Name] of the current combo box selection, create a text box control. Make the text box a calculated control by defining the following expression as the data source for the text box:

```
=[ComboControlName].Column(1)
```

ComboControlName is the name of the combo box. The Column property will make the text box (calculated control) read only.

References:

For more information on how to set up a combo box see the "Microsoft Access User's Guide," version 1.0, Chapter 9, "Designing Forms," pages 233-245.

For more information on the Column property, see the "Microsoft Access Language Reference," version 1.0, "Column Property," pages 72-73.

[References](#)

INF: Field Size of 0 Treated as 255

Article Number: Q93137
CREATED: 24-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

If you set a field size to zero (0) for a text field in table Design view, the field is treated as if the size is set to 255. When you close and save the table, and then reopen it in Design view, the field size will have been changed to 255.

References

PRB: Creating Multiple New Import/Export Specifications

Article Number: Q93136
CREATED: 24-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Creating multiple new Import/Export specifications using the Import/Export Setup [dialog box](#) requires that you either exit the dialog or delete or change each entry in the dialog that does not apply. The dialog does not provide a [method](#) to clear or reset each option in the dialog.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open any [database](#) or create a new one.
2. From the File menu, choose Imp/Exp Setup.
3. Enter information in several fields.
4. From the File menu, choose Save As.

If you want to create another specification at this point, you must either exit this dialog or manually delete all of the irrelevant information.

[References](#)

INF: Using Undocumented CreateForm(),CreateReport() Functions

Article Number: Q93096
CREATED: 23-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article discusses the Microsoft Access undocumented functions CreateForm() and CreateReport(). If you intend to write your own FormWizard or ReportWizard, you can use these functions to create and customize a blank [form](#) or [report](#) to which you can add controls.

This article assumes that you are familiar with [Access Basic](#) and designing forms and reports.

MORE INFORMATION

CreateForm()/CreateReport() is roughly equivalent to choosing "New... Form/New... Report" from the File menu at the top of the screen. When the [function](#) is executed, a new empty form/report appears in an iconized state.

Both functions return an object value that you can use for further manipulation, and neither function requires parameters.

To use these functions, first define a form or report object [variable](#), then assign the variable to the function name. An example of how to do this is:

```
Dim MyForm As Form
Set MyForm = CreateForm()
```

After opening the form/report in design mode after executing the commands above, you can bind the form to a [table](#) or [query](#) by modifying the form's/report's RecordSource property:

```
MyForm.RecordSource = "Categories"
```

With the form/report in design mode, you can access and change any of the other design-time properties of the form/report. You can also access and change properties of each of the form's/report's sections via the Section property. The Section property is actually an array with each array value being a reference to a form's/report's section. For forms, the Section property array is broken down as:

```
Section(0) - Detail Section
Section(1) - Form Header
Section(2) - Form Footer
Section(3) - Page Header
Section(4) - Page Footer
```

For reports, the section property array is broken down as:

- Section(0) - Detail Section
- Section(1) - Report Header
- Section(2) - Report Footer
- Section(3) - Page Header
- Section(4) - Page Footer
- Section(5) - Group Level 1 Header
- Section(6) - Group Level 1 Footer
- Section(7) - Group Level 2 Header
- ...etc.

With this information, you can customize the design of a form/report section programmatically. The following example creates a new report, hides the page [footer](#) by setting its Visible property to false, sets the Height property of the detail section, and enables its KeepTogether property:

```
Dim MyReport As Report
Set MyReport = CreateReport()
MyReport.Section(4).Visible = False
MyReport.Section(0).Height = 1760
MyReport.Section(0).KeepTogether = True
```

References:

For more information on Microsoft Access libraries, query on the following words here in the Microsoft Knowledge Base:

libraries and [debugging](#) and creating

For more information about other ReportWizard and FormWizard functions, query on the following words here in the Microsoft Knowledge Base:

CreateControl and CreateReportControl and CreateForm and CreateReport and Section Property and DeleteControl and DeleteReportControl and CreateGroupLevel

[References](#)

INF: CreateControl() and CreateReportControl() Functions

Article Number: Q93095
CREATED: 23-NOV-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article discusses the Microsoft Access functions CreateControl() and CreateReportControl(). If you intend to write your own FormWizards or ReportWizards, you can use these functions to create controls on a [form](#) or [report](#) that are available in [Design view](#).

This article is for those who are already familiar with [Access Basic](#) and with designing forms and reports.

MORE INFORMATION

You can use the CreateControl() and CreateReportControl() functions to create new controls for a form or report opened in Design view. These functions require two arguments: the name of the form or report as a [string](#) value and a numeric code that represents the [control](#) type.

Here are the control types and their associated numeric codes:

Integer	Control Type
100	label
101	rectangle
102	line
103	(not used)
104	command button
105	option button
106	check box
107	option group
108	bound object frame
109	text box
110	list box
111	text box with drop-down grid
112	subform/subreport
113	graph
114	unbound object frame

CreateControl() and CreateReportControl() both return a control object value. Therefore, you need first to define a control [variable](#), and then set the control variable equal to the [function](#) name.

For example, the following procedure creates a form and then adds a button to the form:

```
Dim MyForm As Form, MyControl As Control
```

```
Set MyForm = CreateForm()  
Set MyControl = CreateControl(MyForm.FormName, 104)
```

When the procedure is finished, you can modify the properties of the new control by using the control variable you defined. For example, you can change the control's Width and Caption properties with these statements:

```
MyControl.Width = 2000  
MyControl.Caption = "&Sum All Records"
```

For controls that are frequently associated with a [field](#) in a [table](#) or [query](#), you can modify the ControlSource property to bind the control to the field.

By default, some controls are created with their Height and Width properties set to zero to make them invisible. Also by default, controls appear in the upper-left corner of the form. You can adjust the size and position of a control immediately after you create it by changing the control's properties. For example, the following code creates, sizes, moves, and then binds a text box to a field by changing the properties:

```
Set MyControl = CreateControl(MyForm.FormName, 109)  
MyControl.Width = 1500  
MyControl.Height = 200  
MyControl.Top = 440  
MyControl.Left = 200  
MyControl.ControlSource = "[Category ID]"
```

In addition to the form name and the code for the type of control, you can also specify the form or report Section where you want Microsoft Access to place the control.

The following is the CreateControl() [syntax](#). The CreateReportControl() syntax is identical.

NOTE: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function CreateControl (FormName As String, _  
                        ControlType As Integer _  
                        [,SectionNumber As Integer])
```

SectionNumber is the Section property setting number that identifies a particular section as follows:

```
Detail section ..... 0  
Form or report header section ..... 1  
Form or report footer section ..... 2  
Form or report page header section ..... 3  
Form or report page footer section ..... 4  
Group-level 1 header section (reports only) ... 5  
Group-level 1 footer section (reports only) ... 6  
Group-level 2 header section (reports only) ... 7  
Group-level 2 footer section (reports only) ... 8
```

If a report has additional group-level sections, the header/footer pairs

are numbered consecutively beginning with 9.
References:

For more information on Microsoft Access Libraries, query on the following words in the Microsoft Knowledge Base.

libraries and [debugging](#) and creating

For more information on other AccessWizard functions, query on the following words in the Microsoft Knowledge Base:

CreateControl and CreateReportControl and CreateForm and
CreateReport and 'Section Property' and DeleteControl and
DeleteReportControl and CreateGroupLevel

[References](#)

INF: DeleteControl, DeleteReportControl, and CreateGroupLevel

Article Number: Q93094
CREATED: 23-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article discusses two Microsoft Access undocumented commands, DeleteControl and DeleteReportControl, and an undocumented [function](#) CreateGroupLevel(). If you intend to write your own FormWizard or ReportWizard, you can use the commands to delete controls on a [form](#) or [report](#) that is available in [Design view](#), and you can use the function to create groups on a report that is available in Design view.

MORE INFORMATION

You can use DeleteControl and DeleteReportControl commands to delete a [control](#) that exists on a form or report in design view. Both commands require a [string](#) value that represents the name of the form or report and a string value that represents the name of the control.

DeleteControl and DeleteReportControl are commands, not functions. They do not return a value.

The example below creates a form, then creates a button on the form, displays a message, and deletes the button on the form:

```
Dim MyForm As Form, MyControl As Control
Set MyForm = CreateForm()
Set MyControl = CreateControl(MyForm.FormName, 104)
MsgBox "About to Delete " & MyControl.ControlName
DeleteControl MyForm.FormName, MyControl.ControlName
```

When you delete a control using DeleteControl or DeleteReportControl, there is no visual feedback that the control is gone if the form or report is not minimized. The control still appears as if it were not deleted even though you can not click it to give it focus. To make the deleted control disappear, you must repaint the form design window by minimizing it and then restoring it.

If you have a report available in design view, you can create groups for it in a procedure by using the CreateGroupLevel() function. Here is the [syntax](#) for CreateGroupLevel():

```
Function CreateGroupLevel (ReportName$, Expression$,
                          HasHeader$, HasFooter$)
```

ReportName is a string [expression](#) representing the report name.

Expression is a string expression representing the name of the [field](#) or

calculated field that you want to [group](#) by.

HasHeader and HasFooter indicate whether the group includes a group [header](#) or group [footer](#), respectively.

On execution, CreateGroupLevel() adds a group at the innermost level. For example, if one group already exists, CreateGroupLevel() creates a second group within the first group. CreateGroupLevel() returns a number indicating which level it created, beginning with zero. In the example just mentioned, CreateGroupLevel() returns one because it is the second group, and the first group was Group zero.

REFERENCES

=====

For more information on Microsoft Access Libraries, [query](#) on the following words here in the Microsoft Knowledge Base:

libraries and [debugging](#) and creating

For more information on other AccessWizard functions, query on the following words here in the Microsoft Knowledge Base:

CreateControl and CreateReportControl and CreateForm and CreateReport and 'Section Property' and DeleteControl and DeleteReportControl and CreateGroupLevel

[References](#)

INF: Users Can Edit but Not Commit Read-Only OLE Objects

Article Number: Q93026
CREATED: 22-NOV-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Users are allowed to edit [OLE](#) objects in a [database](#) that has been opened as [read-only](#), or in a [form](#) that is read-only. However, when a user attempts to move to the next [record](#) or chooses Save Record from the File menu, Microsoft Access displays a message indicating that the record cannot be updated.

To prevent users from attempting to edit OLE objects in read-only forms, set the Enabled property of the object frame to No.

MORE INFORMATION

In a read-only database, a user is allowed to edit and [update](#) OLE objects but is not allowed to commit a record with these changes. At the point the record is committed, the following message appears:

Can't save changes to this object because you don't have permission to write to the record. Copy the object to the [Clipboard](#) if you want to save it, then choose Undo Field.

The user must undo the changes before the record can be committed. To save the changes, the user must copy the object to the Clipboard before choosing Undo Field.

[References](#)

INF: Using DLookup() to Look Up Values in Forms and Reports

Article Number: Q93025
CREATED: 22-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can display multiple fields in a combo or [list box](#) on a [form](#) or [report](#) even when those fields come from a [table](#) that is not bound to the form or report. You can also [update](#) controls such as text boxes with new information based on what a user selects from a combo or list box.

MORE INFORMATION

To accomplish these objectives, use one of these three techniques:

1. Use Row Fix-up in forms. For more information on Row Fix-up, [query](#) on the following words in the Microsoft Knowledge Base:

"Row Fix-up" and "multiple" and "One to Many"

2. Use multiple DLookup() functions in forms and reports. See the "More Information" section below.

3. Use the Column property of a multiple-[column combo box](#) to update a [text box control](#) with new information as focus moves from [row](#) to row in the combo box. For more information on this, query on the following words in the Microsoft Knowledge Base:

"Column Property" and "combo box" and "Update"

MORE INFORMATION

DLookup() Function

Here is the [syntax](#) for DLookup():

```
DLookup(expr, domain [, criteria] )
```

The DLookup() Function returns a value from a [field](#) in a specified set of records called the domain. The DLookup() [function](#) specifies the criteria for the domain. To make the domain dependent on one or more values listed in controls on a form or report, refer to the controls in the DLookup() criteria argument.

The following example looks in the Employees table (the domain) and returns the Last Name of the Employee ID specified in the [ControlName] control on the Form.

NOTE: In the following sample code, an underscore _ is used as a

line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
=DLookup("[Last Name]", "Employees", "[Employee ID] =_  
Forms!FormName![ControlName]")
```

For reports, use Reports!ReportName![ControlName].

DLookup() returns one value from a single field even if more than one [record](#) satisfies the criteria. If no record satisfies the criteria or if the domain contains no records, DLookup() returns [Null](#).

REFERENCES

=====

For more information on DLookup(), search for "DLookup function" using the Microsoft Access Help menu.

[References](#)

INF: Default AutoLabel and AddColon Properties

Article Number: Q93024
CREATED: 22-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The following [table](#) shows the default values for the AutoLabel and AddColon properties for the different [form](#) and [report](#) objects:

Toolbox object	AutoLabel	AddColon
Text box	Yes	Yes
Option group	Yes	No
Toggle button	No	No
Radio button	Yes	No
Check box	Yes	No
Combo box	Yes	Yes
List box	Yes	Yes
Subform/report	Yes	Yes
Bound object frame	Yes	Yes
Command button	No	No

References:

"Microsoft Access User's Guide," version 1.0, Chapter 9, "Designing Forms," and Chapter 18, "Designing Reports"

[References](#)

INF: No Custom Printer Setup for Mailing Labels ReportWizard

Article Number: Q93023
CREATED: 22-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The Mailing Labels ReportWizard allows you to choose from 66 different labels. However, the ReportWizard does not verify whether the labels you select will correctly fit on the paper size that you have specified.

You may need to manually customize your printer setup and/or the number of labels per page after using the ReportWizard.

MORE INFORMATION

The need to customize the printer setup is most common with dot-matrix printers. Label sizes other than .5 inch or 1 inch may not fit evenly on the default 11-inch page.

To manually customize your printer setup and/or the number of labels per page:

1. Open your [report](#) in Print Preview.
2. From the File menu, choose Print Setup.
3. Customize your margins to accommodate the number and/or width of the labels.
4. Choose the More button in the Print Setup [dialog box](#) to customize the number of rows of labels.

[References](#)

PRB: Import/Export Setup Cancel Button Cancels Unsaved Changes

Article Number: Q93022
CREATED: 22-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The Cancel button in the Import/Export Setup [dialog box](#) cancels only those changes that have not been saved. It does not cancel all changes made since entering the dialog box.

CAUSE

If you make a change to the specification and then choose the Save As button, choosing the Cancel button does not invalidate those changes. The Cancel button cancels only those changes that have not been saved and then returns you to the [Database window](#).

NOTE: The OK button performs a "save and exit"; the Save As button saves your changes but does not close the dialog box.

STATUS

This behavior is by design.

[References](#)

PRB: Undo Property Setting Command Appears in Macro Design

Article Number: Q92951
CREATED: 19-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The text string for the Undo command does not correctly reflect the type of object that you are editing. If you are adding or modifying actions in a macro, and then open the Edit menu, the text string that you see for the Undo command is

Undo Property Setting

rather than something similar to "Undo Macro Action."

RESOLUTION

Even though the specific words associated with the Undo command are misleading, the Undo functionality is unaffected.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. Microsoft is researching this problem and will post new information here as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Create a new macro.
2. In the Macro Action column, type "Close" (without the quotation marks) and press ENTER.
3. Open the Edit menu. Note the Undo Property Setting command.
4. Choose the Undo Property Setting command.

The word "Close" is now gone.

[References](#)

INF: Bound OLE Object Blank in Form View

Article Number: Q92950
CREATED: 19-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If a bound object frame on a form has the ControlSource property set to a field that is not of the type OLE Object, this control will be blank when the form is in Form view. In Datasheet view, the cells for this control contain "Invalid Object."

CAUSE

A bound object frame must refer to a field of the type OLE Object in a table or query in order to display anything in Form view. To display an object in this control, the ControlSource property must be set to an OLE Object field. For example, Microsoft Access cannot display the contents of a Text field as an OLE Object. Object frames will appear blank unless there is something to show.

STATUS

This behavior is by design.

[References](#)

INF: Choosing Toolbar Design Button Causes Unexpected Behavior

Article Number: Q92949
CREATED: 19-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

In Form view, pressing the Design button on the toolbar does not return Microsoft Access to Design mode if the form's OnClose property opens another form.

MORE INFORMATION

Perform the following seven steps to demonstrate this behavior.

Steps to Reproduce Behavior

-
1. Create two new forms named Form1 and Form2.
 2. Create a macro named TestMacro to perform two actions: close the open form (Form1) and then open Form2.

Macro Name	Action
TestMacro	Close OpenForm

TestMacro	Actions
-----------	---------

Close
Object Type: Form
Object Name: Form1

OpenForm
Form Name: Form2
View: Form
Data Mode: Edit

3. View the property sheet for Form1. Set the OnClose property to run the macro named TestMacro you created in step 2.

Object: Form

OnClose: TestMacro

4. Save Form1 and close it.

5. Select Form1 from the [Database window](#) and choose the Design button.
6. Choose the View Form button on the toolbar. (The form view is displayed.)
7. Choose the Design button on the toolbar.

At this point, you would expect to return to Design mode. However, Microsoft Access runs the OnClose macro and opens Form2.

[References](#)

INF: Requery Control Err Msg: There Is No Control Called...

Article Number: Q92899
CREATED: 17-NOV-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you try to requery a control by using the Requery macro action, an error appears indicating that the specified control does not exist even though the control clearly exists on the form.

MORE INFORMATION

For the Requery Control Name parameter, Microsoft Access expects a short control name without any brackets. The Help system informs you that you should not use the full syntax, but it does not mention that the brackets must be omitted even if the control name consists of more than one word. Because other scenarios require the use of brackets, this inconsistency is confusing.

The following example reproduces the problem and discusses the solution:

1. Create an unbound form called Form1.
2. Place a combo box bound to NWIND.MDB "Sales Totals Query" on the form and call the combo box Field1.
3. Create a macro called Macro1 with the following Action/Parameter:

```
Action      : Requery
Control Name : [Field1]
```

Note that [Field1] is enclosed in brackets.

4. Save and close Macro1 and drag it on to Form1 in design mode.
5. Switch to Browse mode on the form and choose the button.

The following error message appears:

```
There is no control called '[Field1]'.
```

6. Remove the brackets from the Control Name parameter in Macro1 and try again.

The control should now work properly.

[References](#)

INF: Double-Clicking on Control Moves Control Unintentionally

Article Number: Q92898
CREATED: 17-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you double-click on a control on a form, a property sheet will appear containing the properties for that control. A negative side-effect to this feature is that the control sometimes gets moved when the user double-clicks on it. This article discusses some tips on how to avoid or correct this situation.

MORE INFORMATION

Here are a couple of tips for avoiding this problem...

- Use the TAB key to move between controls.. When you move to the control by using the TAB key, the control will not be moved. Once the desired control has focus, you can choose the Property Sheet button in the toolbar at the top of the screen.

An alternative to tabbing to the control is to select it by drawing a selection around it with your mouse.

- Take advantage of the 'Snap to Grid' feature. This feature disallows any moving of a control unless the move is significant, thus reducing the possibility of moving the control accidentally. Turn 'Snap to Grid' off only when you need fine movement for the control. You can toggle 'Snap to Grid' by choosing Snap to Grid from the Layout menu at the top of the screen.
- If you accidentally move the control, you can choose Undo Move from the Edit menu at the top of the screen immediately after the move.

References

INF: Changing the RowSource of a ListBox/ComboBox at Runtime

Article Number: Q92897
CREATED: 17-NOV-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The RowSource property for a ComboBox or ListBox cannot be modified while the [form](#) is active. However, there is a workaround for changing the recordset that the object is bound to. This article discusses that workaround.

MORE INFORMATION

The workaround for this problem is to bind the object to a [query](#), then manipulate the query's recordset underneath the object. For example, you would bind the object to a query called Query1, then create a new query called Query1 programmatically, while the object is active.

The following example illustrates this. In the following example, there is a [combo box](#) bound to data from the NWIND.MDB Products [table](#). By clicking on one of two check boxes, the order of records in the combo box will be changed.

1. Create a form with an Option Group [control](#) called MyOption, and a combo box called MyCombo.
2. Add two [check box](#) controls to MyOption and specify the following property:

```
AfterUpdate: =NewOrder([MyOption])
```

3. For MyCombo, specify the following properties:

```
RowSourceType: Table/Query  
RowSource: DeleteMe  
ColumnCount: 2  
Locked: True  
Enabled: False
```

4. Add the following [function](#) to a [module](#) in the [database](#):

NOTE: In the following sample code, an underscore(_) is used as a line continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
Option Explicit  
Function NewOrder (WhichSort%)  
    DoCmd Hourglass True  
    Dim D As Database, Q As QueryDef
```

```

Set D = CurrentDB()

Screen.ActiveForm!MyCombo.Enabled = True
Screen.ActiveForm!MyCombo.Locked = False

On Error Resume Next
D.DeleteQueryDef ("DeleteMe")
On Error GoTo 0

If WhichSort = 1 Then
    Set Q = D.CreateQueryDef("DeleteMe",_
        "SELECT [Product ID], [Category ID]_
        FROM Products ORDER BY [Product ID];")
Else
    Set Q = D.CreateQueryDef("DeleteMe",_
        "SELECT [Product ID], [Category ID]_
        FROM Products ORDER BY [Category ID];")
End If

Q.Close
D.Close

DoCmd Requery "MyCombo"
DoCmd Hourglass False
End Function

```

5. Open the form, click one of the check boxes, and drop down the combo box. Notice the order of records in the combo box.
6. Click the other check box and drop down the combo box again. Notice that the order of records has changed.

The key to changing the recordset is found in Step 4 above. By using different SELECT statements for the CreateQueryDef() methods, you can create any kind of query.

[References](#)

INF: Hourglass in Setup Changes to Arrow During Copy

Article Number: Q92896
CREATED: 17-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

During the Microsoft Access setup process, the mouse pointer will change from an hourglass to an arrow if the mouse is moved while files are being copied. The pointer will not return to an hourglass until the Setup program changes to another disk.

CAUSE

This behavior is by design and reflects the ability to switch the Setup program to background processing, which enables you to continue working with other applications running in the Windows environment.

Steps to Reproduce Problem

1. Run SETUP.EXE.
2. Choose Complete Installation. When Microsoft Access begins copying the files, move the mouse. The hourglass will change to an arrow.

[References](#)

INF: CopyObject Appears to Copy Wrong Object

Article Number: Q92895
CREATED: 17-NOV-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The [macro](#) command "CopyObject" always copies the currently selected object from the [Database window](#). When you precede this macro command with a "SelectObject" command, set its "In Database Window" parameter to "No", then run the macro from the Database window, the macro is the selected object and the object copied.

MORE INFORMATION

Steps to Reproduce Behavior

-
1. Create a [table](#) and save it as "Table1".
 2. Create the following macro and call it Macro1. Add the following actions:

Macro1 Actions

SelectObject action
Object Type: Table
Object Name: Table1
In Database Window: No

CopyObject action
New Name: Table2

3. Open and minimize the table Table1.
4. Select and run Macro1.

Result: Macro1 is copied to Table2, not Table1.

NOTE: If 'In Database Window' is set to 'Yes', everything works as expected.

For more information, search for "SelectObject Action" using the Help menu.

REFERENCES

=====

"Microsoft Access Language Reference: Part 2," version 1.0, page 431.

References

INF: Setup Options Dialog Box Lists Disk Space Info Twice

Article Number: Q92894
CREATED: 17-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The Setup Options dialog box will be displayed during a Custom setup or during an Express setup with insufficient disk space for a full installation. The bottom half of the Setup Options dialog box provides disk information (space required and space available).

The bottom left side provides disk information about the drive that you specified for Microsoft Access. The bottom right side provides disk information about the drive that contains your Windows software. If both Microsoft Access and Windows are located on the same drive, the disk information displayed on the left side is also displayed on the right side of the dialog box.

MORE INFORMATION

The Setup Options dialog box is a common dialog box. The same dialog box is opened for installations with Microsoft Access and Windows on the same drive, and Microsoft Access and Windows on different drives. In the former, the disk information for Windows is identical to that for Microsoft Access.

Steps to Reproduce Behavior

1. Run SETUP.EXE for Microsoft Access.
2. Specify an installation path that is on the same disk drive as your Windows directory.
3. Choose Custom Installation (Option 2).
4. The Microsoft Access Setup Options dialog box will open. The bottom half of the dialog box lists Space Required and Space Available information twice.
5. Cancel Setup.

The following steps are optional:

6. Run SETUP.EXE for Microsoft Access.
7. Specify an installation path that is on a different disk drive than your Windows directory.

8. Choose Custom Installation (Option 2).
9. The Microsoft Access Setup Options dialog box will open. The bottom half of the dialog box lists different disk information: one set of information for the Microsoft Access disk drive, and one set of information for the Windows disk drive.
10. Cancel Setup.

[References](#)

INF: Two Error Messages with Database on Workgroups Server

Article Number: Q92893
CREATED: 17-NOV-1992
MODIFIED: 14-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

If you are connected to a Microsoft Windows for Workgroups server and have a [database](#) from the server open on your workstation in Microsoft Access, you may be disconnected by the server and receive error messages from Microsoft Access.

The server can disconnect any client by using Net Watcher, a program that comes with Windows for Workgroups. If you are disconnected, you receive two separate error messages when you attempt to open a [table](#) in Microsoft Access. The first error message is the following:

Internal database error (-1106).

When you choose OK, you receive the second error message:

Disk or Network error.

You must reconnect to the server on which the database is stored in order to use the database.

More Information:

Steps to Reproduce Problem

1. Connect your machine (the client) to a Windows for Workgroups server.
2. Start Microsoft Access on the client and open a database file located on the server.
3. Start Net Watcher on the server.

NOTE: Net Watcher is located in the Accessories [group](#) in Windows Program Manager.

4. From the Connection menu, choose Disconnect to disconnect the client from the server.
5. In the [Database window](#), select a table and try to open it. From the client, the database appears to be open.

Result: You receive the error messages shown above.

References

PRB: Concatenation of Memo Fields Creates Text Field

Article Number: Q92892
CREATED: 17-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you use a Make Table [query](#) to concatenate two memo fields, the query concatenates the memo fields into a text [field](#) in the resulting [table](#). This will result in the truncation of memo field data in the new text field if you have over 255 characters in the memo fields.

CAUSE

Text fields have a limit of 255 characters, but memo fields can contain up to 32,000 characters.

RESOLUTION

A workaround is to perform an Append query to an existing table with a memo field and concatenate the two memo fields into the existing memo field.

MORE INFORMATION

The following are the [SQL](#) statements for the Make Table Query:

```
SELECT DISTINCTROW
    [Memo1]&[Memo2]
AS Concat
INTO NewTable
FROM OldTable;
```

The following are the SQL statements for the Update Query:

```
INSERT INTO ExistingTable
    [ExistingField]
SELECT DISTINCTROW
    [Memo1]+[Memo2]
AS Concat FROM OldTable;
```

In these sample SQL statements ExistingTable and ExistingField are the destination, and OldTable and Memo1 and Memo2 are the source.

The steps to generate the second set of SQL statements are:

1. Create a table with two memo fields:

Table: OldTable

Field Name: Memo1
Data Type: Memo

Field Name: Memo2
Data Type: Memo

2. Create another table:

Table: ExistingTable

Field Name: ExistingField
Data Type: Memo

3. Create a Query based on OldTable. From the Query menu choose Append (to ExistingTable). In the Query Grid place the following properties:

Query: AppendQuery

Field: Concat:[Memo1]+[Memo2]
Append To: ExistingField

This generates the update SQL statement shown above.

[References](#)

PRB: 'Invalid Range' Error When Importing Spreadsheet Range

Article Number: Q92891
CREATED: 17-NOV-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you try to import a range of cells from a spreadsheet, you receive the following error message:

Invalid Range.

The Select File [dialog box](#) appears after the error message.

CAUSE

You specified an invalid range of cells to import, either by improperly referencing them (for example, R0C0:R5C3) or by providing a nonexistent range name.

RESOLUTION

Reselect the file that you want to import, and then choose Import from the File menu. When the Import Spreadsheet Options dialog box is displayed, make sure you type a valid range reference in the Spreadsheet Range box. This range can be a specific range of cells (for example, A5:B12 or R3C3:R50C9) or a defined name that refers to a range of cells on the spreadsheet (for example, MYRANGE).

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 4, pages 72-75

[References](#)

INF: Mail Label Wizard Shows Blank Line in Label Size List Box

Article Number: Q92890
CREATED: 17-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

In the Mailing Label ReportWizard, the Label Size list box has a blank line at the bottom. This blank line shows on standard VGA and super VGA monitors but does not show on model 8515/A monitors.

[References](#)

INF: How to 'Freeze' a Record on a Form

Article Number: Q92889
CREATED: 17-NOV-1992
MODIFIED: 13-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article outlines a [method](#) for retaining the focus on a particular [record](#) so that any attempt to move out of that record fails, until an AfterUpdate [event](#) is triggered. This ensures that the record is updated before the user moves to the next record. You can then programmatically "unfreeze" the focus of the record from some event on the [form](#) to allow the user to move to another record.

MORE INFORMATION

1. Create a new [module](#) and add the following line to the <Declarations> section:

```
Option Explicit  
Global RecordIsLocked$
```

2. Add the following functions to the module:

```
Function LockRecord ()  
    If Len(RecordIsLocked$) Then  
        Screen.ActiveForm.Bookmark = RecordIsLocked$  
    End If  
    RecordIsLocked$ = Screen.ActiveForm.Bookmark  
End Function  
  
Function UnLockRecord ()  
    RecordIsLocked$ = ""  
End Function
```

3. Add the following entry to the OnCurrent event of the form which will utilize this functionality...

```
=LockRecord()
```

4. Add the following entry to the [control](#) or form event which is to "unfreeze" the record:

```
=UnLockRecord()
```

5. Save and Close the form and reopen it in Browse mode.

At this point, you will only be able to move to another record on the form if the "unfreeze" event has occurred.

[References](#)

PRB: Removing User Admin from Admins Group Can't Be Reversed

Article Number: Q92827
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you start Microsoft Access as the user "Admin" (which occurs by default) and then remove the user Admin from the Admins **group**, you will not be able to go back and make the user Admin a member of the Admins group during this session. This is by design.

CAUSE

As soon as you remove the user Admin from the Admins group, this change is written to the Microsoft Access system tables. At this point, the Admin user no longer has permission to add anyone to the Admins group, including itself.

RESOLUTION

To add the user Admin back to the group Admins, you must log on as another user who is in the Admins group.

MORE INFORMATION

Steps to Reproduce Behavior

1. Change the password for the user Admin from blank to some nonblank password. (NOTE: This will allow you to bring up the Log On dialog box at the start of the next session.)
2. IMPORTANT: Create a new user and add this user to the Admins group. (There must always be at least one user in the group Admins. Remember that you are currently logged in as the user Admin.)
3. Remove the user Admin from the Admins group.
4. Try to add the user Admin back to the Admins group. You will receive the error:

No permission for 'Admin'

If you leave the Security Menu and return to change the settings, you will receive the error:

'No Permission for MSysGroups'

5. Quit Microsoft Access and restart it.

The Log On [dialog box](#) will appear because of the change you made in step 1.

6. Log on as the user that you created in step 2.

7. Add the user Admin back to the Admins group.

NOTE: To disable the Log On dialog box, you must change the Admin password back to an empty [string](#) (do not enter any information in the New Password or Verify fields).

References:

"Microsoft Access User's Guide," version 1.0, Chapter 25,
"Administering a Database System."

For more information on Admin, search on "Admin account" and "password" using the Microsoft Access Help menu.

[References](#)

KBCategory:

KBSubcategory:

PRB: MS Access Short and Medium Time Formats Have Leading Zero

Article Number: Q92826
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you use the "Short Time" and "Medium Time" Date/Time formats, a leading zero appears in your data.

RESOLUTION

To eliminate the leading zero, use a custom format, such as "h:mm am/pm."

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a Date/Time field.
2. Format it as Medium Time.
3. Type "4:57pm" (without the quotation marks).
4. Press TAB to leave the field.

The value you entered will be displayed as "04:57 PM".

[References](#)

INF: How to Find Out If a Form Exists in a Database

Article Number: Q92825
CREATED: 16-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article discusses how to obtain a [dynaset](#) containing the names of all the forms in a [database](#) whether or not they are active.

This article assumes that the reader has some understanding of programming with [Access Basic](#) and using recordset (Virtual Table) objects.

MORE INFORMATION

If you want to obtain a complete list of forms whether or not they are active, you can create a dynaset that queries MSysObjects for all the forms stored in the [current database](#). MSysObjects stores the names of every object in a database, and each [record](#) includes the type of object represented by a code and stored in a [field](#) called Type. The code that represents forms in the Type field is -32768. Given this information, you can create a dynaset that obtains the names of all the objects in MSysObjects that are forms:

NOTE: In the following sample code, an underscore `_` is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Dim D As Database, S As Dynaset
Set D = CurrentDB()
Set S = DB.CreateDynaset("SELECT Name FROM MSysObjects_
                        WHERE Type = -32768;")
```

After executing the code above, a dynaset called S is created. This dynaset contains the names all the forms in the current user database. You can then process the information as shown in the example below. The procedure below accepts a [form](#) name and opens it if it exists in the database, or presents a message indicating that it does not exist in the database:

```
Option Explicit
Sub FormOpener (FormToOpen$)
    Dim D As Database
    Dim S As Dynaset
    Set D = CurrentDB()
    Set S = DB.CreateDynaset("SELECT Name FROM MSysObjects _
                            WHERE Type = -32768;")

    S.FindFirst "Name = '" & FormToOpen$ & "'"
```

```
If S.NoMatch Then
  MsgBox FormToOpen$ & " does not exist in this database"
Else
  DoCmd OpenForm FormToOpen$
End If
End Sub
```

[References](#)

INF: Custom Setup May Show Zero or Negative Disk Space Needed

Article Number: Q92824
CREATED: 16-NOV-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The amount of disk space required for each option does not display a consistent number when running a custom installation.

MORE INFORMATION

The setup program calculates and displays the total space that you will need, it does not just display a flat number. If you already have the same version installed the option will display a zero for required disk space. If a previous version of one of the options was much greater, the required disk space will display a negative number. An example of this would be if you added more data or objects to the NWIND.MDB. In the Sample Apps Disk requirement you would notice a negative number.

Steps to Reproduce Behavior

1. Run the setup program for Microsoft Access from Disk 1.
2. Choose Custom Installation.
3. Notice the required disk space for the options: some may be zero or negative.

References:

"Microsoft Access User's Guide," version 1.0, Introduction, "Setting Up and Starting Microsoft Access," pages vi-vii.

[References](#)

INF: Access Table with Time Field Exports Strangely to Excel

Article Number: Q92823
CREATED: 16-NOV-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

A Microsoft Access table that has a Time formatted field displays a strange date format when exported to Microsoft Excel. The Time column on the spreadsheet displays the time as the date 1/0/00. This behavior occurs because of the way Microsoft Excel interprets serial numbers. To display the time correctly, apply a Microsoft Excel time format to the affected information.

MORE INFORMATION

Microsoft Access and Microsoft Excel both use serial numbers to display dates and times. A Microsoft Access table that is exported to Microsoft Excel automatically formats all serial numbers as dates. The spreadsheet column of serial numbers simply needs the format changed to time. To correctly display a time format, highlight the column of dates on the spreadsheet. From the Format menu, choose Number, and select one of the time formats (for example, hh:mm:ss AM/PM). The time will now display correctly.

REFERENCES

=====

Microsoft Excel Help, "Serial Numbers," "Date and Time Numbers"

[References](#)

PRB: One Side of Join Line Not Attached to Table

Article Number: Q92822
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

It appears that the [join](#) line in the [QBE](#) (Query by Example) grid is attached to only one [table](#).

CAUSE

The table is joined to another table that is currently positioned out of view on the [Query window](#).

RESOLUTION

Try maximizing the Query window or using the scroll bars to find the other table, and then move the second table closer to the first table.

STATUS

This behavior is by design.

[References](#)

PRB: Unable to Specify Label Order in Graph Form Wizard

Article Number: Q92821
CREATED: 16-NOV-1992
MODIFIED: 09-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

The labels on a chart created using the Graph Form Wizard are not in order. No matter how you organize the underlying data, the labels always appear in alphabetical order.

CAUSE

Microsoft Access has no way of knowing whether a text [field](#) contains date information or regular text information. The Graph Form Wizard uses the [SQL](#) "Group By" clause, which defaults to alphabetically grouping text fields.

STATUS

This behavior is by design.

RESOLUTION

If you want to chart dates, store the underlying data for the [form](#) in date fields (as date/time data types). Then, when you tell the Graph Form Wizard to GROUP, Microsoft Access will know to [group](#) date fields chronologically.

If you want neither alphabetical or chronological grouping of labels, you will need to manually edit the ColumnHeads property for your [graph](#). This property is available when the form is in Design mode.

MORE INFORMATION

Steps to reproduce behavior

1. Create a [table](#) called Table1 with two fields: a text field called Text and a numeric field called Number.
2. Switch to [Datasheet view](#) and enter the following data in the order shown:

Text	Number
Jan	5
Feb	1

3. Use the Graph FormWizard to create a new form based on Table1

The labels on your chart are in alphabetical, not chronological order.

References:

For more information, search for "Parameter Query," then "GROUP BY clause (SQL)," then "graph labels," then "ColumnHeads" using the Help menu.

[References](#)

PRB: Pressing HOME Key in Text Box Doesn't Move to Beginning

Article Number: Q92819
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you type a great deal of text in a text box control, pressing the HOME key does not move you to the beginning of the text but does position the cursor at the beginning of the current line.

CAUSE

Microsoft Access allows a maximum line length of 20 inches. If your data exceeds this length, or if you have entered a carriage return (CTRL+ENTER) in the text box, pressing HOME will not move you to the beginning of the data.

RESOLUTION

In these cases, press CTRL+HOME instead of HOME.

MORE INFORMATION

Both text boxes bound to text fields and those bound to memo fields act the same with regard to the HOME and END keys.

[References](#)

PRB: 'No Default Printer' Error Message When Printing

Article Number: Q92818
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you create a new [report](#) and no default printer is set in the Windows Control Panel, you get the error message:

There is no default printer. Select or install one using the Windows Control Panel.

CAUSE

Reports have a "Layout for print" property that defaults to "Yes", which enables the "Size to fit" option when designing reports or labels. This option uses the printer font metrics defined in the printer driver. When there is no default printer, Microsoft Access cannot calculate the layout.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. In Windows Control Panel, highlight your default printer and choose Remove.
2. In Microsoft Access, choose Report, then New.
3. Choose either Report Wizard or Blank Report.

You receive the error message immediately when you use the Blank Report option. When you use the Report Wizard, you receive the message after answering all of the Report Wizard information. By choosing OK, you can continue creating the report.

[References](#)

INF: Initial Size of Every MDB File is 64K

Article Number: Q92817
CREATED: 16-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The initial size of every .MDB file is 64K. Microsoft Access allocates additional space in 16 page blocks (2K per page x 16 = 32K).

MORE INFORMATION

The initial size of every .MDB file is 64K, (32 pages). If more space is needed, the additional space is allocated in 16 page blocks. The reason for using 16 page blocks, as opposed to smaller blocks of disk space, is to prevent excessive disk fragmentation.

[References](#)

INF: Converting Julian Dates with Access Basic

Article Number: Q92816
CREATED: 16-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Most government agencies and contractors require the use of Julian dates. (A Julian date starts with a two-digit year, then counts the number of days from January 1, as printed on many desktop calendars.)

Microsoft Access does not have any built-in functions for using or converting Julian dates to standard dates. However, you can write a [function in Access Basic](#) to do this. Below is an Access Basic function which converts a Julian date to a standard serial date in Microsoft Access. Note that the following sample code assumes the year is in the 20th century. The code must be changed to process dates after the year 2000.

MORE INFORMATION

NOTE: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

Before attempting the examples below, create a new [module](#) and enter the following function:

```
Option Explicit
Function ConvertJulian (JulianDate as Long)
    ConvertJulian = DateSerial(1900 + Int(JulianDate / 1000),_
        1, JulianDate Mod 1000)
End Function
```

Example1

1. Create a new unbound [form](#) and add two unbound text boxes (called field0 and field2) to the form.

2. In the Control Source property of field2 type:

```
=ConvertJulian([field0])
```

3. Go to form view and enter 90001 into the first [text box](#). The Julian date is converted into a normal date and displayed in the second text box.

Example2

1. In Microsoft Access, create a table called Table1. Make one field named "Julian", data type "Number".

2. After saving the table, go to Datasheet View and enter the following numbers in the Julian field:

90001
90010
92100

3. Create a new query based on Table1 and drag the Julian field into the query grid.

4. In the Field row of another column of the query grid, enter the following:

Normal: ConvertJulian([Julian])

5. Go to Datasheet view. The Julian dates are converted into normal m/d/y dates.

[References](#)

INF: Using the Generic Print Driver with Forms/Reports

Article Number: Q92815
CREATED: 16-NOV-1992
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

One way to get Microsoft Access output from forms or reports into a text file is to use the Windows Generic print driver and print to a file. However, using the Generic print driver (TTY.DRV) can be a little tricky. Listed below are some hints on how to set up your forms and reports for the Generic print driver.

MORE INFORMATION

The Windows Generic print driver simulates a line-oriented printer with a fixed pitch font, like a TTY device or a line printer. Because of this, it can only print text, and it can only position the text on a grid that is 10 characters per inch in the horizontal direction and 6 lines per inch in the vertical direction. (By contrast, most Windows print drivers can position text on any pixel position.)

This means that any form or report using the Generic print driver to create a text file should be layed out so that everything lines up with this character grid.

Here are some considerations for the form/report layout:

- Ensure that each section of your report has a height that is an integral multiple of 1/6".
- Lay out the text items in your report such that each falls on 1/6" boundaries.
- Since the default hardware font for most printers is a Courier 10cpi font, you should set the font for all of your text items to be Courier 10.
- Do not use Bold fonts, as they will appear three times in your output. (The Generic driver simulates bold text by overstriking.)

Here are some problems that frequently occur when using the Generic print driver to do text output:

- Blank lines appear in the output -- text items or sections are slightly farther apart than 1/6".
- Overlapped lines appear in the output -- text items or sections are slightly closer together than 1/6"

- Text appears three times in the output -- text is formatted as
Bold.

[References](#)

PRB: No Error Msg When BoundColumn Greater Than ColumnCount

Article Number: Q92814
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The default value for the ColumnCount and BoundColumn properties of List Boxes and Combo Boxes is "1". If you change the BoundColumn property to a number that is greater than the number of columns (ColumnCount), you do not get an error message.

CAUSE

Microsoft Access does not enforce any error checking at this point of form or report design. Otherwise, you would always have to enter the correct column count before entering the bound column number.

RESOLUTION

When using List or Combo Boxes, check to ensure that the BoundColumn property setting is an integer between 0 and the number set by the ColumnCount property.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, pages 239-241

For more information, search for "ColumnCount" and "BoundColumn" using the Microsoft Access Help menu.

[References](#)

PRB: Column Widths in Table Design not Saved

Article Number: Q92813
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The browse mode column widths of the table are automatically saved by Microsoft Access. When working on tables in design mode, however, changes made to the design mode grid's column widths are not saved when the form is closed.

CAUSE

Microsoft Access does not currently support saving the column width of the design mode grid for a table.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open a table in design mode.
2. Adjust the column width of "field type" (or one of the other columns).
3. Exit the table.
4. Return to the table in design mode and notice that the width has returned to its original state.

[References](#)

PRB: Cannot Use DoMenuItem for Open Database or New Database

Article Number: Q92812
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you use the [macro](#) command 'DoMenuItem' to open or create a new [database](#), the database is not opened or created, and no error message is shown. Attempting to open a database brings up the File Open dialog box; however, the database you select in this dialog is not opened.

CAUSE

When the macro attempts to open/create another database, Microsoft Access tries to close the [current database](#), then detects that a macro is still executing from the currently open database and, as a result, does not close the database.

RESOLUTION

You can have only one database open at a time in the Database Window. (If you need to open more than one database at a time, you can use [Access Basic](#).) It is possible to close a database via a macro since the close is the final action in the macro. The database from which the macro is running would have to be closed before a new database is opened. The close action could not be followed by an open/create action because the database that contains the code to open/create has already been closed.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create the following macro:

Action

DoMenuItem

Arguments

Menu Bar: Database

Menu Name: File

Command: New Database

2. Run the macro. No database is opened and no error message is given.

[References](#)

PRB: Can't db.OpenTable With Only Read Definition Permission

Article Number: Q92811
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you have only Read Definition permissions on a table, you are unable to open the table in order to examine the table definition from Access Basic. The following error message appears:

```
No permission for 'Tablename'
```

CAUSE

The table cannot be opened from Access Basic when you have only Read Definition permission. You should give users Read Data permission for the table if they need to examine the table definition via Access Basic.

A user with Read Definition permission can manually only look at the table definition by opening the table in design mode as read only via the User Interface (UI).

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce the Problem

1. Open a new database and create a table called Table1.
2. Add a new user named "Joe" and give this user Read Definition permission only on Table1.
3. Exit Microsoft Access, restart Microsoft Access and logon as Joe. (Be sure you have removed all permissions from the group Users since everyone who logs on to Microsoft Access is a member of the group Users.)
4. Enter the following code in a new Module:

```
Option Explicit
Sub Test ()
    Dim Mydb As Database
    Dim MyTable As Table
    Set Mydb = CurrentDB()
    Set MyTable = db.OpenTable("Table1")
```

```
MyTable.MoveFirst
MyTable.Close
Mydb.Close
End Sub
```

5. Run this sub by typing 'test' in the immediate window. The following error message is displayed:

No permission for 'Table1'.

[References](#)

PRB: BeforeUpdate Macro Executing Twice

Article Number: Q92810
CREATED: 16-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The BeforeUpdate [macro](#) appears to be executing twice, rather than only once.

CAUSE

The BeforeUpdate macro will execute whenever an edited [record](#) is saved. And if you add records using the Paste Append command, it will execute when each of those records is added. If you find that the BeforeUpdate macro is executing more than once, check to see exactly what actions you are taking.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a macro with a MsgBox action and then a CancelEvent action.
2. Save this macro as Macro1.
3. Create a bound [form](#) with text boxes.
4. Set the form's BeforeUpdate property to Macro1.
5. Switch the form from Form [Design view](#) to [Form view](#).
6. Enter data in a [field](#), then select the data in that field.
7. From the Edit menu, choose Copy.
8. From the Edit menu, choose Paste Append.
9. Choose the OK button when the Message box (that you created in Macro1) appears.
10. Close the form.
11. Choose the OK button when the Message box (that you created in Macro1) appears.

12. Choose the OK button when the following error message appears:

The record being edited can't be saved. If you close the form, the changes you've made to the record will be lost. Close anyway?

Note that the message box you created in Macro1 reappeared in step 11. The macro that you assigned to the BeforeUpdate property is executing twice: when the record that you added using the Paste Append command is saved, and when when the edited record is saved.

References:

For more information, search for "BeforeUpdate, AfterUpdate Properties" using the Microsoft Access Help menu.

[References](#)

INF: RunApp Action Command has 255 Character Argument Limit

Article Number: Q92809
CREATED: 16-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The RunApp action for macros allows 255 characters.

MORE INFORMATION

MS-DOS has a 127 character limit on command line arguments, but Microsoft Access allows up to 255 characters for the RunApp macro action. This is by design, because there is a 127 character limit for the program name and path, but there is a 127 character limit for the list of arguments to the program. The two argument lists add up to the allowed 255 characters.

[References](#)

INF: Hints for Printing Mailing Labels

Article Number: Q92807
CREATED: 16-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The Microsoft Access ReportWizard for mailing labels makes it quick and easy to create a [report](#) design for many standard mailing labels. However, sometimes you may have difficulty getting the labels to print correctly on your printer. Here are some hints and suggestions for getting labels to print.

MORE INFORMATION

The Microsoft Access ReportWizard for mailing labels creates report specifications for standard mailing labels. However, in order to do so, it makes some assumptions about your printer which may not be true. In particular, it assumes that it can print within a certain distance of each edge of the paper. Every Windows print driver can have a different "printable region," which is the rectangle on the printed page that the application can print into. This rectangle may be smaller than the rectangle that the physical printer can print, due to space and efficiency considerations in the print driver. Therefore you may not be able to print labels as close to the edge of the paper as you would like.

In general, the ReportWizard assumes that it can print within .25 inch of the edge of the paper. If you find that some of the text in your labels is getting cut off or is not getting printed at all, try going into design mode for the report and moving text items until all of the text prints. For example: Select all the controls and move the entire [group](#) down a little within the Detail section to adjust for a top [label](#) being missed. You may have to make the items smaller, choose a smaller font, and move them closer together to get everything to fit without having the label cut off at either the top or the bottom of the printed page.

If your printer has such a large top margin that you can't move your text items down far enough (or if doing so results in a visually crowded label), the next thing to do is change the Print Setup margins so that the first label on each page is not used. To do this, choose Print Setup and add the height of your labels to the top margin provided by the ReportWizard. For example, if you are using 1 inch labels, and the top margin is .03 inch, change it to 1.03 inches to skip the first label on the page. If you are having trouble with the last line of the last label getting cut off, you can use the same technique on the bottom margin to cause the last label on the page to be skipped.

Another common problem is that some labels come on sheets that are 12 inches tall, whereas the print driver only knows about 11 inch paper. In this case, you must ignore the last label on the page and use the labels that fall within the first 11 inches.

[References](#)

INF: Count() Ignores Null Values

Article Number: Q92747
CREATED: 15-NOV-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

NULL values are ignored when using a [query](#) to perform a count.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create Table1 with two text columns as follows:

Column1	Column2
foos	
ball	
foosball	junk
null	
notnull	junk

2. Open a new query based on Table1.

3. From the View menu, choose [SQL](#). Enter the following SQL query:

```
SELECT Column2, COUNT(Column2)
FROM Table1
GROUP BY Column2;
```

4. Run the query. The result of the query is:

Column2	Count(Column2)
	0
junk	2

Workaround

To count NULL values, count on a [primary key column](#), a column that contains unique values, a counter column, or a dummy column where all records equal 1.

The new SQL statement would read:

```
SELECT Column2
COUNT(PrimaryKey)
```

```
FROM Table1  
GROUP BY Column2;
```

References:

Online Help, search on "Count" and select the topic "Calculating Totals in a Query" and "Count Function" for more information.

[References](#)

PRACC9211: Table Export to Excel Does Not Set Database Name

Article Number: Q92746
CREATED: 15-NOV-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

SYMPTOMS

When you export a [table](#) from Microsoft Access to Microsoft Excel, the defined name "Database" is not set in Excel.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem was corrected in Microsoft Access version 1.1.

MORE INFORMATION

To use the exported data as a Microsoft Excel [database](#), the database range must be set manually in Excel. This can be done by highlighting the entire set of data, including the first [row](#) of labels, and choosing Set Database from the Data menu.

For additional information on interaction between Microsoft Access and Microsoft Excel, [query](#) here in the Knowledge Base for the following keywords:

access and excel and database and biff

NOTE: BIFF is the Microsoft Excel Binary Information File Format.

[References](#)

PRB: No Error Given When Linked OLE Object Can't be Found

Article Number: Q92745
CREATED: 15-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Microsoft Access does not warn you if you close an [OLE](#) server without saving a linked embedded object.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Start Microsoft Excel.
2. In SHEET1 enter data in several cells, select and then copy them to the clipboard.
3. Start Microsoft Access.
4. Open a new [form](#) and choose Paste Link to create a linked embedded object.
5. Switch back to Excel and close SHEET1. DO NOT save the changes.
6. Switch back to Access and double-click on the linked embedded object. Excel comes to the foreground; the hourglass is displayed; and Excel is then closed. There is no error message displayed.

[References](#)

INF: Planning Database Security

Article Number: Q92742
CREATED: 15-NOV-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There are two things to remember when dealing with security:

1. Plan your security scheme from the beginning.

Planning a good security scheme from the beginning will ensure you a minimum of work in the future.

2. Assign permissions to Groups, not to Users.

This is very important. By creating appropriate, well thought out groups, a user has the "best" permissions of the groups that (s)he is a member of. For example, if a user is member of Group A, which does not have permissions to an object but also is a member of Group B, which does have the permissions to an object, the user will have Group B's permission to the object.

You will find that maintenance work will decrease significantly if you plan your groups carefully. A typical database setup should have a maximum of 4 to 5 groups. This is not a maximum number of groups limitation for Microsoft Access, merely a guideline for easier security management.

At this time there is no multi-selection functionality in the security dialogs. There is no way can you give a group or user permissions to more than one object at the time; each object must be selected individually and then permissions may be assigned.

MORE INFORMATION

User Name Case Sensitivity

User and group names are not case sensitive when you enter them to start a Microsoft Access session. However, if you are recreating a user account or group you have to match the name exactly. This is due to the way the account information is stored by the system.

NOTE: Passwords are case sensitive.

Reference:

"Microsoft Access User's Guide," version 1.0, chapter 25,

"Administering a Database System."

[References](#)

PRB: Choosing a New Printer Resets Some Printer Settings

Article Number: Q92741
CREATED: 15-NOV-1992
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Microsoft Access retains certain printer settings with Form and Report objects. However, if you select a new printer in the Print Setup dialog your Orientation, Paper and Option settings change to the default values.

CAUSE

This functionality is controlled by the Windows Common Dialog DLL (Dynamically Linked Library.)

STATUS

This behavior is by design.

[References](#)

PRB: Help Gives Error Importing File From Floppy Disk

Article Number: Q92740
CREATED: 15-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Pressing F1 for help, after importing a file from a floppy disk drive, results in a System error:

Cannot read from drive A:.

CAUSE

This problem is caused by a bug in the Microsoft Windows Help engine (WINHELP.EXE) and, as a result, manifests itself in several products, such as Microsoft Word for Windows.

RESOLUTION

To avoid this problem, change the selected drive in the import dialog box back to a fixed-disk drive before closing the [dialog box](#).

STATUS

Microsoft has confirmed this to be a problem in WINHELP.EXE. We are researching this problem and will post new information here as it becomes available.

MORE INFORMATION

Steps to Reproduce Behavior

1. Import a [database](#) file from a file on floppy disk (such as drive A:).
2. Close the Import dialog box.
3. Remove the disk from drive A:.
4. Press F1 for help. You receive the error message:

Cannot read from drive A:

[References](#)

PRB: Out of Disk Space Error Makes Paradox Table Unusable

Article Number: Q92739
CREATED: 15-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you are editing a Paradox [table](#) in Microsoft Access and run out of disk space, the following error message appears:

Out of Disk Space

You continue to receive the same error message, even after you free up additional disk space.

RESOLUTION

Use the following steps to correct this problem:

1. Delete the attachment to the remote Paradox table by highlighting the table name in the table list and pressing the DEL key.
2. Rename both the table (xxx.DB) and its [index](#) (xxx.PX) to some other names.
3. Recreate the attachment, then try attaching to the remote Paradox table again.

Paradox is manufactured by Ansa Software, a Borland company, a vendor independent of Microsoft; we make no warranty, implied or otherwise regarding this product's performance or reliability.

[References](#)

INF: Form View Allowed When ViewsAllowed Set to Datasheet

Article Number: Q92709
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you create a [form](#) and set the ViewsAllowed property to Datasheet, you can still open the form in Form View.

MORE INFORMATION

The DefaultView property in Microsoft Access specifies how to open the form, while the ViewsAllowed property determines what you can do once the form is opened. ViewsAllowed determines whether users can switch between Datasheet View and Form View by choosing either the Form or Datasheet command from the View menu.

[References](#)

INF: CueCards Started Outside Access Does Not Work

Article Number: Q92706
CREATED: 12-NOV-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you select the Microsoft Access Help Icon without having Microsoft Access started and then try to run any topic from the Cue Cards menu, you are returned to the Program Group without any error message.

MORE INFORMATION

This is a limitation of WinHelp's ability to call and respond to DLLs.

Steps to Reproduce Behavior

1. From Microsoft Access Program Group open Microsoft Access Help Icon
2. Select Cue Cards.
3. Select any Cue Cards topic.

Result: You are returned to the Microsoft Access Program Group without any error message.

[References](#)

PRB: Alt+ Any Key Combination Does Not Work With Active Popup

Article Number: Q92703
CREATED: 12-NOV-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When a Microsoft Access popup window is active in Form Browse, pressing any ALT+key combination to access the menu bar does not work.

RESOLUTION

Use the mouse instead of the keyboard to access the menu bar.

MORE INFORMATION

Currently, when a popup window is active, pressing any of the ALT+key combinations will not pass any function to the system menu bar, only to the current active window.

Steps to Reproduce Behavior

1. Run Microsoft Access and open NWIND.MDB.
2. Select the Forms command from the View menu.
3. Open the "Add Products" form into Design mode.
4. Select the Properties command from the View menu. The property sheet will be displayed.
5. Pressing ALT+F to access the menu File option does not work.

[References](#)

INF: Status Bar Text Too Long for Auto Repeat Property

Article Number: Q92702
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you select the Auto Repeat property for a push button, the text that appears in the status bar can get cut off if you are using a display mode that is less than VGA. The text that appears in the status bar should read:

Run macro specified in OnPush property repeatedly while button is pressed?

[References](#)

PRB: Disabled Buttons Not Visible with ATI 1280x1024

Article Number: Q92701
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When using an ATI Ultra Card with the ATI 1280x1024 driver, disabled buttons within Microsoft Access are not visible. This does not apply to push buttons that you create on a [form](#), only with buttons that are part of Microsoft Access.

RESOLUTION

Using a video driver supplied with Microsoft Windows will remedy this problem.

MORE INFORMATION

This occurs only with Microsoft Access buttons and not with user-created buttons because the "text" of the two types of buttons are different. With Microsoft Access buttons, the text is actually text and with user-defined buttons the text is actually a [bitmap](#).

The ATI Video driver is manufactured by a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

PRB: Fields on Popup not Editable when Access is Minimized

Article Number: Q92695
CREATED: 12-NOV-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you minimize the Microsoft Access program window and a Microsoft Access popup [form](#) is displayed, you are unable to place a cursor in an editable [field](#) on the popup form, and as a result, cannot edit its contents.

STATUS

Microsoft is researching this problem and will post new information here as it becomes available.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a form with a Text Box and a Command Button on it, set the form's Popup property to Yes, and save the form as MinPop.
2. Create a [macro](#) with the following macro actions (save this macro as MinPop):

 Actions

 SendKeys

 Keystrokes: "% n"

 Wait: Yes

 OpenForm

 Form: MinPop

3. Run the macro MinPop.

When you run the macro Minpop the Microsoft Access program window is minimized and the MinPop form is displayed. When you click on the text box to edit or tab to it, no cursor is displayed and you are unable to edit or enter data in the [text box](#). If you restore the Microsoft Access program window while the MinPop form is displayed, you then are able to click on the text box, see the cursor, and edit the field.

References:

"Microsoft Access Language Reference" version 1.0, pages 348-351, and 432-435.

[References](#)

INF: Using TransferDatabase to Export to Btrieve Format

Article Number: Q92694
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

According to the online Help for the TransferDatabase Action, it is not necessary to have a Database Name entry in order to export to Btrieve format. This is incorrect.

The third paragraph under "Database Name" should read:

For Btrieve databases, enter the path and file name (FILE.DDF) of the Xtrieve dictionary file for the Btrieve database. If the path you specify doesn't contain an Xtrieve dictionary file, Microsoft Access will create a dictionary file. This is a required entry. If you don't know the location of an Xtrieve dictionary file, enter a space <using the spacebar> and Microsoft Access will create this file for you.

MORE INFORMATION

You must physically enter either a space or an invalid path before Microsoft Access will create a dictionary file for you. An Xtrieve dictionary file is required when exporting a table to Btrieve format.

[References](#)

INF: Inconsistent Save and Save As Menus with Forms

Article Number: Q92693
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

With a Form open in Design view, the File menu has options for Save and Save As. With a Form open in Form view or Datasheet view, the File menu has options for Save Form and Save Form As.

MORE INFORMATION

These options are available so that the user will be less likely to think that saving the form while in Form view or Datasheet view will save the actual data displayed at that time. While in Design view, no data is displayed, and it is clear that saving the form will not save any data from the form's dynaset. In all the cases, only the information which defines the form itself will be saved by choosing either save option from the File menu.

[References](#)

INF: Copying Columns In Microsoft Access

Article Number: Q92692
CREATED: 12-NOV-1992
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

In Microsoft Access Table Datasheet Browse/Edit view, you cannot select a [column](#) of data, copy it to the clipboard, and then edit it, as you can with Microsoft Excel. You can, however, change or rearrange data without copying it to the clipboard.

This article describes procedures for the following three methods of manipulating data in a Microsoft Access [table](#):

- Rearranging the column order
- Adding a column
- Copying a column of data to another application

REARRANGING THE COLUMN ORDER

=====

Use the following steps to rearrange the order of the columns in a Microsoft Access table:

1. Use the column indicator to select the column.
2. Drag the column to the desired position, and then release it.

For more information, search for "Column Width (datasheets)," then "Changing a Datasheet's Appearance" in Microsoft Access online help.

ADDING A COLUMN

=====

This section describes how to add a column of information from one Microsoft Access table to another.

The first procedure describes how to [update](#) information in Table1 with information from Table2. The second procedure describes how to create two tables (MASTER and EXTRAINFO), and then add the information from EXTRAINFO to the MASTER table.

To add a column to a Microsoft Access table, you must run an [update query](#). If you are also adding new records, you must run an [append query](#) (refer to step 3, in either procedure) before you run the update query.

Copying a Column from One Table to Another

Update information in Table1 with information from Table2 as follows:

1. Set the same primary key for both Table1 and Table2.
2. Manually add the new column names from Table2 to Table1. Make sure they are the same data type. They do not have to be the same name as the corresponding fields in Table2, but it saves a lot of typing if the names are the same.
3. Use Table2 to create an append query.

NOTE: Skip this step if Table2 contains only records that exist in Table1.

- a. Select the Query button, then select New. Add Table2 to the query, then close the Add Table dialog box.
 - b. From the Query menu, choose Append.
 - c. In the Query Properties box, specify Table1 as the Append To Table Name. Make sure that the Unique Values checkbox is selected, then choose the OK button.
 - d. Add all the field names from Table2 to the QBE grid.

If the field names are spelled correctly, Microsoft Access automatically fills in the Append To field with the name of the correct field from Table1. If not, you must manually select the Table1 field name in the Append To field.
 - e. Run the query. New records from Table2 are added to Table1.
4. Create an update query that contains both Table1 and Table2.
 - a. Select the Query button, then select Update. Add both Table1 and Table2 to the query, then close the Add Table dialog box.
 - b. Join the two tables by dragging the key field from Table1 to Table2.
 - c. From the Query menu, choose Update.
 - d. Drag the new fields from Table1 and drop them into the QBE grid, then type the [Table2]![<fieldname>] into the Update From field. (You must manually type this entry.)
 - e. Run the query to update the new fields in existing Table1 records.

Creating Two Tables and Adding Information from One to Another

1. Create two tables called MASTER and EXTRAINFO. The MASTER table has two fields: KEY and ITEM1. The EXTRAINFO table has two fields: KEY

and ITEM2. Add data as follows:

MASTER:	KEY	ITEM1	EXTRAINFO:	KEY	ITEM2
	1	AAAAA		1	11111
	2	BBBBB		3	33333

2. Open MASTER in Design view and add the field ITEM2. It should look like this:

MASTER:	KEY	ITEM1	ITEM2
	1	AAAAA	
	2	BBBBB	

3. Create the append query as follows:
 - a. Select the Query button, then select New. Add EXTRAINFO to the query, then close the Add Table dialog box.
 - b. From the Query menu, choose Append.
 - c. In the Query Properties dialog box, select MASTER as the Append To Table Name. Select the Unique Values check box, then choose the OK button.
 - d. Drag the KEY field and the new field, ITEM2, and drop them into the QBE grid. Microsoft Access fills in the Append To field names.
 - e. From the Query menu, choose Run.
 - f. When Microsoft Access displays the following message, choose the OK button:

2 rows will be appended.

- g. When Microsoft Access displays the following message, choose the OK button:

1 record(s) were lost due to key violations

NOTE: This is expected; the record with KEY=1 couldn't be appended. The update query (step 4) will copy this record.

MASTER should now look like this:

MASTER:	KEY	ITEM1	ITEM2
	1	AAAAA	
	2	BBBBB	
	3		33333

4. Create the update query as follows:
 - a. Select the Query button, then select Update. Add both EXTRAINFO and MASTER to the query, then close the Add Table dialog box.
 - b. Join the tables by dragging the KEY field from EXTRAINFO to MASTER. A line will connect the two tables.

- c. From the Query menu, choose Update.
- d. Drag the new field, ITEM2, from MASTER and drop it into the QBE grid. Then, type [EXTRAINFO]![ITEM2] in the Update From field.
- e. From the Query menu, choose Run.
- f. When Microsoft Access displays the following message, choose the OK button:

2 rows will be updated.

MASTER should now look like this:

MASTER:	KEY	ITEM1	ITEM2
	1	AAAAA	11111
	2	BBBBB	
	3		33333

For more information on append and update queries, search on the following topics in Microsoft Access online help:

- "Appending Records to a Table"
- "Update Queries, Updating Records as a Group"
- "Joins, Joining Tables in a Query"

COPYING A COLUMN OF DATA TO ANOTHER APPLICATION

=====
If need to copy a column of data from Microsoft Access to another application, such as Microsoft Excel, try one of the following methods:

- Use Copy & Paste to transfer the records you want to copy. Then delete any unwanted columns in the other application.
- Use the Hide Columns command from the Layout menu to suppress the columns that you do not want copied. Then use Copy & Paste to transfer the records that you want to copy. You must Unhide each column individually.
- Create a Query that contains only the columns that you want. Then use Copy & Paste to transfer the records that you want to copy. This [method](#) is useful if you plan to copy data on a regular basis.

For more information, search for "Copy Command" in Microsoft Access online help.

REFERENCES

=====
For more information when rearranging the order of columns, read "Changing Column Width and Order" in Chapter 14 of the "Microsoft Access User's Guide" for version 1.0 or 1.1.

For more information when adding columns to tables, read "Creating Multiple-Table Queries" in Chapter 6, and "Appending Records" and "Updating Tables" in Chapter 7 of the "Microsoft User's Guide for version 1.0 or 1.1."

[References](#)

INF: Unsupported SQL Keywords Not Listed in Access

Article Number: Q92691
CREATED: 12-NOV-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

In Appendix C of the "Microsoft Access User's Guide", there is a list of all [SQL](#) keywords that are supported in Microsoft Access version 1.x. However, there is no list of SQL keywords that are not supported in Microsoft Access version 1.x.

MORE INFORMATION

There are over 200 ANSI SQL keywords that Microsoft Access does not support. Rather than list them all, only the supported keywords are listed in Appendix C.

References:

"Microsoft Access User's Guide," version 1.0, Appendix C, "Expressions", pages 649-650, 655-656.

[References](#)

INF: Limit of AND's in SQL Select Statement

Article Number: Q92690
CREATED: 12-NOV-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There is an undocumented limit on the number of ANDs you can use in a [SQL](#) Select statement. The limit is 40 for the Where clause and 40 for the Having clause. Microsoft Access returns the following error message when you exceed this limit:

Out of Memory

References:

For more information on limits, search on the following word here in the Microsoft Knowledge Base:

limitations

[References](#)

INF: Queries Sort by Order of IN() Parameters

Article Number: Q92689
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

If you use the IN() operator in an unsorted query, the output will be grouped on the associated field. In addition, Microsoft Access will place the groups in the same order as the parameters are listed in the IN() operator.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a table with Firstname and Lastname fields (data type "text"). Enter the data listed below:

JOE	SMITH
JANE	DOE
ABE	LINCOLN
JOHN	DOE

2. Create a query on this table. In the Criteria cell for Lastname, type:

```
IN("LINCOLN","DOE")
```

The query will generate these records:

ABE	LINCOLN
JANE	DOE
JOHN	DOE

You can have the records appear in a different sequence by changing the order of the parameters in the IN() operator, or you can explicitly specify a SORT order.

[References](#)

PRB: Report or Form Bound to a Table Locks the Table

Article Number: Q92687
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You may receive the following message when you try to open a [table](#) in [Design view](#):

Can't open table 'table name' for modifications. Open it [read-only](#)?

This message occurs by attempting to open a table in design view when the table is the [record](#) source for a [form](#) or [report](#), and one of the following conditions is also true:

1. A report is opened in Print Preview or a form is opened in Browse view that has the same record source, or is based on a [query](#) with the same record source.
2. A form is opened in Browse view first and then switched to Design view.
3. A report is opened in design view, then switched to Print Preview and back to design. (You cannot switch to design view if you opened the report in Print Preview first.) This message does not occur when opening the form or report in Design view first, (without ever switching to Browse view or Print Preview), and then opening the table in Design view.

STATUS

This behavior is by design.

MORE INFORMATION

You must close the Form or [Report window](#) to be able to open the table in Design view to allow changes. Search in Help for "error messages: reference" for more information regarding this error.

[References](#)

PRB: Opening a Database Read Only Forces Others to Read Only

Article Number: Q92686
CREATED: 12-NOV-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you open a [database](#) as Read Only, all other users attempting to open the same database as Exclusive are forced to open it Read Only also. This applies to multiple copies of Microsoft Access running on one workstation as well as the multi-user environment.

RESOLUTION

To avoid opening the database in Read Only mode, do not select Exclusive when opening the second instance of the database.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open Microsoft Access and open the NWIND database with Read Only selected and Exclusive not selected.
2. Minimize the first instance of Microsoft Access and start a second instance of Microsoft Access.
3. In the second instance of Microsoft Access, open the NWIND database with Exclusive selected and Read Only not selected.

Result: the database opens as Read Only, even though the Read Only flag wasn't selected.

[References](#)

PRB: GoToRecord Macro Doesn't Work on Subform

Article Number: Q92685
CREATED: 12-NOV-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you create a GoToRecord macro that points to a subform and run that macro from the main form, you get the following error message:

```
Object 'Subform Name' isn't open
```

where 'Subform Name' is the subform that you specified in the GoToRecord macro.

STATUS

This behavior is by design.

RESOLUTION

Use a GoToControl macro and specify the subform, then use the GoToRecord action to specify the specific record within the subform.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open NWIND.MDB and create the following macro:

Macro Name	Condition	Action

Macro1		GoToRecord

Macro1 Actions

```
GoToRecord
  Object Type: Form
  Object Name: Categories Subform
  Record: Next
```

2. Open the Categories form and from the File menu, choose Run Macro, then select Macro1. You will get the following error message:

```
Object 'Categories Subform' isn't open
```

3. Now change Macro1 to the following:

Macro Name	Condition	Action
------------	-----------	--------

```
-----  
Macrol                               GoToControl  
                                       GoToRecord
```

```
Macrol Actions
```

```
-----  
GoToControl  
  Control Name: Categories Subform
```

```
GoToRecord  
  Record: Next
```

Run the macro in the same way as above. The macro will go to the second record in the Categories subform.

[References](#)

INF: How to Change the Caption on the Access Window

Article Number: Q92684
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article shows you how to change the default title bar of the Microsoft Access window with a function called from a macro.

This macro can also be named "Autoexec", which would then change the window caption automatically whenever this database is opened.

MORE INFORMATION

To create this function:

1. Open a new database and name it TEST.MDB.
2. Open a new blank form. Place at least one control on the form. Close and save the form as "MainForm".
3. Open a new macro and enter the following action (Close and save the macro as "Macro1"):

Action	Argument
RunCode	ChangeAccessCaption("<Your Custom Caption Here>")

4. Open a new module, and enter the following declarations:

```
Option Explicit
Declare Function GetParent% Lib "USER" (ByVal hwnd%)
Declare Sub SetWindowText Lib "USER" (ByVal h%, ByVal lpstr$)
```

5. This will allow you to enter the code for the ChangeAccessCaption function. Enter the following:

```
Function ChangeAccessCaption (Caption$)
    Dim X%
    Dim hParent%

    ' Here's an example of how to set the caption bar in Access.
    DoCmd OpenForm "MainForm"           'Open the main startup form...
    X% = Forms!MainForm.hwnd            'Get its handle...
    hParent% = GetParent(GetParent(X%))  'Get the forms parent...Access
    Call SetWindowText(hParent%, Caption$)

End Function
```

6. Save the module as "Module1".
7. After closing and saving the module, run Macro1. The Window title for the Microsoft Access window should change to the text you specified as the function parameter in step 3.

[References](#)

INF: '#Name?' or Control is Blank on Form Based on a Query

Article Number: Q92683
CREATED: 12-NOV-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

If you have a [form](#) based on a [query](#), and the query contains multiple fields with the same name (such as FieldA from Table1 and FieldA from Table2), the [control](#)(s) on the form that are bound to that [field](#) name will be blank or may display the error "#Name?".

MORE INFORMATION

Because the same field name appears multiple times in the query that the form is based on, Microsoft Access does not know which field to [link](#) to the control(s). Therefore, instead of making an assumption about which field to bind to, it leaves the control(s) linked to duplicate field names blank or displays the error "#Name?" in the control.

To correct this problem, rename the field in the query.

To change a field name in a query:

1. In a query's [Design view](#), place the insertion point in front of the first letter of the field name in the [QBE](#) grid.
2. Type the new name followed by a colon. Delete any names assigned by Microsoft Access, but don't delete the field name or [expression](#).

NOTE: Changing a field name in a query's Design view changes the heading in the query's [Datasheet view](#) and the field name in a Form or Report based on the query. However, the underlying field name in the [table](#) doesn't change.

Reference(s) :

"Microsoft Access User's Guide," version 1.0, page 103.

[References](#)

INF: Parameter Queries, Wildcards, and SQL Server

Article Number: Q92682
CREATED: 12-NOV-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The asterisk (*) and the question mark (?) wildcard characters are not valid in [SQL](#) Server parameter queries and thus are interpreted by SQL Server as text. Unless these characters are present in the [field](#) restricted by the parameter, the [query](#) will return an empty set.

Instead of using the asterisk (*) and the question mark (?) wildcard characters in parameter queries, use the percent sign (%) and the underscore (_), the SQL Server wildcard characters.

MORE INFORMATION

SQL Server uses the percent sign (%) and the underscore (_), the ANSI standard wildcard characters, whereas Microsoft Access uses the asterisk (*) and the question mark (?) wildcard characters. When executing a [parameter query](#), Microsoft Access does not translate the wildcard equivalents; the [string](#) you type in the Query Parameters [dialog box](#) is passed verbatim to SQL Server. Microsoft Access translates only strings that are part of the [QBE](#) grid.

References:

For more information, search for "parameter query" using the Help menu.

[References](#)

INF: Defragment and Compact Database to Improve Performance

Article Number: Q92681
CREATED: 12-NOV-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Compacting your [database](#) and defragmenting your hard drive will improve the performance of Microsoft Access.

MORE INFORMATION

Since the data on your hard disk becomes fragmented over time, you should run a disk defragmentation utility (or "defragmenter") periodically. Within the file, portions of the database may also become fragmented due to Microsoft Access. Because of this, you should also run Compact from within Microsoft Access.

A disk defragmenter will place the database file in contiguous clusters on your hard disk, making file access quicker in general. If you do not defragment your hard disk, the time it takes for MS-DOS to retrieve your file may increase since it must go to several physical locations on the disk to retrieve the entire file.

The Compact option under the File menu will also aid in increasing the performance of Microsoft Access. Compact makes a copy of the database, rearranging how the database file is stored on disk if it is fragmented.

File storage can become fragmented when you make many changes to a database. The compacted database file is usually smaller. You can use the same name as the original for the compacted database file, or you can use a different name to create a separate file. If you use the same name and the database is compacted successfully, Microsoft Access automatically replaces the original file with the compacted version.

Limitations of Compact

- The compact operation succeeds only if you have enough storage space on your disk for both the original and compacted versions of the database.
- You cannot compact a database when it is open. In a multi-user environment, the compact operation fails if another user has the database open.

Fragment or Compact First?

If you compact after running a defragmenter, you would theoretically leave open disk space immediately after the .MDB file on the disk, thereby allowing MS-DOS to place any additional information in the next physical clusters (this would be very fast). However, if you defragment after running Compact, your .MDB might be placed on the first part of the disk followed by the rest of your files, with no open disk space until the end (the inside tracks) of the disk. This would result in somewhat slower disk access.

[References](#)

INF: Currency is Significant to Four Digits

Article Number: Q92680
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When using the currency [data type](#) within [Access Basic](#), be aware that numbers are significant to four digits to the right of the decimal point.

MORE INFORMATION

The reason for this is that most financial information, such as mortgage rates and stocks, require four digit precision.

Currency variables are stored as 64-bit numbers (8 bytes) in a two's complement integer format and scaled by 10,000 to give a fixed-point number with 15 digits to the left of the decimal point and 4 digits to the right. This representation provides a range of -922,337,203,685,477.5808 to 922,337,203,685,477.5807. The Currency data type is extremely useful for calculations involving money and for fixed-point calculations in which accuracy is particularly important.

[References](#)

PRB: Out Of Memory When Pasting Text Into Module

Article Number: Q92679
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

An "Out of Memory" message appears when pasting in more than 64K of ASCII text from the [Clipboard](#).

CAUSE

The implementation of the Clipboard paste in Microsoft Access cannot paste more than 64K of ASCII text. This message appears even if the code translates into less than 64K of pseudocode.

RESOLUTION

Paste the text into the [module](#) in more than one step. Each paste must contain less than 64K of text.

[References](#)

PRB: Access Trims Spaces Off Database Names

Article Number: Q92678
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When naming a [database](#) that has one or more spaces at the front of the database name, Microsoft Access strips off the leading spaces.

CAUSE

By default, the Windows Common Dialog strips leading spaces off names.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new database.
2. Give the database a name that begins with a space.
3. Save the database.
4. Reopen the database. The space has been stripped off the name.

[References](#)

PRB: Scrolling to Bottom of Module Window Goes Past Typed Info

Article Number: Q92677
CREATED: 11-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In a Module window, moving the vertical scrollbar thumb control all the way to the bottom takes you past the end of your program. You must manually scroll back up to locate the end of your code.

CAUSE

The scrollbar indicates your position relative to the module's data boundaries, rather than your position relative to the current information in the module. Therefore, it is likely that at some point you entered information some distance below the end of your current code. Even though that information has been deleted, the module's boundaries still reflect the furthest position of entered information.

RESOLUTION

Once a sheet's boundaries have been extended, there is no way to shrink them. A possible workaround would be to copy and paste your code into a new module window.

STATUS

This behavior is by design.

[References](#)

PRB: Subform not Updated

Article Number: Q92676
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

As you scroll through records in the main form, your subform is not updated. The records on the main form and the subform are not synchronized.

CAUSE

Your subform is not linked to the main form.

RESOLUTION

You should use the LinkMasterFields and LinkChildFields form properties to link your main form and subform automatically. You may manually update the subform by pressing the F9 function key (Recalculate).

STATUS

This behavior is by design.

MORE INFORMATION

Recalculation occurs automatically for controls that reference other fields on the same form, or fields in subforms. Recalculation does not occur automatically for subform controls that only reference fields on the main (master) form, or in other subforms.

This is because subforms notify the master form of any changes, but the master form does not notify the subforms of changes. Nor do subforms on the same main form notify one another of any changes.

References:

For more information, search for "Link: main form and subform" then "function keys" using the Microsoft Access Help menu.

[References](#)

PRB: Insert Object Dirties Record

Article Number: Q92675
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Placing your cursor in a field with the data type OLE Object, choosing Insert Object from the Edit menu, and then clicking the cancel button in the Insert Object dialog box will "dirty" a record. A dirty record is one that has been edited.

If the field is part of an existing record, the previous contents of the field are not deleted or modified, but the record is rewritten when you move off of it.

STATUS

This behavior is by design.

MORE INFORMATION

Steps To Reproduce Behavior

-
1. Open a table that includes an OLE object field.
 2. Place your cursor in the OLE object field.
 3. Select Insert Object from the Edit menu, then click Cancel.
 4. The record selector now indicates that the record is dirty by displaying the pencil indicator.

[References](#)

PRB: TransferDatabase Macro Accepts Invalid File Names

Article Number: Q92674
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When using the TransferDatabase [macro](#) to export a [database](#), Microsoft Access allows you to enter file names that include spaces in the Destination argument. Once the database is exported with a file name that includes a space, Microsoft Access will not be able to import that file. If you try to import the file, the following error message appears:

The filename is not valid

RESOLUTION

If you export a file with a blank embedded in the filename, renaming the file without the space, using File Manager or MS-DOS, enables Microsoft Access to import the file.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a [table](#) and call it Table1.
2. Create a new macro:

Macro Name	Condition	Action
Macro1		TransferDatabase

Macro1 Actions

TransferDatabase Arguments	Attribute
Transfer Type:	Export
Database Type:	Dbase or Paradox
Database Name:	c:\access
Object Type:	Table
Source:	Table1
Destination:	Test Table
Structure Only:	No

3. This macro creates a file called Test Tab.xxx where xxx is the appropriate extension for the export type.
4. Try to import this file. Microsoft Access displays the message:
Test Tab.xxx This filename is not valid

[References](#)

INF: Error Message 'Unable to Recover'

Article Number: Q92671
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You may receive the following error message when attempting to open a table in design view:

Unable to recover from attempt to switch view. Window will be closed.

Microsoft Access returns focus to the Database window.

MORE INFORMATION

This error message occurs only in rare scenarios. It is an informational message only, not a program error.

Steps to Reproduce Behavior

1. Two users (User A and User B) both open the same database non-exclusively.
2. Both users open the same table in Datasheet view.
3. Both users simultaneously switch to Table Design view.
4. User A succeeds, but User B receives the following error message:

Table is in use, cannot lock by user '<user B's logon name>' on machine <user B's machine name>'.
5. User B tries to reopen the table in Datasheet view, and receives the following error message:

Couldn't lock table. Table '<table name>' is exclusively locked by user '<user A's logon name>' on machine '<user A's machine name>'.
6. User B is returned to the Database window with the following error message:

Unable to recover from attempt to switch view. Window will be closed.
7. Microsoft Access returns User B's focus to the Database window.

When in Datasheet view (step 3), both users have the table open non-exclusively. When they try to go into Table Design view, they are actually closing the table and attempting to re-open it exclusively.

In the above scenario, User A succeeds and User B fails. Thus the error message shown in step 4.

When User B then tries to re-open the table (non-exclusively) in Datasheet view, User A already has the table exclusively locked. Thus the error message shown in step 5.

The error message in step 6 is an information message designed to let User B know that Microsoft Access is unable to open the table in either Table Design or Datasheet view. Microsoft Access returns the focus to the Database window.

[References](#)

PRB: Default Width in Report Design Does Not Reflect Margins

Article Number: Q92670
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The default grid area is independent of how a report or form is printed. The controls you place in the document determine the size. In the USA, the default width of the grid area is set to 5 inches, regardless of the print settings (paper size, orientation, margins). You may think that because the default, US, print settings call for a page size of 8.5" x 11", portrait orientation, with 1" margins that the default grid area would be 8.5" - 2" = 6.5".

STATUS

This behavior is by design.

References:

For more information on how to change the desired layout of reports or forms, search for "Creating a New Template for Forms and Reports" using the Microsoft Access Help menu.

[References](#)

PRB: Can't Exit Extend Mode in Parameters Dialog Box

Article Number: Q92669
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you press ESC to cancel Extend mode while in the Parameters dialog box of a query, you cancel the Parameters dialog box. Your work is not saved.

CAUSE

Pressing F8 turns Extend mode on (EXT appears in the status bar). Once Extend mode is on, pressing F8 repeatedly extends the selection to the word, the field, the record (in Datasheet and Form view), and all records (in Datasheet and Form view). Pressing SHIFT+F8 reverses the selection, but will not take you out of Extend mode. To turn off Extend mode, press ESC.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open a new Query (don't need to add any tables.)
2. Choose Parameters from the Query menu.
3. Enter a couple parameters with more than one word in each.
4. Click in the center of a word and press F8 to enable Extend mode.
5. Press ESC to cancel Extend mode. The Parameters dialog box is cancelled.

[References](#)

INF: No Permission to Modify Data in a Table

Article Number: Q92667
CREATED: 12-NOV-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you try to modify data in a [table](#) in [Datasheet view](#), you may receive the error message

No [permissions](#) for '<name of table>'

even though you were able to open the table without receiving any error messages.

MORE INFORMATION

This error message indicates that you have only Read Definitions and Read Data permissions for the table. Because you have permission to view the data, no error message is displayed when you first open the table in Datasheet view, which is the default.

When you try to open the table in [Design view](#), the following error message is displayed:

You don't have permission to modify '<name of table>'.
Open as [read-only](#)?

This message appears because you have no [security](#) permissions to open the table in Design view.

References:

"Microsoft Access User's Guide," version 1.0, page 621

For more information on permissions, search for "permissions command" using the Help menu in Microsoft Access.

[References](#)

PRB: CTRL+HOME Navigation Problem in Macro Design Mode

Article Number: Q92665
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you press the key combination CTRL+HOME when you are in design mode of a new **macro**, the cursor might not move to the beginning of the macro, as you might expect.

CAUSE

If you click in the Action **column** you are in Edit mode for the current macro line; pressing CTRL+HOME moves the cursor to the beginning of the drop down list. If you click in any Comments column which is empty, you are already at the beginning of the **field**.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new macro.
2. Click in the Comments column of any **row** but the first.
3. Press CTRL+HOME.

Nothing happens. If you navigate with the arrow keys after clicking in the Comment section, pressing CTRL+HOME positions you on the first line of the macro.

[References](#)

INF: How to Create a Chart Without a Title

Article Number: Q92655
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When creating a [graph](#) in Form Design mode using the [GraphWizard](#), Microsoft Access fills in the default value "Graph" if no title is specified for the chart.

MORE INFORMATION

There is a workaround for this by changing the font color for the chart title:

1. Launch the Microsoft Graph Server from Form Design mode by selecting Insert Object from the Edit menu.
2. Double-click on the chart title.
3. Select Font from the Format menu.
4. Change the font color to the same color as your [form](#) background color.
5. Close Microsoft Graph and answer "Yes" to [update](#).
6. Save the form.

The title now blends in with the form background.

[References](#)

PRB: Can't Attach to Paradox Table on Read-Only Drive

Article Number: Q92653
CREATED: 12-NOV-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

You cannot attach to a Paradox [table](#) (*.DB) located on a [read-only](#) drive. If you try to attach the table, you receive the following error message:

```
Couldn't open file 'FILENAME#DB'
```

CAUSE

Neither Paradox nor the Paradox indexed sequential access [method](#) (ISAM) shipped with Microsoft Access support databases located on read-only shares.

STATUS

This behavior is by design.

MORE INFORMATION

Paradox is manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

PRB: Excel Custom Date Format Not Detected by Import

Article Number: Q92651
CREATED: 12-NOV-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you import a Microsoft Excel spreadsheet containing custom date formats into a new Microsoft Access [table](#), Microsoft Access gives the fields a Double [data type](#), instead of a Date/Time data type.

CAUSE

The import protocol supports date and currency guessing only on cells formatted with a system-supplied (built-in) format.

RESOLUTION

One possible workaround is to import the Microsoft Excel spreadsheet into an existing table in which that [field](#) has been formatted for Date/Time. Microsoft Access will not recognize the custom format, but will display the date in the standard Date/Time format (mm/dd/yy).

Another workaround is to open the imported table in [Design view](#) and change the data type from Number to Date/Time.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new Microsoft Excel spreadsheet.
2. Enter a date in Cell A1.
3. Assign a new date format, such as mmmmmm/dd/yyyy.
4. Apply the format to the A [column](#).
5. Import the spreadsheet to Microsoft Access.

Result: The date is formatted as Number (Double), not as Date/Time.

References:

"Microsoft Access User's Guide," version 1.0, chapter 4, pages 72-75

[References](#)

PRB: OLE Excel Format Changes when Updated in Microsoft Access

Article Number: Q92650
CREATED: 12-NOV-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

New [OLE](#) Objects created in Microsoft Excel are displayed without [row](#) or [column](#) headings in Microsoft Access. Once the object has been updated, the display changes and row and column headings are displayed.

MORE INFORMATION

Steps To Reproduce Behavior

1. Start Microsoft Excel.
2. Enter "Test" in [cell](#) A1.
3. Select cell A1 and choose Copy from the Edit menu.
4. Start Microsoft Access and open a [database](#).
5. Create a new [form](#) and choose Paste from the Edit menu.
This creates an OLE Object which displays cell A1.
6. Double-click on the OLE Object to activate Excel.
7. Modify the contents of cell A1.
8. From the File menu choose Update.
9. Activate Microsoft Access.

The OLE Object now displays row and column headings.
You may need to resize the OLE Object to see the updated contents of cell A1.

[References](#)

INF: How to Remove Relationships

Article Number: Q92649
CREATED: 12-NOV-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article explains how to delete a [relationship](#) in Microsoft Access once it has been created.

MORE INFORMATION

The sample [database](#) NWIND.MDB contains a relationship between the Employees and the Orders tables. The steps that follow describe how to delete this relationship.

1. Open the NWIND database by selecting Open Database from the File menu.
2. From the Edit menu, choose Relationships.
3. From the Primary Table [combo box](#), choose Employees.
4. From the Related Table combo box, choose Orders.

Since there is a relationship already defined for these two tables, Microsoft Access automatically displays Employee ID in the Select Matching Fields combo box and grays out the Add button.

5. Choose the Delete button to remove the existing relationship.
6. Choose the Close button to exit the Relationships [dialog box](#) and return to the [Database window](#).

NOTE: In addition to the Relationships dialog box, you can view existing relationships by creating a [query](#). Include all the tables in your database in the query. Microsoft Access automatically displays [join](#) lines when there are existing relationships between tables.

References:

For more information on listing relationships in a database, query on the following words in the Microsoft Knowledge Base:

relationships and list

[References](#)

INF: Differences Between Native and Attached Tables

Article Number: Q92644
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Attached tables created in other applications (such as Paradox, dBase IV, and so forth) are different from native Microsoft Access tables in the following four ways:

- In the [table](#) list, icons for attached tables contain black arrows.
- When you try to open an [attached table](#) in [Design view](#), the following error message appears:

Table 'xxxxx' is an attached table. Some properties can't be modified. Open it anyway?
- When you delete an attached table, its contents are not deleted; only the table entry is deleted from the list. However, the error message is the same as for a deleted Microsoft Access table, in which the contents actually are deleted.
- You cannot define relationships between attached tables, nor between native Microsoft Access tables and attached tables.

Paradox is manufactured by Ansa Software, a Borland company, and dBASE IV is manufactured by Borland International, Inc., vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

INF: How to Take a Button Out of Tab Order

Article Number: Q92640
CREATED: 12-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes two methods for removing push buttons from the [tab order](#) on a [form](#).

MORE INFORMATION

1. Place the buttons in a [form header](#) or [footer](#). Each section on a form has it's own tab order, so the form user would have to explicitly move from the detail section to the [header](#) or footer.
2. Place an unbound, hidden [text box](#) before the buttons in terms of tab order on the form. In the OnEnter property of the hidden text box, call a [macro](#) that sets focus to the first [control](#) on the form.

This solution is one that FormWizards also use. Put all of your regular controls in the detail section of the form first in the tab order, then place an unbound, hidden text box on the form, and finally place any buttons on the form. In the OnEnter property of the hidden text box, use a macro to set focus to the first of your regular controls.

References:

For more information, search for "tab order" then "Tab Order Command (Edit Menu)" or "Setting Tab Order" using the Microsoft Access Help menu.

[References](#)

PRB: Cannot Install Cue Cards Without Installing Help

Article Number: Q92639
CREATED: 12-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

During a custom installation, you can select and deselect options through the Microsoft Access Setup Options dialog. If you deselect Help, the size of the Cue Card files increases from 1688K to 5112K. However, the total disk space required does not increase and the size of Help does not decrease from 3424K to 0K.

CAUSE

You must have Help loaded in order to use Cue Cards. Therefore, when you deselect Help, the size of the Cue Cards files increases to include the size of the Help files.

Help and Cue Cards are separate selections because it is possible to install and use the Help files without installing the Cue Card files.

STATUS

This behavior is by design.

[References](#)

PRB: FormWizard Labels Placed on Left Margin Grow Down

Article Number: Q91721
CREATED: 03-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

New labels added at the left margin of a form created by FormWizards using a single column format appear to "grow" down instead of out towards the right margin. In fact, the default label is aligned right for all single-column FormWizards.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a bound form using a FormWizard and choose Single Column.
2. Open the form in Design Mode
3. Create a label on the leftmost side of the form in any section.

Results: As the user enters text for the label, it grows down with each keystroke.

[References](#)

INF: Form Sizing Grid Accepts Non-Integer Values

Article Number: Q91720
CREATED: 03-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you use the Grid option in design mode for either a [form](#) or a [report](#), non-integer values are accepted, but are rounded to the nearest integer.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open a report or form in design mode
2. Change the Grid option to any non-integer value
3. Upon exiting the [field](#), the value you entered is rounded to the nearest integer

[References](#)

PRB: Two Labels on Top of Each Other Both Print

Article Number: Q91716
CREATED: 03-NOV-1992
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you have two labels on top of each other, both labels may print using the following Windows printer drivers and the hardware printer fonts:

Win 3.0
 CANONIII
 DESKJET
 IBM4019

Win 3.0 SDL
 BROHL24
 CANONIII

Win 3.1
 HPPCL5A
 IBM4019
 LBPIII (Canon)

CAUSE

This is a limitation of the printer driver.

RESOLUTION

Do not place labels on top of each other.

STATUS

This behavior is by design.

[References](#)

PRB: Text May be Hard to Read on an EGA Screen

Article Number: Q91715
CREATED: 03-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Text may be hard to read or appear slightly off center when using an EGA video driver. This condition may worsen if you use a gray background. This is particularly true of Option Buttons.

STATUS

This behavior is by design.

[References](#)

PRB: Toolbars Do Not Use Control Panel Button Background

Article Number: Q91714
CREATED: 03-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Changing the button background in the control panel does not change the Toolbar's background. This behavior is consistent with the status bar in Microsoft Access and with other applications' toolbars, such as Microsoft Excel (with the exception of Microsoft Word for Windows.)

MORE INFORMATION

Toolbars in Microsoft Access are predefined forms. There is no way to specify the Button Background as a color on a form. If you would like to change colors within Microsoft Access, you can open and edit the forms contained in UTILITY.MDA. If you modify UTILITY.MDA, however, extreme care should be taken beforehand by creating one or more backups of it.

[References](#)

PRB: Grid Lines Not Visible in Access When Using Wizards

Article Number: Q91713
CREATED: 03-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The grid is not visible in design mode when using FormWizards or ReportWizards.

CAUSE

ReportWizards and FormWizards will each use default GridX and GridY values of 64. If these settings are greater than 16, the grid is not visible. The default settings are in the visible range when you don't use the wizards.

RESOLUTION

You can make the grid visible by selecting the properties box for the **form** or **report** and changing the GridX and GridY unit settings so that they are less than or equal to 16 units per inch.

MORE INFORMATION

If you are creating a report or a form, the grid will be visible when you do not use AccessWizards, but will not be visible if the AccessWizards are used. This is true only in design mode. The grid can be made visible when you use the AccessWizards by changing the GridX and GridY property settings.

Grids are dots that appear in the design views of a report or a form. These help you position controls more precisely by letting you increase or decrease the fineness of the grid. The number used in the GridX and GridY properties reflect the number of dots per unit of measurement. This unit of measurement is usually inches, but will vary depending on the **control** panel settings of the individual. If the numbers in the grid properties are greater than 16 dots per inch or 6 dots per centimeter, they will not be visible. The allowable values are from 1 to 64.

[References](#)

INF: Height/Width Settings Do Not Affect Size of Objects

Article Number: Q91712
CREATED: 03-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Setting the Height and Width properties for an object within an [option group](#) does not affect the size of the Option button or Check box; instead, it affects only the [label](#) associated with the Option button or Check box.

MORE INFORMATION

Adjusting the Height and Width properties for an object within an [option group](#) increases or decreases the area you can click with the mouse and select the specific Option button or Check box. Setting these properties does not affect the visible size of the object.

References:

"Microsoft User's Guide" version 1.0, chapter 9, "Designing Forms", pages 247-252.

For additional information, search for "Option Button Properties" and select the topic "Option Button Control Properties" in Microsoft Access Help.

[References](#)

INF: 'Couldn't Open File' Error Message Using Paradox Table

Article Number: Q91711
CREATED: 03-NOV-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

This article describes why you may be unable to open an attached Paradox [table](#) from within Microsoft Access, or why you may experience an 'Unexpected Condition' in Paradox when you attempt to view a table.

Paradox and Microsoft Access are able to simultaneously access the same Paradox tables if there is only one PARADOX.NET file in use by both programs.

Microsoft Access

If you are in Microsoft Access and attempt to open an attached Paradox table you may receive the following error:

 Couldn't open file 'filename'

This may be caused by already having the table open in Paradox. Other situations could also cause this error message. For more information, search for 'Error Messages Reference' using the help menu.

Paradox

If you are in Paradox and attempt to open a table, you may experience an error condition and be returned to the DOS prompt or Windows. A message similar to the following appears:

 Unexpected condition: Fileopen (I104) - c:\pdoxdata\yourfile.db
 (Code 5) -- Leaving Paradox

This may be caused by Microsoft Access having the table open.

In both situations the possibility exists that there are two different PARADOX.NET files in use by the applications.

MORE INFORMATION

When you attach to a Paradox version 3.0 or 3.5 table (<filename>.DB) in Microsoft Access, file locking information is maintained by PARADOX.NET the Paradox [network control](#) file. Paradox requires this file in network environments but it is also used in standalone configurations.

Microsoft Access places a PARADOX.NET file in the Microsoft Access program directory during the setup process. An entry is made in the MSACCESS.INI file in the [Paradox ISAM] section pointing to the location of the file. The MSACCESS.INI file should be in your Windows directory. The entry follows:

```
[Paradox ISAM]
ParadoxNetPath=C:\ACCESS\
```

Paradox also installs a PARADOX.NET file during normal installation. By default it will be located in the Paradox program directory. The location of this file is determined by the network administrator during a network installation.

To redirect Microsoft Access to the location of the PARADOX.NET file that Paradox installed, edit the MSACCESS.INI file in any text editor such as Notepad, and change the path. Do not include the name of the file.

To change the Paradox configuration to point to the Microsoft Access PARADOX.NET file, you could run the NUPDATE.EXE utility. Refer to the Paradox "Network Administrator's Guide" for more information on this topic.

[References](#)

PRB: No Implicit Parameters on CrossTab Queries

Article Number: Q91710
CREATED: 03-NOV-1992
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you create a crosstab (XTAB) [query](#) using an implicit parameter [XXX] in a [WHERE clause \(criteria\)](#), Microsoft Access returns the following error when you attempt to run the query:

```
Can't bind name '[XXX]'
```

If you use an explicit parameter, the query works as expected.

CAUSE

A [crosstab query](#) dynamically generates [column](#) names. Therefore, Microsoft Access cannot tell whether [XXX] is referring to a parameter or a column name until after the query is bound.

When you build a [graph](#), Microsoft Access uses a crosstab to build the data to graph. If the query is a [parameter query](#), you have to define the parameters explicitly as you would for a normal crosstab query.

RESOLUTION

To avoid this error, define [XXX] as an explicit parameter by adding it to the Query Parameters [dialog box](#).

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the sample Northwind Traders [database](#) (NWIND.MDB).
2. Create a new crosstab query on the Order Details [table](#).
3. Drag the fields Unit Price, Discount, and Order ID to the [QBE](#) grid, and set the values in the grid as follows:

```
Crosstab Query: Query1
```

```
-----  
Field: Unit Price  
Total: Group By
```

Crosstab: Row Heading
Criteria: > [Enter an Amount:]

Field: Discount
Total: Group By
Crosstab: Column Heading

Field: Order ID
Total: Count
Crosstab: Value

4. Run the query. The following error message is displayed:

Can't bind name '[Enter an Amount:]'.

5. Choose OK to return to query design, and then select Parameters from the Query menu.

6. Enter the following under Parameters:

[Enter an Amount:]

7. Select Currency as the Data Type, and choose OK.

8. Run the query and type any amount (for example, type "2").
The query executes.

REFERENCES

=====

For more information, search for "parameter query" then "Creating a Parameter Query" using the Microsoft Access Help menu.

[References](#)

PRB: Changing Caption on Title Label Leaves Shadow Unmodified

Article Number: Q91709
CREATED: 03-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Changing the Caption on the Title Label leaves the shadow of the original Title Bar present. This occurs when creating a [form](#) with any Form Wizard (other than Graph) using Embossed View.

STATUS

This behavior is by design. There are two labels associated with this caption. The shadow is really another separate [label](#) that gives the impression of a shadow.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new form using any Form Wizard, except Graph.
2. Select Embossed as the View for the form.
3. Open the form in Design mode.
4. Select the title and change the Caption to something shorter than the current title.
5. Change to a smaller [point size](#) to make the shadow more visible.
6. Switch to [Form view](#).

In order to use the Embossed View and the new title correctly, you must also select the shadow and change it.

[References](#)

PRB: Query Design Status Bar Does not Have Help Message

Article Number: Q91708
CREATED: 03-NOV-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In Query Design Mode, the status bar does not inform the user that "F6=Switch Panes F1=Help" as it does in Table and Macro design modes.

STATUS

This behavior is by design.

MORE INFORMATION

Query design mode offers the use of F1 to display help, and F6 to switch panes in the design window.

References:

For more information, search for "F1" then "Keyboard: Navigation Keys" using the Microsoft Access Help menu.

[References](#)

INF: Use CTRL+M to Open Objects in Design Mode

Article Number: Q91707
CREATED: 03-NOV-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can use the CTRL+M key combination to force any object in the [Database window](#) to open in Design mode.

MORE INFORMATION

Steps to Reproduce Behavior:

1. Select any object in the Database window.
2. While holding down the Control key (CTRL), type the letter M.

The object that you select will now be opened in design mode rather than the default browse mode.

[References](#)

PRB: Expression in Query Grid Doesn't Show All Text

Article Number: Q91705
CREATED: 03-NOV-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When entering an [expression](#) in the Query Grid window, Microsoft Access does not fill a [cell](#) all the way to its right. On long expressions, it may look like your expression or [column](#) name is cut off before the end of the cell.

CAUSE

Microsoft Access reserves a portion of the cell to display the drop down arrow. Your expression remains unchanged.

RESOLUTION

Click the cell and press SHIFT+F2 to view the entire expression in the Zoom box. You can also change the size of the cell by clicking and dragging the cell border at the top of the grid cell.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open a Database (such as NWIND.MDB.)
2. Create a new Query (you do not need to add any tables).
3. Enter the following expression in the first [field](#) cell:

[Unit Price]*[Quantity]

4. Press tab or enter to move to the next column.

The expression appears truncated.

References:

For more information, search for "Zoom Box" using the Microsoft Access Help menu.

[References](#)

INF: SQL Server Triggers and Key Values to and from Access

Article Number: Q91179
CREATED: 29-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If a Microsoft Access procedure sends a record to SQL Server to insert into a SQL Server table, and the unique key for the record is created by a trigger on that table, the Microsoft Access procedure will lose reference to the record.

MORE INFORMATION

When information is sent to a SQL Server table, Microsoft Access looks at the record and determines a unique key value using the information in that record. When the data is passed to a SQL Server backend to be inserted into a table and a trigger creates a unique key value for the record, Microsoft Access will lose reference to that record because the unique key stored by Microsoft Access is different than the unique key created by the trigger on SQL Server. Until it is queried on using the new unique key value, Microsoft Access views the record just sent to SQL Server as having been deleted.

[References](#)

INF: Transparent Text Boxes Not Supported

Article Number: Q91045
CREATED: 27-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The default label control associated with a text box control has a BackStyle property that can be changed to Clear on the color palette, but the text box itself has no such property and cannot be changed to Clear on the color palette.

To work around this limitation, change the BackColor property of the text box to the same color as the background it overlays.

[References](#)

PRB: Cannot Scroll Left in Form's Design View with Rulers

Article Number: Q90990
CREATED: 26-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In a form's Design view, with both Microsoft Access and the form maximized, turning the rulers off and then selecting and dragging a control on a form to the left does not give expected results.

The control will not scroll left until the ruler is turned back on. You can, however, move and scroll by using the "thumb" (scroll bar box) manually. Turning the rulers off and selecting and dragging a control right widens the form.

CAUSE

Microsoft Access will scroll left if the edge of the form is encountered while dragging a control. With everything maximized, the control being dragged does not run into the edge of a form or window; it runs into the edge of the screen. The other edges have Microsoft Access objects (scroll bars or menu bars) so Microsoft Access is aware that the control is being dragged and will scroll accordingly.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 8, "Form Basics," and Chapter 9, "Designing Forms"

[References](#)

INF: How To Tell If An Object Exists in the Database

Article Number: Q90989
CREATED: 26-OCT-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

It may be necessary to check for the existence of an object before the creation of a new Table, Form, Report, etc., using [Access Basic](#) (AB).

This article explains how functions can check for the existence of a given object.

MORE INFORMATION

Definition of Variables Used

ObjectFound% - a test [variable](#). If -1 then the object exists, otherwise it does not.

WhatObjectType\$ - the object type of the item in question.

ObjectName - the name of the item in question.

Follow these four steps to create a user-defined [function](#) to check for the existence of a given object:

1. Choose Module from the [database](#) window.
2. Choose New to create a new [module](#).
3. Type the following function that checks for the existence of a given object:

```
Function ObjectExists% (WhatObjectType$, ObjectName$)
  Dim ObjectFound%

  On Error Resume Next

  If WhatObjectType = "Form" Then
    DoCmd SelectObject A_FORM, ObjectName, True
  ElseIf WhatObjectType = "Table" Then
    DoCmd SelectObject A_TABLE, ObjectName, True
  ElseIf WhatObjectType = "Macro" Then
    DoCmd SelectObject A_MACRO, ObjectName, True
  ElseIf WhatObjectType = "Module" Then
    DoCmd SelectObject A_MODULE, ObjectName, True
  ElseIf WhatObjectType = "Report" Then
    DoCmd SelectObject A_REPORT, ObjectName, True
```

```

ElseIf WhatObjectType = "Query" Then
    DoCmd SelectObject A_QUERY, ObjectName, True
Else
    MsgBox WhatObjectType & " type does not exist!"
    WhatObjectType = "False"
End If

If Err <> 2544 And WhatObjectType <> "False" Then
'Object was found in the Database Container (DBC)
    Err = 0      'Reset the error level
    ObjectFound% = -1
Else
    'Object was not found in the DBC
    ObjectFound% = 0
    MsgBox "Error #" & Err & ": " & Error
    'Give error number and message
End If
ObjectExists% = ObjectFound%
End Function

```

This function returns a value of True (-1) if the object exists. If the object does not exist, or an error is encountered, the function will return a value of False (0).

To use the function `ObjectExists`, do the following:

Choose Immediate Window from the View Menu while in a module, then type:

```
?ObjectExists("object type","Object name")
```

NOTE: Where "object type" is "Form", "Table", "Macro", "Module", "Query", or "Report" and "object name" is the name of the object in question. An sample call would be:

```
?ObjectExists("Table","Employees")
```

If the object exists, the database window becomes active and the object is selected. If the object does not exist a Message Box pops up stating the error that occurred and that the object does not exist. If the object type is incorrect, a Message Box pops up stating that the object type does not exist.

NOTE: One side effect of the above function is that it will make your Database window visible and have the focus if the table exists. This can be corrected with a test function to find the Visible state of the Database window and change it and the focus as desired.

References:

"Microsoft Access Basic: An Introduction to Programming", version 1.0, chapters 1-5.

"Microsoft Access Language Reference", Part 1, version 1.0.

"Microsoft Windows Software Development Kit", Microsoft Press, 1992

"Programming Windows: the Microsoft Guide to Writing Applications for

Windows 3", Charles Petzold. Microsoft Press, 1990

"Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference" Volumes 1 - 6, Microsoft Press, 1992

[References](#)

INF: How to Use Get and Write PrivateProfileString

Article Number: Q90988
CREATED: 26-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not have a [function](#) to access or store various settings in user created applications (that is, a log of users, window position settings, etc.). However, you can use the Microsoft Windows 3.1 Application Program Interface (API) through [Access Basic](#) (AB) to create user-defined functions to store settings in a file outside the [database](#).

MORE INFORMATION

The variables are defined in this section of the article, followed by the six steps necessary to create user-defined functions to store settings in a file outside the database.

Definition Of Variables Used

Variables Required By GetPrivateProfileString and WritePrivateProfileString:

- | | |
|---------------------|--|
| lpfilename\$ | - Filename looked for in Windows directory by default. |
| lpapplicationname\$ | - Section title in square brackets in the INI file. |
| lpkeyname\$ | - The section in the INI file where the information is located. |
| lpDefault\$ | - What information to give if there is no value. |
| lpreturnedstring\$ | - Information asked for from the INI file. |
| nSize% | - Max number of characters the buffer holds of information returned. |
| lpString\$ | - The information saved in the INI file. |

User Defined Variables:

- | | |
|-----------|---|
| GotInfo | - Information retrieved from the level requested |
| GetInfo\$ | - Used to pass the lpkeyname\$ to function GetInfo |
| Where\$ | - Used to pass the lpkeyname\$ to function WriteIni |
| Info\$ | - Used to pass the lpString\$ to function WriteIni |

Steps to follow:

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

1. Choose Module from the [Database window](#).
2. Choose New to create a new [module](#).
3. Type the declaration statements that follow in the Global [Declarations section](#).

```
Option Explicit
Declare Function GetPrivateProfileString% Lib "Kernel" _
    (ByVal lpapplicationname$, _
    ByVal lpkeyname$, _
    ByVal lpDefault$, _
    ByVal lpretreturnedstring$, _
    ByVal nSize%, ByVal lpfilename$)

Declare Function WritePrivateProfileString% Lib "Kernel" _
    (ByVal lpapplicationname$, _
    ByVal lpkeyname$, _
    ByVal lpString$, _
    ByVal lpfilename$)
```

5. Type the following function that gets the information from the file.

```
Function GetIni (GetInfo$)
    lpfilename$ = "YOUR.INI"
    lpapplicationname$ = "Your Section"
    lpkeyname$ = GetInfo$
    lpDefault$ = "No Value"
    lpretreturnedstring$ = Space$(255)
    nSize% = Len(lpretreturnedstring$)

    GetIni = GetPrivateProfileString(lpapplicationname$, _
        lpkeyname$, lpDefault$, _
        lpretreturnedstring$, nSize%, _
        lpfilename$)

    GotInfo = lpretreturnedstring$
End Function
```

6. Type the following function that writes the information to the file.

```
Function WriteIni (Where$, Info$)
    lpapplicationname$ = "Your Section"
    lpkeyname$ = Where$
    lpString$ = Info$
    lpfilename$ = "YOUR.INI"

    WriteIni = WritePrivateProfileString(lpapplicationname$, _
        Where$, Info$, lpfilename$)
End Function
```

To use these functions follow the two steps below:

1. When used in a [macro](#), the function WriteIni creates the entry First=TIME in the file YOUR.INI under a section named [Your Section]. The following line calls the function:

```
=WriteIni("First","TIME")
```

In the file YOUR.INI the entry looks like the following:

```
[Your Section]
First=TIME
```

2. Use the function GetIni in a Macro to return "TIME" (without the quotes) in the [variable](#) GotInfo.

```
=GetIni("First")
```

The variable GotInfo now holds the value "TIME" (without the quotes) and can be used anywhere needed in the application.

NOTE: This example does not have error trapping. Unexpected results may occur if the declarations and the variable types are not correct or your file is not in the location specified or does not exist.

References:

"Microsoft Access Basic: An Introduction to Programming", version 1.0, chapters 1-5.

"Microsoft Access Language Reference", version 1.0, Part 1.

"Microsoft Windows Software Development Kit", Microsoft Press, 1992.

"Programming Windows: the Microsoft Guide to Writing Applications for Windows 3", Charles Petzold, Microsoft Press, 1990.

"Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference" Volumes 1 - 6, Microsoft Press, 1992.

[References](#)

INF: Common Dialog Boxes Allow Only Lowercase Typing

Article Number: Q90987
CREATED: 26-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The File Name box in the File New Database, File Open Database, File Compact Database, File Encrypt/Decrypt Database, and File Repair Database [dialog box](#) allows typing in lowercase only.

This behavior occurs because the File Name box in the common dialog box template has the lowercase flag set so as to allow more characters to be displayed in the given width. This behavior has no effect on the outcome of the given tasks because Microsoft Access does not differentiate between lowercase and uppercase characters in filenames.

[References](#)

PRB: Change Option Affects Only One Instance of Many of Access

Article Number: Q90986
CREATED: 26-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When two instances of Microsoft Access are run on the same machine, they both access the same SYSTEM.MDA, but the options for both do not always appear to be the same after being changed in one instance. However, the View option shows them both as being changed.

CAUSE

When there are two instances of Microsoft Access running on the same machine, only the routines in the instance where the preferences are changed are called. The other instance does not have its memory copies updated.

STATUS

This behavior is by design.

MORE INFORMATION

Many options are stored in memory so that they do not have to be read from the preferences [table](#) every time they are accessed. The mechanism by which the memory and disk copies are kept in synchronization is the array of routines that is called whenever a preference is changed.

When there are two instances of Microsoft Access running, only the routines in the instance changing preferences are called. The other instance does not have its memory copies updated, but the View options shows them as being changed.

In order to make the preference take effect in another instance on the same system, the user either needs to shutdown and reopen this instance, or change the preference and resave it.

This is not an issue for users on separate systems, since their preferences are stored per user.

References:

"Microsoft Access User's Guide," version 1.0, chapter 25,
"Administering a Database System."

"Microsoft Access User's Guide," version 1.0, Appendix D, "Setting Up Microsoft Access on a Network."

[References](#)

PRB: Alert With Wrong Filename Given When Opening File Copied

Article Number: Q90985
CREATED: 26-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

On a machine with a shared drive, if a large [database](#) file is copied to a new name or location, when another machine attempts to open that database during the copy procedure, an error message occurs stating the incorrect database name is corrupt or is not a Microsoft Access database file.

Depending upon how the file was opened, an error message below is given where "Modules" is not the correct name of the file being opened.

RO=Read Only	Error Message
1. RO: No Exclusive: No	--> "Database 'Modules' is corrupted."
2. RO: No Exclusive: Yes	--> "Couldn't access file 'filename'."
3. RO: Yes Exclusive: No	--> "Database 'Modules' is corrupted."
4. RO: Yes Exclusive: Yes	--> "Couldn't access file 'filename'."

CAUSE

Due to the way file locking is placed upon a file by MS-DOS during the copy procedure, Microsoft Access has no way of knowing what is happening to the file being opened.

RESOLUTION

This behavior is by design.

[References](#)

PRB: Two Users Can Open the Same dBASE Database Exclusive

Article Number: Q90882
CREATED: 25-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Attaching to a dBASE file in Microsoft Access using the Exclusive [check box](#) allows another user to attach to the same file over the [network](#).

CAUSE

When two users or applications are running on separate machines on a multiuser network, they can both open the same dBASE or Paradox file without receiving an error message. However, if two instances of Microsoft Access are running on one machine and both instances try to open the same dBASE or Paradox file, one instance will receive an error message saying the [database](#) is locked.

In dBASE there is no [exclusive](#) database lock. In dBASE, records in files are locked, not the entire file. In other words, one user can open the first database exclusively, and a second user can open it shared or exclusively. Therefore, the exclusive database lock is only recognized when Microsoft Access controls both attempts to open the same file from the same machine.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 4

[References](#)

INF: DDE Link Can't Find Data; Opens Second Copy of Server App

Article Number: Q90881
CREATED: 25-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Using Microsoft Access and Dynamic Data Exchange ([DDE](#)) to access information from an open DDE server application, such as Microsoft Excel, where the data does not exist causes the following error message to be displayed:

Remote data not accessible. Start application Excel?

If you choose the Yes button in the message [dialog box](#), a second copy of Microsoft Excel is started with the name of the nonexistent spreadsheet in its [title bar](#).

MORE INFORMATION

This behavior is correct unless you already have a spreadsheet with the name given in the DDE [link](#) open in Microsoft Excel. If the DDE link cannot find the spreadsheet, a second copy of Microsoft Excel will be launched using the filename given in the DDE statement. However, the DDE standard is not to have running applications open documents.

This behavior is by design.

References:

"Microsoft [Access Basic](#): An Introduction to Programming," version 1.0, Chapter 9, "Dynamic Data Exchange"

[References](#)

PRB: Long WHERE Clause Is Truncated After View SQL Command

Article Number: Q90880
CREATED: 25-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In Microsoft Access' Query By Example (QBE) grid, having a WHERE clause greater than 255 characters causes that clause to be truncated if OK is selected after the SQL command on View menu is used to see the SQL statement, or the next time the query is opened in Design view. The following error message appears:

Expression is too long for the QBE grid and has been truncated...

The beginning of the expression is listed after the error message.

CAUSE

This problem most often occurs when a long WHERE clause using the OR operator is entered into the QBE grid on the Criteria and Or lines. To optimize performance, Microsoft Access may combine criteria that are entered on separate lines. The limit for each QBE grid cell is 255 characters.

RESOLUTION

When two or more query lines are combined together and the total number of characters is greater than 255, do not view the SQL statement.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 6

[References](#)

INF: Empty Table Created by SELECT INTO with No Matching Rows

Article Number: Q90879
CREATED: 25-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Using a SELECT INTO or APPEND query with criteria that has no matching records causes an empty table to be created.

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 6

[References](#)

PRB: Primary Keys Not Used to Order Data in Combo/List Boxes

Article Number: Q90878
CREATED: 25-OCT-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If a [table](#) has a [primary key](#) and you use the table in a RowSource property in a [list box](#) or [combo box](#), you will see the data in insertion order, not primary key order.

RESOLUTION

For historical reasons of the [database](#), the order of insertion may have importance. To view the table's contents in primary key order, design a [query](#) that sorts on the primary key. Use the query as the RowSource property for the list box or combo box.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, pages 21-33 and 87-114

[References](#)

INF: Copy/Paste Operations from Forms to Queries

Article Number: Q90877
CREATED: 25-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you wish to copy and paste [SQL](#) statements from the RowSource properties of a [control](#), [form](#), or [report](#) to a [query](#), you must use the CTRL+C [method](#) to copy to the clipboard.

MORE INFORMATION

Using the pulldown menu commands Edit/Copy and Edit/Paste will not work in this case. You must use CTRL+C to copy the RowSource information to the clipboard from the form property, and then use CTRL+V to paste this information into the queries SQL statement. This method must be reversed to go from queries to forms because the View SQL [dialog box](#) is [modal](#).

Initially, after pasting the information into the RowSource, the statement may appear truncated or incorrect. Moving the focus off of the RowSource should correct this.

[References](#)

INF: Displaying List or Combo Boxes with More Than One Field

Article Number: Q90876
CREATED: 25-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article discusses information needed to display more than one [field](#) in a [list box](#) or combo list box.

MORE INFORMATION

If you enter a [SQL](#) command in the RowSource property for the [control](#), Microsoft Access can display more than one field in a list box or combo list box. For example, if you want to combine two fields into one list box, the [syntax](#) for the SQL command could be as follows:

```
Select [FieldName1] & [FieldName2] from [TableName];
```

If you are using a bound [column](#) in the list, the bound column would now apply to the placeholder in the SQL statement. In the above example, a bound column setting of 2 would equate to FieldName2.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 9

[References](#)

INF: Headers On Non-Continuous Forms Disable Scroll Bar

Article Number: Q90875
CREATED: 25-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you add a large [form header](#) to a non-[continuous form](#), the [vertical scroll bar](#) disappears and you will not have access to the detail below the [header](#).

MORE INFORMATION

It is better to size the header so that it fits within the limits of the scrollbar range so that the form behaves in form view just as it does in design view. The user must scroll the form down to see the rest of the form. For example, this problem occurs if you want a summary chart in the header, then detail charts below the header. The scroll bars consider the header and the form to be all one object.

[References](#)

INF: Correct Procedure for Inserting New Field Above Existing

Article Number: Q90874
CREATED: 25-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

On page 51 of the Microsoft Access "Getting Started" manual, the procedure for inserting a new field above an existing field is not correct. The procedure outlined in the manual inserts the field under the existing field instead of above it.

Step 1 for "Inserting a new field above an existing field" should be changed from

Click the row above which you want the new row to appear...

to:

Click the row below which you want the new row to appear...

[References](#)

INF: ESC Key Turns Off Extend (Selection) Mode

Article Number: Q90873
CREATED: 25-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

When you enter extend (EXT) mode by pressing F8 within any object, except for Modules, the only way to turn off extend mode is by pressing the ESC key (this is reflected in the status bar).

References:

For more information on Extend Mode search for "Extend mode", then "Selection Keys" using the Help menu.

Or search for the following keyword here in the Microsoft Knowledge Base:

extend

[References](#)

INF: Page Header Will Span Full Width Of MultiColumn Report

Article Number: Q90870
CREATED: 22-OCT-1992
MODIFIED: 20-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can design a [report](#) in Microsoft Access that has a page [header](#) that spans the entire width of the page, a [group](#) section header that is the width of a [column](#), plus has a multiple-column detail section.

To do this, the columns are defined in the Print Setup option as either Vertical (also called snaking columns,) or Horizontal. This article presents the steps to create a report with 3 across, 2 inch wide snaking columns.

MORE INFORMATION

To create a grouped, multiple-column report where the [page header](#) spans the width of the page, follow these steps:

1. In report [Design view](#), drag the right edge of any report section to the desired width of the report. For this example, drag the edge over to the 7 inch mark on the ruler.
2. Choose Sorting and Grouping from the View menu and enter a [field](#) or [expression](#) in the Field/Expression box to group your data by.
3. In the Group Properties section of the Sorting and Grouping dialog box, enter 'Yes' for Group Header.
4. Using the Toolbox, create a Label and place it in the Page Header section. Enter the text for the [label](#) in the Caption property. Size the [text box](#) so it is the same width as the Page Header section.
5. Using the Toolbox, create a text box bound to the group field. Place the text box in the Group Header. Size this [control](#) 2 inches wide or less, and place the left edge of the box along the left edge of the group section.
6. In the detail section place your [bound control](#)(s) within the first 2 inches.
7. From the File menu, choose Print Setup.
8. In the Print Setup [dialog box](#), choose the More button.
9. In the Items Across box, type the number of columns you want in the detail section. For this example, enter 3. Leave [row](#) and column spacing as they are.

10. In the Item Size section, clear the Same as Detail [check box](#).
11. In the Width box, type the number of inches you want to allow for each column of data. For this example, enter 2.
12. Under Item Layout, select Vertical. This creates a snaking column report.
13. Choose OK.
14. Click on the group Header Section and display the [property sheet](#). For the "New Row Or Col" property, choose Before Section. This causes the [group header](#) to print at the top of the column one time for each new group.
15. Print or Preview the report.

For other articles describing how to print the group header at the top of continued columns, search in this Knowledge Base for:

print and group's and name and subsequent and column and page

References:

"Microsoft User's Guide," version 1.0, chapter 18, "Designing Reports," and chapter 19, "Sorting and Grouping Data."

For additional information on columnar reports search for "Multiple-column reports" and "NewRowOrCol" using Microsoft Access Help.

[References](#)

PRB: ReportWizard Select No Fields for Form Still Shows All

Article Number: Q90869
CREATED: 22-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Using the ReportWizard and not choosing anything for the fields you want on your report will give you a report that includes all the fields.

CAUSE

Based on feedback from users, this is the desired result of selecting the Go To Last Page button, which bypasses the entire design questioning of AccessWizards.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Choose the Report button and the New button in the Database window.
2. Select a table or query, then choose the ReportWizards button.
3. Select one of the three types of reports, then choose the OK button.
3. Choose the Next button without selecting any fields.
4. Select one of the three available looks for your report, then choose the Next button.
5. Choose the Print Preview button.

Note that your report includes all the available fields even though you did not choose any fields for your report.

References

PRB: Sizing Rows and Columns to Zero Works Differently

Article Number: Q90868
CREATED: 22-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In Table [datasheet](#) view, if you attempt to size a [row](#) height to zero, you receive the error message:

The setting you entered isn't valid for this property.

However, you will be able to set [column](#) widths to zero.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Problem

-
1. Open a new or existing [table](#).
 2. Choose a row and select Row Height from the Layout menu. Set the height to zero and choose OK.

[References](#)

PRB: Select All Not Grayed Even When No Items Are Available

Article Number: Q90867
CREATED: 22-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In [Form view](#), the Select All option on the Edit menu is never unavailable (dimmed), even when there are no items to select.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. From the [Database window](#), choose the Form button.
2. Choose the New button and then the Blank Form button to create a new [form](#) with no controls.
3. From the Edit menu, choose Select All. Select All is available, even though there is nothing on the form to select.

[References](#)

INF: Commonly Asked Questions About Access Version 1.0

Article Number: Q90866
CREATED: 22-OCT-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

Commonly asked questions about Microsoft Access version 1.0 are listed below.

MORE INFORMATION

- 1.Q. Is it possible to customize the Help file in Microsoft Access?
- A. Customized Help files can be used only in Microsoft Access forms. Use the HelpContextId and HelpFile properties of the [form](#) to reference a Help file previously constructed with the Microsoft Windows Help Compiler. For more information on this, search for "HelpContextId" in the Microsoft Access online Help.
- 2.Q. Is it possible to set up a many-to-many [relationship](#)?
- A. No. The only kinds of relationships you can define are one-to-many and one-to-one. It is generally not a good [database](#) design to have many-to-many relationships. An alternative is to build a linking [table](#) between two tables that have a many-to-many relationship. For example, if you had an "Employees" table and a "Job Skills" table, each employee may have many skills, and each skill could belong to more than one employee--a many-to-many relationship. To work around this, you would build a linking table "EmpSkill" that had two columns, EmpID and SkillID, to relate the two tables.
- 3.Q. Where are [OLE](#) objects (such as the pictures in the Employees table in NWIND.MDB) stored?
- A. OLE objects can be either linked or embedded. Embedded objects are stored inside the .MDB file. Linked objects point to a file located external to the database, and are therefore not stored in the .MDB file. Since the Employees' pictures within NWIND.MDB are embedded OLE objects, the pictures are stored in NWIND.MDB.
- 4.Q. How can I tell if an [OLE object](#) is embedded or linked?
- A. To tell if an object is linked or embedded, first select the object, then choose the last command on the Edit menu. The command name varies depending on the object type. (For example, "Paintbrush Picture Object," "Microsoft Excel Worksheet Object," and so on.) When the option is highlighted or selected, a [submenu](#) appears to the right of the menu. If the object in question is an

embedded object, the Change Link option will be unavailable (dimmed). If it is a linked object, the Change Link option will not be dimmed.

- 5.Q. Is all the information (tables, queries, reports, and so on) that appears as part of a database actually stored in the database?
- A. All database information is stored in the database, with the exception of database "preferences" (choose Options from the View menu) and certain security information (users, their passwords, and group information), which are stored in the SYSTEM.MDA file.
- 6.Q. Is it possible to break out the information that is contained in a database (that is, tables) into individual files?
- A. You can use the Export option to export tables and other data to separate files (ASCII, dBASE, and so on). You can also export Microsoft Access objects to other Microsoft Access databases.
- 7.Q. Is some type of tool available that helps document applications? (For example, variables used, macros used, properties of different interface elements, and so on.)
- A. A tool called "Database Analyzer" is included with Microsoft Access that aids in this task. For information on using this tool, look in the PSSKB.TXT file, question number 23, or choose the Microsoft Access Q&A icon in the Access group in the Microsoft Windows Program Manager.
- 8.Q. Are memo fields part of a table, or are they stored in a separately linked file as is done in FoxPro, which has .DBF and .FPT files?
- A. They are part of a table and thus stored in the .MDB file.
- 9.Q. Can Microsoft Access create stand-alone applications?
- A. Currently, Microsoft Access does not create stand-alone applications. Since there is not yet a run-time version of Microsoft Access, a developer and his or her customers both require full working copies of Microsoft Access to run Microsoft Access applications. The developer distributes only the .MDB file(s) for the application to the customers.

A run-time version of Microsoft Access is planned for the near future that a developer can distribute with his or her .MDB file(s) so that customers will not require a full working copy of Microsoft Access to run applications.

[References](#)

PRB: GP Fault If MSACCESS.INI Installable ISAM Path Uses '..'

Article Number: Q90865
CREATED: 22-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A general protection (GP) fault occurs if the path in the "Installable ISAMs" section of the MSACCESS.INI file (located in the WINDOWS directory) contains the characters "\". For example:

```
[Installable ISAMs]
dBASE IV=..\ACCESS\DBSISAM.DLL
```

The GP fault occurs whether or not the path and/or file is valid.

RESOLUTION

Use explicit paths when referring to files in the MSACCESS.INI file. For example:

```
[Installable ISAMs]
dBASE IV=C:\ACCESS\DBSISAM.DLL
```

STATUS

Microsoft has confirmed this to be a problem in Microsoft Windows operating system version 3.1. We are researching this problem and will post new information here as it becomes available.

[References](#)

INF: Access Saves Printer Information with Each Form/Report

Article Number: Q90864
CREATED: 22-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access saves printer driver information with each form or report that it generates. This means you do not have to set up the printer with the correct driver in order to print out a form or report--the form or report "remembers" the correct driver and uses it. For example, you can save one form using a PostScript printer driver and another form using the generic text driver.

NOTE: If the Default Printer option button is selected, the form will print to the Windows default printer.

MORE INFORMATION

If the printer that the printer driver requires is not currently connected, Microsoft Access displays a dialog box stating that the printer is not available and allows you to select another printer.

Bear in mind that some text formatting depends on the current printer driver. For example, if you format a document with a PostScript printer font instead of a software font (such as a TrueType font) and send the document to a PostScript printer, everything should print out correctly. However, if you send the document to a LaserJet, you will get output errors because some PostScript printer fonts are not available on a LaserJet.

A driver-related problem could be something such as the sudden appearance of the Zapf Dingbats font in a form or report set up to use the Windows System font and sent to a PostScript printer. This problem occurs because the Windows System font is not available on PostScript printers. When a font is unavailable, Windows attempts to find an equivalent among those on the printer, searching through them alphabetically. In this case, it arrives at the Zapf Dingbats font last and uses it because there are no other options.

The best way to avoid these problems is to create all your forms and reports with TrueType fonts, which appear correctly in print preview and are processed the same way by all printers.

References

INF: Older Versions of Shared DLLs Cause Problems with Access

Article Number: Q90863
CREATED: 22-OCT-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Shared dynamic-[link](#) libraries (DLLs) in your system older than those supplied with Microsoft Access version 1.0 or Microsoft Windows version 3.1 can cause unexpected errors with Microsoft Access.

To avoid problems, make sure that all shared DLLs are at least as current as those supplied with Microsoft Windows version 3.1 or Microsoft Access version 1.0, and that they are all located in the WINDOWS\SYSTEM subdirectory.

MORE INFORMATION

The shared DLLs used by Microsoft Access are:

COMMDLG.DLL	89248 bytes	3/10/92	Windows 3.1
COMMDLG.DLL	89248 bytes	10/25/92	Access 1.0
COMMDLG.DLL	89248 bytes	5/21/93	Access 1.1
COMMDLG.DLL	97984 bytes	10/1/92	Windows for Workgroups
OLECLI.DLL	83456 bytes		Same for all
OLESVR.DLL	24064 bytes		Same for all
DDEML.DLL	36864 bytes	3/10/92	Windows 3.1
DDEML.DLL	36964 bytes	10/25/92	Access 1.0
DDEML.DLL	36964 bytes	5/21/93	Access 1.1
DDEML.DLL	38400 bytes	10/1/92	Windows for Workgroups
SHELL.DLL	41600 bytes	3/10/92	Windows 3.1
SHELL.DLL	41600 bytes	10/25/92	Access 1.0
SHELL.DLL	41600 bytes	5/21/93	Access 1.1
SHELL.DLL	41520 bytes	10/1/92	Windows for Workgroups
VER.DLL	9008 bytes		Same for all

(These DLLs are used by Windows applications other than Microsoft Access, as shown above. The DLLs for which no date is provided are the same for all applications.)

Microsoft Access supplies the same DLL versions as Windows 3.1, and it requires these (or newer) versions for correct operation; older versions can cause errors.

Microsoft Access and Windows copy shared DLLs into the WINDOWS\SYSTEM

subdirectory, and that is where Microsoft Access first looks for them. If it does not find one or more of the shared DLLs, it looks in the directory where Microsoft Access is installed, and then in the current directory.

Often, Microsoft Access finds outdated or incorrect DLLs because third-party software with old or foreign DLLs has been installed or reinstalled, overwriting the correct DLLs. Some third-party software packages copy DLLs into directories other than WINDOWS\SYSTEM, creating multiple copies and making the problem harder to track and resolve.

Microsoft Access will not reload DLLs previously loaded by another software package; it will only load shared DLLs not already loaded. If the previously loaded DLLs are incorrect, they can cause problems with Microsoft Access.

If the Microsoft Access Setup program fails to locate a shared DLL, or detects an older version of one, it issues an error message such as:

Outdated XXXX.DLL found. Please reinstall MSAccess.

-or-

Can't find XXXX.DLL.

[References](#)

INF: Using the Windows 3.1 API to Connect to Network Resources

Article Number: Q90862
CREATED: 22-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The Microsoft Windows version 3.1 application programming interface (API) can be used to connect to and disconnect from [network](#) drives and printers in an [Access Basic module](#).

MORE INFORMATION

To use the API functions, follow the steps below. You will probably want to use variables for the parameters.

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

1. Declare the functions required to add and remove network connections. Type the lines in a module in the Global Declarations section.

```
'-----  
Option Explicit  
Declare Function WNetAddConnection% Lib "User" (ByVal lpszNetPath$, _  
                                                ByVal lpszPassword$, _  
                                                ByVal lpszLocalName$)  
  
Declare Function WNetCancelConnection% Lib "User" (ByVal lpszName$, _  
                                                  ByVal fForce%)  
  
Const WN_SUCCESS=0           ' The function was successful.  
Const WN_NET_ERROR=2        ' An error occurred on the network.  
Const WN_BAD_PASSWORD=6     ' The password was invalid.  
  
Dim Results%, Force%  
'-----
```

2. Create a function that makes the connection.

```
Function AddConnection ()  
    ' lpszLocalName$ can be in the form D: or E: and LPT1 or LPT2.  
    Results% = WNetAddConnection("\\server\share", "password", "y:")  
End Function
```

Some of the possible return values for Results% are WN_SUCCESS, WN_NET_ERROR, and WN_BAD_PASSWORD.

For more information on how to obtain an unused drive resource programmatically, search the Knowledge Base for the following keywords:

network and getdrivetype

3. Create a function that cancels the connection. The parameter `fForce%` specifies whether any open files or open print jobs on the device should be closed before the connection is canceled. If this parameter is `FALSE` and there are open files or jobs, the connection will not be canceled.

```
Function CancelConnection ()
    Force%=1
    ' NOTE: If Force%=0 and files were open,
    '       the connection will not be 'canceled.

    Results% = WNetCancelConnection("y:", 1)
End Function
```

Two of the most common return values for `Results%` are `WN_SUCCESS` and `WN_NET_ERROR`.

References:

"Microsoft Access Introduction to Programming," version 1.0, Chapter 2, "Writing a New Function," pages 8-14

[References](#)

INF: Connecting to the First Available Network Drive

Article Number: Q90861
CREATED: 22-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

To make a [network](#) connection with the Windows API, you can use an unused logical drive letter or a UNC name (\\SERVER\SHARE). You can call the Windows API [function](#) GetDriveType from an [Access Basic module](#) to find the first available drive letter and then make a network connection to this available drive.

MORE INFORMATION

Use the following steps to find the first free drive and make a network connection.

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

1. Declare the API functions.

```
'-----  
Option Explicit  
  
Declare Function GetDriveType% Lib "kernel" (ByVal nDrive%)  
Declare Function WNetAddConnection% Lib "User" (ByVal lpszNetPath$, _  
                                                ByVal lpszPassword$, _  
                                                ByVal lpszLocalName$)  
'-----
```

2. Write a function to return the first free drive and make the new connection.

```
Function FreeDrive ()  
    Dim DriveNum, FirstFreeDrive  
    Dim FirstDrive%, Results%  
  
    DriveNum = -1  
  
    Do  
        DriveNum = DriveNum + 1  
        FirstDrive% = GetDriveType(DriveNum)  
    Loop Until FirstDrive% = 0  
  
    FirstFreeDrive = Chr$(DriveNum + 65) + ":"  
  
    'Substitute the appropriate server share and password.
```

```
Results% = WNetAddConnection("\\server\share", "password", _  
                                FirstFreeDrive)  
End Function
```

References:

"Microsoft Access Introduction to Programming," version 1.0, page 8

[References](#)

PRB: #Error When ControlName Used in ControlSource Expression

Article Number: Q90860
CREATED: 22-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

A [field](#) resulting in "#Error" will be created when the ControlSource property (bound field) contains the ControlName property in an [expression](#). However, in an unbound [text box](#), the same expression generates the expected value rather than "#Error."

CAUSE

There is a circular dependency in either a [form](#) or a [report](#). Control names take precedence over field names when binding. Microsoft Access currently does not provide a more descriptive error for circular references.

RESOLUTION

If you need to reference the ControlName property in an expression, use an unbound text box. Otherwise, reconstruct the ControlSource property expression so that it does not reference the ControlName property.

STATUS

This behavior is by design.

MORE INFORMATION

If you create a bound form and drag a numeric field onto the form, by default the ControlSource property and ControlName property will be given the name of the actual field in the underlying [table](#) or [query](#). If you change the ControlSource property to an expression that includes the ControlName property, you will see "#Error" in this field when you switch to Browse or Datasheet mode. The following demonstrates an expression within the ControlSource property:

```
ControlName: Person_ID  
ControlSource: =[Person_ID] + 1
```

[References](#)

PRB: Fonts May Change Zooming Out in Print Preview

Article Number: Q90859
CREATED: 22-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Zooming out when Previewing a Report may cause the font to change to a generic font.

CAUSE

Microsoft Access chooses fonts so that the font displayed will always be readable. The normal font will print as expected.

RESOLUTION

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, pages 406-413.

[References](#)

PRB: Mail Label Wizard Cuts Off First Field at Top of Page

Article Number: Q90858
CREATED: 22-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The first line on each page in a mailing [label](#) created with the ReportWizard may not print or show in Print Preview. Forms and reports may also exhibit this behavior.

CAUSE

All printers have non-printable areas. Microsoft Access reads the default Windows printer driver to determine what to print or show in Print Preview. If the label layout exceeds the printable area, the labels will be truncated and the first line will not print.

RESOLUTION

If you experience this problem, you should increase the height of your Page Header to align your labels. In some cases the first [row](#) of labels will be unusable.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.00, page 401.

[References](#)

INF: Cut and Paste Removes Blank Lines in Macro's Design View

Article Number: Q90857
CREATED: 22-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you cut and paste or copy and paste multiple rows, and several of the rows are blank, the blank rows will be removed in the [macro's Design view](#).

Although Microsoft Access removes the blank lines, the resulting code is functionally equivalent.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 21

[References](#)

PRB: CancelEvent Action in OnClose Macro Prevents Exit

Article Number: Q90856
CREATED: 22-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

After a [form](#) is opened, Microsoft Access and Microsoft Windows cannot be closed (quit).

CAUSE

If there is a CancelEvent action in an OnClose [macro](#), you will be unable to close Microsoft Access and Windows.

RESOLUTION

If a form's OnClose property specifies a macro that executes a CancelEvent action, you won't be able to close the form. You must either correct the condition that caused the CancelEvent action or open the macro and delete the CancelEvent action. If the form is a [modal](#) form, you won't be able to open the macro.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, pages 549-552

[References](#)

INF: Shrinking Empty or Null OLE Object on Report

Article Number: Q90855
CREATED: 22-OCT-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If a report has object linking and embedding (OLE) objects that are empty or null, you can set up the report so that empty or null OLE objects will not appear.

MORE INFORMATION

To illustrate how this is done, we will use the sample database NWIND.MDB:

1. Start Access and open NWIND.MDB. You may want to make a copy of NWIND.MDB before proceeding.
2. From the View menu, choose Table. Open the Employees table and add a few records without pictures to the end of the table. Close the table.
3. From the View menu, choose Macros. Create a new macro and enter the lines below. Save the macro as Macro99.

Condition	Action
Reports![Report1]![photo] Is <u>Null</u>	SetValue
Reports![Report1]![photo] Is Not Null	SetValue

Macro99 Actions

SetValue
Item: [photo].visible
Expression: False

SetValue
Item: [photo].visible
Expression: True

Line 1 of this macro checks to see if the OLE object is null or empty. If it is, the Visible property is set to 0 (False), making the object invisible. For Line 2, if the OLE object is not null or empty, the Visible property will be set back to -1 (True) and the picture will be visible.

4. From the View menu, choose Report.

5. In the Database window, choose the New button.
6. When prompted, base the report on the Employees table and select Blank Report instead of the Report Wizard.
7. Drag Employee ID, Last Name, First Name, and Photo from the list of fields and place them in the detail section of the report.
8. If the property sheet is not showing, choose Properties from the View menu, or click the Properties button on the toolbar.
9. Click the detail section bar (the bar labeled "Detail"). On the property sheet, change the following properties:

 On Format: Macro99
 Can Shrink: Yes
10. In the toolbox, click the text box button. Add an unbound text box to your report that completely overlaps the OLE object. The text box is necessary to cause the items below it to be pulled up.
11. For this unbound text box, change the following properties:

 Visible: NO
 CanShrink: YES
12. From the Layout menu, choose Send To Back. The OLE object should now be on top of the unbound text box.
13. Save the report as Report1.
14. Scroll through the pages of the report in print preview. Scroll to the last page and note how the CanShrink property works with null or empty OLE objects.

When an OLE object is visible, the CanShrink property will prevent the empty text box from shrinking. When the OLE object is invisible, the text box will shrink to nothing.

References

INF: Double-click Also Executes OnPush Twice

Article Number: Q90853
CREATED: 22-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

On a [form](#) using the OnPush and OnDbClick events, the double-click runs the OnPush [event](#) before and after the OnDbClick event.

MORE INFORMATION

If you want the OnDbClick event to execute without the OnPush event, you must put an explicit CancelEvent action at the end of the OnDbClick [macro](#). Otherwise, both clicks will execute the macro assigned to the OnPush event.

Steps to Reproduce Behavior

1. Create a macro with a beep action. Save it as Macro1.
2. Create a second macro with a MsgBox action. Save it as Macro2.
3. Create a blank form and add a [command button](#).
4. Set the OnPush property for the command button to Macro1 and the OnDbClick property to Macro2.
6. From the View menu, choose Form.
7. Double-click the command button.

Note that the OnPush event is activated before and after the OnDbClick event.

[References](#)

INF: ComboBox On Toolbar Do Not Use Window Background Color

Article Number: Q90852
CREATED: 22-OCT-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The Microsoft Access Combo Box on the [toolbar](#) does not use the window background color, so changing the background color from the Color icon within the Control Panel does not have any effect.

MORE INFORMATION

Microsoft Access toolbars are predefined forms. There is no way to specify the windows background as a color on a [form](#). If you would like to change colors within Microsoft Access, you can open and edit the forms within UTILITY.MDA. If you modify UTILITY.MDA, extreme care must be taken by creating backups of UTILITY.MDA before making any modifications.

Steps to Reproduce Behavior

1. Change the Window Background color in the Control Panel.
2. Start Microsoft Access and open the sample [database](#) NWIND.MDB.
3. Select the Table command from the View menu.
4. Select the Customer [table](#) by double clicking on it with the mouse.

At this point the "Field:" dropdown [combo box](#) still has a white background.

[References](#)

INF: Network Users Can Open Their Own SYSTEM.MDA Exclusively

Article Number: Q90851
CREATED: 22-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Two users on the same network can open the same SYSTEM.MDA file exclusively. Microsoft Access ignores the exclusive lock on a SYSTEM.MDA file in a multiuser installation.

MORE INFORMATION

Microsoft Access opens the system database nonexclusively on the network. Subsequent requests to open the database ignore the read-only and exclusive flags.

Since the SYSTEM.MDA file is opened whenever a user logs into the database, opening the SYSTEM.MDA file exclusively would lock out all other users on the network.

[References](#)

INF: Setting Form's Dynaset Property Puts Bookmark at EOF

Article Number: Q90850
CREATED: 22-OCT-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

In [Access Basic](#), you can make a copy of the [dynaset](#) that a [form](#) may currently be using for data manipulation.

The [syntax](#) to accomplish this is:

```
Dim MyDB as Database
Dim MyDyn as Dynaset
Set MyDB = OpenDatabase("NWIND.MDB")
Set MyDyn = Forms!Categories.Dynaset
```

In this case, you may assume that the bookmark is at the beginning of the Categories dynaset. However, the bookmark is at the end-of-file (EOF) marker. In fact, after a [variable](#) is set to the dynaset of a form, there is no [current record](#) because the bookmark is actually at the EOF marker.

MORE INFORMATION

Before doing any further actions, the two dynasets should either be synchronized or the bookmark should be set to the first [record](#). This is done with the following statement:

```
MyDyn.Bookmark = Forms!Categories.Bookmark ' Synch dynasets.
```

-or-

```
MyDyn.MoveFirst ' Move to first record.
```

After one of these statements has been issued, there will be a current record and the bookmark will be pointing at this current record.

References:

"Microsoft Access Language Reference," version 1.0, Chapter 8,

[References](#)

INF: Attached Table Must Be Opened as Dynaset from Modules

Article Number: Q90817
CREATED: 22-OCT-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

In Access Basic modules, a table can be opened as the Table data type in the following manner:

```
Dim MyDB as Database
Dim MyTable as Table
Set MyDB = OpenDatabase("NWIND.MDB")
Set MyTable = MyDB.OpenTable("Categories")
```

However, if you have an attached table that is in the database, it cannot be opened as the Table data type. An attached table must be opened as the Dynaset data type in the following manner:

```
DIM MyDB as Database
DIM MyDyn as Dynaset
Set MyDB = OpenDatabase("NWIND.MDB")
Set MyDyn = MyDB.CreateDynaset("AttachedTable")
```

If you try to execute the OpenTable action on an attached table, you will receive the following error message:

```
Can't perform operation; it is illegal
```

MORE INFORMATION

When you refer to a Table data type, the table must reside in the .MDB file. This is why an attached database must be opened as a dynaset. An attached table only has a pointer to where the table resides.

References:

"Microsoft Access Language Reference," version 1.0, Chapter 8

INF: Freestanding Labels Do Not Appear in Datasheet View

Article Number: Q90816
CREATED: 22-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

The final sentence on page 221 of the "Microsoft Access User's Guide" is misleading. It should read as follows:

Freestanding labels do not appear in Datasheet view. Freestanding (unbound) text boxes will appear in Datasheet view with their control names as column headings.

[References](#)

INF: How to Create For Next Loops in a Macro

Article Number: Q90815
CREATED: 22-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

To perform a For Next loop in a Microsoft Access [macro](#), you need to use a combination of correct macro structure and the RunMacro action with appropriate information supplied in either the Repeat Count or Repeat Expression parameters.

To enhance your macro, you can use a different macro structure and include a call to an [Access Basic function](#) that maintains a counter, or use the SetValue macro action to maintain and [update](#) a counter value in a [text box](#) on a hidden [form](#). The benefit of maintaining a counter is that you can use the value of the counter in your macro processing just as in a Access Basic For Next loop.

MORE INFORMATION

To perform a For Next loop in a macro, you would use a macro similar to the following examples. To create the following sample macros, open a new macro and choose Macro Names from the View menu. Add the following macro names and actions, then save this macro [group](#) as "For_Next_Loops".

For Next Loop Using Repeat Count

=====

Macro Name	Action

For_Next_Loop1 Loop1	RunMacro MsgBox
For_Next_Loop1 Actions	

RunMacro	
MacroName: For_Next_Loops.Loop1	
Repeat Count: 10	
Loop1 Actions	

MsgBox	
Message: ="Loop"	

In this example, the For_Next_Loop1 macro calls the Loop1 macro 10

times. The number of times the macro is called is specified in the Repeat Count parameter of the RunMacro action.

The limitation of this example is that a counter value is not available to indicate which instance of the loop is being executed. As a result, there is no way to tell if this is the first, fifth, or tenth time the loop is being executed.

For Next Loop That Maintains a Counter in a Text Box on a Form

Macro Name	Action
------------	--------

For_Next_Loop2	SetValue RunMacro
Loop3	MsgBox SetValue

For_Next_Loop2 Actions

SetValue

Item: Forms![CounterForm]![Counter]
Expression: 1

RunMacro

MacroName: For_Next_Loops.Loop2
Repeat Expression: Forms![CounterForm]![Counter]<=10

Loop2 Actions

MsgBox

Message: ="Loop Count: " & Forms![CounterForm]![Counter]

SetValue

Item: Forms![CounterForm]![Counter]
Expression: Forms![CounterForm]![Counter]+1

In this example, the For_Next_Loop2 macro uses the SetValue macro action to initialize (set to a value of 1) a counter in a text box called Counter on a form called CounterForm. It then runs the Loop2 macro until the Repeat Expression parameter evaluates to false. In this example, the loop executes 10 times. The Loop2 macro displays a message box that indicates which instance of the loop is being executed by referring to the counter value in the Counter text box. The Loop2 macro then increments the counter (adds 1 to the current counter value) by using the SetValue macro action.

The benefit of this example is that a counter value is available to indicate which instance of the loop is being executed. The drawback to this example is that it depends on a form that must be created and opened prior to the execution of the loop. The form can be hidden or visible, but it must be open for this example to work.

For Next Loop Using Access Basic Function to Maintain a Counter

```

-----
Macro Name      Action
-----
For_Next_Loop3  RunCode
                  RunMacro
Loop3           MsgBox
                  RunCode

```

```

For_Next_Loop3 Actions
-----

```

```

RunCode
  Function Name: =LoopCount(1)

```

```

RunMacro
  MacroName: For_Next_Loops.Loop3
  Repeat Expression: LoopCount(3)<=10

```

```

Loop3 Actions
-----

```

```

MsgBox
  Message: ="Loop Count: " & LoopCount(3)

```

```

RunCode
  Function Name: =LoopCount(2)

```

In this example, the For_Next_Loop3 macro runs the LoopCount(1) function by using the RunCode macro action. This causes the function to initialize the counter to a value of 1. It then runs the Loop2 macro using the RunMacro action until the Repeat Expression parameter is false. LoopCount(3), which is used in the conditional expression, returns the current value of the counter. Therefore, when the counter reaches 11, the loop has executed 10 times, and when the RunMacro action checks the condition, it evaluates to false and will not call the Loop3 macro again. The Loop3 macro displays a message box that indicates which instance of the loop is being executed by referring to the counter value returned by the LoopCount(3) function call. The Loop3 macro then increments the counter using the RunCode macro action, which calls the function LoopCount(2).

The benefits of this example are that a counter is available that indicates which instance of the loop is being executed and you do not need to create and open a form with a control on it to contain the counter value. The drawback to this example is that you need to create the LoopCount() Access Basic function prior to using this example; however, after you create the function it is available anywhere in Microsoft Access.

The Access Basic function procedure LoopCount() returns the current value of a counter that the function declares and maintains. Depending on which parameter is passed (1, 2, or any other number), LoopCount() will initialize, increment, or return the counter's current value. Below is the Access Basic code for the LoopCount() function procedure.

```
Function LoopCount (Action)
    Static LoopCounter          ' Static variable to hold counter.
    If Action = 1 Then          ' Initialize counter.
        LoopCounter = 0
    ElseIf Action = 2 Then      ' Increment counter.
        LoopCounter = LoopCounter + 1
    End If
    LoopCount = LoopCounter     ' Return value.
End Function
```

References:

"Microsoft Access Basic: Introduction to Programming," version 1.0,
chapter 3, pages 6-7, 10-11

"Microsoft Access Language Reference," version 1.0, pages 246-248,
316-317, 412-414, 437-438

[References](#)

INF: How to Create Do While/Do Until Loops in a Macro

Article Number: Q90814
CREATED: 22-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

To perform a Do While or Do Until loop in a Microsoft Access [macro](#), you need to use a combination of correct macro structure and the RunMacro action with appropriate information supplied in the Repeat Expression parameter. These requirements are described in more detail below.

MORE INFORMATION

There are several types of Do loops. Each type handles the looping procedure and conditions differently. The different types of Do loops are:

1. Do While <condition>... Loop

This loop will execute while the condition is true. If the loop is encountered and the condition is already false, the loop will not be performed.

2. Do Until <condition>... Loop

This loop will execute until the condition is true. If the loop is encountered and the condition is already true, the loop will not be performed.

3. Do... Loop While <condition>

This loop will execute the first time unconditionally, then loop while the condition is true. If the loop is encountered and the condition is already false, the loop will be performed once.

4. Do... Loop Until <condition>

This loop will execute the first time unconditionally, then loop until the condition is true. If the loop is encountered and the condition is already true, the loop will be performed once.

To perform a Do While or Do Until loop in a macro, you would use a macro similar to the following examples. To create the following examples, open a new macro and choose Macro Names from the View menu. Add the following macro names and actions and save this macro [group](#) as Do_Loops.

Do While <Condition>... Loop and Do Until <Condition>... Loop

Macro Name Action

Do_Loop1 RunMacro
Loop1 MsgBox

Do_Loop1 Actions

RunMacro
 MacroName: Do_Loops.Loop1
 Repeat Expression: <your_condition>

Loop1 Actions

MsgBox
 Message: ="Loop"

In this example, the Do_Loop1 macro calls the Loop1 macro while the Repeat Expression parameter of the RunMacro action is true.

Do... Loop While <Condition> and Do... Loop Until <Condition>

Macro Name Action

Do_Loop1 RunMacro
 RunMacro
Loop1 MsgBox

Do_Loop1 Actions

RunMacro
 MacroName: Do_Loops.Loop1

RunMacro
 MacroName: Do_Loops.Loop1
 Repeat Expression: <your_condition>

Loop1 Actions

MsgBox
 Message: ="Loop"

In this example, the Do_Loop1 macro calls the Loop1 macro once unconditionally, then continues to call the Loop1 macro while the Repeat Expression parameter of the RunMacro action is True.

Conditions

The condition used in the loop, which is supplied in the Repeat Expression parameter of the RunMacro action, can be based on a value in a field on a form, a property of a control on a form, or the value returned from a Microsoft Access Basic function. If the condition is based on a field in a form or a property of a control on a form, it will have syntax similar to:

```
Forms![CounterForm]![Counter]<=10
```

-or-

```
Forms![EntryForm]![InvoiceNo].Visible=True
```

If the condition is based on the value returned from a Access Basic function, it will have syntax similar to:

```
Time()>=TimeEntry()
```

Do While Loops vs. Do Until Loops

Whether a loop is a Do While loop or a Do Until loop depends on the condition. The RunMacro action runs the macro until the condition in the Repeat Expression parameter is false. This behavior is exactly what is needed for a Do While loop. It is, however, the opposite of what is needed for a Do Until loop. Therefore, to make a condition for a Do Until loop work correctly, precede the condition in the RunMacro action with the Not operator.

References:

"Microsoft Access Language Reference," version 1.0, pages 316-317, 412-414

[References](#)

INF: MoveSize Arguments in Wrong Order in Language Reference

Article Number: Q90813
CREATED: 22-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The documentation on the [syntax](#) for the MoveSize [macro](#) action incorrectly states the order of arguments as:

```
[right][,down][,height][,width]
```

The correct order of the arguments, as stated in the Microsoft Access Help topic for the MoveSize action, is:

```
[right][,down][,width][,height]
```

References:

"Microsoft Access Language Reference," version 1.0, pages 315-316.

For more information, search for "MoveSize" then "[Access Basic](#)" using the Microsoft Access Help menu.

[References](#)

PRB: Can't Refer to Record in Function Called by OnClose Event

Article Number: Q90812
CREATED: 22-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you attempt to refer to the last record that was displayed by a form in a function or macro that was triggered by the OnClose event of the form, instead of receiving information about the last record that was displayed, you will receive information about the first record of the bound table or query.

CAUSE

This behavior occurs because the OnClose event is the last action that occurs before the form is closed. By the time the OnClose event occurs, the form has released its pointer to the record in the bound query or table and can no longer refer to it. If you were stepping through the macro or code that was called by the OnClose event, you would still see the information in the form, but because the pointer to that data no longer exists, you would not be able to access that data.

RESOLUTION

If you need to refer to the last record that was displayed by a form prior to closing, you will need to use a method that uses another form or Microsoft Access Basic code to maintain a duplicate copy of the current record that is displayed in the form. The macro or function that does this should be called by the form's OnCurrent event so that every time you change the record information, the macro makes a copy on the separate form or in Microsoft Access Basic variables. Then, when the OnClose event is triggered, your macro or function can use the data on the separate form or in the Microsoft Access Basic variables because they will still be available.

STATUS

This behavior is by design.

References

INF: How to Check a Menu Item Using Access Basic

Article Number: Q90811
CREATED: 22-OCT-1992
MODIFIED: 13-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Microsoft Access does not have any built-in [macro](#) or Basic commands that allow you to place a check mark beside or remove a check mark from a menu command. To do this in Microsoft Access, you must use [Access Basic](#) code to call Microsoft Windows application programming interface (API) functions.

More Information:

The following Windows API [function](#) is used to check and uncheck a menu item:

CheckMenuItem% (hMenu%, wIDCheckItem%, wCheck%)

This function places check marks next to or removes check marks from menu items in the pop-up menu specified by the hMenu% parameter.

hMenu%	Identifies the menu.
wIDCheckItem%	Specifies the menu item to be checked.
wCheck%	Specifies how to check the menu item. The wCheck% parameter can be a combination of the MF_CHECKED or MF_UNCHECKED with MF_BYPOSITION or MF_BYCOMMAND flags. These flags can be combined by using the bitwise OR operator. The values are described as follows:
Value	Meaning
-----	-----
MF_BYCOMMAND	Specifies that the idCheckItem parameter gives the menu-item identifier (MF_BYCOMMAND is the default).
MF_BYPOSITION	Specifies that the idCheckItem parameter gives the position of the menu item (the first item is at position zero).
MF_CHECKED	Selects the item (adds check mark).

MF_UNCHECKED Clears the item (removes check mark).

Return Value The return value specifies the previous state of the item. It is either MF_CHECKED or MF_UNCHECKED. The return value is -1 if the menu item does not exist.

Note: Top-level menu items (items on the Microsoft Access menu bar) cannot have a check mark.

When determining the position of a menu command, separator bars count as commands. For example, to get to the Import menu command on the File menu in the Database window, use the arguments 0 (for the File menu) and 7 (for the Import command). Positions 3 and 6 correspond to the separator bars on the File menu.

Example

The example below creates a menu, associated with a form, that can be used to check and uncheck a menu command.

1. Create a new macro. After adding the following actions and their associated properties, save the macro and name it "Menu Checking Macro." To display the "Macro Name" column, choose Macro Name from the View menu.

Macro Name	Action	Function Name
&Check	RunCode	Check_Menu_Item(0,0)
&Uncheck	RunCode	Uncheck_Menu_Item(0,0)

Note: The first parameter in the function indicates which menu on the menu bar to use. All menus are zero based. For example, if your menu bar contains the File, Edit, Window, and Help menus, and you specify 2 as the first parameter for one of these functions, the function will work on the Window menu. A 2 indicates the Window menu instead of the Edit menu because the parameter uses base 0 instead of base 1; therefore, 0 indicates the File menu, 1 indicates the Edit menu, and so on.

The second parameter in the function indicates which menu command (on the menu indicated in the first parameter) to place the check mark beside. This parameter also uses base 0. To refer to the first command on the menu, use 0; to refer to the second, use 1; and so on.

2. Create a new macro. After adding the action below and its associated property, save the macro as "Custom Menu".

Action

AddMenu
Menu Name: &Check
Menu Macro Name: Menu Checking Macro

3. Create a new blank form and display the Properties window of the form by choosing Properties from the View menu.
4. Set the OnMenu property to "Custom Menu".
5. Close the form by choosing Close from the File menu. Save the form as "Menu Checking Form".
6. Create a new module from the Database window. In the new module, enter the Access Basic code listed at the end of this document. Save the module as "Menu Checking Code".
7. From the Database window, select the "Menu Checking Form" form and then choose the Open button to display the form in Form view. The normal Microsoft Access menu will disappear and be replaced by the custom menu that was designed above.
8. There are two options on the Check menu. If you choose the Check command, the command will be checked. If you choose the UnCheck command, the Check command will be unchecked.

Note: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```

'*****
'Declarations section of the module.
'*****
Option Explicit

Declare Function GetMenu% Lib "user" (ByVal hWnd%)
Declare Function GetSubMenu% Lib "user" (ByVal hSubMenu%, ByVal nPos%)
Declare Function CheckMenuItem% Lib "user" (ByVal hSubMenu%,_
    ByVal nPos%, ByVal Flag%)
Declare Function FindWindow% Lib "user" (ByVal lpClassName As Any,_
    ByVal lpCaption As Any)
Declare Function IsZoomed% Lib "user" (ByVal hWnd%)

Const MF_BYPOSITION = &H400
Const MF_BYCOMMAND = &H0
Const MF_CHECK = &H8
Const MF_UNCHECKED = &H0
Const MyNull = 0&
Const ClassName = "OMain"

Dim ChWnd%           ' Handle to the Microsoft Access window.
Dim hMenuTop%        ' Handle to the Microsoft Access menu.
Dim hSubMenu%        ' Handle to the pop-up menu associated with hMenuTop%.
Dim ItemID%          ' Command ID associated with a specified menu item.
Dim ReturnVal%       ' Function return value may be used depending on
                    ' the function.

'=====
' This function displays a check mark next to the menu item.
'=====
Function Check_Menu_Item% (TopLevel%, SubLevel%)

```

```

'If the form is maximized, the system menu is added to the forms
'menu bar, so increment the actual TopLevel%
If (IsZoomed(Screen.ActiveForm.hWnd)) Then
    TopLevel% = TopLevel% + 1
End If

Call Get_Menu_Handles(TopLevel%)
' The following statement must appear on one line.
Check_Menu_Item = CheckMenuItem(hSubMenu%, SubLevel%,_
    MF_BYPOSITION Or MF_CHECK)
End Function

'=====
' This function initializes:
'   - The window handles associated with the Microsoft Access form.
'   - The handle to the menu of the specified window.
'   - The menu handle of the specified pop-up menu of the Window menu.
' The variables here are global to the database.
'=====
Sub Get_Menu_Handles (TopLevel%)
    ChWnd% = FindWindow(Classname, MyNull)
    hMenuTop% = GetMenu(ChWnd%)
    hSubMenu% = GetSubMenu(hMenuTop%, TopLevel%)
End Sub

'=====
' This function will uncheck a menu item.
'=====
Function UnCheck_Menu_Item% (TopLevel%, SubLevel%)
    Call Get_Menu_Handles(TopLevel%)
    ' The following statement must appear on one line.
    UnCheck_Menu_Item% = CheckMenuItem(hSubMenu%, SubLevel%,_
        MF_BYPOSITION Or MF_UNCHECKED)
End Function

```

References:

"Microsoft Access User's Guide," version 1.0, pages 596-602

For more information on similar programming features, [query](#) on the following words here in the Microsoft Knowledge Base:

GetSubMenu

[References](#)

INF: How to Force a New Line in a MsgBox Statement

Article Number: Q90810
CREATED: 22-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

To force a new line in a message box, enter the message as a [string](#) formula and concatenate a carriage return/line feed, Chr\$(13) + Chr\$(10), into the message anywhere you want the message to start a new line.

MORE INFORMATION

For example, if you had the message:

Notice: This is an Important Message!

And instead of having it all display on one line, or letting Microsoft Access determine where to break the message onto a new line, you want to force it to display as:

Notice:
This is an Important Message!

To force the message to display like this you need to enter the message as a string formula. If you call the message box from a [macro](#) the [syntax](#) would be:

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
Action
-----
MsgBox
    Message: ="Notice:" & Chr(13) & Chr(10) & "This is an Important_
            Message!"
```

If you call the message box from a macro the syntax would be:

```
MsgBox "Notice:" & Chr(13) & Chr(10) & "This is an Important_
      Message!"
```

These formulas place "Notice:" on the first line and insert 1 carriage return/line feed to force a hard carriage return between the 2 lines of text.

References:

"Microsoft Access Language Reference," version 1.0, pages 19, 66, 316-320.

[References](#)

INF: How to Use a Query to Filter Unique Data

Article Number: Q90809
CREATED: 22-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you want to filter a table to eliminate duplicate data, you need to use a query that uses one of the aggregate functions--First(), Last(), Min(), or Max()--in the fields that DO NOT contain the duplicate data. The fields that contain the duplicate data should contain the Group By() function.

MORE INFORMATION

For example, suppose you import an inventory table from an application and discover that the data has duplicates in it. Your goal is to get the data back down to a baseline, such as one record per product, so you can then reinventory and have a correct and complete set of data. You can use a query to filter the data.

Your table might look like this:

ProdID	Description	Cost	MarkUp	Quantity
1	A Product	\$1.50	0.5	10
2	B Product	\$2.50	0.7	100
3	C Product	\$1.59	0.9	25
2	D Product	\$4.59	0.8	30
5	E Product	\$1.99	0.7	40
6	F Product	\$2.69	0.4	60
9	G Product	\$4.95	0.8	20
8	H Product	\$6.79	0.9	32
9	I Product	\$6.89	0.7	0
1	J Product	\$2.99	0.5	11

If you want to filter the table down to the point where it has a unique ProdID code and take the first entry from each of the other fields, you can create a query to do this, as follows:

1. Create a new query based on the original table by selecting it from the table list and clicking the New Query button on the toolbar.
2. Drag all the fields from the Field list to the query grid.
3. From the View menu, choose Totals.
4. Set the Total line of the query grid to First() for every field except ProdID, which is set to Group By().

5. From the View menu, choose Datasheet.

The data you see should be a list of unique ProdID data with the first value that is encountered for that product in each of the other fields. If you use this procedure on the above data your result should be:

ProdID	Description	Cost	MarkUp	Quantity
1	A Product	\$1.50	0.5	10
2	B Product	\$2.50	0.7	100
3	C Product	\$1.59	0.9	25
5	E Product	\$1.99	0.7	40
6	F Product	\$2.69	0.4	60
8	H Product	\$6.79	0.9	32
9	G Product	\$4.95	0.8	20

To obtain different results, you can use Max(), Min(), or Last() instead of First().

To generate a unique table from this query, you could change the query to a [make-table query](#) during the query design.

References:

"Microsoft Access User's Guide," version 1.0, pages 139-144

[References](#)

PRB: HelpContextID Does Not Validate Negative Values

Article Number: Q90808
CREATED: 22-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Entering a value of less than 0 in the HelpContextID property of a [form](#) does not generate an error message.

STATUS

This behavior is by design.

MORE INFORMATION

The acceptable value range is 0 to 2,147,483,647. Numbers above 2,147,483,647 generate an error message, but numbers less than 0 do not. The HelpContextID is just a number that Microsoft Access passes to the Microsoft Windows application programming interface (API) WinHelp() [function](#), which takes an unsigned long. Microsoft Access property sheets do not recognize unsigned longs, so large numbers show up as negative numbers instead.

Steps to Reproduce Behavior

1. Create a new item on a form.
2. Open the [property sheet](#).
3. Enter a negative number for the HelpContextID; for example, -1000.

Microsoft Access will not generate an error.

[References](#)

PRB: GoToControl to Control with Focus in AfterUpdate Ignored

Article Number: Q90806
CREATED: 22-OCT-1992
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When a [control](#) has the focus, and the GoToControl action is executed with the same control name that has the focus, the focus will move to the next control in the [tab order](#). The focus will not stay on the control that currently has the focus, as expected.

RESOLUTION

To prevent the focus from leaving the [field](#), you can use an OnExit [macro](#) that performs the CancelEvent action. Or, if you want to prevent the field from being saved (and also prevent the focus from leaving the control), you can use a [validation rule](#) or a BeforeUpdate macro.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. In NWIND.MDB, create the following macro:

Condition	Action	Control Name
[Category ID]="aaaa")	GoToControl	[Category ID]

2. Save the macro as Macro1.
3. Open the Categories [form](#) in design mode. Call the macro in the AfterUpdate property of the CategoryID control.
4. View the Categories form in Browse mode.
5. Type "aaaa" into the [Category ID] field and press TAB.

The focus moves to the next field in the tab order. However, if you had moved to another control before pressing TAB, the focus would not have changed.

[References](#)

INF: Cannot Change Borders of Up or Pressed Down Controls

Article Number: Q90804
CREATED: 22-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Setting the border color or border width of an up or pressed down control has no effect. The border color and size is determined by Microsoft Access and cannot be altered by the developer. If you would like to control the border width or the border color, the appearance must be set to normal.

This behavior is by design.

[References](#)

PRB: Object Buttons in Database Window Do Not Change Colors

Article Number: Q90803
CREATED: 22-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Changing the default button colors in Windows Control Panel does not affect the colors of the buttons in the [Database window](#).

CAUSE

These buttons are actually hard-coded bitmaps, and there is no way to dynamically change their face, shadow, and highlight colors.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

-
1. Change one of the button color properties by choosing the Color icon in Control Panel.
 2. Return to Microsoft Access and go to the Database window. All the object buttons (Table, Form, Query, and so on) still use the default colors.

[References](#)

PRB: Invalid Use of Null Alert Assigning Null Control

Article Number: Q90802
CREATED: 22-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Default fields values are set to Null. If you use the Eval() function on a field without a value you will get the following error message:

Invalid Use of Null Alert

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a form (Form1) with a textbox (Field0).
2. Switch to Browse view.
3. Open a module.
4. Type the following line in the immediate window:

```
stForm$ = Eval("Forms!Form1!Field0").
```

Results: Invalid Use of Null Alert.

[References](#)

PRB: Giving Controls Duplicate Name Does not Generate Error

Article Number: Q90725
CREATED: 21-OCT-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you give a control the same name as another control, and you press TAB or ENTER, the following error message appears:

You already have a Control name <control name>

However, if you click outside the property sheet instead of pressing TAB or ENTER, the error message does not appear and the control returns to its original name.

Also, when giving a control a duplicate name in a form and then closing and exiting that form, you will also not receive an error message. The control name will return to it's original name.

CAUSE

Because Microsoft Access cannot prevent you from clicking outside the property sheet, it was decided that clicking outside the property sheet would silently commit any changes and if the changes violated integrity, the changes would be canceled. If Microsoft Access hadn't been designed in this manner, the error message would appear when you were in the wrong window, which might be confusing.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

-
1. Place two text boxes on a form.
 2. Set the control name of one text box to Field0.
 3. Set the control name of the second text box to Field0, but instead of pressing ENTER, click outside the property sheet.

The error message does not occur, and the control reverts to its original name as if you had canceled the event.

References

PRB: Focus Doesn't Shift When Find Accelerators Are Depressed

Article Number: Q90724
CREATED: 21-OCT-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Using the Hot keys on the Find [dialog box](#) to select the Find First button does not change the focus from the Find Next button to the Find First button. However, clicking on the Find First button with the mouse does change the focus.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Start Microsoft Access & Open NWIND.
2. Open "Order Details" [table](#).
3. Click on the FastFind Button (binoculars on the Toolbar.)
4. Set Search as formatted to TRUE.
5. Set Search Text for "10002".
6. The focus is on Find Next so press the accelerator combo for Find First (ALT+S).

Result: The Find First button is depressed as expected, but the focus remains on the Find Next button which originally had the focus. If you select the Find First button using the mouse, the Find First button then receives the focus.

[References](#)

PRB: Load From Query Dialog Title Does Not Match Menu

Article Number: Q90723
CREATED: 21-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The title of the 'Load From Query' [dialog box](#) does not match the menu name that opened the dialog box. Instead 'Applicable Filter' is displayed in the dialog box.

Steps to Reproduce Behavior

-
1. Open a [form](#).
 2. Select the EditFilter command from the Records menu.
 3. Select LoadFromQuery command from the File menu.

RESOLUTION

This behavior is by design.

[References](#)

PRB: Reason Val() Function Returns Invalid Use of Null Message

Article Number: Q90722
CREATED: 21-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you create a [function](#) that uses the [Access Basic](#) function Val() and evaluates a null [field](#) on a [form](#), the following error message is generated:

Invalid Use of [Null](#)

RESOLUTION

The Val() function is giving an [expression](#) evaluation error when NULL fields are encountered.

The Val() function returns the numeric value of a [string](#) of characters.

STATUS

This behavior is by design.

References:

"Microsoft Access Language Reference," version 1.0, page 489.

[References](#)

INF: How to Make Access Opening Screen Display

Article Number: Q90170
CREATED: 11-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

To have the initial opening screen appear when starting Microsoft Access, you can edit the MSACCESS.INI file. Open MSACCESS.INI in Windows Notepad (or any other text editor), and search for "Tutorial=" under the [Microsoft Access] section.

Change the entry from

```
Tutorial=0
```

to

```
Tutorial=1
```

For example:

```
[Microsoft Access]  
Tutorial=1
```

This causes the selection in the option box on the opening screen to be canceled, which means that the initial Access screen will display every time it is started. If you select the box again, the initial screen will not be displayed on startup.

[References](#)

INF: Temporarily Disabling a Macro Action

Article Number: Q90167
CREATED: 11-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access version 1.0 and 1.1

SUMMARY

Programming languages, such as [Access Basic](#), typically provide a way to temporarily disable any [syntax](#) on a line of code. For example, in Access Basic, you can place a REM command at the beginning of a line of code so that it is not executed. This feature is useful when you are [debugging](#) several lines of code because it allows you to determine which line is causing an error by disabling one line at a time until the error no longer occurs.

This [method](#) can also be useful for debugging macros. This article describes how you can temporarily disable a [macro](#) line.

MORE INFORMATION

To disable a macro action line without deleting it:

1. Open the macro sheet containing the line you want to disable.
2. If there is no Conditions [column](#), choose Conditions from the View menu to open the Conditions column.
3. In the Conditions column, on the line you want to disable, type "False" (without quotation marks) if there is nothing in the column for that line, or "False And" (without quotation marks) if a condition is already entered in the column.

Now, when you run the macro, the Condition for the macro action will be evaluated as false and the line will be ignored.

[References](#)

PRB: SCROLL LOCK Has No Effect in Form Browse

Article Number: Q90162
CREATED: 11-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Pressing the SCROLL LOCK key in Microsoft Access has no effect in Form or Table Browse mode.

STATUS

This behavior is by design.

MORE INFORMATION

Microsoft Access does not use the SCROLL LOCK key in the same way as Microsoft Excel. Pressing SCROLL LOCK in Excel locks the current position of the cursor and moves the data around within this area. This behavior was designed into Microsoft Excel, but not into Microsoft Access.

[References](#)

PRB: Opening Screen Size Is Different from Earlier Session

Article Number: Q90161
CREATED: 11-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

When starting Microsoft Access, the window position and size may not be the same as it was when Microsoft Access was last closed. This behavior is by design. Microsoft Access does not currently have any support to maintain the window state between sessions.

MORE INFORMATION

Microsoft Windows determines where an application appears. Some applications remember if they were maximized, and some keep window state information relative to the screen and size of a window. Microsoft Access remembers if it was closed while maximized, and will restart that way if it was.

[References](#)

INF: Minimized Popup Forms Cannot be Easily Restored

Article Number: Q90160
CREATED: 11-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you use the Minimize action in a [macro](#) to minimize a Popup [form](#) in Microsoft Access, it may be difficult to restore the Popup form.

MORE INFORMATION

If the mouse double click is on the icon's caption, it will restore the Popup form as any other icon restores the window state.

Windows does not allow the restore of a window unless it has the Minimize Box on the window. Popup forms in Microsoft Access do not have a Minimize Box and therefore they work differently.

References:

"Microsoft Access User's Guide" version 1.0, Chapter 9, Designing Forms.

[References](#)

PRACC9210: Find Toolbar Button Not Grayed After Pressing

Article Number: Q90158
CREATED: 11-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

After pressing the F7 key the Find Button on the Toolbar is not grayed out.

RESOLUTION

Instead, if the Find command on the Edit menu is selected, or or the user presses the Find button, the Find button is grayed out.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here as it becomes available.

Additional References:

"Microsoft Access Getting Started Guide," version 1.0, page 117.

[References](#)

INF: Border Widths Not Supported with Raised or Sunken Effects

Article Number: Q90156
CREATED: 11-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Border width properties are not applicable to controls with raised or sunken properties set. You can effectively mimic these features of Microsoft Access by using the line [control](#).

MORE INFORMATION

When you are in Form Design mode, setting the BorderWidth property in the palette screen will not have any affect on a control if the raised or sunken effects are enabled.

To work around this behavior, you can enable these properties for the control(s) in question and then, using the line control, create your own sunken or raised effect. You can set the width of the line to whatever you desire. The sunken and/or raised effect is accomplished by placing lines around a standard control.

Additional References:

"Microsoft Access User's Guide," version 1.0, chapter 9.

[References](#)

INF: Horizontal Sizing Controls Availability on Controls

Article Number: Q90155
CREATED: 11-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

After placing a [control](#) on a [form](#) or [report](#), the horizontal sizing controls will only be enabled if the Height property of the control is greater than or equal to .26 inches. The default size for many controls is less than .26 inches, so horizontal sizing points will not appear.

This means it could be very difficult to alter the horizontal size of a control if the snap to grid option is enabled, because the slightest movement causes a snap to grid to occur.

One workaround is to temporarily set the size of a control to .26 inches, alter the horizontal size, then reset the Height property to its original value. Alternatively, the Height property can be set as the default for the control. To do this, select a control on the tool bar, display the Properties list, and change the default Height property.

References:

"Microsoft Access User's Guide," version 1.0, chapter 9, pages 213-268.

[References](#)

INF: Form Must Close for Modal & Popup Settings to Take Effect

Article Number: Q90154
CREATED: 11-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you change a [form](#)'s Modal or PopUp property while designing the form, the changes will not take effect until after the form is closed and re-opened. Simply switching to browse mode will not reflect the changes, even after the properties are entered.

References:

"Microsoft Access Users Guide," version 1.0, chapter 9, pages 213-268

[References](#)

INF: Bitmaps Hide the Caption of a Push or Option Button

Article Number: Q90153
CREATED: 11-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article discusses the limitations of using Microsoft Access push button or [option button](#) controls with bitmaps.

MORE INFORMATION

Microsoft Access has a Picture Property setting on command buttons that can be used to import a [bitmap](#) that will be displayed on the button. This setting will place the caption "behind" the picture and the caption will no longer be visible.

References:

"Microsoft Access Users Guide," version 1.0, chapter 8, pages 188-211.

[References](#)

INF: Paradox Tables, Indexes, and Multiple Connections

Article Number: Q90152
CREATED: 11-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If a Paradox table does not have a primary key, then it cannot be opened twice, even exclusively. This means it cannot be used in more than one place in a query or code.

MORE INFORMATION

If you try to attach to a Paradox table that has no primary key, you will get an error stating that the operation is not supported on Paradox tables without a primary key. This can be confusing, because you can create the query that accesses the table twice, but if you try to use it, you will get this error. In addition, if anyone tries to get the output column list of the query, this error will occur.

[References](#)

INF: Windows for Workgroups, Novell, SQL Server, and Access

Article Number: Q90151
CREATED: 11-OCT-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you are running [SQL](#) Server on a NetWare [network](#) and try to add Microsoft Access and Windows for Workgroups, you will not be able to talk to SQL Server from Microsoft Access version 1.0. Windows for Workgroups and SQL Server have NetWare support, but the two are currently incompatible.

MORE INFORMATION

Named pipes (the communication protocol used by SQL Server) is serviced from NETAPI.DLL. The Windows for Workgroups NETAPI.DLL overrides the Novell NETAPI.DLL, so communication with SQL Server is broken.

You can work around this problem by using the Netware Integration Kit (NIK) for Novell. The NIK enables Microsoft Access and Windows for Workgroups to communicate with SQL Server on a Novell network.

You can order the NIK from Microsoft Consumer Sales at (800) 426-9400. Microsoft Consumer Sales is open Monday through Friday, 6:30 A.M. to 5:30 P.M., Pacific time.

[References](#)

INF: Custom Menus Cannot Be Implemented with Pop-up Forms

Article Number: Q90150
CREATED: 11-OCT-1992
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

A [form](#) will not process the [macro](#) associated with the OnMenu property if the form is designed with the PopUp property set to True. A form that is designed as a pop-up window does not have any menus associated with the form. If you want a custom menu when the pop-up window is created, you must have a separate non pop-up window create and maintain the custom menu.

According to "The Windows Interface - An Application Design Guide," dialogs (pop-up) do not generally have menu bars.

References:

"The Windows Interface - An Application Design Guide," page 57, Microsoft Press, 1992.

[References](#)

INF: Exporting to SQL Server Does Not Create Indexes

Article Number: Q90149
CREATED: 11-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access can [update](#) an attached [SQL](#) Server [table](#) only if the table has a unique [index](#). However, Microsoft Access does not build indexes on the SQL Server table when exporting information.

MORE INFORMATION

Microsoft Access allows you to export data to SQL Server backends. However, Microsoft Access does not build the index on export, thus creating potential problems and confusion. For example, if you then attach the table, updates are not allowed and the data is displayed as read only in [datasheet](#) or [form](#) view.

To work around this export problem, manually create the SQL Server table indexes. After the indexes are created, you can create forms based on the attached SQL Server tables and the data will be updateable.

References:

For more information on creating indexes on SQL Server [database](#) tables, see the SQL Server "System Administrator's Guide" or pages 67-72 in the SQL Server "Language Reference."

[References](#)

PRB: Access Not Prompting for Second Parameter in Query

Article Number: Q90148
CREATED: 11-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

Microsoft Access does not prompt you for the second parameter in your query. But it did prompt you for the first parameter in your query.

CAUSE

The criteria fields are identical for both parameters. Each prompt for a user-entered parameter must contain a unique text string.

RESOLUTION

Modify the text in the criteria field for one of your parameters. Then Microsoft Access will be able to distinguish between the two parameters.

MORE INFORMATION

For example, a query prompts the user for the Product ID and Product Name from the Products table in the Northwind Traders sample database shipped with Microsoft Access. This query will find information based on both values or either value. However, the prompt values placed in the criteria fields of the query grid cannot be the same for both fields. The following example is incorrect:

Field:	Product ID	Product Name
Sort:		
Show:	X	X
Criteria:	[Enter Value:]	
or:		[Enter Value:]

Microsoft Access considers the two parameter prompts, [Enter Value:], to be one prompt. Once the first is filled with the user's search criteria, Microsoft Access thinks the second is filled as well.

The following is an example of a multiple-parameter query that prompts for both values:

Field:	Product ID	Product Name
Sort:		
Show:	X	X
Criteria:	[Enter Product ID:]	
or:		[Enter Name:]

If these prompts are used, Microsoft Access will prompt for both values.

[References](#)

INF: Form OnOpen Event May Not Set Values of Unbound Controls

Article Number: Q90146
CREATED: 11-OCT-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

If you have a [form](#) or [report](#) with unbound controls that you want to set by a [macro](#) called from the OnOpen [event](#), that macro should include a GoToControl or GoToRecord action before the SetValue action(s). The OnOpen event occurs as the object(form or report) is opened but before the first [record](#) is displayed. At this time there is no [current record](#) or [control](#).

You must have a current record or control before Microsoft Access can refer to any controls on the form or report.

MORE INFORMATION

The OnOpen event for forms and reports occurs when the object is opened but before there is a current record. To force the focus to a specific control in order to set its value before the first record is displayed, insert a GoToControl or GoToRecord action before the SetValue action in the macro called from the OnOpen event.

If the controls' values should come from a specific record, use the GoToRecord action in your macro before the SetValue action. If you wish to set the value of individual controls, use the GoToControl action.

Actions such as GoToControl or GoToRecord, by default, refer to the object the macro was called from. Therefore, the [action argument](#) "Item" for each GoToControl and SetValue action should contain only the Control Name of the control you wish to set the value of.

References:

For more information, search for "OnOpen" then "OnOpen, OnClose Properties>" using the Microsoft Access Help menu.

"Microsoft User's Guide," version 1.0, chapter 22, "Using Macros with Forms," pages 528, 538-542

[References](#)

PRB: Memory Block Not Freed If Cue Cards Not Custom Installed

Article Number: Q90143
CREATED: 11-OCT-1992
MODIFIED: 14-JUL-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SYMPTOMS

If you did not install [Cue Cards](#) during a custom installation of Microsoft Access, and later you copy the CUECARDS.DLL file to the directory in which you installed the MSACCESS.EXE file, an error occurs. When you choose one of the Cue Card buttons and then choose OK to exit Cue Cards, the following error message is displayed:

1 Global memory block is not freed.

CAUSE

Cue Cards have been installed incorrectly.

RESOLUTION

When you quit the current session of Microsoft Access, this block of memory is released. To correctly install Cue Cards, rerun the Setup program and choose Custom Installation. Select only the Cue Card installation.

STATUS

This behavior is by design.

[References](#)

PRB: Toggle Button with CancelEvent on BeforeUpdate Property

Article Number: Q90142
CREATED: 11-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you have a form containing a toggle button with the BeforeUpdate property set for a macro that performs the CancelEvent action, the toggle button will behave differently than text boxes with the same BeforeUpdate property setting.

Open a form that has an toggle button defined in this manner. Click the toggle button, then tab to the next control. You will see that the record is dirtied and the toggle button is not pressed down. Move to another record. If you click a text box, then tab to another control, the record is not dirtied.

CAUSE

Text boxes have a state in which they hold a dirty value that cannot be saved or validated. This is why you can click a text box and, without making a change, move to another control.

Toggle buttons do not have a static state holding a dirty value that cannot be validated. When the CancelEvent occurs, the button is left in its original, up state. The button code does not consider the button to be dirty after the save failed. It has the original clean value.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, page 529

[References](#)

PRB: DoCmd OpenReport Opens Rpt in 3 Views or Prints to File

Article Number: Q90141
CREATED: 11-OCT-1992
MODIFIED: 07-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

You can enter an Access Basic command in the Immediate window to open a report in a view that is not available when you create a macro using the OpenReport action. The Output As dialog box appears when you type the following command in the Immediate window:

```
DoCmd OpenReport "My Report", 3
```

The following error message is displayed if you try to save the report using the Output As dialog box:

```
Action OpenReport was canceled.
```

STATUS

This behavior is by design.

MORE INFORMATION

You can create a macro that will open an existing report in one of three views. The view arguments supported in macros are as follows:

```
0 Print
1 Design
2 Print Preview
```

The following Access Basic command does not cause an error message to appear, even though it calls an argument that is not supported from within a macro:

```
DoCmd OpenReport "My Report", 3
```

The Output As dialog box is displayed for the user to specify the name of the text file to which that output should be directed.

REFERENCES

=====

"Microsoft Access Language Reference," version 1.0, pages 355-356

INF: 'Err Msg: Operation Not Supported on Paradox Table...'

Article Number: Q90140
CREATED: 11-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you create a Microsoft Access [report](#) based on a nonkeyed Paradox [table](#), or when you attempt to run a Microsoft Access [query](#) with the table added twice, you will receive the following error message:

Operation not supported on Paradox table with no [primary key](#).

MORE INFORMATION

In Microsoft Access, if you attach to a Paradox version 3.0 or 3.5 table, it is best to have a key [field](#) defined in the table. Paradox creates an [index](#) (.PX) file that should always be in the same directory as the table (.DB) file. Important information is stored in the .PX file. Attaching to nonkeyed Paradox tables limits the report and query capabilities of Microsoft Access.

Query Limitation

You can create a nonupdatable query based on a nonkeyed Paradox table. However, you can add the table only one time to the query. If you add the table twice, you will receive the error mentioned above when you attempt to run the query.

Report Limitation

You cannot use or design a Microsoft Access report based on a query using nonkeyed Paradox tables.

When you create a new report and select the query to base the report on, the error above will appear. Choose the OK button to close the error [dialog box](#) and view the report in Design mode. If you look at the Field List for the report, you will not see any fields listed.

The above limitations occur because attached non-keyed Paradox tables can only be opened once in Microsoft Access. Use of the table in both of these scenarios requires the table to be opened more than one time in order to perform the specific Microsoft Access task.

Paradox is manufactured by Borland International, Inc, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

References:

For more information, search for "Paradox" using the Microsoft Access Help menu.

"Microsoft Access User's Guide," version 1.0, pages 62-63.

[References](#)

PRB: #Error Appears Using TRIM() in Forms or Reports

Article Number: Q90139
CREATED: 11-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you have a [form](#) or [report](#) with the RecordSource property defined as a [table](#) that contains no records, "#Error" will be displayed when you [preview](#) the report or browse the form if a bound [control](#)(s) is defined using the TRIM() [function](#).

CAUSE

When an [expression](#) is used to define the ControlSource property for a form or report, Microsoft Access evaluates the expression and presents the result for the control. If a blank [field](#) is evaluated, "#Error" appears.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a table with two text fields named Field 1 and Field 2.
2. Create a report bound to this table.
3. Place the fields in the detail section of the report.
4. Modify the ControlSource property for Field 1. Enter the following:

```
=Trim([Field One])
```

5. Preview the report.

You will see "#Error" where data would appear for Field 1. Field 2 will be blank.

6. Create a form bound to the table.
7. Place the fields in the detail section of the form.
8. Repeat step 4.
9. Browse the form.

You will see the same results as for step 5.

References:

"Microsoft Access User's Guide," version 1.0, page 485-486

[References](#)

PRB: Pressing CTRL+ALT+SPACEBAR Enters Control's Default Value

Article Number: Q90138
CREATED: 11-OCT-1992
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

When you create a [form](#) based on a [table](#) that has one or more fields defined with the [field](#) property DefaultValue, pressing CTRL+ALT+SPACEBAR inserts the [control](#)'s default value defined in the form, not the table's default value. If no value has been specified for the control's Default Value property, then CTRL+ALT+SPACEBAR has no effect.

When you are entering data in the table's [datasheet](#) mode, pressing CTRL+ALT+SPACEBAR enters the field's default value, as defined in the table.

CAUSE

When you create a form based on a table with one or more fields defined with the field property DefaultValue, new controls bound to such a field will have the control's DefaultValue property filled in with the same default value. You may keep this value, change it, or delete it.

When you enter new records using form Browse mode, any control that has a DefaultValue property will be displayed with the control's default value. If you decide that you want to have the default value that was defined in the table entered, you will have to type it in. In form Browse mode, when you press CTRL+ALT+SPACEBAR, the default value defined for the control is entered.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a table with a text field called City and a text field called State.
2. For the State field, enter "NC" for the DefaultValue property.
3. Create a form based on this table.
4. Add text boxes for the City and State fields.

5. Change the DefaultValue property for the control bound to the State field to "WA".

6. Browse the form.

The first record has "WA" displayed in the State field.

7. Delete "WA" and enter "CA". Commit the record.

8. Delete "CA" and press CTRL+ALT+SPACEBAR. Note that "WA" is inserted.

References:

"Microsoft Access User's Guide," version 1.0, pages 229-230

[References](#)

PRB: Report Cannot be Opened or Previewed

Article Number: Q90137
CREATED: 11-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you open a report in Design mode and switch to Preview mode, the report is not previewed. In addition, if you select the report from the Database window and choose to open the report, nothing happens.

CAUSE

The report's OnOpen event may have a macro attached to it that has the first action listed as StopAllMacros. This macro action acts like the CancelEvent action, which prevents the report from opening.

The StopAllMacros action stops all macros that are currently running. In this case, the report's OnOpen event calls a macro that is immediately terminated, which aborts the preview of the report.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new macro and add the StopAllMacros action.
2. Save the macro as Macro1.
3. Open a report in design mode.
4. Set the Report Property OnOpen event to Macro1.
5. Preview the report. Note that nothing happens.

References:

"Microsoft Access User's Guide," version 1.0, pages 522, 566.

[References](#)

PRB: Limited to 65535 Pages in Print Dialog Box

Article Number: Q90136
CREATED: 11-OCT-1992
MODIFIED: 23-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

In the Print dialog box the largest value allowed in the Pages To: control is 65535. If a number greater than 65535 is entered, the following message is displayed:

The 'To' field contains non-numeric characters or a number greater than 65535

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide." Version 1.0, pages 31-32, 209, 410.

[References](#)

PRB: Form Displays Data from Deleted Field in Query

Article Number: Q90135
CREATED: 11-OCT-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you have a [form](#) based on a [query](#) and you remove a [field](#) from the query grid, you may still see the data from the deleted field in Form view.

You will not see the data from the deleted field if you save and the reopen the form. The [text box](#) will contain the "#Name?" error message.

CAUSE

A form is based on the compiled query. The query must be recompiled before the form recognizes the absence of the field. Microsoft Access does not recompile queries "on the fly."

RESOLUTION

When the form is closed and reopened, the query is recompiled. The following procedure describes another way to have the form recompile the query:

1. With the form in [Design view](#), place your cursor in the text box of the newly deleted field.
2. Press ENTER.
3. From the View menu, choose Form. Note the error message in the Title text box.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Start Microsoft Access and open the sample [database](#) NWIND.MDB.
2. Create a new query based on the Employees [table](#).
3. Drag the LastName and Title fields to the query grid.
4. Save the query as Query1.

5. Create a new form based on Query1.
6. Drag all the columns from the field list to the form.
7. From the View menu, choose Form.
8. Give the query focus. Delete the Title column from the grid. Save the query.
9. Give the focus back to the form (in Form view).

REFERENCES

=====

"Microsoft Access User's Guide," version 1.0, page 100

[References](#)

PRACC9211: Cannot Cascade or Tile with 50 Windows in Access

Article Number: Q90134
CREATED: 11-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you have 50 Microsoft Access objects open and you choose Cascade or Tile from the Window menu, the screen does not change. No error message is displayed.

CAUSE

There is not enough available memory for Windows to perform the operation.

RESOLUTION

Close some of the open windows.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here as it becomes available.

MORE INFORMATION

To duplicate this scenario, create 49 [query](#) windows. The Database window counts as one window. Choose Cascade or Tile from the Window menu. Nothing will change.

[References](#)

INF: Access Databases Truncated w/ LAN Manager 2.1a & Earlier

Article Number: Q90133
CREATED: 11-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
 - Microsoft LAN Manager versions 2.1a and earlier
-

SUMMARY

There is a known problem in Microsoft LAN Manager versions 2.1a and earlier that causes file truncation when the workstation (client) machine is faster than the server. This problem is not specific to Microsoft Access, but can cause irreversible damage to a networked database if Microsoft Access is used in conjunction with Microsoft LAN Manager versions 2.1a and earlier.

MORE INFORMATION

When a client workstation sends information to the server through LAN Manager version 2.1a or earlier, LAN Manager performs a "lazy" write (that is, the data is buffered in memory and then written to disk at a later time). If the client machine sends information to the server too quickly, the server sends a message to the client telling it to send information slower or to wait. With LAN Manager versions 2.1a and earlier, this "slow down" message is misinterpreted by the client redirector and the client writes 0 bytes to the file, effectively truncating it wherever the write was occurring.

This problem is specific to the lazy writing process. When the client writes to the server, it sends the data in a burst. This burst overloads the server and elicits the slow down message.

To work around this problem, disable the lazy write function of the LAN Manager client. This can be done by adding the following line to the [Workstation] section in the LANMAN.INI file:

```
;.....012345678901234567890123456789012  
wrkheuristics=X0XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

NOTE: The top line with the numbers is only a comment line for spacing. The total number of x's and 0's (zeros) is 33 decimal.

This workaround should be used with Microsoft LAN Manager versions 2.1a and earlier. This problem was corrected in Microsoft Windows for Workgroups.

NOTE: This information is also located in the README.TXT file.

References

PRB: Combo Boxes Open When Window Sized to Vertical Minimum

Article Number: Q90132
CREATED: 11-OCT-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Combo box drop-down list boxes still open even when the window is sized to a vertical minimum. Therefore, they appear as list boxes in the middle of the screen, apparently unattached to any object.

CAUSE

Even though the [combo box](#) itself isn't visible, it still has the focus and still responds to key combinations that affect it, such as ALT+DOWN ARROW.

RESOLUTION

To work around this behavior, make sure the [Query window](#) or Access window is not smaller than the [query](#) grid, or change the focus to another [control](#) prior to shrinking the window size.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce:

1. Create a new query based on some [table](#), such as the Categories table in the Northwind [database](#) (NWIND.MDB.)
2. Place the cursor in the first [cell](#) of the query grid ("Field:").

Note that when you press ALT+DOWN ARROW, the combo box list drops down.

3. Using its window size handle, shrink either the Query window or the Access window by pulling the bottom of the window up to its title bar.
4. Press ALT+DOWN ARROW.

Note that the combo box list still drops down, yet appears not to be attached to any combo box.

[References](#)

PRB: Grid Disappears When Measurement Changed to Metric

Article Number: Q90131
CREATED: 11-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

The grid disappears, after changing the measurement option under Control Panel/International from "English" to "Metric".

CAUSE

The grid is only invisible; it is still active and functioning correctly. (This behavior only occurs if the Grid option of the View menu is activated.)

The grid is not visible because it becomes too dense to display when the GridX/GridY properties are set to over 6 divisions per centimeter. Since the default value for GridX is 10, and 12 for GridY, the values make the grid too dense when measurement is switched to metric. The largest number of divisions per centimeter the grid can be set to and still be visible, is 6, (using the [form](#)'s GridX/GridY properties) regardless of the video card or monitor resolution.

RESOLUTION

Either set the number of divisions in the [form properties](#) to less than 7, or switch the measurement value (in Control Panel/International) to "English". However, be aware that even when using the "English" measurement option, setting the GridX/GridY values to anything over 16 divisions per inch will also make the grid disappear.

STATUS

This behavior is by design.

[References](#)

PRACC9210: Transaction Limit Can Cause 'Out of Memory' Error

Article Number: Q90130
CREATED: 11-OCT-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you attempt to make a change to a [database](#), an "Insufficient Disk Space or Memory" error message occurs, even if there are ample system resources and/or disk space.

CAUSE

Microsoft Access has a 4MB [transaction](#) limit. All [table](#) design changes are embedded within a transaction to ensure database integrity. As a result, you will get the "Out of memory" error if you attempt to [update](#) the contents of a table this large or larger.

RESOLUTION

If, for example, a table contains a [field](#) of type "double" that you want to change to "integer," there are two ways to do this:

Method 1

1. Select and copy the desired table to the [Clipboard](#).
2. From the Edit menu, choose Paste (paste only the table structure to a new table name).
3. Change the [data type](#) of the desired [column](#) in this new table.
4. From the Edit menu, choose Paste (paste append the data to this new table).

Method 2

1. Add a new column to the table with the data type you wanted.
2. Execute an update [query](#), where the new column is assigned the value of the column you wanted to change.
3. Delete the old column from the table.

Note: You may get similar "out of memory" alerts when using both of these methods, but the message boxes should give you the option of continuing or canceling.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. We are researching this problem and will post new information here as it becomes available.

[References](#)

INF: How Access Deletes Objects from a Database

Article Number: Q90129
CREATED: 11-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The following is a description of what happens internally when an object (table, form, query, and so on) is deleted from a database in Microsoft Access.

MORE INFORMATION

When you delete an object from the Database window, it is not physically deleted from the database at that moment. Internally, it is renamed to a temporary name (for example, ~TMPCLP1) and hidden, so that it disappears from the Database container. (The renaming operation occurs very quickly, regardless of the database's size.) The object is physically deleted from the database once you take any action other than Undo Deletion.

Objects are deleted this way in order to allow you to undo an accidental deletion, unless some other action has been taken that then makes "undoing" an object deletion impossible. (Microsoft Access can undo commands only one level back; it cannot queue up multiple commands to undo actions.)

Since the physical object deletion occurs on the next action taken, instead of when Delete is chosen, the machine may appear to be hung while it is performing the delete because the next action taken might be one that should not take a long time to perform, such as creating a new table, pasting, browsing a form, and so on. If the object is large enough, the deletion can take some time, making the machine appear to be hung.

NOTE: If the machine is rebooted before the deletion is completed, the hidden temporary object remains in the database, wasting space. To remove hidden objects, compact the database.

Speed of Deletion

The time it takes to delete an object depends on how large it is and how fast the computer is that you are using. Smaller objects are deleted faster than larger ones. If the database that contains the object to be deleted is on a network file server, the deletion will take longer.

How long an object deletion will take cannot be accurately predicted, since the speed of deleting the object is also affected by the nature

of the data contained in it. (For example, with tables, number columns are deleted more quickly than memo and OLE columns.)

Internally, Microsoft Access does not provide a status callback for object deletion, so a progress indicator in the status bar is not possible. Since there is no progress indicator, Microsoft Access will not yield to the system until the object deletion procedure is complete. Therefore, you cannot abort the operation or switch to another application during object deletion.

[References](#)

PRB: Microsoft Access GP Fault with ATI Ultra Video Card

Article Number: Q90121
CREATED: 08-OCT-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If you are using an ATI Graphics Ultra video driver and select 1024 x 768 resolution, you will get a general protection GP-fault when you create a new [form](#) and choose Align from the Layout menu unless first selecting a [control](#) or [field](#) on the form. This error cannot be reproduced using an ATI Graphics Ultra video driver at a lower resolution or with any of the video drivers packaged with Microsoft Windows version 3.1.

RESOLUTION

To work around this problem, select a control on the form before choosing Align from the Layout menu.

STATUS

This behavior does not occur with the latest ATI video drivers (build59 or later). These drivers are available from ATI or on CompuServe by entering GO ATITECH.

MORE INFORMATION

Microsoft has only been able to reproduce this problem on the ATI Graphics Ultra video card running in 8514 mode (that is, 1024 x 768).

Steps to Reproduce Behavior

-
1. Create a new form.
 2. From the Layout menu, choose Align.

The following error message appears:

```
MSACCESS caused a General Protection Fault in module
<unknown>022F:D7C8@|MSACCESS will close.
```

[References](#)

PRB: CTRL+BREAK Makes Mouse Disappear With a Module Active

Article Number: Q90120
CREATED: 08-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Pressing CTRL+BREAK causes the mouse to disappear when a [module](#) is active.

The following steps can be used to reproduce this behavior

1. Start Microsoft Access.
2. Open a [database](#).
3. Open a module.
4. Press CTRL+BREAK.

At this point the mouse will disappear until the mouse pointer is moved.

RESOLUTION

This behavior is by design.

MORE INFORMATION

The module window makes the mouse disappear when you type so that the mouse will not obscure what you are typing. The use of CTRL+BREAK triggers this behavior.

[References](#)

PRB: Cannot Undo Property Settings in Table Design Mode

Article Number: Q90119
CREATED: 08-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Choosing Undo from the Edit menu while in Table Design mode will not completely undo adding a field.

RESOLUTION

To completely Undo the operation, choose Delete Row from the Edit menu.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Choose New-Table from the File Menu.
2. Enter a field name then Tab twice.
3. Choose Undo from the Edit Menu. Note that the Undo command states 'Undo Property Setting'.

The field name will be removed, but the data type will remain.

References:

"Microsoft Access Getting Started," version 1.0, chapter 3, "Creating a Table," pages 36 - 45

[References](#)

PRB: List Box Not Dimmed When Disabled

Article Number: Q90117
CREATED: 08-OCT-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

The contents of a disabled [list box](#) are not dimmed (grayed) as they are in other disabled controls.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open the Northwind [database](#) (NWIND.MDB).
2. Create a new [form](#).
3. Place a list box [control](#) on the form and set its RowSource property to "Customers".
4. Set the Enabled property to "No".
5. Choose the Form View button.

Note that the list box's [label](#) is dimmed, but the contents of the list box are not dimmed, as you would expect from a disabled control.

The problem, however, is only visual; the list box control is truly disabled and you cannot alter its contents.

[References](#)

PRB: LimitToList Combo Box Will not Let You Leave After Delete

Article Number: Q90116
CREATED: 08-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you delete an entry in a [combo box](#), the following error message is displayed when you attempt to leave the [control](#):

Entry must match value in list

CAUSE

This behavior is by design in order to prevent Access from adding nulls to the list just as it would any other new value. Undoing the [record](#) will cause the record to revert to the original stored value.

RESOLUTION

To add a blank entry to a combo box list, include a blank value in the combo box's source [table](#). You can then select the blank value from the list rather than deleting the entry in the [field](#).

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create an unbound [form](#) with a combo box [row](#) source attached to any table. Set the LimitToList property to True.
2. Browse the form. Choose a value from the combo box and press ENTER. Note that the field is committed.
3. Choose the combo box again. Delete the value and attempt to leave the control. The error mentioned above will appear.

References:

"Microsoft Access Getting Started," pages 83-87

"Microsoft Access User's Guide," pages 233-245

[References](#)

PRB: OnDbIcIck Macro Requires CancelEvent to Change Focus

Article Number: Q90115
CREATED: 08-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

The macro action GotoControl does not successfully change the focus to another control if the macro is attached to the OnDbIcIck property of a text box, list box, or combo box.

CAUSE

This behavior occurs because the second click of a double-click is not processed until after the event is completed. The second click on the same field forces Microsoft Access to return the focus to that field.

RESOLUTION

To prevent the second click in the double-click from being processed, add a CancelEvent action to the macro after the GotoControl action.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 21, page 541.

[References](#)

PRB: Nulls Not Concatenated with '+' Operator

Article Number: Q90114
CREATED: 08-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When the "&" operator is used, text can be concatenated to either side of a null. However, when this is done with the "+" operator, a null is returned.

CAUSE

The & operator treats nulls as empty strings. The + operator propagates nulls.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

-
1. Create a form called Form1.
 2. Place a text box field called Field0 on this form.
 3. Switch Form1 to Browse mode.
 4. Add the function:

```
Function Demonstrate ()  
  
    Forms!Form1!Field0 = Null + "This is a string"  
    Debug.Print Forms!Form1!Field0  
    Forms!Form1!Field0 = Null & "This is a string"  
    Debug.Print Forms!Form1!Field0  
  
End Function
```

5. In the Immediate window, type the following and press ENTER:

```
?Demonstrate()
```

The Immediate window will now show the following results:

```
?Demonstrate()  
#NULL#  
This is a string
```

References:

"Microsoft Access User's Guide," version 1.0, Appendix C

[References](#)

PRB: Make-Table Query Causes Existing Table to Be Deleted

Article Number: Q90112
CREATED: 08-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Running a [make-table query](#) that creates a table with the same name as the table the [query](#) is based on deleted the table that the query was based on.

CAUSE

Microsoft Access deletes a table before creating a new table with the same name.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Save an existing table as "Author".
2. Create a make-table query based on Author and type "Author" (without the quotation marks) for the name of the table to be created.
3. Run the make-table query. A [dialog box](#) with the following warning appears:

Existing Table 'Author' will be deleted before running the query. Continue anyway?

If you choose the Yes button, a dialog box with the following error message will appear:

Couldn't find input table or query 'Author'.

At this point, the table "Author" will no longer exist and the query will fail. You will not be able to retrieve the table, even by choosing the Undo command from the Edit menu.

References:

"Microsoft Access User's Guide" version 1.0, Chapter 7, "Designing Action Queries and Parameter Queries," page 164

[References](#)

PRB: 'Couldn't Find Installable ISAM' Error

Article Number: Q90111
CREATED: 08-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Importing, exporting, or attaching to a file using an indexed sequential access [method](#) (ISAM) driver returns an error if Microsoft Access cannot find the ISAM driver.

CAUSE

An ISAM driver is used by Microsoft Access to [update](#) file formats other than its own. If the path to the ISAM driver in the MSACCESS.INI file is invalid or the ISAM driver does not exist, you will receive the following error:

 Couldn't find [installable ISAM](#).

RESOLUTION

Edit the MSACCESS.INI file to specify the correct path, or reinstall Microsoft Access using the Custom option.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide" version 1.0, Chapter 4, "Importing, Exporting, and Attaching," pages 60-69.

[References](#)

PRB: Database Window Loses its Focus

Article Number: Q90110
CREATED: 08-OCT-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

If no documents are open and you have the Database window minimized, pressing ENTER opens the document highlighted in the Database window. If there is another document opened and minimized, nothing happens when you press ENTER on the minimized Database window icon.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open NWIND.
2. Minimize the DataBase Container.
3. Press ENTER.

At this point a table will open.

If you close the table and minimize the object, then press ENTER again, nothing happens. But if you restore and minimize the Database window, the same results occur.

[References](#)

PRB: Pressing SHIFT While Dragging Not Working on New Control

Article Number: Q90109
CREATED: 08-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Pressing SHIFT while dragging a control is used to restrict movement horizontally or vertically. However, this procedure works only if the item is not selected beforehand.

RESOLUTION

A better method, if the control is already selected, is to click anywhere in the form, then reselect the control and move it by pressing SHIFT while dragging the control.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a new form.
2. Place a text box on the form.
3. Hold the SHIFT key down.
4. Click the border of the text box and move it around.

If the text box was not selected before you performed the above steps, you will notice the restricted movement (either horizontal or vertical, depending on which text box border you clicked).

[References](#)

INF: Links to Formatted DDE Items Result in Text

Article Number: Q90108
CREATED: 08-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Dynamic data exchange ([DDE](#)) links to numeric items that have been formatted will result in text. To perform calculations on the results of these links, you must convert the results to a numeric [data type](#), as described below.

MORE INFORMATION

This article will first demonstrate the problem, using an example, then list the Microsoft Access functions that can be used to convert the text value to a numeric value.

Example

Create the formatted value as follows:

1. Start Microsoft Excel.
2. In [cell](#) A1 of Sheet1, type "80" (without the quotation marks).
3. Select cell A1, and choose Number from the Format menu.
4. From the Format Codes list, select a currency format, such as \$#,##0_);(\$#,##0), and choose the OK button.
5. Select cell A1, and choose Copy from the Edit menu.

Create the DDE [link](#) as follows:

1. Start Microsoft Access and open a [database](#).
2. Create a new [form](#).
3. From the Edit menu, choose Paste Special.
4. From the Data Type list, select Text.
5. Choose the Paste Link button. This creates a [text box](#) with the following formula:

```
=DDE("Excel","Sheet1","R1C1")
```

6. If the Properties box is not showing, choose Properties from the

View menu.

7. Select the text box created in step 5 and verify that "Field0" appears in the Control Name property.

8. Create a new text box and enter the formula:

```
=Field0+100
```

9. Switch to browse mode.

Note that the first text box, Field0, displays "\$80". The second displays "#Error".

Convert the text value to a numeric value as follows:

1. Switch back to design mode.

2. Change Field0 to read:

```
=CCur(DDE("Excel","Sheet1","R1C1"))
```

3. Switch back to browse mode to verify that the second text box now contains 180.

Conversion Functions

Part 1 of the "Microsoft Access Language Reference" contains several tables. Table 1 lists functions and statements by programming task. Under the task "Conversion" you will find the following references:

Convert string to number: Val

Convert one numeric data type to another: CCur, CDbl, CInt,
CLng, Csng, Fix, Int

For more information on using these formulas, see the "Microsoft Access Language Reference."

PRB: Minimizing, Maximizing the Desktop Flashes Slightly

Article Number: Q90107
CREATED: 08-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you are using a [bitmap](#) for the Windows desktop, the desktop flashes slightly when you minimize or maximize Microsoft Access.

CAUSE

Microsoft Access is setting the system palette to its own, which changes the colors of all the bits on the screen. You will observe the same behavior if you run Paintbrush, load a 256-color bitmap, and do the same steps as above.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Minimize all extra applications running.
2. Set your desktop to a bitmap (for most noticeable flash, use the 256-color bitmap that comes with Windows 3.1).
3. Open Microsoft Access.
4. Maximize Microsoft Access.
5. Minimize Microsoft Access.

Note the flash as the desktop is repainted.

6. Click the Microsoft Access icon.

NOTE: If you click another minimized application after you click the Microsoft Access icon, the screen again repaints inversely, then repaints correctly.

[References](#)

PRB: Criteria Sections Combined into Same Column in QBE Grid

Article Number: Q90106
CREATED: 08-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Microsoft Access combines multiple columns of the Criteria section in the Query By Example (QBE) grid if two or more columns have the same field name. You will see this combination after saving the query and opening it a second time.

This behavior will typically not present a problem, unless you need to have more than 255 characters in one cell of the QBE grid.

As stated above, the problem occurs when the query is saved and opened a second time. Microsoft Access combines the two Criteria sections into one, making the expression longer than 255 characters. This then causes the following error message to be displayed:

```
Expression is too long for the QBE grid and has been truncated:  
'<Expression>'
```

CAUSE

In Microsoft Access, each cell of the QBE grid is limited to 255 characters. The QBE grid combines expressions when possible so that there are no repetitive columns.

RESOLUTION

To work around the 255-character limit, break the large statement into two smaller statements with the same field name. Make sure that the two smaller statements are on the same Criteria grid line. The query will execute correctly and the two statements will be in their correct places.

To break the statement into two smaller statements, choose one of the two columns that have the same field name and put parentheses around the field name. The following is a sample query:

```
Query1  
-----  
FieldName: UserID  
Criteria: Like "DON*"  
  
FieldName: UserID  
Criteria: Like "DONFU*"
```

When you save and reopen the query, the criteria will be set as:

```
Query1
-----
FieldName: UserID
Criteria: Like "DON*" And Like "DONFU*"

FieldName: UserID
```

If you save the query as follows, the two-column design will be retained:

```
Query1
-----
FieldName: UserID
Criteria: Like "DON*"

FieldName: (UserID)
Criteria: Like "DONFU*"
```

Note that when you save the query and then reopen it in design mode, the second field name will read as follows:

```
Query1
-----
FieldName: UserID
Criteria: Like "DON*"

FieldName: Expr: ([UserID])
Criteria: Like "DONFU*"
```

NOTE: If the field name is more than one word, place brackets around the field name before putting the field name in parentheses.

Putting parentheses around the field name turns it into an expression. This will differentiate the columns and prevent Microsoft Access from combining the two expressions.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 5, "Query Basics."

[References](#)

INF: Couldn't Find SYSTEM.MDA or UTILITY.MDA File

Article Number: Q90105
CREATED: 08-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When Microsoft Access is started, one or both of the following error messages may be displayed:

 Couldn't find file 'SYSTEM.MDA'

 Couldn't find file 'UTILITY.MDA'

These errors occur if the SystemDB and UtilityDB settings in the [Options] section of the MSACCESS.INI file point to a directory that does not contain the SYSTEM.MDA or UTILITY.MDA file. These errors also occur if the directory points to a share in which there is no connection.

If the directory does not exist, the following error message is displayed:

 Invalid Path 'pathname'

MORE INFORMATION

A good way to manage the MSACCESS.INI file is to keep backup copies that pertain to particular setups. Have one setup for starting in a workgroup on a [network](#) and have another that allows for starting on a local machine. To change setups, copy the appropriate backup .INI file over the MSACCESS.INI file.

References:

For more information, search for "MSACCESS.INI" using the Microsoft Access Help menu.

[References](#)

PRB: OpenTable Action Ignores Data Mode If Table Open

Article Number: Q90104
CREATED: 08-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

In a [macro](#), the OpenTable action ignores the Data Mode parameter if the [table](#) being opened is already active. If table A is opened in Read Only mode and then it is opened a second time in Edit Mode without being closed first, the table will still be in Read Only mode.

CAUSE

If OpenForm, OpenTable, or OpenQuery is called, and the window is already opened, you will switch to the new view, but the data mode cannot be switched.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Create a macro, as follows:

MacroName	Action
-----------	--------

Test1	OpenTable
-------	-----------

OpenTable Actions

Table Name:	(Give an existing table name here)
View:	Datasheet
Data Mode:	Read Only

2. Save and run the macro.
3. Leave the table open and return to the [Macro window](#).
4. Make the following change to the OpenTable action of MacroName Test1.

OpenTable Actions

Table Name: (Give an existing table name here)
View: Datasheet
Data Mode: Edit

5. Save the macro and run it again.

When the macro is run a second time (step 5) with a different data mode, the Table window is activated but it is still in Read Only mode. Again, this behavior only occurs if the table is already open when executing the OpenTable action.

References:

"Microsoft Access User's Guide," version 1.0, Chapter 22, "Using Macros with Forms"

[References](#)

INF: How to Abort Creating a Field in Table Design

Article Number: Q90103
CREATED: 08-OCT-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When designing a new table, a valid data type name from the data type combo box must be associated with each field, otherwise, you get the error message:

The text you enter must match an entry in the list.

This is by design, due to field validation performed on exiting the field.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open a new table.
2. Select any data type for the first row from the data type drop-down combo box.
3. Remove the data type selected using either the delete or the backspace key.
4. Try to move to the next row. You get the error message:

The text you enter must match an entry in the list.

The only way to move to another row is to do one of the following:

1. Fill in the Data Type field by selecting (or typing in) a data type from the data type drop-down combo box.
2. Cancel the entry by pressing the ESC key (which fills in the Data Type field with a default entry of "Text").
3. Select "Delete Row" from the Edit menu.

References:

"Microsoft Access User's Guide." Version 1.0, page 27

[References](#)

INF: How to Get SQL Pass-Through Functionality Using Q+E

Article Number: Q90102
CREATED: 08-OCT-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access version 1.0 does not directly support the use of pass-through [SQL](#) (the ability to pass SQL commands directly to a server). However, you can use the Microsoft [Access Basic](#) programming language to perform dynamic data exchange ([DDE](#)) with Q+E, which does support pass-through SQL. This allows you to use pass through SQL commands in Microsoft Access to execute stored procedures and other SQL commands.

MORE INFORMATION

To use Q+E's pass-through SQL capabilities in Microsoft Access, create the following [function](#) in a Microsoft Access Basic [module](#).

NOTE: The second DDEExecute command, which contains the OPEN and USE commands, should be entered on one line similar to the first DDEExecute command.

```
Function SQLPasThru (SQLCommandText, SQLDatabaseText)
    On Error Resume Next
    Err = 0
    chan = DDEInitiate("QE", "System")
    If Err Then
        Err = 0
        Result = Shell("QE", 2)
        If Err Then
            ' Note the following statement should be on one line.
            MsgBox "ERROR: Unable to start Q+E. Q+E must be in your_
                PATH statement", 16, "SQL PasThru"
            Exit Function
        End If
        chan = DDEInitiate("QE", "System")
    End If
    DDEExecute chan, "[LOGON(SQLServer)]"
    DDEExecute chan, "[OPEN('USE " & SQLDatabaseText & "; " &
        SQLCommandText & "', 'SQLSERVER')]"
    DDEExecute chan, "[EXIT()]"
    DDETerminate chan
End Function
```

The SQLPasThru function attempts to initiate a DDE channel with Q+E. If Q+E is not available, SQLPasThru attempts to start Q+E. If Q+E cannot be started, SQLPasThru displays an error and ends the function. If Q+E can be started, a DDE channel is initiated.

SQLPassthru then tells Q+E to display its Logon [dialog box](#) to enable you to log on to SQL server. The SQL USE command is executed to ensure that SQLPassthru is in the [database](#) you specify, and SQLPassthru executes the SQL command. After the command is executed, SQLPassthru exits the Q+E application and terminates the DDE channel the function established.

A call to this function would resemble the following:

```
=SQLPassthru("sp_addlogin ted, ted","master")
```

In this example, the function will start Q+E and initiate a DDE channel, if possible, and prompt you to log on a SQL server. After you have logged on, it will use the master database on the SQL server and execute the stored procedure sp_addlogin, which will add a login for ted with a password of "ted". After the command is executed, the function will exit the Q+E application and terminate the DDE channel it established.

NOTE: You must include quotation marks around both the parameters to ensure that they are used properly. You may also have to include additional quotation marks in the SQL command you are sending to the SQL server to ensure that the command is being processed correctly. For examples of the types of adjustments that you may need to make to the SQL command [string](#), see the "Q+E for Microsoft Excel User's Guide" or the application note "A Practical Guide to Using Q+E," which can be obtained by calling Microsoft Product Support at (206) 454-2030.

You can modify and use this function in many different ways, such as:

- To accept additional parameters and have additional corresponding DDEExecute commands, thereby allowing the execution of multiple SQL commands with one function call
- To use as a Sub in which you can hard-code the parameters for the command and SQL database in the Basic code
- To retrieve information from the SQL server, or send information to the SQL server by using the DDERequest and DDEPoke commands
- To use in conjunction with a [form](#) to allow on-line SQL command execution
- To break into multiple functions with different capabilities, such as one function to initiate the channel, another function to execute the commands, and another function to terminate the channel
- To access database types that Microsoft Access does not yet support but that Q+E does, such as Oracle and Rdb

References:

"Microsoft Access Language Reference," version 1.0, pages 115-126

"Microsoft Access Basic: An Introduction to Programming," version 1.0, Chapter 2, pages 1-8

For additional information on using Q+E to access Oracle and Rdb, see the "Q+E for Microsoft Excel User's Guide," version 4.0, pages 101-102

[References](#)

PRB: When Column Selected and Deleted, Query Grid Disappears

Article Number: Q90101
CREATED: 08-OCT-1992
MODIFIED: 19-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

The Query By Example (QBE) grid, which appears when you open a Microsoft Access query in Design view, disappears when you do the following:

1. Select the grid by highlighting the column header.
2. Extend your selection to the end of the query grid.
3. From the Edit menu, choose Delete.

CAUSE

When you select an entire column on the query grid and choose Delete from the Edit menu, Microsoft Access deletes the columns, rather than deleting the data within the columns.

RESOLUTION

To display a blank column after the query grid has disappeared, choose Insert Column from the Edit menu.

To display a column in which the fields already contain data, either select a field from one of the lists displayed and drag it to the query grid, or double-click the field.

STATUS

This behavior is by design.

REFERENCES

=====

"Microsoft Access User's Guide" version 1.0, pages 92-100

[References](#)

PRB: SQL Server Tables Are Read-Only Without a Unique Index

Article Number: Q90100
CREATED: 08-OCT-1992
MODIFIED: 02-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you try to edit an attached [SQL](#) Server [table](#) that does not have a unique [index](#) defined for it, Microsoft Access beeps and the status bar displays the following error message:

Form is read only.

This error occurs whether you are trying to edit a [form](#) or edit a table in [Datasheet view](#) and applies to all back-end databases.

CAUSE

Microsoft Access requires that a unique index be defined for each table. To edit the records, it is necessary to reference them in the table.

RESOLUTION

With Microsoft SQL Server, you can define a unique index for each table using the SQL command CREATE UNIQUE INDEX. The basic [syntax](#) for this command is as follows:

```
CREATE UNIQUE INDEX <index_name>  
ON [[<database.>]<owner.>]<table_name.> (<column_name>  
[, <column_name>]...)
```

This procedure cannot be completed from within Microsoft Access; it must be done in the system administration facility (SAF) on a SQL Server machine or by using a program such as Q+E, which comes with Microsoft Excel and can execute SQL commands directly on the SQL Server machine.

After you create the unique index, detach from SQL Server, then reattach to implement the change.

STATUS

This behavior is by design.

References:

"Microsoft Access User's Guide," version 1.0, pages 66-68

[References](#)

INF: Hiding Rows and Columns in Tables

Article Number: Q90099
CREATED: 08-OCT-1992
MODIFIED: 17-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

To decrease the width of a column, drag the right side of the column to the left. When you want to "hide" the column, or make it not visible, drag the right side beyond the left side.

To decrease the height of a row, drag the bottom of the row toward the top. Note: If you drag the bottom of the row above the top of the row, the sizing does not take effect and the row is not hidden because row sizing in script design (or table/menu design or tabular forms) resizes all rows, which would leave the client window with no rows. This behavior is by product design.

References:

"Microsoft User's Guide," version 1.0, chapter 14, pages 355-357.

[References](#)

INF: Creating a Snapshot Is Slower Than Creating a Dynaset

Article Number: Q90098
CREATED: 08-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

A [dynaset](#) is a dynamic set of records that results from running a [query](#) or applying a [filter](#). A snapshot is a static image of a set of data, such as the records displayed as the result of a query. Snapshots are not updateable, whereas dynasets are updateable.

Creating a snapshot takes longer than creating a dynaset because the snapshot [table](#) is populated with data, and the dynaset is populated with bookmarks, which are significantly smaller than data.

MORE INFORMATION

The results from testing have resulted in only slight speed differences between the dynaset and snapshot, but the dynaset has been found to be faster.

[References](#)

INF: AccessWizards Button Grayed Out

Article Number: Q90097
CREATED: 08-OCT-1992
MODIFIED: 01-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

When creating forms or reports in Microsoft Access version 1.0, the FormWizards or ReportWizards button is grayed out. This occurs because in the MSACCESS.INI file, the section [Libraries] must contain the entry wizard.mda = ro. If this entry is missing, the FormWizards or ReportWizards button will be grayed out. The entry in the MSACCESS.INI file should look like the following:

```
[Libraries]
wizard.mda = ro
```

Another reason for the FormWizards and ReportWizards button to be grayed out is that there are no [table](#) or [query](#) objects in the [database](#) window.

[References](#)

PRB: Update Queries Do Not Check Validation Rules

Article Number: Q90096
CREATED: 08-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

During update queries, validation rules are not checked, and no error log is generated for the invalid records during an update query.

RESOLUTION

To validate the data, a macro will need to be executed to check validity.

STATUS

This behavior is by design.

MORE INFORMATION

Update queries are queries that change existing data in a table. Validation rules are rules placed on a field when it is defined to prevent that field from accepting data that does not meet the specified rules.

When an update query is executed, as long as the data that is being changed is the correct data type for the field, it will be changed even if it violates the validation rules for field.

References:

"User's Guide," version 1.0, pages 174-178

[References](#)

PRB: Margins Do Not Relate to Printable Area

Article Number: Q90095
CREATED: 08-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you print or print preview a report with the margins set to fall outside the printable area defined by your printer driver, the information in the report falling outside of the printable area will not be printed correctly.

Depending on the printer and driver being used, the output may be truncated, shifted so it falls within the printable area, or flow onto another page. Also, output from the second page to the end of the report may vary from that of the first page. For example, the printer may shift the output on those pages so the top and left sides of the report correspond to the top and left sides of the printable area.

CAUSE

Access sets margins based on the page size selected, not on the printable area of that page. Therefore, if you set your margins to have output go beyond the printable area, the information that would be placed in that area will not print correctly.

RESOLUTION

Print preview any output from Access to make sure that no information is truncated. If information is truncated, increase the margins by choosing the Setup button and changing the margin settings.

For example, if your top margin is set to 0 and the information on the tops of the pages appear to be truncated in print preview, increase the top margin (for example, .5) and print preview the information again. Continue increasing the margins until the information is no longer truncated in print preview.

STATUS

Microsoft has confirmed this to be a problem in Access version 1.0. We are researching this problem and will post new information here as it becomes available.

[References](#)

PRB: ESC While Creating Control Does not Deselect Control

Article Number: Q90094
CREATED: 08-OCT-1992
MODIFIED: 10-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Pressing ESC while creating a control on a form doesn't deselect the control in the toolbox.

CAUSE

The toolbox is not reset until the mouse button is released. This behavior allows you to press ESC so that you can realign a control on the form.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open a new form.
2. From the View menu, choose Toolbox.
3. Using the left mouse button, select the text box control in the toolbox and drag the control to the new form, but DO NOT release the left mouse button.
4. Press ESC.

The text box is canceled but the text box control remains selected in the toolbox.

References:

"Microsoft Access User's Guide," version 1.0, pages 214-230

[References](#)

INF: Close Action has Blank Line in Object Type Argument

Article Number: Q90092
CREATED: 08-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SYMPTOMS

The Close Action has a blank line as the first choice in the drop down list for the Object Type argument in a [macro](#). To reproduce this do the following:

1. Create a macro.
2. Add the action Close.
3. Select the drop down menu for the Object Type argument.

RESOLUTION

This behavior is by design to allow the Close Action to execute on the current window.

References:

"Microsoft Access User's Guide." Version 1.0, Chapter 21 "Macro Basics"

[References](#)

PRB: Arrange Icons do Not Locate Icons in Correct Place

Article Number: Q90091
CREATED: 08-OCT-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Icons do not always stay where you expect them to. See "More Information" below for more details.

CAUSE

Icons and their placement are handled entirely by Windows. The same behavior can be reproduced in Program manager.

RESOLUTION

This behavior is by design.

MORE INFORMATION

1. In the Data Base window, create several objects, maybe five, such as forms, queries, and macros.
2. Minimize all the objects to icons.
3. Randomly place the icons around the screen. Make sure that a couple of the icons are place off the Microsoft Access client area. You can do this by dragging the icon as far right as possible. When the icon is touching the window's border, a horizontal scroll bar will appear. Move the scroll bar's thumb as far to the right. Repeat this process until you can move the scroll bar's thumb back to the left and no longer see the moved icon.
4. Select Windows, Arrange Icons.
5. Double click on one of the icons.
6. The document from step 5 should be open. Now minimize it.

Result: The icon is not in the same place as before step 5. If you do this with each icon they all end up moving. If you do a second Arrange Icons after step 6 they move to the correct location. Arrange Icons with icons on Microsoft Access windows with horizontal or vertical scroll bars displays this problem.

References:

"Microsoft Access Getting Started," version 1.0, pages 154, 158.

[References](#)

INF: Allowing an Ampersand to Appear in a Caption

Article Number: Q90090
CREATED: 08-OCT-1992
MODIFIED: 06-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

A caption containing an ampersand (&) will result in a value that replaces the ampersand with an underscore character (_). For example, entering the caption

Jack & Jill

in the Property Sheet will result in the following caption on the [form](#) or [control](#):

Jack _Jill

MORE INFORMATION

Microsoft Access allows you to underline a character in a caption, giving it ALT key access, by specifying an ampersand before the character that is to be underlined. Because the ampersand appears before the space in the following caption

Jack & Jill

Microsoft Access assumes that the intention is to underline the space following the ampersand.

If an ampersand is to appear literally in a caption, you must specify two ampersand characters in a [row](#). For example, entering the following caption in the Property Sheet

Jack && Jill

will result in the following caption in the resulting form or control:

Jack & Jill

References:

"Microsoft Access User's Guide," version 1.0, page 598.

[References](#)

PRB: 'Field0' Appears in a Query But Not in the Source Table

Article Number: Q89686
CREATED: 28-SEP-1992
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

The resulting [datasheet](#) view of a [query](#) reveals that a [field](#) called "Fieldn" (where n is a number, for example, "Field0") exists, but you do not have a field by that name in the source [table](#). The SELECT statement of this query includes an asterisk (*) to indicate that all fields should be displayed.

CAUSE

The resulting data set of a query cannot contain two columns with the same name. When you create a [SQL](#) statement with an asterisk as the result of dragging the asterisk from the fields list to a [column](#) in the query grid, then you include another field from the same table, there are in effect two fields with the same name. The first field is from the [group](#) of fields represented by the asterisk, and the second field is the one that was explicitly included in the query grid. (Note that this could also occur if you try to add a field to the grid that is already there explicitly.)

Rather than present an error message indicating that there are two columns with the same name in the view, Access substitutes the name of the second and subsequent duplicate fields with "Fieldn".

RESOLUTION

If you want to have the datasheet display a more meaningful name than "Fieldn," do the following (the steps below assume that you have included an asterisk in the query grid and that you are ready to add the duplicate field):

1. In the column of the query grid where the duplicate field is to appear, type the name of the heading you want to see in the datasheet followed by a colon (:). For example:

EmployeeName2:

2. Following the colon, type the name of the field value you want to appear in this column.

When you display the query in datasheet mode, the field will display with the [alias](#) name given in step 1.

[References](#)

INF: Flexible Formatting of Phone Numbers

Article Number: Q89685
CREATED: 28-SEP-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The format capabilities in forms are sufficient for normal usage, but a developer may not be sure of the format in which data will be entered. Because input masks are not available in Access, a developer cannot rely on masks to guide the user's input. This article uses functions to determine what format is best suited for a particular situation.

This article assumes you have a working knowledge of [Access Basic](#) code and functions.

MORE INFORMATION

The objective of this example is to take the phone number value entered by the user and determine if an area code was included. If not, it will not display the parentheses. Of course, this example does not take into account phone extensions or international numbers.

By creating two functions, this and other types of intelligent formatting are possible. The Format_Phone() [function](#) massages the phone number into a common [string](#) (void of parentheses, spaces, and dashes) and then determines the most appropriate format. It returns the phone number back to the [form](#) in that format. The SStrip() function is a very important concept. It continues to operate until it eliminates a specified character (StripStr) from the string (InWord). This is used to eliminate extra characters (dashes, spaces, and so on) from the phone number.

Create the following two functions:

```
Function Format_Phone (PhoneNumber As Control)
    '* Exit the function if there is no information passed
    If IsNull(PhoneNumber) Then
        Exit Function
    End If

    '* Need to strip out unwanted characters leaving only numbers
    PhoneNumber = SStrip(PhoneNumber, "-")
    PhoneNumber = SStrip(PhoneNumber, " ")
    PhoneNumber = SStrip(PhoneNumber, ")")
    PhoneNumber = SStrip(PhoneNumber, "(")

    '* Reformat the character string
    Select Case Len(PhoneNumber)
```

```

Case 7
    Screen.ActiveControl = Format(PhoneNumber, "@@@-@@@@")
Case 10
    Screen.ActiveControl = Format(PhoneNumber, "(@@@) @@@-@@@@")
Case 11
    'Note that this should be on one line.
    Screen.ActiveControl = Format(PhoneNumber, "@ (@@@) @@@- _
                                @@@@")

Case Else
    MsgBox "This is not a valid phone number."
    Screen.ActiveControl = PhoneNumber
End Select
End Function

```

```

Function SStrip (InWord, StripStr)
    Do While InStr(InWord, StripStr)
        'Note that this should be on one line
        InWord = Mid(InWord, 1, InStr(InWord, StripStr) - 1) & _
                Mid(InWord, InStr(InWord, StripStr) + 1)
    Loop
    SStrip = InWord
End Function

```

Along with these two functions, bind the control (Text Box) to the function by setting the following property:

```
AfterUpdate:    =Format_Phone([<control name>])
```

Here is an xample for a field with the control name [Text1]:

```
AfterUpdate:    =Format_Phone([Text1])
```

[References](#)

PRB: 'Compile All' is Enabled After Saving a Module

Article Number: Q89684
CREATED: 28-SEP-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article explains why the Compile All menu item is re-enabled after saving a [module](#).

MORE INFORMATION

After you compile a module by choosing Compile All from the Run menu, the Compile All menu item is grayed out. This is expected since you do not need to recompile a module until you make a change to it.

However, if you save the module by choosing Save from the File menu, the Compile All menu item is no longer grayed out even though you have not made any changes to the module.

Modules are always saved in an uncompiled state. When you save the module, some of the program code in memory actually is "de-compiled" for internal requirements for saving the module. Since there may be uncompiled code in memory, the Menu item is enabled again.

Note that this does not affect startup performance.

References:

For more information, [query](#) on the following words in the Microsoft Knowledge Base:

compile all and parser

[References](#)

PRB: In Extend Mode, Text to Right of Indicator Is Selected

Article Number: Q89683
CREATED: 28-SEP-1992
MODIFIED: 10-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you try to select any text in Microsoft Access (for example, data in text boxes), all the text to the right of the indicator is also selected, even if you are not holding down the left mouse button.

CAUSE

This occurs because Microsoft Access has been set to Extend mode, indicated by the letters "EXT" on the status bar. You can turn on Extend mode by pressing the F8 key.

Extend mode allows you to select large blocks of text without holding down the right mouse button and is a feature of Microsoft Access and other desktop applications, such as Microsoft Excel and Microsoft Word.

RESOLUTION

Press the ESC key to turn off Extend mode.

REFERENCES =====

For more information, search for "extend" using the Help menu.

[References](#)

INF: Obtain Name of Current User Database Using Access Basic

Article Number: Q89681
CREATED: 28-SEP-1992
MODIFIED: 12-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access does not inherently provide any way to determine the name of the database that the user currently has open. This article provides a function to do this.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

MORE INFORMATION

Although you can obtain the object variable of the current user database using the CurrentDB() function, you are unable to obtain the text representation of the database name. The example provided shows how a user-defined function called GetUserDBName() obtains a text representation of the user database name.

The GetUserDBName() function returns the name of the database with no path or extension. You must add your own path or extension string to the result of the GetUserDBName() function when necessary.

NOTE: The code to determine the current database does not work with the Access Run Time (MSARN110.EXE).

Example

1. Enter the sample code listed below into a module.
2. From the Debug window of the module, type the following, followed by a carriage return.

```
?Database_Name()
```

3. You should see the name of the database.

Sample Module Code

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

NOTE: You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.

```
Option Compare Database 'Use database order for string comparisons
Option Explicit
```

```
Declare Function FindWindow% Lib "User" (ByVal lpClassName As Any, _
    ByVal lpWindowName As Any)
Declare Function GetWindow% Lib "User" (ByVal hWnd%, ByVal wCmd%)
Declare Function GetClassName% Lib "User" (ByVal hWnd%, _
    ByVal lpClassName$, ByVal nMaxCount%)
Declare Function GetWindowText% Lib "User" (ByVal hWnd%, _
    ByVal lpString$, ByVal aint%)
Declare Function GetParent% Lib "User" (ByVal hWnd%)
```

```
Global Const GW_CHILD = 5
Global Const GW_HWNDNEXT = 2
```

```
Function Database_Name () As String
```

```
    Dim hWnd%
    Dim ClassLen%
    Dim CaptionLen%
    Dim wStart%
    Dim fNotMDI%
    Dim ClassName$
    Dim Caption$, Start%
```

```
    hWnd = FindWindow("OMain", 0&)
    hWnd = GetWindow(hWnd, GW_CHILD)
    fNotMDI = False
    Do While hWnd And Not fNotMDI
        ClassName = Space$(128)
        ClassLen = GetClassName(hWnd, ClassName, Len(ClassName))
        ClassName = Mid$(ClassName, 1, ClassLen)
        If ClassName = "MDIClient" Then
            fNotMDI = True
        Else
            hWnd = GetWindow(hWnd, GW_HWNDNEXT)
        End If
    Loop
```

```
    hWnd = GetWindow(hWnd, GW_CHILD)
    Do While hWnd
        ClassName = Space$(128)
        ClassLen = GetClassName(hWnd, ClassName, Len(ClassName))
        ClassName = Mid$(ClassName, 1, ClassLen)
        If ClassName = "ODb" Then
            Caption = Space$(256)
            CaptionLen = GetWindowText(hWnd, Caption, Len(Caption))
            Start = InStr(Caption, ":") + 2
            Database_Name = Mid$(Caption, Start, CaptionLen - Start _
                + 1)
            Exit Function
        End If
        hWnd = GetWindow(hWnd, GW_HWNDNEXT)
```

Loop

Database_Name = ""

End Function

[References](#)

INF: Cut/Paste for Immediate Window and SQL Window

Article Number: Q89680
CREATED: 28-SEP-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

In the [Immediate window](#) and in the [SQL](#) View window, you do not have access to the Cut, Copy, or Paste menu items from the Edit menu.

More Information:

Use the following keystrokes for cutting, copying, and pasting:

Cut	Shift-Delete
Copy	Ctrl-Insert
Paste	Shift-Insert

This will work for Windows 3.0 and 3.1. You could also use the following keystrokes for Windows 3.1:

Cut	Ctrl-X
Copy	Ctrl-C
Paste	Ctrl-V

[References](#)

INF: How to Use a Toggle Button to Hide and Display a Subform

Article Number: Q89622
CREATED: 27-SEP-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

In Microsoft Access you can use a Toggle Button to hide and display a subform. This article shows how to use the Toggle Button to hide and display the Quarterly Orders subform in the Quarterly Orders form located in NWIND.MDB.

MORE INFORMATION

The following steps can be used to add a toggle button:

1. Open the NWIND.MDB sample database.
2. Choose Macros from the Database window.
3. Choose New to create a new macro.
4. Choose Conditions from the View menu.
5. The items below are required for the macro.

MacroName	Condition	Action
Toggle It	[Toggle Button] <> 0	SetValue
	[Toggle Button] = 0	SetValue
Toggle It Actions		

SetValue		
Item	[Quarterly Orders Subform].Visible	
Expression	No	
SetValue		
Item	[Quarterly Orders Subform].Visible	
Expression	Yes	

6. Save the macro as TOGGLE.
7. Press the F11 function key on the keyboard to return to the Database window.
8. Choose Forms from the Database window.
9. Open the Quarterly Orders form in Design mode.

10. Choose Properties from the View menu. On the form property sheet, under Default Editing, change the setting to Allow Edits.
11. Display the Toolbox.
12. From the Toolbox, choose the Toggle button.
13. Place the mouse somewhere on the Quarterly Orders form in an open area to position the Toggle button.
14. Display the Object Properties box for the Toggle button.
15. The items below are required for the Object Properties of the Toggle button.

Toggle Button

Control Name: Toggle Button
Caption: Toggle Subform
After Update: Toggle.Toggle It

When the Quarterly Orders form is shown in browse mode the Toggle Subform toggle button hides and displays the Quarterly Orders subform.

References:

"Microsoft Access User's Guide" Chapter 22, "Using Macros with Forms" and Chapter 9, "Designing Forms;" Appendix C

[References](#)

PRB: Avery Labels 4143, 4160, and 5096 Don't Work on Laser Ptr

Article Number: Q89621
CREATED: 27-SEP-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When using Microsoft Access ReportWizards for mailing labels, Avery [label](#) numbers 4143, 4160, and 5096 print incorrectly when printed by a laser printer.

CAUSE

These labels are meant for dot-matrix printers.

STATUS

This behavior is by design.

[References](#)

INF: Import/Export Accepts Only 24-Hour Time Format

Article Number: Q89618
CREATED: 27-SEP-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

When importing or exporting time values, Microsoft Access works only with 24-hour time values.

More Information:

If you try to import a time value such as 1:00 P.M. into an existing time column, the record will not be loaded and you will get an Import Errors table that has a "Type Conversion Failure" on the value. To import a time value such as 1:00 P.M., the value must be in the 24-hour format (in this case, 13:00).

To work around this behavior, import your time values into an existing text column and use Microsoft Access Basic code to manipulate the values into the correct data type format.

If you try to export a value such as 1:00 P.M. to an ASCII text file, the value will be exported in the 24-hour format (13:00).

[References](#)

INF: Delete Command on Form Open if Counter has Focus

Article Number: Q89617
CREATED: 27-SEP-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

When a form is open in browse mode and the first control to have the focus is a counter, the delete command will be enabled on the Edit menu, even though you cannot execute this command.

More Information:

This may cause the following error message if you run a macro that executes the DoMenuItem action and then select Delete:

Command not Available: Delete

The Delete command in the Edit menu will be disabled after this error message appears. One workaround would be to create a macro that uses the DoMenuItem command and a Select Record from the Edit menu prior to running the macro that calls Delete from the Edit menu.

[References](#)

INF: Tips on How to Debug Microsoft Access Macros

Article Number: Q89610
CREATED: 27-SEP-1992
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article presents five methods of troubleshooting a Microsoft Access [macro](#). When a macro fails to perform correctly, there are several tools that can be used to help isolate where the problem is:

1. Single Stepping - This feature steps through a macro one action at a time with a pause between each action. To activate single step mode, choose the Single Step button on the [toolbar](#) or choose Single Step from the Macro menu. When any macro is run, it will execute in single step mode until you turn this feature off or choose the Continue button in the Macro Single Step [dialog box](#). This feature make it easy to determine if the action items are listed in the correct sequence. During single step mode, Microsoft Access displays the Macro Single Step dialog box, which displays the macro name, the name of the first action in the macro, and the arguments for this action. To continue macro execution, you can choose the Step, Halt, or Continue button.
2. Action Failed Dialog Box - This feature displays an error message caused by an invalid action. The same information displayed by the Single Step dialog box is displayed in this dialog box. To return to the [Macro window](#) and correct the problem, choose the Halt button.
3. MsgBox Action - Insert the MsgBox action in your macro wherever you want to display the value of a [control](#). The [action argument](#) for the message will be the name of the control you want to check. For example, to display the value of an [unbound control](#) named Field1 in Form1, enter the [expression](#) "=Forms!Form1!Field1" (without quotation marks) for the message.
4. StopMacro Action - This macro action takes no arguments and will stop the macro. Use this action to check the results that have been calculated so far. Use this feature together with the Immediate window to display calculated values.
5. Immediate Window - This tool can be used to display any values that have been declared. The [Immediate window](#) is associated with [module](#) objects. To display the Immediate window, open an existing module, or create a new module, then choose Immediate window from the View menu. To display a value, type a question mark, then the [variable](#) name, and then press the ENTER key.

For information on displaying the Immediate window without displaying

the Module window, search on the following words in the Microsoft Knowledge Base:

immediate and findwindow and showwindow

References:

"Microsoft Access User's Guide," version 1.0, pages 523-524

[References](#)

INF: OLE: New Graph Object Frames are Blank

Article Number: Q89607
CREATED: 27-SEP-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When you create a [graph](#) on a [form](#) or [report](#) it will appear as a blank object frame until you switch to browse mode of a form or either print or print-[preview](#) a report for the first time.

MORE INFORMATION

All graphs are bound objects. The Row Source may be a [table](#), [query](#), or a list of values. A graph may not be based only on the values in other controls. For this reason the query which retrieves the data for the graph is not run when the graph is created; it is executed at the time the form or report is browsed. This is consistent with all bound objects in Microsoft Access.

Steps To Reproduce Behavior

1. Start Microsoft Access and open the sample [database](#) NWIND.MDB.
2. Click on the New Form button on the Toolbar.
3. Select Category Sales for 1991 and click on the FormWizards button.
4. Choose Graph and press OK.
5. Choose Next> to accept the default graph.
6. Click on the double right arrows (>>) to add both fields and then choose Next.
7. Click on Design.

You now have a form with a blank Object Frame. To view the graph switch to Browse mode. Once the query has been executed for the first time the graph will display data from the most recent recalculation

[References](#)

PRB: New Label Controls Inherit FormWizards' Label Text Align

Article Number: Q89604
CREATED: 27-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

After creating a [form](#) with certain FormWizards (Single-Column and Main/Subform), TextAlign properties of new Label controls placed on the new form will have different defaults than they do when using the Tabular and Graph FormWizards, or when creating a new form without using FormWizards. (Tabular and Graph FormWizard labels, as well as labels on forms created without FormWizards, default to "General" text alignment.)

CAUSE

It is a feature of Microsoft Access that controls placed on a form created with FormWizards default to the same properties as the Wizard-created controls.

For example, when you create a blank form (not using a FormWizard), a Label [control](#)'s TextAlign property normally defaults to "General". With the Single-[column](#) and Main/Subform FormWizards, however, [label](#) controls placed on the new form by the FormWizard default to "Right" alignment. As a result, all subsequent new labels placed on that form will also default to "Right" alignment. Please bear in mind that this is only a default value, and can be changed.

STATUS

This behavior is by design.

[References](#)

PRB: Close Action from AB Does Not Trigger OnClose Errors

Article Number: Q89603
CREATED: 27-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If a [form](#) refers to a nonexistent Microsoft [Access Basic function](#) in its OnClose property, no error message is generated when exiting the form through a Close action called from a Microsoft Access Basic function.

RESOLUTION

To work around this problem, call a [macro](#) that issues the Close action, rather than calling a function that issues the Close action. (An example of how to do this is shown below.)

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. We are researching this problem and will post new information here as it becomes available.

MORE INFORMATION

Steps to Reproduce Problem

1. Create a [module](#) with the function "CloseMe()":

```
Function CloseMe()  
    DoCmd Close  
End Function
```

2. Save the module with the default name, "Module1".
3. Create a new form.
4. Set the following property of the new form:

Form Property

```
OnClose: =Bogus()
```

5. Place a [command button](#) on the form.
6. Set the following property of the command button:

Command Button

OnPush: =CloseMe()

7. Click the Form Browse button on the [toolbar](#).
8. From the File menu, choose Close.

The error message, "Unknown function name in 'OnClose' [expression](#)," appears. However, if you browse the form again, and click the button again, you will exit the form without an error message, even though the same offending call to a nonexistent function is still in the form.

Instead of using an Access Basic function to exit the form, use a macro to issue the Close action:

1. Create a new macro called "Closer" and add the following action and property:

Action	Object Type	Object Name

Close	Form	Form1

2. Set the following properties of the command button:

Command Button

OnPush: Closer

Note that the correct error message is now generated when the button is clicked.

[References](#)

INF: AutoExec Macro Invoked Only When Database Is Opened

Article Number: Q89602
CREATED: 27-SEP-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

The AutoExec macro will not execute when a previously hidden database is unhidden or when a minimized database window is maximized again.

The AutoExec macro will execute only when a database that has an AutoExec macro is opened. It will not execute at any other time unless it is called from Access Basic or another macro.

[References](#)

INF: END Key Behavior in Edit Boxes

Article Number: Q89599
CREATED: 27-SEP-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Pressing the END key in a form or table sometimes places the focus on the last control in the tab order, while at other times it places the cursor at the end of a string of characters in an edit box (for example, a text box, list box, or combo box in a form, or a field in a table.)

This behavior is by design.

More Information:

Once text is typed into an edit box (or table field), the Microsoft Access text editor goes from "enter" mode to "edit" mode. "Enter" mode means that the arrow (direction) and cursor-movement keys (such as END, HOME, and so on) will move the focus from control to control. "Edit" mode means the arrow and cursor-movement keys affect cursor control within that control only.

Therefore, if no characters have been typed in the box, the END key moves the focus to the last control in the tab order. If a character has been typed in the box, the END key will move the cursor to the end of the character string in the edit control.

Once the "edit" mode has been invoked, (pressing the function key F2) pressing the TAB key moves the cursor to the next control and exits "edit" mode.

References

INF: ENTER in Text Box Does Not Move to Next Line

Article Number: Q89598
CREATED: 27-SEP-1992
MODIFIED: 19-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

The cursor in a multiline text box does not move to the next line in a text box when you press the ENTER key; instead it moves to the next control in the tab order on the form.

CAUSE

This behavior is by design.

RESOLUTION

To move the cursor to the next line of a multiline text box, press CTRL+ENTER.

[References](#)

INF: Maximizing, Minimizing, and Restoring Access in Modules

Article Number: Q89597
CREATED: 27-SEP-1992
MODIFIED: 01-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Use Microsoft Windows API calls in Microsoft Access modules to maximize, minimize, and restore Microsoft Access. Microsoft Access does not provide a [function](#) to perform these actions in a [module](#) or [macro](#).

MORE INFORMATION

Once these functions are defined in a module they can be used in a macro as a RunCode action. Define the functions using the following steps.

1. Choose New from the File menu. Then choose Module. In the declarations section add both of the following declarations.

Note: In the following sample code, an underscore `_` is used as a line continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
Option Explicit
Declare Function GetActiveWindow% Lib "User" ()
Declare Function ShowWindow% Lib "User" (ByVal hWnd%, _
                                         ByVal nCmdShow%)
```

2. Create the function MaximizeAccess().

```
Function MaximizeAccess ()
    Dim ActiveWnd%, MaxIt%
    ActiveWnd% = GetActiveWindow()
    MaxIt% = ShowWindow(ActiveWnd%, 3)
End Function
```

3. Create the function MinimizeAccess().

```
Function MinimizeAccess ()
    Dim ActiveWnd%, MinIt%
    ActiveWnd% = GetActiveWindow()
    MinIt% = ShowWindow(ActiveWnd%, 2)
End Function
```

4. Create the function RestoreAccess().

```
Function RestoreAccess ()
    Dim ActiveWnd%, RestoreIt%
    ActiveWnd% = GetActiveWindow()
```

```
RestoreIt% = ShowWindow(ActiveWnd%, 1)
End Function
```

5. As an example of creating a macro that will minimize the Microsoft Access window, add the following action and it's properties to a macro.

Action	FunctionName
RunCode	MinimizeAccess()

References:

"Microsoft Access Basic: An Introduction to Programming", Chapters 1-5

"Microsoft Access Language Reference", Part 1

"Microsoft Windows Software Development Kit", Microsoft Press, 1992

"Programming Windows: the Microsoft Guide to Writing Applications for Windows 3", Charles Petzold. Microsoft Press, 1990

"Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference" Volumes 1 - 6, Microsoft Press, 1992

[References](#)

INF: Using Microsoft Access Macro to Quit Windows

Article Number: Q89596
CREATED: 27-SEP-1992
MODIFIED: 03-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

It is possible to quit Microsoft Windows from within a Microsoft Access program by using the Windows application programming interface (API) [function](#) ExitWindows(). To do this, you must create a [module](#) that declares the ExitWindows() function and then can call the Microsoft Access function in a [macro](#) using the RunCode action.

MORE INFORMATION

The call to the Windows dynamic-[link](#) library (DLL) behaves in the same way as does the Exit command from the File menu in Program Manager. Each application must agree to be closed; for example, if you choose Cancel when you are prompted to save a file, your exit request is also canceled.

To create a Microsoft Access function to quit Windows, use the following steps:

1. From the File menu, choose New and then Module.
2. In the [Declarations section](#), type the following statements:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
Option Explicit
Declare Function ExitWindows% Lib "user" (ByVal dwReserved&,_
                                           ByVal wReturnCode%)
```

3. Create a Microsoft Access function that calls the ExitWindows() function, as follows:

```
Function ExitNow ()
    DimRetVal as Integer
   RetVal = ExitWindows(0, 0)
End Function
```

4. Save the module.
5. From the File menu, choose New and then Macro. Add the following action and its properties:

Action	Function Name
--------	---------------

RunCode ExitNow()

6. Save the macro.

You can now add the macro to a [form](#) and use it as you would any other Microsoft Access macro.

Microsoft does not recommend exiting from Microsoft Access using Windows API calls, though exiting can be safely accomplished. In some cases, temporary files can be left in the Windows Temp directory; however, you can safely delete the Temp files. To automatically remove temp files each time the computer is started, add the following line to the AUTOEXEC.BAT file:

```
del c:\windows\temp\~*.tmp
```

NOTE: This line should be placed before the line that begins "Windows...." This line assumes that c:\windows\temp is the location of the Windows Temp directory specified by the Set Temp statement in the AUTOEXEC.BAT file.

REFERENCES

=====

"Microsoft Windows Software Development Kit," Microsoft Press, 1992

"Programming Windows: The Microsoft Guide to Writing Applications for Windows 3," by Charles Petzold, Microsoft Press, 1990

"Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference," Volumes 1-6, Microsoft Press, 1992

[References](#)

INF: Using Scroll Bars Doesn't Change Record Pointer

Article Number: Q89595
CREATED: 27-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

In Datasheet view, the record pointer does not change if you scroll using the scroll bars; likewise, the record number in the VCR control does not change. However, keyboard actions, such as pressing PAGE DOWN, do change the record pointer and record number.

To change the record pointer after scrolling, use the mouse to select the record that you want to make the current record. This also updates the record number in the VCR control.

More Information:

By using the scroll bars, you can view large amounts of information without changing the record pointer. You can view other records by either clicking one of the scroll bar arrows or by moving the thumb on the scroll bar.

In contrast, keyboard actions always remain relative to the record pointer regardless of what records are being viewed in Datasheet view.

[References](#)

INF: How to Display Immediate Window Without Module Window

Article Number: Q89594
CREATED: 27-SEP-1992
MODIFIED: 15-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

Summary:

Microsoft Access does not display the Immediate window unless the Module window is visible. This can be a problem if the user wants to print to the Immediate window while in Datasheet view of a form. By calling two Windows application programming interface (API) functions from Access Basic, the Immediate window can be displayed at any time, even without the Module window visible.

More Information:

To display the Immediate window at any time, you need to call the FindWindow() API function to get the handle to the Immediate window, and then call the ShowWindow() API function to make the actual window visible. You can attach the Access Basic function that includes these API functions to a command button, a RunCode action in the Autokeys macro, or add it to your toolbar.

To display the Immediate window, do the following:

1. Create a new module, or open an existing module.
2. Add the following declarations to the Global section of the module:

Note: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

Option Explicit

```
Declare Function ShowWindow% Lib "user" (ByVal hWnd%, ByVal nCmd%)
Declare Function FindWindow% Lib "user" (ByVal lpClassName As Any, _
ByVal lpWindowName As Any)
```

3. Add the following ShowImmediateWindow function:

```
Function ShowImmediateWindow ()
    Dim IhWnd%                ' Handle to the Immediate Window.
    Dim ApiResults%          ' Returns the previous state of IW.
    Const MyNull = 0&
    Const SW_SHOW = 5        ' Internal constant to show window.
    Const ClassName = "OImmediate" ' Internal ClassName of IW.

    IhWnd% = FindWindow(ClassName, MyNull)

    If IhWnd% = 0 Then
```

```
MsgBox ("You need to open the IW once for this to work.")  
End If
```

```
ApiResults% = ShowWindow(IhWnd%, SW_SHOW)  
End Function
```

4. Attach the code to a command button of a form.
5. When you first start Microsoft Access, the Immediate window is not displayed. When you open a module and display the Immediate window, then close it, Microsoft Access sets the window's Visible property to False. Calling ShowWindow resets the Visible property to True. If you call this function without first displaying the Immediate window at least once, you receive the error message because Microsoft Access has not created the Immediate window yet and therefore cannot return a window handle.

For more information on how to display the Immediate window automatically from a module, [query](#) on the following words in the Microsoft Knowledge Base:

immediate and window and macro

With the information in the article referenced above, you can open the Immediate window with macros once, then use the code from this article to display the Immediate window at any time.

6. If you have registered the Immediate window with Windows by opening it at least once, pressing the command button of the form while in Datasheet view displays the Immediate window. Any Debug.Print statements should then be visible.

Note: This is unsupported code, and there may be instances when this example does not work.

[References](#)

PRB: Autokeys Macros with the Same Name Are Still Executed

Article Number: Q89593
CREATED: 27-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Under certain circumstances, DoEvents and SendKeys are not supported when called by user-defined functions (UDFs) in [Access Basic](#).

More Information:

DoEvents and SendKeys are not supported in UDFs in the following situations:

- When calculating a [field](#) in a [query](#)
- When calculating a [control](#) on a [form](#) or [report](#)
- From an Embedded Basic Fill [function](#) for a [list box](#), [combo box](#), or [OLE object](#)

The DoEvents statement will not allow messages to be processed in these circumstances; instead, it will immediately return. Thus, any keys sent through SendKeys statements or functions will not be processed until all the computations listed above have been completed and responses have been returned to the user.

[References](#)

INF: DoEvents, SendKeys May Not Be Supported in UDFs

Article Number: Q89592
CREATED: 27-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Under certain circumstances, DoEvents and SendKeys are not supported when called by user-defined functions (UDFs) in [Access Basic](#).

More Information:

DoEvents and SendKeys are not supported in UDFs in the following situations:

- When calculating a [field](#) in a [query](#)
- When calculating a [control](#) on a [form](#) or [report](#)
- From an Embedded Basic Fill [function](#) for a [list box](#), [combo box](#), or [OLE object](#)

The DoEvents statement will not allow messages to be processed in these circumstances; instead, it will immediately return. Thus, any keys sent through SendKeys statements or functions will not be processed until all the computations listed above have been completed and responses have been returned to the user.

[References](#)

INF: Error() Function Does Not Return Detailed Information

Article Number: Q89591
CREATED: 27-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

When errors are trapped in [Access Basic](#), the Error() [function](#) returns a minimal message to the user. To receive full extended error messages, use Error or Error\$. For example, Error() or Error\$() might return "Table not found," whereas Error or Error\$ will return "Table MyTable not found."

More Information:

The following is an example of a common format for error handling in Access Basic using MsgBox to display the error text.

```
MsgBox "Error #" & Err & ": " & Error(Err)
```

This line would say something like "Error #15: File | Not found" if Err had trapped a "file not found" error.

To get the extended error message information, use:

```
MsgBox "Error #" & Err & ": " & Error
```

Note the calling of Error and not Error(Err).

[References](#)

INF: Making ENTER Add Lines in a Text Box

Article Number: Q89590
CREATED: 27-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Text boxes in Microsoft Access do not support using the ENTER key to add lines within a field. When ENTER is pressed in a text box, the focus is placed on the next field or control. This article explains how to change the functionality of pressing ENTER in text boxes so that it adds lines instead of changing the focus.

More Information:

The following function changes the behavior of the ENTER key in text boxes to allow adding lines instead of changing the focus to a different field or control. This code should be attached to the BeforeUpdate event of the text box.

NOTE: There must be some change to the text box before this code will work because Microsoft Access evaluates controls only after they have been changed (dirtied).

In the Declarations section of a Module, type the following:

```
Declare Function GetKeyState% Lib "user.exe" (ByVal nKey%)  
Const VK_RETURN = &HD
```

The following is the actual code:

```
Function MakeEnterWork()  
    If GetKeyState(VK_RETURN) < 0 Then  
        DoCmd CancelEvent  
        SendKeys "^{ENTER}"  
    End If  
End Function
```

[References](#)

INF: Topics Supported by Access as a DDE Server

Article Number: Q89586
CREATED: 27-SEP-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

Microsoft Access supports dynamic data exchange ([DDE](#)) both as a client and as a server. This article lists the five topics that Microsoft Access supports as a DDE server and the items that are valid on each of those topics.

This article does not explain what DDE is or how to implement it. For more information on DDE, [query](#) on the following words in the Microsoft Knowledge Base:

DDE and <the name of application you want to use as a DDE client>

For information on using Microsoft Access as DDE client, refer to Chapter 13 of the "Microsoft Access User's Guide." For information on using [Access Basic](#) code to exchange data with other applications, see to Chapter 9 of "Microsoft Access: An Introduction to Programming."

MORE INFORMATION

The five topics supported by Microsoft Access are "System", <Database>, <Table Name>, <Query Name>, and "[SQL](#) <statement>". Below are descriptions of the five topics and the items that are supported in each.

System

System is a standard topic for all DDE server applications. It returns a list of the topics supported by Access. Valid items with the System topic are as follows:

SysItems - Returns a list of items supported by the topic "System"
Formats - A list of the formats Microsoft Access can post to the clipboard.
Status - Busy or Ready.
Topics - A list of all open databases.

In addition to requesting the four items above, you may send the name of a [macro](#) to Access over a DDE channel initiated on the topic System. If the macro is contained in the currently open [database](#) it will be executed. A macro name is a valid item in each of the five topics supported by Access.

<Database>

<Database> is the filename of an existing database. This can be a fully qualified pathname to the file and may include the .MDB extension. You cannot query the SYSTEM.MDB via DDE. After you have initiated a DDE [link](#) with a database, you may request a listing of the objects in that database.

TableList - List of [table](#)*
QueryList - List of queries
MacroList - List of scripts
ReportList - List of reports
FormList - List of forms
ModuleList - List of modules

MacroName - You may also send the name if a macro to be executed.

<Table Name> and <Query Name>

These topics support the same items. When requesting the contents of a table or the results of a query, you must specify the database the object is in, followed by a semicolon (;), the keyword TABLE (or QUERY), and then the name of an existing table or query:

[db Name];TABLE <tbl/qry name> or [db Name];QUERY <tbl/qry name>

Examples include: NWIND;Table Categories

c:\Access\NWIND.MDB;Query Category List

NOTE: Do not place brackets ([or]) around table or query names.

You may specify the following items:

All - All the data in the table including the [column](#) names.
Data - Returns all rows of data without the column names.
FieldNames - A list of Columns.
NextRow - Returns only the next [row](#) in the table/query. When the conversation is first started NextRow returns the first row. If the current row is the last [record](#) and NextRow is executed the request will fail.
PrevRow - Returns only the previous row. If PrevRow is the first request ever a new channel the last row of the table/query is returned. If the current row is the first record the request for PrevRow will fail.
FirstRow - Returns data in the first row of the table/query.
LastRow - Returns data in the last row of the table/query.
FieldCount - Count of columns in the table/query.
MacroName - You may also send the name if a macro to be executed.

"SQL <[expression](#)>"

This topic will return the results of the specified SQL expression.

The [syntax](#) for SQL is [db Name];SQL <expression>.

NOTE: All SQL statements must end with a semicolon (;).

The following example uses Excel to call NWIND and returns the result in [cell](#) B1.

Example:

```
chan=INITIATE("MSACCESS","c:\Access\NWIND.MDB;  
    SQL Select [first name] from Employees;")  
=REQUEST(chan, "firstrow")  
=FORMULA(A2,B1)  
=TERMINATE(chan)  
=RETURN()
```

The Items supported on the Topic SQL are the same as those for <Table Name> or <Query Name>.

[References](#)

PRB: Reserved Words won't Format (Capitalize) in Module Window

Article Number: Q89576
CREATED: 27-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you type reserved words into a module window, the words usually format themselves once you move to another line so that they are easier to read. For example, if you type the following into a module window,

```
dim q as querydef
```

the line will reformat to look like this:

```
Dim q As QueryDef
```

However, not all reserved words are formatted automatically.

CAUSE

This is by design. Microsoft Access capitalizes methods, such as Close and FindNext. However, properties, such as SQL, Format, RecordCount, and LastUpdated are not capitalized.

References

PRB: Table Design Doesn't Check Validity of Default Value

Article Number: Q89549
CREATED: 24-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

The default value is not checked according to the validation of the field. For example, if the default value for a numeric field is 100 and the validation rule says that the field must be greater than 1000, table design will not check the validity of what is contained in the default value. If you add a new record and simply tab/move over the field that has the default value in it without changing it, the validation rule will not flag the value as an error. Once you change the default value, the validation rule will be in effect.

RESOLUTION

This behavior is by design.

[References](#)

PRB: Floating Objects in Access Cause Unusual Effects

Article Number: Q89548
CREATED: 24-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

Floating objects such as the Toolbox, the Properties window, and the Immediate window cause unrecognizable effects in Microsoft Access. Objects can obscure underlying data from view, and an underlying object can be obscured from view even if it has the focus or is selected, and the drop down menu for the object's corresponding option on the menu bar can appear separated from the object.

CAUSE

These effects are caused by the different Windows style properties of the windows.

RESOLUTION

To minimize the impact of this behavior, keep the floating objects from overlaying your work area by moving or closing them when not needed.

STATUS

This behavior is by design.

MORE INFORMATION

Steps to Reproduce:

Menu not Connected to the Option on the Menu Bar

1. Bring up a form in design mode.
2. Display the color palette by choosing the palette button or choosing Palette from the View menu.
3. Position the palette so it partially obscures a choice on the menu bar.
4. Choose the menu by clicking on it with the mouse or using the appropriate keystroke combination.
5. The drop down menu related to that menu bar choice should then be displayed. At this time, top level menu item will not overlap the palette, but the popup menu will overlap, giving a disjoint look.

Immediate Window Obscures Code During Replace Operation.

1. Bring up a [module](#) in design mode.
2. Display the immediate window by choosing Immediate Window from the View menu.
3. Position the immediate window so it obscures a section of code in the module window.
4. Start a replace operation that would attempt to replace a piece of code in the section of code which is obscured by the immediate window. To do this choose Replace from the Edit menu, enter the information to find and replace, and choose Verify.
5. When Microsoft Access finds the information obscured by the immediate window, the [dialog box](#) that asks you to verify that you want to change the information is displayed, but you cannot see the code.

References:

"Microsoft Access User's Guide" version 1.0, page 195.

"Microsoft Access: Introduction to Programming" version 1.0, Chapter 1 pages 2-3, Chapter 4, pages 9-10.

[References](#)

PRB: Arrange Icons Doesn't Work on More Than 40 Windows.

Article Number: Q89546
CREATED: 24-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When Microsoft Access has more than 40 Windows open, selecting Window, Tile or Cascade does not work.

CAUSE

This is functionality provided by Windows. Windows allocates memory to do the arrange function of icons. If you have enough windows open the system sometimes fails to do the allocation.

RESOLUTION

Successive tries sometimes makes it work. Otherwise, you can close some of the open windows.

MORE INFORMATION

Steps to Reproduce Behavior

1. Open and minimize 40 document windows
2. Move the icons around a bit
3. Choose Arrange Icons from the Windows menu

Nothing happens.

[References](#)

INF: Deciding When to Use Macros and When to Use Access Basic

Article Number: Q88999
CREATED: 06-SEP-1992
MODIFIED: 03-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Because Microsoft Access provides two forms of programming, macros and [Access Basic](#), there is the potential for confusion as to when you should use one [form](#) of programming rather than the other. This article discusses some tips on how to choose between [macro](#) programming and Access Basic programming.

MORE INFORMATION

Microsoft Access has a very effective user interface that is powerful enough to accommodate most needs. However, like many other [database](#) products, Microsoft Access also provides the user with the ability to develop his or her own way to manipulate the power of the product.

One distinct advantage of Microsoft Access over other products is that the user has a choice of two levels of programming: macros and Access Basic. These two programming languages are designed to complement each other.

As a rule, you should write your applications using macros and then supplement the application with SUB and FUNCTION procedures when you encounter a programming situation that a macro cannot easily accommodate.

AccessWizards are an example of a program you can write that is beyond the capability of a macro. AccessWizards require functionality that cannot be duplicated by a series of macro actions. Having to resort to Access Basic should not be a common occurrence for most mainstream database applications, however.

Knowing when to use Access Basic modules instead of macros is easier when you understand how Microsoft Access works. If you understand and are familiar with Microsoft Access, you will find that it is easier to determine when modules are required to accomplish a specific task.

New users of Microsoft Access will likely be surprised at the ease, power, and versatility that Microsoft Access macros can provide.

[References](#)

INF: How to Dim (Gray out) Menu Items with Access Basic

Article Number: Q88940
CREATED: 03-SEP-1992
MODIFIED: 16-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Access Basic does not have a command that allows you to dim (make unavailable) a menu item. There are also no properties associated with a form that enable you to set this menu characteristic.

To change such characteristics of a menu item, you can use Windows application programming interface (API) functions.

NOTE: This will only work for custom menus created with macros. It will not work for the default Microsoft Access menus. This technique will also not work for custom menu items created with the DoMenuItem action. For example, if you add a command to the menu with the DoMenuItem action and specify the PrintPreview arguments for the action, Microsoft Access overrides any API function that you write to check or gray this menu command.

MORE INFORMATION

The following Windows API function is used to dim a menu item:

EnableMenuItem% (hMenu%, wIDEnableItem%, wEnable%)

This function enables, disables, or grays a menu item.

hMenu%	Specifies the menu.
wIDEnableItem%	Specifies the menu item to be checked. The wIDEnableItem% parameter can specify pop-up menu items as well as menu items.
wEnable%	Specifies the action to take. It can be a combination of MF_DISABLED, MF_ENABLED, and MF_GRAYED. These values can be combined by using the bitwise 'OR' operator.
Return Value	The return value specifies the previous state of the menu item. The return value is -1 if the menu item does not exist.

Example

The following example designs a menu that dims a menu item:

1. Create a new macro. After adding the following actions and their associated properties, save the macro and name it "Menu

Manipulation Macro." To display the "Macro Name" column, choose the Macro Name command from the View menu.

Macro Name	Action	Function Name
GrayItem	RunCode	Gray_Menu_Item(0,0)
UnGray	RunCode	UnGray_Menu_Item(0,0)

When determining the position of a menu command, separator bars count as commands. For example, to get to the Import menu command on the File menu in the Database window, use the arguments 0 (for the File menu) and 7 (for the Import command). Positions 3 and 6 correspond to the separator bars on the File menu.

2. Create a new macro. After adding the action below and its associated property, save the macro as "Custom Demo Menu."

Macro Name	Action	Function Name
Top Level Menu	AddMenu	

[Top Level Menu].AddMenu Action Arguments

Menu Name	&Gray
Menu Macro Name	Menu Manipulation Macro

3. Create a new blank form and display the Properties window of the form by choosing the Properties command from the View menu.
4. Set the OnMenu property to "Custom Demo Menu."
5. From the File menu, choose Close to close the form. Save the form as "Menu Manipulation Form."
6. Create a new module from the Database window. Within the new module, enter the Access Basic code listed further below. Save the module as "Menu Manipulation Code."
7. From the Database window, select the Menu Manipulation form and then choose the Open button to display the form in Form view. The normal Microsoft Access menu disappears and is replaced by the custom menu you created in the previous steps.
8. There are two options on the menu. Choose the GrayItem command and the command is dimmed (grayed). Choosing the UnGray command makes the GrayItem menu command available.

Note: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```

'*****
'Declarations section of the module.
'*****
Option Explicit

```

```

Declare Function FindWindow% Lib "user" (ByVal lpClassName_
    As Any, ByVal lpCaption As Any)
Declare Function GetMenu% Lib "user" (ByVal hWnd%)
Declare Function GetSubMenu% Lib "user" (ByVal hSubMenu%,_
    ByVal nPos%)
Declare Function EnableMenuItem% Lib "user" (ByVal hMenu%,_
    ByVal wItem%, ByVal wEnable%)
Declare Function IsZoomed% Lib "User" (ByVal hWnd%)

Const MF_BYPOSITION = &H400
Const MF_GRAYED = &H1

Const MyNull = 0&
Const ClassName = "OMain"

Dim ChWnd%           'handle to the Microsoft Access window.
Dim hMenuTop%       'handle to the Microsoft Access menu.
Dim hSubMenu%       'handle to the pop-up menu
Dim ItemID%         'command ID associated with menu item.
Dim ReturnVal%

'=====
'This function initializes:
'
' - The window handles associated with the Microsoft Access form.
' - The handle to the menu of the specified window.
' - The menu handle of the specified pop-up menu of the window menu.
'
'The variables here are global to the database.
'=====
Sub Get_Menu_Handles (TopLevel%)

    ChWnd% = FindWindow(ClassName, MyNull)
    hMenuTop% = GetMenu(ChWnd%)
    hSubMenu% = GetSubMenu(hMenuTop%, TopLevel%)

End Sub

'=====
'This function dims a menu item. The text of a dimmed
'menu item appears in light gray text on the menu,
'but does not allow the user to select the item either by
'mouse or keypad. The macro action associated with the
'menu item does not execute when the user tries to select
'the menu item.
'=====
Function Gray_Menu_Item (TopLevel%, SubLevel%)

    'If the form is maximized, the system menu is added to the forms
    'menu bar, so increment the actual TopLevel%
    If (IsZoomed(Screen.ActiveForm.hWnd)) Then
        TopLevel% = TopLevel% + 1
    End If

    Call Get_Menu_Handles(TopLevel%)
    Gray_Menu_Item = EnableMenuItem(hSubMenu, SubLevel%,_
        MF_GRAYED Or MF_BYPOSITION)

```

End Function

```
'=====
'This function does not ungray a menu item that also enables
'the menu item so the user can select the item and run the
'macro associated with the menu.
'=====
```

```
Function UnGray_Menu_Item% (TopLevel%, SubLevel%)
```

```
    Call Get_Menu_Handles(TopLevel%)
```

```
    UnGray_Menu_Item = EnableMenuItem(hSubMenu, SubLevel%, _
        Not MF_GRAYED And MF_BYPOSITION)
```

End Function

References:

For more information about the structure of the Microsoft Access menu system, [query](#) on the following words in the Microsoft Knowledge Base:

menu and system and findwindow and getmenu and getsubmenu

[References](#)

INF: Generating a Random Temp File Name

Article Number: Q88929
CREATED: 03-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article discusses a [function](#) that can generate a random file name usable as a temp file.

This article assumes that you have some experience with writing procedures in [Access Basic](#).

More Information:

Sometimes it is necessary for an application to create a file that is used temporarily, then deleted. The name of temporary files are important because you do not want your application to inadvertently overwrite an existing file.

In such cases, you can create and use the user-defined function discussed below called MakeTempFileName(). This function accepts a file extension and returns a [string](#) representing a non-existent, random filename with the file extension you specified. The filename includes a path to your Windows TEMP directory so that the actual temp file is placed in the same directory Windows uses to write its temp files. This is handy for those who perform periodic maintenance on their hard drive partially by deleting any leftover temp files in the Windows directory.

In the following example, which you can type in the [Immediate window](#), MakeTempFileName() returns a name that is used as a text file. The text file is filled with some text, then deleted:

```
TempFileName$ = MakeTempFileName("TMP")
Open TempFileName$ For Output As #1
Print #1, "This is a line of text"
Close #1
Kill TempFileName$
```

The MakeTempFileName() function is listed below:

```
Function MakeTempFileName (Extension As String) As String
    On Error Resume Next
    IsFile% = False
    FHandle% = FreeFile

    Do
        WinTemp$ = Environ("TEMP") & "\"
        For Cntr% = 1 To 8
```

```
WinTemp$ = WinTemp$ & Mid(LTrim(Str(CInt(Rnd * 10))), 1, 1)
Next

TF$ = Trim(WinTemp$) & "." & Extension

Open TF$ For Input As #FHandle%
Loop While Err > 0
Close #FHandle%
MakeTempFileName = TF$
End Function
```

[References](#)

INF: Sorting/Grouping a Query Using a Portion of Field's Value

Article Number: Q88927
CREATED: 03-SEP-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

You can sort or group a query using only a portion of a field's value. For example, in the sample database NWIND.MDB, you can create a query based on the Employees table that uses the Mid() function to sort by six characters, beginning with the fourth character in the Last Name field. You can use the Left() and Right() functions to return the first and last characters of a string argument.

MORE INFORMATION

The following procedure explains how to use the Mid() function to sort data using only a portion of a field's value:

1. Enter the following in the Field row in the first column of the query grid:

```
Sort String:Mid([Last Name],4,6)
```

2. Select the Sort cell in the Last Name column and choose Ascending or Descending.
3. Choose the Totals button on the toolbar. Group By now appears in the Totals row of the query grid.
4. Switch to Datasheet view. "Sort String" appears as the name of the column, but you can change this header to any other name.

Note that the employee records are sorted by the fourth letters of the employees' last names.

This technique is useful if a field contains values with a set number of leading or trailing characters that are insignificant to your sorting task. An example of this is a part number field called PartNo that contains the following values:

```
PartNo  
-----  
A-453-34567  
A-123-45675  
B-234-75658  
B-645-65759 and so forth
```

To sort these part numbers by the third through fifth characters only, use the following expression:

Expr:MID([PartNo],3,3)

REFERENCES

=====

For more information on string functions and related functions, search for "String," "Left," "Right," and "Len" using the Help menu.

[References](#)

INF: Cannot Group By Memo or OLE Object Fields in Query

Article Number: Q88926
CREATED: 03-SEP-1992
MODIFIED: 09-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

If you attempt to create a select, [update](#), or [append query](#) that groups by a Memo [field](#) or [OLE object](#), you receive the following error message:

Can't [group](#) on Memo or OLE object (<clause>).

Microsoft Access cannot group by a Memo field or OLE object in a [query](#). Also, the GROUP BY clause of a [SQL](#) statement cannot include Memo or OLE Object fields.

To include a Memo or OLE Object field in a Totals query, drag the field to the [QBE](#) grid and select the [aggregate function](#) First or Last in the Total [row](#) for that field.

More Information:

You can group by the first <n> characters in a Memo field by using an [expression](#). To group by the first 25 characters in a Memo field called Description, use the following steps:

1. Enter the following expression in the QBE grid:

```
Expr:Left([Description],25)
```

2. Choose the Totals button on the [toolbar](#).

3. In the Total row, select Group By.

4. Switch to [Datasheet view](#). Note that "Expr" is the [column header](#).

References:

Search for "Calculating Totals in a Query" using the Help menu.

[References](#)

INF: The Immediate Window for More than Access Basic

Article Number: Q88925
CREATED: 03-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

Summary:

The Immediate window is helpful for debugging Access Basic code in that you can type out an expression to see if it returns the expected value or any value. The Immediate window can also be helpful for debugging macros, learning how to use expressions, and more. Even though the Immediate window has many uses, it is only available when the focus is on a module window. This article discusses how you can make the Immediate window available at all times.

More Information:

The Immediate window allows you to type an expression and see the results of that expression without having to write and run an Access Basic procedure. For example, you can use the Access Basic '?' command in the Immediate window as shown below:

```
? 2 + 2
```

When you type this statement in the Immediate window and press ENTER, Microsoft Access displays the number 4 just below the expression. Another example of using the Immediate window is to set the value of a variable or to test for the value of a variable. For example:

```
MyVar = 2  
? MyVar
```

The '?' statement returns the number 2 since that is the value you gave to the variable 'MyVar'.

The Immediate window is helpful for debugging Access Basic code. One reason is because you can suspend execution of the procedure and set values of variables in the procedure to see what kind of result you would get depending on different values. Another way an Immediate window can be helpful is to suspend a procedure and check for the value of a given variable or expression to see if it is calculating as expected.

In the same way that an Immediate window can be helpful for debugging code, it can also be helpful for more common tasks, such as:

- Trying out a form expression such as Forms!MyForm!MyControl to see if it returns a value.

- Trying out a query expression such as a DSum().
- Trying out a macro action in the Immediate window using DoCmd to see if a certain macro action produces the desired results.

The Immediate window is not always accessible via the menu at the top of the screen. In fact, it is only visible and accessible when focus is on a module window. In order to have the Immediate window available at all times, follow these steps:

1. In the database window, choose the 'Macro' button.
2. If you already have a macro called AutoKeys, open it up in design mode. Otherwise, choose the 'New' button.
3. If the 'Macro Name' column does not appear in the macro sheet, choose macro command from the View menu.
4. If the 'Condition' column does not appear in the macro sheet, choose the Conditions command from the View menu.
5. Add the following action and its corresponding property:

Macro Name	Condition	Action

^I		SendKeys DoMenuItem SendKeys
	MsgBox ("IW already on the screen?")=7	DoMenuItem

SendKeys Action Parameters

 KeyStrokes: {F11}
 Wait: Yes

DoMenuItem Action Parameters

 Menu Bar: Database
 Menu Name: View
 Command: Modules

SendKeys Action Parameters

 KeyStrokes: %N
 Wait: Yes

DoMenuItem Action Parameters

 Menu Bar: Module
 Menu Name: View
 Command: Immediate Window

6. Close and save the macro. If the macro is not already called AutoKeys, save it with that name.

From this point on, you can press Ctrl-I to activate the Immediate window. Follow these steps for an example of how this works:

1. Open any form in design mode
2. Press Ctrl-I

After some screen activity, a prompt appears asking if the Immediate window is already on the screen. If you do not see the Immediate window on the screen, choose 'No'. Doing so causes the macro to execute one extra step that would otherwise not be necessary.

Notice that a module window, sized to only a portion of the caption bar, appears at the upper right-hand side of the screen.

The Immediate window is now available to use.

3. Click once on the form design window or on the database window.

The Immediate window disappears.

4. Click once on the module window mentioned in step 2.

The Immediate window reappears.

You should only need to use Ctrl-I once to make the Immediate window and shrunken module window appear. Every subsequent request for the Immediate window can be made by clicking once on the module window. If the module window is closed, press Ctrl-I to bring up the Immediate window again.

[References](#)

INF: Using the GetSystemMetrics API Call in Access Basic

Article Number: Q88922
CREATED: 03-SEP-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

In the Windows environment, various display resolutions may cause screens to appear out of proportion. As a developer, you can obtain the width and height of various elements of the window display using the Windows application programming interface (API) [function](#) GetSystemMetrics. Incorporating this function into a Microsoft Access application gives the developer more information in designing the user interface.

This article describes the GetSystemMetrics API function and discusses a sample application that uses the function to display various object sizes in a message box.

More Information:

The Windows GetSystemMetrics API function retrieves information about the system metrics (the width and height of various display elements of a particular window). The GetSystemMetrics function can also return flags that indicate whether a mouse is present or if the meaning of the left and right mouse buttons have been reversed. System metrics are dependent upon the system display, and may vary from display to display.

To use the GetSystemMetrics functions, do the following:

1. Place the following Declare statement in the global section of the [Access Basic](#) Code window:

```
Declare Function GetSystemMetrics% Lib "user" (ByVal nIndex%)
```

2. Depending on what Window property you want to determine, you must define the correct [constant](#) to pass to the GetSystemMetrics function. Below are sample declarations of the constants and their appropriate meaning:

```
Const SM_CXSCREEN = 0           ' Width of screen  
Const SM_CYSCREEN = 1          ' Height of screen  
Const SM_CXFULLSCREEN = 16     ' Width of window client area  
Const SM_CYFULLSCREEN = 17     ' Height of window client area  
Const SM_CYMENU = 15          ' Height of menu  
Const SM_CYCAPTION = 4         ' Height of caption or title  
Const SM_CXFRAME = 32         ' Width of window frame  
Const SM_CYFRAME = 33         ' Height of window frame  
Const SM_CXHSCROLL = 21       ' Width of arrow bitmap on
```

```

Const SM_CYHSCROLL = 3      ' horizontal scroll bar
                             ' Height of arrow bitmap on
                             ' horizontal scroll bar
Const SM_CXVSCROLL = 2     ' Width of arrow bitmap on
                             ' vertical scroll bar
Const SM_CYVSCROLL = 20    ' Height of arrow bitmap on
                             ' vertical scroll bar
Const SM_CXSIZE = 30       ' Width of bitmaps in title bar
Const SM_CYSIZE = 31       ' Height of bitmaps in title bar
Const SM_CXCURSOR = 13     ' Width of cursor
Const SM_CYCURSOR = 14     ' Height of cursor
Const SM_CXBORDER = 5      ' Width of window frame that can't
                             ' be sized
Const SM_CYBORDER = 6      ' Height of window frame that can't
                             ' be sized
Const SM_CXDOUBLECLICK = 36 ' Width of rectangle around the
                             ' location of the first click. The
                             ' second click must occur in the
                             ' same rectangular location.
Const SM_CYDOUBLECLICK = 37 ' Height of rectangle around the
                             ' location of the first click. The
                             ' second click must occur in the
                             ' same rectangular location.
Const SM_CXDLGFRAME = 7    ' Width of dialog frame window
Const SM_CYDLGFRAME = 8    ' Height of dialog frame window
Const SM_CXICON = 11       ' Width of icon
Const SM_CYICON = 12       ' Height of icon
Const SM_CXICONSPACING = 38 ' Width of rectangles the system
                             ' uses to position tiled icons
Const SM_CYICONSPACING = 39 ' Height of rectangles the system
                             ' uses to position tiled icons
Const SM_CXMIN = 28        ' Minimum width of window
Const SM_CYMIN = 29        ' Minimum height of window
Const SM_CXMINTRACK = 34   ' Minimum tracking width of window
Const SM_CYMINTRACK = 35   ' Minimum tracking height of window
Const SM_CXHTHUMB = 10     ' Width of scroll box (thumb) on
                             ' horizontal scroll bar
Const SM_CYVTHUMB = 9      ' Width of scroll box (thumb) on
                             ' vertical scroll bar
Const SM_DBCSENABLED = 42  ' Returns a non-zero if the current
                             ' Windows version uses double-byte
                             ' characters, otherwise returns
                             ' zero
Const SM_DEBUG = 22        ' Returns non-zero if the Windows
                             ' version is a debugging version
Const SM_MENUDROPALIGNMENT = 40 ' Alignment of popup menus. If zero,
                             ' left side is aligned with
                             ' corresponding left side of menu-
                             ' bar item. If non-zero, left side
                             ' is aligned with right side of
                             ' corresponding menu bar item
Const SM_MOUSEPRESENT = 19 ' Non-zero if mouse hardware is
                             ' installed
Const SM_PENWINDOWS = 41   ' Handle of Pen Windows dynamic link
                             ' library if Pen Windows is
                             ' installed

```

```
Const SM_SWAPBUTTON = 23      ' Non-zero if the left and right  
                               ' mouse buttons are swapped
```

The following sample call will return the height of the caption bar of a form:

```
HeightY% = GetSystemMetrics(SM_CYCAPTION)
```

[References](#)

INF: Running MS-DOS SHARE with Windows for Workgroups

Article Number: Q88914
CREATED: 03-SEP-1992
MODIFIED: 09-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0

SUMMARY

If you are running a regular version of Microsoft Windows 3.x (not Windows for Workgroups or Windows NT) during the installation of Microsoft Access, Setup automatically inserts the following MS-DOS command in the AUTOEXEC.BAT file:

```
<MS-DOS directory>\SHARE.EXE /L:500
```

If you are running Windows for Workgroups, we recommend that you not load this SHARE command.

MORE INFORMATION

Setup automatically detects Windows for Workgroups and does not install SHARE if Windows for Workgroups is found.

Windows for Workgroups has its own sharing mechanism (VSHARE.386), which is loaded with Windows by an SYSTEM.INI file entry. VSHARE.386 is intended as a replacement for SHARE.EXE.

SHARE.EXE is dependent on the number of locks specified by the /L: parameter, with a default of 20. VSHARE.386, on the other hand, allocates the number of locks dynamically.

Because VSHARE.386 is a 386 enhanced mode virtual device driver (VxD), it contains features that improve file sharing when running in 386 enhanced mode. Because it was designed to work with Windows for Workgroups, VSHARE.386 has greater usability features than SHARE.EXE, especially regarding access control.

If SHARE.EXE is loaded and you start Windows for Workgroups in 386 enhanced mode, VSHARE.386 takes over file sharing control until you exit Windows. If you usually run Windows for Workgroups in 386 enhanced mode, you may not need to run SHARE.EXE at all, saving you 5K or more of conventional memory. The number of locks available is especially important if you are using your Windows for Workgroups machine as a network server.

[References](#)

INF: How to Modify the Toolbar

Article Number: Q88907
CREATED: 03-SEP-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Although Microsoft Access does not include any end-user features for modifying the [toolbar](#) at the top of the screen, you can modify the toolbar if you know how to design a [form](#) and write [Access Basic](#) code. This article discusses where toolbars are stored, how they work, and how you can modify them.

NOTE: Modification of the toolbar was not intentionally designed into Microsoft Access, so you may run into situations where your modifications do not work. Test your modifications on noncritical databases, since making these modifications is not guaranteed to work correctly in every instance. Modifications to toolbars are not supported by Microsoft.

This article assumes that you have some experience with designing forms and writing procedures in Access Basic.

MORE INFORMATION

The various toolbars that appear when you are using Microsoft Access are nothing more than forms stored in a [database](#)--the same kind of forms that you create in Microsoft Access. These forms are stored in UTILITY.MDA, a system database file that is loaded into memory when you start Microsoft Access.

You cannot simply start Microsoft Access and open UTILITY.MDA to modify these forms, however. This is because UTILITY.MDA is loaded as a [library database](#) when you are using Microsoft Access and you are not allowed to open a library database while it is active. To work around this, do the following:

1. Quit Microsoft Access if you have it open and make sure that no instances of Microsoft Access are active.
2. Choose the MS-DOS Prompt icon in Program Manager and type the following commands at the MS-DOS [command prompt](#) (the example below assumes that your Microsoft Access system files are stored in a directory called C:\ACCESS):

```
C:>CD \ACCESS  
C:\ACCESS>COPY UTILITY.MDA UTILITY.NEW
```

3. Start Microsoft Access.

At this point, you can begin modifying the toolbars in UTILITY.NEW.

Open UTILITY.NEW as a database (a couple alerts will appear about loading duplicate procedure names; ignore these alerts by choosing the OK button). In the Database window, switch to Form view and note the names of the forms. The forms in the Database window that end with the characters "TB" are toolbar forms. At this point, you can open a toolbar form and make a modification. After the modification is made, you can change a setting in your MSACCESS.INI file that will force Microsoft Access to use UTILITY.NEW instead of UTILITY.MDA as its utility database.

It is important to point out that any functionality you add to new controls on the toolbars must be supported by Access Basic--do not use macros, as they will not work for this purpose.

The example below illustrates how you can add a toolbar button to your form design toolbar that will toggle the toolbox on and off:

1. Open the form called FDTB in design mode. This is the toolbar that appears when you bring up a form in design mode, as you are doing at this point.
2. Add a small command button next to right of the "paint palette" toolbar button on the form. Make sure your button does not overlap any other controls and that it does not alter the height of the form.
3. For the button's OnPush property, specify:

```
=ToggleToolBox()
```

4. Save and close the form.
5. Create a new module called NewToolBarFunctions and add the following function:

```
Function ToggleToolBox ()  
    DoCmd DoMenuItem 3, 2, 8  
    ' The DoCmd command above invokes the forms design menu item  
    ' View->Toolbox  
End Function
```

6. Save and close the module and close the database.
7. Quit Microsoft Access.
8. From Program Manager, open Notepad.
9. In NotePad, open the file MSACCESS.INI from your Windows program directory.
10. In MSACCESS.INI, locate the [Options] section and modify the UtilityDB= line to read:

```
UtilityDB=C:\ACCESS\UTILITY.NEW
```

11. Save and close MSACCESS.INI and start Microsoft Access.
12. Open any database, and open any form in design mode. Note that the button you added appears in the toolbar. Click the button to turn the toolbox on and off.

In the same way you modified your form design toolbar using the steps above, you can make other modifications to add functionality to your Microsoft Access toolbars.

[References](#)

INF: Importing Several dBASE Databases Simultaneously

Article Number: Q88764
CREATED: 31-AUG-1992
MODIFIED: 12-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article explains how to import dBASE databases in a batch process using [Access Basic](#).

Many new Microsoft Access users want to immediately import data from the systems they are currently using. The Import [dialog box](#) in Microsoft Access allows you to import one [table](#) at a time, which usually is sufficient. However, some users have numerous tables to import and or may want to import multiple tables regularly.

This article assumes that you have some experience working in the Access Basic programming environment and that you can create and use Microsoft Access tables for data entry. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

More Information:

Importing tables with a batch process can be done quickly using an Access Basic procedure and a table. This procedure is designed for importing only dBASE databases, but it can easily be modified to accommodate other file formats.

In summary, a "batch" table lists the tables that you want to import. The Access Basic procedure reads the table and imports each foreign table listed there. To do this, use the following procedure:

1. Create a table called Batch Import with the following structure:

Field Name	Type	Length
Source Directory	Text	50
Source Database	Text	50
Imported Name	Text	50
Type of Table	Text	50

2. Enter information about the tables that you want to import. The list below shows what data you should add to each [field](#) in the Batch Import table:

Source Directory - This is the full path for the location of the [database](#) file (for example, C:\DBASE).

Source Database - This is the name and extension of the dBASE

database you want to import (for example, CUSTOMER.DBF).

Imported Name - This is the name you want the table to have once it is imported into Microsoft Access (for example, Customers).

Table Type - This can be either dBASE III or dBASE IV. Specify dBASE III for both dBASE III and dBASE III PLUS databases.

For example, suppose you wanted to import a dBASE IV database called C:\DBASE4\EMPLOYEE.DBF and a dBASE III database called D:\DBASE3\DATA\ORDERS.DBF. You would add the following two records to the table:

Source Directory	Source Database	Imported Name	Table Type
C:\DBASE4	EMPLOYEE.DBF	Employee Table	dBASE IV
D:\DBASE3\DATA	ORDERS.DBF	Orders Table	dBASE III

3. Create a new module as follows:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Sub BatchImport ()
  Dim B_DB As Database, B_TBL As Table
  Set B_DB = CurrentDB()
  Set B_TBL = B_DB.OpenTable("Batch Import")

  DoCmd Hourglass True
  B_TBL.MoveFirst
  Do Until B_TBL.EOF
    DoCmd TransferDatabase A_IMPORT, B_TBL![Type of Table],_
      B_TBL![Source Directory], A_TABLE, B_TBL![Source Database],_
      B_TBL![Imported Name], False
    B_TBL.MoveNext
  Loop
  DoCmd Hourglass False
End Sub
```

4. Import your foreign databases as follows:

- a. Open a module.
- b. From the View menu, choose Immediate Window.
- c. Type the following command:

```
BatchImport
```

The pointer becomes an hourglass and remains so until all of your databases are imported. This process may take several minutes, depending on the sizes of the databases.

dBASE III, dBASE III PLUS, and dBASE IV are manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no

warranty, implied or otherwise, regarding these products' performance or reliability.

[References](#)

INF: Using the LIKE Operator with Parameter Queries

Article Number: Q88671
CREATED: 30-AUG-1992
MODIFIED: 26-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article describes how to search for all company names beginning with a certain letter.

More Information:

An easy way to prompt a user for a character to search for is to create a [parameter query](#). The LIKE operator is used along with the wildcard symbol (*) to accomplish this task.

First create a [query](#) for the desired tables and then place the following in the CRITERIA for the [field](#):

```
LIKE "[Enter the first char to search by: ]|*"
```

Now whenever the users run this query they will be prompted with the message you specified with the LIKE statement, and type in a C to view all the companies that start with the letter C.

References:

For more information, search for "parameter query", then "Creating a Parameter Query" using the Help menu.

[References](#)

INF: Using TransferDatabase Macro to Attach to SQL Server Data

Article Number: Q88658
CREATED: 30-AUG-1992
MODIFIED: 08-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article describes how to use the TransferDatabase [macro](#) action to attach to data in backend [SQL](#) tables and what information is necessary to execute this process.

This article assumes the reader is fluent with Microsoft Access macro programming.

More Information:

The macro action TransferDatabase can be used to import from, export to, or attach to external data sources. This article specifically describes how to attach to data in a SQL Server [table](#).

When you create a macro with the TransferDatabase action, you will need to supply the macro-specific arguments:

Transfer Type	For this type of transfer, choose ATTACH.
Database Type	This is the type of the external database source that is being attached. In this case, select <SQL DATABASE>.
Database Name	With SQL Server, a connect string supplies the necessary information.
Object Type	The type of object being attached in this case is a TABLE.
Source	The name of the table in the SQL Server table goes here.
Destination	This is the name of the table as it is seen in the Microsoft Access database.
Structure Only	Depends on whether the data should be attached, also. In most cases (as well as this case) this should be NO.

The connect string (Database Name) is a string of items or parameters that define the remote data source and the user's identification.

The following are items within the connect string:

DATA SOURCE NAME (DSN) The name of the SQL Server machine.

DATABASE QUALIFIER (DATABASE) The name of the SQL Server database.

WORKSTATION ID (WSID) The name of the workstation that is making the connection (optional).

USER ID (UID) The login id*.

PASSWORD (PWD) The user password*.

* This item is optional within the connect string. If it is not present, the system will prompt the user with a [dialog box](#) to enter this information at the time the macro is executed.

Note: SQL Server is case sensitive when dealing with password information.

The following is a sample connect string to SQL Server. Note that it starts with "[ODBC](#)", which tells the system that it will use the ODBC driver (as opposed to an [installable ISAM](#) driver).

```
ODBC;DSN=ServerX;Database=Pubs;WSID=wrkstn1;UID=JohnDoe;PWD=MARCH
```

This string would be placed in the Database Name argument of the TransferDatabase action.

[References](#)

INF: Using Date/Time Data Type and Calculating Elapsed Time

Article Number: Q88657
CREATED: 30-AUG-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how Microsoft Access stores the date/time data type and provides several examples of calculating elapsed time using different functions.

MORE INFORMATION

Microsoft Access uses double-precision, floating-point numbers (up to 15 decimal places) to store date/time values. Using this numbering system, you can add, subtract, and compare date/time values as you would other numeric values. The integer portion of the double-precision number represents the date; the decimal portion represents the time.

The acceptable range for date information is January 1, 100 A.D. (-657,434) through December 31, 9999 A.D. (2,958,465), where January 1, 1900 is 2. Negative numbers represent dates prior to December 30, 1899.

The acceptable range for time information is between 0:00:00 and 23:59:59, or 12:00:00 A.M. and 11:59:59 P.M., inclusive. Time values representing 24 hours are stored internally as double-precision fractional numbers between 0.0 and 0.99999. Below are some examples of these fractional numbers:

2.0 represents 1/1/1900 12:00:00 A.M.

0.25 represents 6:00:00 A.M. (0.25 * 24 hours = 6 hours)

33914.125 represents 11/6/1992 3:00:00 A.M. (0.125 * 24 hours = 3 hours)

24794.84375 represents 11/18/67 8:15:00 P.M. (0.84375 * 24 hours = 20.25 or 20 hours and 15 minutes (.25 * 60 minutes = 15 minutes)

Using Null Values

Microsoft Access recognizes when a portion of the data in a date/time field is missing. For example, if you enter only a time value, Microsoft Access does not display the false (0) date value of 12/30/1899. Calculations that include a null date/time field return a null value.

Formatting and Storing Date/Time Values

Date/time fields can show the date, the time, or both, depending on how you format and display the date/time field. Microsoft Access always stores both the date and the time.

For more information on the formatting options available, search for "Format Property Date/Time Values" using the Help menu.

Comparing Dates Using Functions

Microsoft Access also uses double-precision, floating-point numbers to calculate functions that contain date or time arguments. This [method](#) can occasionally yield unexpected results.

For example, if you test the following [expression](#) in the Immediate window of a [module](#), the result is 0 (false), even if today's date is 1/31/92:

```
?Now()=DateValue("1/31/92")
```

This [function](#) returns -1 (true) only when the current date and time are 1/31/92 12:00:00 A.M. This occurs because the Now() function returns the double-precision number of the current date and time. This number may not be equal to the integer number of the date only, which is returned by the DateValue() function.

To avoid this problem, use either of the following statements to compare today's date with another date:

```
?Date()=DateValue("1/31/92")  
?Int(Now())=DateValue("1/31/92")
```

The Int() function strips the fractional portion of the double-precision number from the Now() function, returning only the number representing the date.

Computing and Displaying Elapsed Time

Below are two sample functions, HoursTest() and ElapsedTest(). They return the amount of elapsed time between two given date/time values, using different display formats. Both functions require you to specify time1 and time2 as parameters.

The first example, HoursTest(), displays the elapsed time in hours, minutes, and seconds:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when recreating this code in [Access Basic](#).

```
Function HoursTest()  
' This function displays the total hours elapsed in this format:  
' 48:57:04
```

```

Time1 = CVDate("1/3/92 4:05:06")
Time2 = CVDate("2/2/92 5:02:10")
HoursTest = Format(Int(Abs(Time1 - Time2) * 24), "###:") + _
             Format(Abs(Time1 - Time2), "nn:ss")
End Function

```

The HoursTest() function uses the CVDate() function to convert the string expression to a date value. The HoursTest() function then takes the absolute value of the difference between the two date/time values and multiplies this new value by 24:

```
Abs(Time1 - Time2) * 24
```

HoursTest() uses the Int() function to return only the integer portion of the date/time value. The resulting integer is the total number of hours elapsed:

```
Int(Abs(Time1 - Time2) * 24)
```

The first Format() function returns the elapsed hours. The second Format() function takes the absolute value of the difference between the two time values and uses the "nn:ss" argument to capture only the minutes and seconds. HoursTest() concatenates the results of two Format() functions, as follows:

```
Format(Int(Abs(Time1 - Time2) * 24), "###:") + _
      Format(Abs(Time1 - Time2), "nn:ss")

```

The second example, ElapsedTest(), adds the number of elapsed days to the resulting value:

```

Function ElapsedTest()
' This function displays the elapsed time as the total number of
' days, hours, minutes, and seconds. For example: 4 Days 01:04:05 Hours.
Time1 = CVDate("1/31/92 4:01:05")
Time2 = CVDate("2/4/92 5:05:10")
ElapsedTime = Abs(Time1 - Time2)
ElapsedDays = Int(ElapsedTime)
ElapsedPartialDay = (ElapsedTime - ElapsedDays)
ElapsedTest = ElapsedDays & " Days " & Format(ElapsedPartialDay, _
        "HH:NN:SS") & " Hours"
End Function

```

The ElapsedTest() function uses a process similar to that used by HoursTest(), but displays its result in a different format.

Microsoft Excel Dates vs. Microsoft Access Dates

Microsoft Excel also uses double-precision numbers to store date/time values, but Microsoft Excel values are limited to the range 1/1/1900 through 12/31/2078. The Microsoft Excel date and time system begins with the double-precision number 1.0, which represents 1/1/1900 00:00:00 (12:00:00 A.M.).

Originally, Microsoft Access supported the same date range as Microsoft Excel (1900-2078). But when Microsoft extended the date

range in Microsoft Access to 100-9999, the 1900 leap year anomaly was corrected. This means that for dates between 1/1/1900 and 2/29/1900, the internal integer portion of the date values for Microsoft Access and Microsoft Excel differ by one.

[References](#)

PRB: Scroll Box Does Not Move Down Scroll Bar in Combo Box

Article Number: Q88656
CREATED: 30-AUG-1992
MODIFIED: 09-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

When you are using a [combo box](#) on a [form](#) and the data source is an [attached table](#), the scroll box does not always move down the scroll bar when you click the scroll arrow. Once you reach the bottom of the list, the scroll box works properly; it jumps to the bottom of the scroll bar and then moves with you as you scroll up and down the list.

CAUSE

When you click the scroll arrow to view the list, the combo box [control](#) does not always know how many records are in its ControlSource [table](#) or [query](#).

When you click the scroll arrow, Microsoft Access begins to read the data in the ControlSource table or query. In order for the scroll box to move down the scroll bar, the combo box must know how many items are in the list so that it can calculate the correct physical location of the scroll box. For example, if you have 100 records in your table and you are viewing the 50th [record](#) in the list, the scroll box should appear halfway down the scroll bar.

Once you have scrolled to the bottom of the list, the combo box knows the number of items included in the list. The scroll box now adjusts itself automatically, relative to any item in the list that you select.

RESOLUTION

This behavior is by design.

[References](#)

thumb

INF: How Access Uses SQL Server Connections

Article Number: Q88655
CREATED: 30-AUG-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how Microsoft Access uses connections to [SQL](#) Server and how application developers can minimize use of these connections.

This article assumes the reader is highly knowledgeable about the low-level interaction between applications and SQL Server.

MORE INFORMATION

The conventional application that accesses a server does it in a simple single-tasking manner, requiring a single connection to the server. Microsoft Access, however, is not a typical conventional server front-end. Many of its features, including updatable views (dynasets), joins between local and server data, execution of complex expressions that the server may not be able to handle, and seamless transition from one server to another, require Microsoft Access to obtain more connections than a conventional front-end.

Although some servers, such as Microsoft SQL Server, are not stingy giving out connections, there are other servers that place strict limits on the number of connections an application can open. (In some installations, SQL Server also places absolute restrictions on connections.) When designing applications for the more restrictive servers, it is important to understand how Microsoft Access uses connections.

Connections are opened by Microsoft Access when it needs to execute a [query](#) on the server (which it must do to build dynasets or static views), obtain the data in a [dynaset](#), or [update](#) data on the server. These connections can be categorized into two types: connections needed to build the working set and connections needed to service dynasets.

Both types of views available in Microsoft Access (static views and dynasets) are built by executing a query (called a "local" query to differentiate it from other types of queries discussed below). This local query processes Microsoft Access and foreign ISAM data (both locally and file server based) and remote server data, and pulls it into a temporary [table](#) that represents the view. To build static views, the actual data is stored in the temporary table, whereas to build dynasets, pointers to the actual data are stored.

When server data is included in the local query, execution of the query involves asking the server (or servers) for data, which is done by opening connections and sending queries to the server. Each query

sent to the server requires that a connection be opened. If more than one query is to be sent to the server, a single connection could be used and the queries executed serially. However, this would typically require the user to wait for all of the server queries to complete before any data would appear. Therefore, Microsoft Access executes the server queries in parallel, requiring a connection for each.

The number of queries sent to a server may not be entirely obvious. Much of the time, access to all of the remote base tables will be combined into a single query, if all of the base tables reside on the same server. In the worst case, each server base table referenced by the local query will require a connection. There are several factors that could compel Microsoft Access to split a local query into several server queries:

- If server data is joined to data from other sources (that is, local, foreign ISAM, and other servers), the Microsoft Access optimizer may decide that it is more efficient not to combine references to two tables on the same server into a single query. For example, if the Customers and Order Items tables are on a server and the Orders table is in a local Microsoft Access database, then it is likely that the optimizer will decide to ask for the Customers and Order Items tables separately, rather than asking for the cross-product of the two, to join to the Orders table. As you can tell from this example, this is unlikely to happen in a real scenario.
- If an expression involved in the query cannot be executed on the server (either because the server does not support it or because the semantics provided by the server differ greatly from those provided by Microsoft Access), the expression will be executed locally. An extreme example of this is a local query with a restriction that calls a user-defined function. Although this may not require separating remote tables into separate server queries, it will if the expression that is not executable on the server is somehow involved with the join between the two tables. Note that in Microsoft Access, expressions containing a conjunction that can be executed on the server will be executed locally in their entirety. These expressions are not split to execute part locally and part on the server.
- The parameters used to attach tables on the same server to a Microsoft Access database differ. For example, if two tables are in different databases, or if they are being accessed using different user names and passwords, then separate connections will be required to execute a local query referencing both of them.

Connections opened in the execution of a local query will be kept open until the working set containing the queries is closed.

Connections are also opened in support of dynasets. But, unlike the server queries sent during the execution of a local query, the queries sent to the server in support of a dynaset are executed entirely and very quickly. This is because they are used to fill portions of the dynaset with data (given the pointers to the data provided by the local query), and to do updates to the server. Therefore, Microsoft Access opens only one connection per attached server, provided that the link attachment information is the same

for each server tables.

As a result of the above rules, when Microsoft Access executes a query, it typically requires only one (for a static view) or two (for a dynaset) connection. This does not represent an extravagant use of server connections, but good applications could involve the execution of many Microsoft Access queries that could occupy many server connections very quickly. There are some steps the designer can take to minimize the use of connections:

- Reduce the number of queries required by the application. Each form, subform, data sheet, report, and list box requires a query, and each of these queries that accesses server data will require at least one or two connections. Close forms, data sheets, and reports as soon as they are no longer needed. Also, bring server data to a local database where practical, especially for filling list boxes.
- Avoid queries that join server data to local data to server data, as described above. These types of queries will most likely be executed using a connection for each server table.
- Avoid expressions that must be executed locally when joining server data together. Each of the tables involved with such a join expression, or that is restricted by a WHERE clause containing such an expression, will need to be executed locally. It should be noted that separating expressions into separate query objects will not help alleviate this problem, because the optimizer combines all query objects in a single query before execution for efficiency.
- Use static working sets when seeing other users' changes, or when making changes yourself is not required. This will avoid the connections needed to support dynasets.

With an understanding of how Microsoft Access uses connections, and with prudent application design, you should not run out of server connections.

[References](#)

INF: Performing an SQL Bulk Action Query from Access Basic

Article Number: Q88654
CREATED: 30-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Access Basic does not support SQL statements to access your data outside of having to define a virtual table (VT) object to execute such a statement.

For example, you cannot execute the following SQL command on a line by itself even though it would be valid in a Microsoft Access query:

```
UPDATE Cust SET Cust.[Phone] = "(206) " & [Phone];
```

This article discusses how to perform these kinds of bulk action queries and also includes a subprocedure called PerformSQLAction you can add to your program. This subprocedure allows you to pass a bulk action SQL statement to the procedure and have it executed.

This article assumes that you have some experience with writing procedures in Access Basic.

More Information:

Access Basic does not support SQL statements in the sense that you cannot expect a SQL command to execute on a line by itself. For example, the following SQL command, although valid in a Microsoft Access query, will generate a syntax error in an Access Basic program:

```
UPDATE Cust SET Cust.[Phone] = "(206) " & [Phone];
```

Instead of including the SQL command on a line by itself, you have to follow these steps in order to execute the SQL command:

1. Dim a Database and QueryDef variable as shown below:

```
Dim MyDB As Database  
Dim MyQuery As QueryDef
```

2. SET the database variable to the current user database as shown below:

```
Set MyDB = CurrentDB()
```

3. SET the QueryDef variable to use the CreateQueryDef method, assigning an arbitrary query name such as "TempQuery" and including the desired SQL statement. The query you create using

this step is only for the purpose of executing the bulk action query and will be deleted later. An example of how to use CreateQueryDef for this purpose is shown below:

Note: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Set MyQuery = MyDB.CreateQueryDef("TempQuery" ,UPDATE_  
    Cust SET Cust.[Phone] = "(206) " & [Phone];
```

4. Execute the TempQuery query you created in the previous step by using the Execute method:

```
MyQuery.Execute
```

5. Delete TempQuery by using the DeleteQueryDef method as shown below:

```
MyDB.DeleteQueryDef("TempQuery")
```

All of these steps, along with some error trapping, have been combined in the subprocedure PerformSQLAction listed at the bottom of this article. By creating a generic procedure such as PerformSQLAction, you can make use of SQL commands to perform bulk action queries in your Access Basic program.

PerformSQLAction requires only the SQL command you want to execute as a [string](#) value. For example:

```
PerformSQLAction "UPDATE Cust SET Cust.[Phone] = '(206) ' & [Phone];"
```

Note that the SQL command is always followed by a semicolon (;).

The following is a listing of PerformSQLAction:

```
Sub PerformSQLAction (SQLStmt As String)  
    Dim LocalDB As Database, LocalQry As QueryDef  
  
    On Error Resume Next  
    Set LocalDB = CurrentDB()  
    LocalDB.DeleteQueryDef("TempQuery")  
    On Error GoTo 0  
  
    Set LocalQry = LocalDB.CreateQueryDef("TempQuery", SQLStmt)  
    LocalQry.Execute  
    LocalQry.Close  
    LocalDB.DeleteQueryDef("TempQuery")  
End Sub
```

[References](#)

INF: How to List the Related Tables in a Database

Article Number: Q88653
CREATED: 30-AUG-1992
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Although Microsoft Access allows you to create relationships among tables in your [database](#), there is no inherent command to list these relationships. Sometimes this information can be of use, particularly when trying to find the source [table](#) of a [referential integrity](#) violation or other such example.

This article discusses how you can use a series of queries to display a list of tables that are related in a database.

More Information:

Information on the relationships among the tables in a database can sometimes be very helpful, particularly when trying to find the source table of a referential integrity violation or other such example. This information is also good for documenting your database.

Although Access allows you to create relationships among tables in your database, there is no inherent command that will gather this information and present it in a meaningful way. All of this information can be made available by gathering and relating information from the system tables MSysObjects, MSysColumns, and MSysIndexes. This article discusses how you can create a [query](#) containing two columns. The first [column](#), called "Primary.Name," will contain a table name. The second column, called "Foreign.Name," will represent the table that the Primary.Name table is related to.

Because the information will be coming from the system tables mentioned above, you must make sure you have access rights to these tables. MSysObjects has these rights by default, so the instructions below indicate how you can give yourself access rights to MSysIndexes and MSysColumns.

1. In the Database Window menu, select the View menu, then choose Options.
2. Change the Show System Objects setting to Yes, then choose OK.
3. From the Database Window menu, choose the Permissions command from the Security menu.
4. In the following [dialog box](#), choose Table for the Object Type prompt, then choose MSysIndexes for the Object Name prompt.

5. In the Permissions box at the bottom of the dialog, select the Full Permissions box.
6. Click Assign and then choose the Close button. Repeat steps 4 and 5 for MSysColumns.
7. From the View menu at the top of the screen, choose Options.
8. Change Show System Objects to No, then choose the OK button.

At this point, you can begin building the queries that will retrieve the relationship information. To do this, follow the instructions below:

1. In the database window, choose the Query button, then choose New. A query design screen appears, along with a dialog asking for a Table/Query to add to the query. Choose Close so that the query design grid is blank.
2. From the View menu, choose SQL.
3. Delete any contents that appear in the SQL dialog box.
4. Enter the following into the SQL window:

```
SELECT DISTINCTROW
    MSysObjects.Name, MSysObjects.Id
FROM MSysObjects
WHERE ((MSysObjects.Type=1)
AND (MSysObjects.Flags Is Null Or MSysObjects.Flags<>2))
ORDER BY MSysObjects.Name;
```

5. Choose the OK button. Close and save the query as "psi User Tables."
6. Repeat steps 1 through 4, substituting the SQL statement in step 4 with the following SQL statement:

```
SELECT DISTINCTROW [Tables].Name AS Table,
    MSysColumns.Name AS Field,
    [Tables].Id AS [Primary Id]
FROM [psi User Tables] AS Tables, MSysColumns, Tables
INNER JOIN MSysColumns ON [Tables].Id = MSysColumns.ObjectId
ORDER BY [Tables].Name, MSysColumns.PresentationOrder;
```

7. Choose the OK button, then close and save the query as "psi User Tables Field List."
8. Repeat steps 1 through 4, substituting the SQL statement in step 4 with the following SQL statement:

```
SELECT DISTINCTROW Primary.Name, Foreign.Name
FROM [psi User Tables] AS Primary, MSysIndexes, [psi User Tables]
    AS Foreign, MSysIndexes AS LKeyName, LKeyName
INNER JOIN MSysIndexes ON LKeyName.Idxid = MSysIndexes.Idxid,
    LKeyName
INNER JOIN MSysIndexes ON LKeyName.ObjectId = MSysIndexes.ObjectId,
```

```
Primary
INNER JOIN MSysIndexes ON Primary.Id = MSysIndexes.ObjectId,
Foreign INNER JOIN MSysIndexes ON Foreign.Id =
MSysIndexes.ObjectIdReference
WHERE ((MSysIndexes.ObjectIdReference<>0)
AND (MSysIndexes.Operation=2)
AND (LKeyName.Operation=0))
ORDER BY Primary.Name, Foreign.Name;
```

9. Choose the OK button. Close and save the query as "psi Relationships."
10. To view the Relationships table, highlight the query called "psi Relationships" in the database window and choose the Open button.

With this query, you can create reports or use the query in any other way that you would normally use an Access query to make use of the information.

A tool called "Database Analyzer" is included with Microsoft Access that is another method for displaying database information. For information on using this tool, look in the PSSKB.TXT file, question number 23, or choose the Microsoft Access Q&A icon in the Access group in the Microsoft Windows Program Manager.

[References](#)

INF: Creating a Proper Function for Data Entry

Article Number: Q88652
CREATED: 30-AUG-1992
MODIFIED: 25-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

There is no built-in format (or [Access Basic](#)) [function](#) that capitalizes the first character of a word (similar to Excel's Proper function).

However, this can be done with a one line function attached to the AfterUpdate property of a [form control](#).

NOTE: You can call this function from forms, reports and queries, but not from tables.

MORE INFORMATION

To create an example, create a form based on some desired [table](#). Change the following property of the form:

```
AfterUpdate: =Proper(<control name>)
```

Where <control name> is the name of the control on the form. This will call the function Proper() whenever the value on the form is changed.

Now create a [module](#) and enter the following function:

```
Function Proper(Ctl as Control)
    Ctl = UCase(Left(Ctl, 1)) & LCase(Right(Ctl, Len(Ctl) - 1))
End Function
```

Whenever you edit the [field](#) in the form the first letter will be capitalized. What you have done is pass program control by reference to the function, which parses the [string](#) and capitalizes the first character. You may use this function from any number of controls.

[References](#)

INF: Practice Using Object Analyzer Commands

Article Number: Q88651
CREATED: 30-AUG-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article provides an example of how to use the Object Analyzer commands discussed in the article titled "Programming with the Object Analyzer Engine and Utilities." For more information on the commands themselves, search on the following words here in the Microsoft Knowledge Base:

DumpTableInfo or DumpQueryInfo or DumpFormOrReport or
DumpMacroInfo or DumpModuleInfo

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

MORE INFORMATION

As discussed in "Programming with the Object Analyzer Engine and Utilities," you can take advantage of the Object Analyzer engine by using the commands in the Object Analyzer [module](#).

To get some practice using these commands, follow the steps below to create your own Database Analyzer in the sample [database](#) NWIND.MDB:

Creating a Form

1. Create a new [form](#) called Little Analyzer.
2. Create a [text box control](#) on the form. Change the ControlName property to Name.
3. Create an option [group](#) control called Options on your form.
4. Add six toggle buttons to the Options [option group](#) and modify the properties of the toggle buttons as described below:

Control	Option Value	Caption

First Button	1	Table
Second Button	2	Query
Third Button	3	Form
Fourth Button	4	Report

Fifth Button	5	Macro
Sixth Button	6	Module

- For the AfterUpdate property of the Options option group, specify the following:

```
=AnalyzeIt(Name, Options)
```

- Save and close the form.

Creating a Module

- Create a new module in NWIND.MDB.
- Add the following [function](#) to the module:

```
Function AnalyzeIt (InName As String, InType As Integer)
' This function accepts the name of a Microsoft Access object
' and a number representing what type of object it is, then uses
' the Object Analyzer Dump commands to analyze the object.

DoCmd Hourglass True
Select Case InType
Case 1
DumpTableInfo "NWIND.MDB", "@Table", InName, False
Case 2
DumpQueryInfo "NWIND.MDB", "@QuerySQL", "@Query", InName
Case 3
DumpFormOrReport "NWIND.MDB", "@Form", "@FCtrls", InName, True
Case 4
DumpFormOrReport "NWIND.MDB", "@Report", "@RCCtrls", InName, False
Case 5
DumpMacroInfo "NWIND.MDB", "@Macro", InName
Case 6
DumpModuleInfo "NWIND.MDB", "@Procs", "@Vars", InName
End Select
DoCmd Hourglass False
End Function
```

- Save and close the module.

Combine Form and Module to Create Object Analyzer

- Open the Little Analyzer form.
- In the text box, type "Categories" (without quotation marks).
- Choose the Tables button. You have created an Object Analyzer.

The Object Analyzer analyzes the Categories [table](#) and creates a table called @Table, which contains information about the structure of the Categories table.

If you have the Database Analyzer library loaded, you can use these commands to build your own object-analysis tool, or you can

incorporate these commands into your application to make it self-documenting.

[References](#)

INF: How to Create a Password Textbox

Article Number: Q88650
CREATED: 30-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

Summary:

Some applications require password protection and should provide a dialog to the user at startup which prompts for a password. For [security](#) purposes, this prompt should not show the characters that the user is typing, but should give the user some idea of how many characters have been typed so that the user does not lose track of what part of the password is being typed.

More Information:

To modify the password textbox on your [form](#) to behave like this, follow these instructions:

1. Open your form in design mode.
2. Click once on the textbox on the form which you are using to receive the password so that the textbox has focus.
3. In the [toolbar](#), drop down the list of fonts located directly to the right of the Paint Palette toolbar button.
4. From the list, choose 'Courier'.
5. In the toolbar, drop down the list of font sizes located directly to the right of the list of fonts from which you just made a selection.
6. From the list, choose the value 10.
7. From the palette window, click on the color White for both the Fill [row](#) and Text row of colors.
8. Save and close your form.

After making these modifications, you will notice that characters typed in the password textbox do not appear, but the cursor is present. The Courier font assigned to the textbox makes cursor movement and position clearly definable so that the user has an idea of how many characters have already been typed.

[References](#)

INF: Opening Databases as Read-Only Causes Error Message

Article Number: Q88649
CREATED: 30-AUG-1992
MODIFIED: 09-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

Summary:

You open a [database](#) under the following three conditions:

- The file is [read-only](#), whether or not its file attribute is set to read-only.
- The directory in which the file resides is also read-only.
- You are using the /Ro [command line](#) parameter.

You receive the following error message:

Database '<Database_Name>' is read-only. You won't be able to save changes made to data or object definitions in this database.

Note also that when a database is opened as read-only, an .LDB file is never created.

More Information:

The only workaround currently available is to implement [security](#) measures. Instead of physically opening the file as read-only, you would open all objects in the database as read-only. The procedures to implement database security are outlined in the "Microsoft Access User's Guide," version 1.0, chapter 25.

[References](#)

INF: Easy Way to Add Items to the Help Menu

Article Number: Q88648
CREATED: 30-AUG-1992
MODIFIED: 23-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

Summary:

The Help menu item that appears at the top of the screen in Microsoft Access can be supplemented so that your own custom menu items appear. This can be done through the use of an MSACCESS.INI entry.

More Information:

The section is called [Menu Add-Ins]. You can use this feature to invoke a [macro](#), an [Access Basic](#) program, or an Access Basic [function](#) from a Help menu item. To add this section to MSACCESS.INI, follow these instructions:

1. Close Microsoft Access if it is open.
2. From your Windows Program Manager, start Notepad.
3. In Notepad, open the file MSACCESS.INI, (this file is located in the same directory that you installed Windows).
4. Scroll to the bottom of the text, add a new line, and type the following:

```
[Menu Add-Ins]
```

You are now ready to add the menu items that you want to appear in the Help menu. You can do this by adding lines to the [Menu Add-Ins] section that follow this convention:

```
<menu label>=<menu action>
```

menu label - String that appears in the menu.

Menu action - Can be one of the following:

1. Macro name
2. User-defined function
3. Access Basic function

For example, if you wanted to add a menu item called 'Customers Form' that runs a macro called 'OpenCustForm', you would add the following line to the [Menu Add-Ins] section:

```
Customers Form=OpenCustForm
```

If you wanted to add an item called 'Run Daily Update' which runs an Access Basic function you wrote called 'RunUpdate()', you would add the following line to the [Menu Add-Ins] section:

```
Run Daily Update==RunUpdate()
```

The second '=' in this example is required for running user-defined functions or other Access Basic functions.

You can also include an underlined character accessible with the ALT key in the menu item label by placing an ampersand (&) in front of the character that should be underlined. For example, suppose you wanted a menu item in the Help menu called 'Show Message' with the 'M' underlined. This menu item would call Access Basic's MsgBox() function to display a message:

```
Show &Message==MsgBox("This is a message")
```

An example of the [Menu Add-Ins] section as it should appear in MSACCESS.INI given the example menu items above follows:

```
[Menu Add-Ins]
Customers Form=OpenCustForm
Run Daily Update==RunUpdate()
Show &Message==MsgBox("This is a message")
```

Once you have made these modifications to MSACCESS.INI and restart Microsoft Access, you can pull down the Help menu from the main menu to see the resulting added menu items.

[References](#)

PRB: Irregular Characters in Attached dBASE IV Memo Field

Article Number: Q88647
CREATED: 30-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SYMPTOMS

If you have an attached dBASE IV table with a memo field, a strange character resembling an "i" (ASCII 161) appears randomly throughout the memo field when viewed in Microsoft Access.

CAUSE

This problem is the result of the way dBASE IV handles its memo fields and how this translates when you view the memo field in Microsoft Access.

When you open a memo field in dBASE IV, you are presented with an editor that is fixed at 65 characters per line and cannot be adjusted. In order to format your memo data so that it fits comfortably in the memo editor, dBASE IV inserts invisible characters that act as carriage returns. These characters become an actual part of the data in the .DBF file. Microsoft Access does not use these characters to format the memo data because the size of a memo control is adjustable, so they are ignored and consequently appear in the memo control on your form.

RESOLUTION

These characters can be stripped out of the memo field by using an Access Basic procedure. The procedure would go through each record in the table, grab the memo field, and copy it to a temporary holding area character for character, ignoring all instances of Chr(161), then copy the string from the holding area back into the memo field.

STATUS

This behavior is by design.

MORE INFORMATION

The product included here, dBASE IV, is manufactured by a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

INF: How to Force a Cascading Delete

Article Number: Q88635
CREATED: 30-AUG-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

The capabilities of referential integrity in Microsoft Access do not extend to performing cascading deletes. A cascading delete is a method of deleting all records of a child table when the corresponding record in the master table is deleted. This article discusses how to write a macro to perform a cascading delete upon deleting a record from a related master table.

MORE INFORMATION

As an example of a cascading delete, suppose you are viewing the Categories table in the Northwind Traders sample database NWIND.MDB and you decide to delete the "BEVR" Category ID. The Categories table is related to the Products table with referential integrity enabled and contains many child Category ID records with the value "BEVR." Therefore Microsoft Access prevents you from deleting the record in the Categories table until you delete all the child "BEVR" records in the Products table. With the ability to perform a cascading delete, all of the Products' "BEVR" records would be deleted automatically before the actual deletion of the Categories "BEVR" took place.

To force a cascading delete using the Categories/Products example, first create a macro and attach it to the OnDelete event of the Categories form:

1. In the Database window, choose the Macro button, then choose New.
2. If the Condition column does not appear in the macro sheet, choose the Conditions command from the View menu.
3. Add the following actions and their corresponding properties:

Condition	Action
MsgBox("Delete record & all child records?",33)<>1	CancelEvent
...	StopMacro
	SetWarnings
	SendKeys
	RunSQL

SetWarnings Action

Warnings On: No

SendKeys Action

Keystrokes: {enter}

RunSQL Action

SQL Statement: DELETE * FROM [Products] WHERE [Category ID] =
Forms!Categories![Category ID];

Note the semicolon (;) at the end of the SQL
statement.

4. Close and save the macro, naming it "Cascade".

To use this macro with an existing form, do the following:

1. Open the form in design mode.
2. From the Edit menu, choose Select Form.
3. In the property sheet, specify "Cascade" for the OnDelete event.
4. Close and save the form.

Now when you use the form, deleting any records will delete any
matching child records in a related table.

To delete a record on the Categories form (because there are no Record
Selectors on this form), choose the Select command from the Edit menu,
then choose the Delete command from the Edit menu.

[References](#)

INF: Programming with the Object Analyzer Engine and Utilities

Article Number: Q88620
CREATED: 30-AUG-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

When the Database Analyzer library is loaded into memory, you not only have access to the Database Analyzer tool, but also to all the core commands that do the actual analysis. This article discusses those commands and functions and how you can use them to include this sort of data analysis in your custom applications or to write your own [database](#)-object analysis tool.

This article assumes that you are familiar with [Access Basic](#) and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

MORE INFORMATION

Included with Microsoft Access is a tool called the Database Analyzer. You can use this tool to retrieve information on the design of the objects in your database, such as tables, queries, forms, and so forth. For more information on what the Database Analyzer is and how to install and use it, choose Contents from the Help menu and then choose the Product Support topic.

Once you have loaded the Database Analyzer library into memory (by following the instructions in the Help file), you not only have access to the Database Analyzer tool, but also to all the core commands that do the actual analysis. Before learning the commands and command [syntax](#) available, you should know the basic makeup of the Database Analyzer tool.

The Database Analyzer consists of three pieces:

- Database Analyzer

This is the "front-end" [form](#) that you see when you start the Database Analyzer tool. It includes the code that gives it the ability to display and select objects for analysis.

- Object Analyzer

This is the core engine that actually creates the object tables, performs the analyses, and writes the resulting information to the object tables.

- Object Analyzer Utilities

This piece contains global procedures and functions that support some of the commands in the Object Analyzer.

The purpose of the Database Analyzer is to find out which objects the user wants analyzed, then call the Object Analyzer to perform the actual analyses using one of the following commands:

DumpTableInfo
DumpQueryInfo
DumpFormOrReport
DumpMacroInfo
DumpModuleInfo

Assuming you have loaded the Database Analyzer library in memory, these commands are available to you from the Immediate window. To use one of these commands, do the following:

1. Open the sample database NWIND.MDB in Microsoft Access.
2. In the Database window, choose Modules. Select the Introduction To Programming module and choose the Design button.
3. From the View menu, choose Immediate Window.
4. In the Immediate window, type the following command, then press ENTER:

```
DumpTableInfo "NWIND.MDB", "MyTableInfo", "Employees", False
```

Upon executing this command, you will notice some disk activity. Once the disk activity has stopped, you will find a new table called MyTableInfo in the Database window. This table will contain the structure information for the Employees table.

Described below are the functions that you can call with the Object Analyzer.

```
DumpTableInfo <TargetDB>, <DetailsTable>, <ObjectName>, <IsAttached>
```

<u>Formal Parameter</u>	<u>Definition</u>
TargetDB\$	The name of the database that will receive the table.
DetailsTable\$	The name of the table that will receive the information.
ObjectName\$	The name of the table to <u>report</u> on.
IsAttached%	True if the table being reported is an Attached Table; otherwise False.

```
DumpQueryInfo <TargetDB>, <SQLTable>, <DetailsTable>, <ObjectName>
```

Formal Parameter	Definition
------------------	------------

TargetDB\$	The name of the database that will receive the table. SQLTable is the name of the table that will receive the <u>SQL</u> representation of the <u>query</u> .
------------	---

DetailsTable\$	The name of the table that will receive the detailed information of the query.
----------------	--

ObjectName\$	The name of the query to report on.
--------------	-------------------------------------

DumpFormOrReport <TargetDB>, <PropsTable>, <DetailsTable>,
<ObjectName>, <IsForm>

Formal Parameter	Definition
------------------	------------

TargetDB\$	The name of the database that will receive the table.
------------	---

PropsTable\$	The name of the table that will receive the information on the form's or report's properties.
--------------	---

DetailsTable\$	The name of the table that will receive the detail information about the form or report.
----------------	--

ObjectName\$	The name of the form or report.
--------------	---------------------------------

IsForm%	True if reporting on a form, False if on a report.
---------	--

DumpMacroInfo <TargetDB>, <DetailsTable>, <ObjectName>

Formal Parameter	Definition
------------------	------------

TargetDB\$	The name of the database that will receive the table.
------------	---

DetailsTable\$	The name of the table that will receive the information.
----------------	--

ObjectName\$	The name of the <u>macro</u> to report on.
--------------	--

DumpModuleInfo <TargetDB>, <ProcsTable>, <VarsTable>, <ObjectName>

Formal Parameter	Definition
------------------	------------

TargetDB\$	The name of the database that will receive the
------------	--

table.

ProcsTable\$ The name of the table that will receive the listing of procedures.

VarsTable\$ The name of the table that will receive the listing of variables.

ObjectName\$ The name of the module to report on.

NOTE: DumpModuleInfo will not work if it is executed from an Immediate window.

References:

For more information, query on the following words in the Microsoft Knowledge Base:

practice and using and object and analyzer and commands

[References](#)

INF: Creating, Debugging, and Using an Access Library

Article Number: Q88175
CREATED: 19-AUG-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SUMMARY

This article describes a Microsoft Access library and discusses some basics on how to create and debug a library, as well as some things to watch out for.

This article assumes that you are familiar with [Access Basic](#) and with creating Access applications with the programming tools provided with Access.

MORE INFORMATION

Access Basic Library Defined

When you write an Access application, such as the NWIND application with the package, the application works only within the [database](#) in which it was created. This is satisfactory for many applications that specifically use the data that resides in the application's database.

However, many Microsoft Access developers write generic applications, programs, and utilities that are designed to work on any user database. An example of this is Wizards. Wizards are Access Basic programs that reside in their own database, but are available to the user in any database the user has open. If this weren't the case, you could not use a Wizard outside of the database that the Wizard program and system objects reside in. In order to use a program (such as a Wizard) so that its code and objects are available to any user database, you must load the database containing the program and its objects as a library.

To load a database as a library, you must open MSACCESS.INI and add an entry to the Libraries section. When you open MSACCESS.INI initially, you will probably see a Libraries section with an entry for the Wizards library:

```
[Libraries]
wizard.mda=ro
```

NOTE: If there is no Libraries section, add it to the end of the file and continue. The MSACCESS.INI file can be found in your Windows directory.

The "ro" in the Wizard entry means that the library is [read-only](#). If you have an application that uses system tables that are to be written

to at any point in your program, you would specify "rw" rather than "ro". For example, suppose you have an application in a file called STOCKAPP.MDB that employs the use of system tables that can be modified. You would add the following entry to use the MDB file as a library:

```
[Libraries]
wizard.mda=ro
stockapp.mdb=rw
```

Given these library entries, the WIZARD.MDA library will be loaded read-only, and the STOCKAPP.MDB library will be loaded as read-write. You can now open an [Immediate window](#) in a user database and invoke sub and [function](#) procedures from STOCKAPP.MDB, or open tables, queries, forms, or reports with DoCmd commands. Even though you can access the code and the [database objects](#), you cannot see them in the Database window.

When a database is loaded as a library, it cannot be opened as a user database.

Writing and Debugging Access Basic Library Code

When you write an Access Basic application for use as a library, you are doing little more than writing the application in a user database with the intention of using it as a library at a later point. Because of this, a rule of thumb is to make sure the application works completely before trying to use it as a library.

Although this rule of thumb is enough to successfully create many types of library applications, there are some important pitfalls to watch for when writing a library application, even if the application works perfectly as a user database.

Debugging an Error in a Library Database

If the [library database](#) generates an error that only occurs while it is a library, it can be very difficult to locate. An error might occur that gives you some idea of the general area of the problem, but there may be little or no indication of the offending line. Because you cannot set and use breakpoints and stepping in library applications, you should design error traps that convey meaningful messages and indicate the location of the problem.

Another [debugging](#) tip is to place MsgBox's at milestone areas of the code so that you always have an idea of which code is being executed.

CodeDB() Versus CurrentDB()

Access Basic includes the CodeDB() function for opening library databases. CodeDB() works identically to CurrentDB() if you are running the application as a user database. However, if you are running the application as a library, CodeDB() returns the database

object for the library database from which it was called, while CurrentBD() returns the database object from the current open user database. Because of this, it is easy to confuse one for the other, which results in logical errors that do indicate this is the problem.

Domain Functions

Domain functions include a parameter that is used as criteria for applying the function to a specified set of records. The criteria is in the form of a SQL WHERE statement and assumes first that any table in the criteria is located in the user database. This could pose a problem for your application if you intend to perform a domain function on a library table that happens to have the same name as a user table. Of course, this does not happen frequently; but it becomes very important when you work with Microsoft Access system tables which will have the same name in both databases. If you are to include a table within your library, you should give this table a unique name that you do not expect a user to duplicate such as "My Library Table1."

Macros in a Library Cannot Be Called from a User Database

Of all the objects you can create in a Microsoft Access database, macros are the only type of object you cannot use in a library application. The most obvious problem this presents is that forms require the use of macros in order to have menus. Because of this limitation, you must make sure to use only Access Basic code for programming.

[References](#)

INF: How to Obtain Access Startup Directory with API Calls

Article Number: Q88174
CREATED: 19-AUG-1992
MODIFIED: 31-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

[Access Basic](#) does not have an inherent command to return the Microsoft Access startup directory. However, this can be accomplished within a Microsoft Access Basic program by making Windows API calls.

This article describes the steps involved in creating a [function](#) that returns the Microsoft Access startup directory.

For more detailed information on Windows API calls, [query](#) on the following words in the Microsoft Knowledge Base:

handle and api and dll and kernel and user and gdi

MORE INFORMATION

The Access Basic CurDir\$ function returns the current directory. Because Microsoft Access can be started from a directory other than the current directory, the CurDir\$ function cannot be used to determine the startup directory.

The following program [module](#) makes use of the Windows API functions GetModuleHandle and GetModuleFileName. With the module handle, the path can be obtained with the GetModuleFileName function.

Module

To create an Access Basic function that returns the Microsoft Access startup directory, do the following:

NOTE: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

1. From the File menu, choose New, and select Module.
2. In the [Declarations section](#) of the module, enter the following Declare statements:

```
Declare Function GetModuleHandle% Lib "kernel" (ByVal FileName$)
Declare Function GetModuleFileName% Lib "kernel" (ByVal hModule%, _
    ByVal FileName$, ByVal nSize%)
```

3. Enter the following commands to create the function:

```
Function StartUp_Dir ()
    hModule% = GetModuleHandle("MSACCESS.EXE")
    Buffer$ = Space$(255)
    Length% = GetModuleFileName(hModule%, Buffer$, Len(Buffer$))
    Buffer$ = Left$(Buffer$, Length%)
    Msg$ = "Startup path and filename: " + Buffer$
    MsgBox Msg$
End Function
```

4. From the Run menu, choose Compile All to compile the function.

5. From the File menu, choose Save, and enter "Startup Directory" as the name of your module. Choose OK.

6. From the View menu, choose Immediate Window to invoke the function. Type the following command into the Immediate window, then press ENTER:

```
? StartUp_Dir()
```

Uses and Variations

The Startup directory is displayed when

```
? StartUp_Dir()
```

is issued in the Immediate window. This function can also be incorporated into other program modules and used in expressions. For example, entering =Startup_Dir() as the OnPush property of a button on a form returns the startup directory of Microsoft Access whenever the button is chosen.

For additional functionality, you can change the ACCESS.EXE argument for the Windows API GetModuleHandle function so the function returns the startup directory of another program started within the Windows environment. Furthermore, you can pass a program name as a variable to the Windows API function, giving even more flexibility to the function.

References:

"Microsoft Windows Software Development Kit", Microsoft Press, 1992

"Programming Windows: the Microsoft Guide to Writing Applications for Windows 3", Charles Petzold, Microsoft Press, 1990

"Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference" Volumes 1 - 6, Microsoft Press, 1992

[References](#)

INF: ODBC Setup for Access and SQL Server

Article Number: Q88173
CREATED: 19-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.10
 - Microsoft [SQL](#) Server versions 1.10, 1.11, and 4.20 for MS-DOS and Microsoft OS/2
 - Microsoft SQL Administrator version 4.20 for Windows
-

Summary:

Installation and setup of Microsoft [ODBC](#) for Microsoft and Sybase SQL Server requires the use of ISQL [command line](#) procedures to properly run the INSTCAT.SQL script.

If INSTCAT.SQL has not been properly implemented in SQL Server, the following error will be displayed while attaching a SQL [table](#) in Microsoft Access:

```
[Microsoft][ODBC SQL Server Driver] The ODBC catalog stored
procedures installed on server <server_name> are version xx.xxxx;
version xx.xx.xxxx is required to ensure proper operation. Please
contact your system administrator.
```

More Information:

To properly configure SQL Server for use with Microsoft ODBC, you must run the SQL script file INSTCAT.SQL (shipped with the ODBC disk) to set up the proper stored procedures that provide catalog information used by Microsoft ODBC. Microsoft SAF for MS-DOS and OS/2 are limited to 511 lines of code in a SQL script. INSTCAT.SQL has well over 511 lines of code in it.

SQL Administrator does not recognize the GO command used in INSTCAT.SQL.

The proper way to install the catalog stored procedures using INSTCAT.SQL is to run INSTCAT.SQL from the command line using the SQL Server facility ISQL (Interactive SQL).

The ISQL facility is run from either the MS-DOS or OS/2 command prompt. The [syntax](#) for this procedure is:

Note: In the following sample code, an underscore `_` is used as a line continuation character. Remove the underscore when re-creating this code in [Access Basic](#). Do not include the angle braces `<>` in the command.

```
isql /U <sa login name> /n /P <password> /S <SQL server name> /i_
<drive:\path\INSTCAT.SQL> /o <drive:\path\output file name>
```

/U The login name for the system administrator
/n Eliminates line numbering and prompting for user input
/P Password used for the system administrator
/S The name of the server to set up
/i Provides the drive and fully qualified path for the location of
INSTCAT.SQL
/o Provides isql with an output file destination for results or
the
process including errors

(Note: The /P option is case sensitive.)

Example

```
isql /U sa /n /P skier /S DUMMY_SERVER /i d:\SQL\INSTCAT.SQL /o_  
d:\SQL\output.txt
```

After you run INSTCAT.SQL, you should run the RECONFIGURE command against the MASTER [database](#) using SAF. See pages 205-212 of the "Microsoft SQL Administrator's Guide" version 4.2 manual for more information.

Note: This information is available in the Microsoft Access README.TXT file.

[References](#)

INF: Norton Desktop 1.x Is Not Compatible with Access

Article Number: Q88172
CREATED: 19-AUG-1992
MODIFIED: 01-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

Running Norton Desktop for Windows versions 1.x with Microsoft Access results in unpredictable behavior in Microsoft Access (for example, you may lose the background screen or experience a general protection [GP] fault). To resolve these problems, upgrade to version 2.0 of Norton Desktop for Windows.

The product included here, Norton Desktop for Windows, is manufactured by a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

[References](#)

INF: Using SetSysModalWindow API in MS Access Basic

Article Number: Q88171
CREATED: 19-AUG-1992
MODIFIED: 19-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

In Microsoft [Access Basic](#), you can make an application [modal](#) so that users cannot switch to other Windows-based programs while the Microsoft Access application is active. This functionality is achieved with the following Windows application program interface (API) functions:

SetSysModalWindow

-and-

FindWindow

This article discusses the use of the SetSysModalWindow API [function](#) and provides an example of its use in Microsoft Access Basic.

For information when using the FindWindow API function, [query](#) on the following words here in the Microsoft Knowledge Base:

Access and FindWindow

MORE INFORMATION

Microsoft Windows is designed so that the user can switch between applications without terminating one program to run another. There may be times, however, when a Microsoft Access application needs to restrict the user from switching to another application. Examples of this are a [security](#) system or time-critical application that needs to run uninterrupted.

SetSysModalWindow Usage Notes

Consider all of the following usage notes before making a window system-modal.

1. If an application passes the handle of the window as an argument to the SetSysModalWindow, the user is limited to that window. This prevents the user from moving to other applications or bringing up the Task List (either with the mouse or by pressing CTRL+ESC) .
2. If an application uses the SetSysModalWindow function in conjunction with the GetSystemMenu function, the Control (system) menu is disabled and the user is prevented from quitting Windows

by pressing ALT+F4.

3. A window that is system-modal cannot be minimized, maximized, moved, or sized. Microsoft Access help is created with a separate Windows Help file; therefore, help windows created with the Windows Help Compiler are not available to the user.
4. When an application uses the SetSysModalWindow API function, the system-modal window must be closed before other applications or windows can be accessed. Therefore, when using this function, be sure that the application has a means to exit Microsoft Access.
5. Any child window that is created by a system-modal window becomes a system-modal window. For example, if Microsoft Access is system-modal and a form is opened, that form becomes system-modal until it is closed.

How to Use the SetSysModalWindow Function

To use the SetSysModalWindow API function within Microsoft Access Basic, create the sample code below.

NOTE: In this sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

1. Within the Declarations Section, enter the following:

```
Option Explicit
```

```
Declare Function SetSysModalWindow% Lib "User" (_  
    ByVal hwnd%)
```

```
Declare Function FindWindow% Lib "User" (_  
    ByVal lpzClassName As Any, _  
    ByVal lpzWindow As Any)
```

2. Create a procedure by entering the following:

```
Function System_Modal_Test ()  
  
    Const lpClassName = "OMain"  
    ' Declares the class name constant used in  
    ' the FindWindow API function.  
  
    Dim AccessHandle  
    Dim Success As Integer  
  
    ' FindWindow returns the Handle for Microsoft Access.  
    AccessHandle = FindWindow(lpClassName, 0&) 'Note this is a zero  
  
    ' The Handle for Microsoft Access is sent then Microsoft Access  
    ' becomes modal.  
    Success = SetSysModalWindow(AccessHandle)  
  
End Function
```

3. From the File menu, choose Save and save the module as "Modal Access" (without the quotation marks). Choose the OK button.
4. From the Run menu, choose Compile All.
5. From the View menu, choose Immediate Window. In the Immediate window, type the following, then press ENTER:

```
?System_Modal_Test()
```

After running the program, test the modal functionality by trying to select other windows or resize or move the system-modal window.

REFERENCES

=====

For more information on the Modal Property in Microsoft Access, version 1.1, refer to online help.

For more information on API functions, refer to:

- "Microsoft Windows Software Development Kit," Microsoft Press, 1992
- "Programming Windows: The Microsoft Guide to Writing Applications for Windows 3," Charles Petzold, Microsoft Press, 1990
- "Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference," Volumes 1-6, Microsoft Press, 1992

For more detailed information on Windows API calls, query on the following words here in the Microsoft Knowledge Base:

handle and API and DLL and kernel and user and GDI

[References](#)

INF: Determining the Existence of a Query in Access

Article Number: Q88170
CREATED: 19-AUG-1992
MODIFIED: 22-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

SUMMARY

This article describes how to determine the existence of a [query](#) with [Access Basic](#) code.

MORE INFORMATION

Microsoft Access does not have an intrinsic [function](#) to determine the existence of a specific query. Using Access Basic code, you can easily determine if a query exists in the [database](#).

Example

To create a function that will return the existence of a query, do the following:

1. Within the NWIND.MDB database, create a new [module](#) by choosing New from the File menu, and then select Module from the list.
2. Enter the following function into the Code window:

```
Function QueryExists%(QueryName as String)  
    On Error GoTo ItExists
```

```
    Dim MyDb as Database  
    Dim MyQry as QueryDef  
    Set MyDb = CurrentDB()
```

```
    '-----  
    'If there is a query by this name, an error will occur  
    'on the MyDb.CreateQueryDef method when you try to create the  
    'existing query. Code execution will transfer to the ItExists  
    'label if this is the case.  
    '-----
```

```
    Set MyQry = MyDb.CreateQueryDef(QueryName)
```

```
    '-----  
    'If the query was create, we need to clean up by deleting the  
    'newly create query. You may elect to have this function  
    'create the query. If this is the case, delete the  
    'My.DeleteQueryDef line from this function.  
    '-----
```

```
MyDB.DeleteQueryDef(QueryName)

QueryExists = False

Exit Function

ItExists:
'-----
'The query exists. Return True and exit from the function.
'-----

QueryExists = True
Exit Function

End Function
```

3. From the View menu, choose Immediate Window. Enter the following command, then press ENTER:

```
?QueryExists("Catalog")
```

4. "-1" should be printed in the Immediate window (-1 in Access represents True).
5. Experiment by calling the function with a nonexistent query. This should print "0" (which represents False).

[References](#)

INF: Incrementing the Numeric Portion of a String

Article Number: Q88169
CREATED: 19-AUG-1992
MODIFIED: 12-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article describes a formula that sets the default value of a bound text field in a form to the next higher numeric value (default value = default value + 1).

The sample formula in this article follows the procedure below:

1. Finds the highest numeric value used in a table.
2. Strips any leading text characters.
3. Adds 1 to the value found in step 1.
4. Reconnects the leading text characters.

NOTE: This example assumes that the number of leading text characters is known at the time the form is designed.

More Information:

Use the following procedure to increase the default value of a bound text field on a form:

1. Open an existing database, or create a new database by choosing New from the File menu.
2. Create a new table.
3. Add a field called Book ID with a Text data type. From the Edit menu, choose Set Primary Key.
4. Add a second field of any data type to the table.
5. Save the table as Increment.
6. Switch to Design view and enter the following values in three separate rows:

Book ID	Second Field
BO-110	
BO-111	
BO-112	

7. Close the table.
8. Create a new form based on the Increment table.
9. From the View menu, choose Field List.
10. Drag the fields Book ID and Second Field from the [field list](#) to the form.
11. Change the form's Default View property to Single Form.
12. Set the Default Value property of the Book ID [text box](#) to the following:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

```
= "BO-" & Right(DMax("[Book ID]","Increment"),Len(DMax_  
("[Book ID]","Increment"))-3)+1
```

13. Switch to [Form view](#) and enter a new [record](#).

Each portion of the above formula completes a task, as follows:

- DMax("[Book ID]","Increment") locates the highest value in the table.
- Len(DMax(...)) finds the length of the value returned by the DMax() [function](#).
- Right(DMax(...),Len(...)-3) strips the three leading text characters.
- +1 increments the result of the Right() function.
- "BO-" and concatenates the text "BO-" on to the incremented number.

Note that this example works correctly when the form's default property is Single View; it may not work consistently in Continuous View. When you move to a new record and begin to enter data, Microsoft Access displays the next empty record. The default values for this record are calculated before the record you are currently editing is committed. Because the DMax() function calculates the maximum value from those records stored in the table, the Book ID of the record you are currently editing is repeated.

If you are working in a multiuser environment, it is possible that more than one user may receive the same calculated Book ID value. Although you can manually change the Book ID, you can also maintain the highest numeric value in a separate table, using a [macro](#) or Access Basic code.

References:

For more information about multiuser counters, [query](#) on the following words in the Microsoft Knowledge Base:

multiuser and counter

[References](#)

INF: Error Msg: Can't Hide the Control that Has Focus

Article Number: Q88168
CREATED: 19-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

If you change the Visible property of a [control](#) to False, and that control has focus, you will receive the following error message:

Can't hide the control that has focus

Although you can't trap this error message within a [macro](#), you can call [Access Basic](#) code that can trap this error and return the results to a macro.

More Information:

Steps to Reproduce Behavior:

1. From the File menu, choose New to create a new [form](#).
2. Using the Toolbox while in [Design view](#), place a [command button](#) on the form.
3. Set the following properties for the command button:

Control name:	Controller
Caption:	Turn Off
On Push:	TurnOff

Note: To display the command buttons properties, choose Properties from the View menu. Once the Properties window is displayed, choose the command button with the mouse. The caption of the Properties window should be "Command Button" at this time.

4. Close and save the form as "Turn Off Controller".
5. Repeat steps 2-3 for the second form, but use the following properties for the new command button:

Control name:	Subject
Caption:	Subject

6. Close and save the form as "Turn Off Controller Subject".
7. From the File menu, choose New to create a new macro.
8. From the View menu, choose Conditions to display the Condition

field.

9. Add the following actions to this macro:

Condition	Action	Comment
TurnOff()<>2509	MsgBox	Successfully turned off
...	StopMacro	Note the ellipsis (...)
	MsgBox	Message here

MsgBox Action

Message: Successfully turned off

StopMacro Action

none to set

MsgBox Action

Message: Message here

10. Close and save the macro. Save the macro as "Turn Off."
11. From the File menu, choose Module to create a new module.
12. Within the Code window of the module, add the following code:

```
'-----  
'Global Section  
'-----  
Options Explicit  
  
'-----  
'Functions  
'-----  
  
Function TurnOff ()  
    'If a control has focus, then this will trap error 2509  
  
    On Error GoTo errhandler  
  
    DoCmd SetValue Forms![Turn Off Subject]![Subject].Visible, False  
    TurnOff = True  
  
Exit Function  
  
errhandler:  
  
    If Err = 2509 Then  
        TurnOff = 2509    'trap the "Has focus scenario"  
    Else  
        TurnOff = 0
```

End If
Exit Function

14. Close the module window and save as "Turn Off."
15. Open both the "Turn Off" and the "Turn Off Subject" forms.
16. With the mouse, choose the Controller button.

Because "Subject" has focus on the form within the scoping of the form, this will error out and display the message box:

Message here

[References](#)

INF: How to Determine if a Specific Windows Program Is Running

Article Number: Q88167
CREATED: 19-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

There are times when you only want one instance of an application to run within Windows. If you place a [command button](#) on a [form](#) that starts the Windows Calculator (CALC.EXE) program, the user can start many instances of Calculator. This is very inefficient on memory and system resources. If your program determines if the application is already running, it can either start the application for the first time, or bring the existing application to the front.

To determine if a specific program is running, call the Windows application programming interface (API) [function](#) FindWindow. FindWindow returns the handle of the window whose class is given by the lpClassName parameter and whose window name, or caption, is given by the lpCaption parameter. If the returned value is zero, the application is not running.

More Information:

When an application is started from the Program Manager it registers the class name of the form. The window class provides information about the name, attributes, and resources required by your form. The Access window has a class name of "OMain." For additional command class names, see further below.

By calling FindWindow with a combination of a specific program's class name and/or the [title bar](#) caption, your program can determine whether that specific program is running.

You can determine the class name of an application by using SPY.EXE, which is supplied with the Microsoft Windows Software Development Kit (SDK) version 3.1.

If the window has a caption bar title, you can also use the title to locate the instance of the running application. This caption text is valid even when the application is minimized to an icon.

The following example shows three ways to determine if the Windows 3.1 Calculator is running. To create the program, do the following:

1. From the Program Manager File menu, choose Run and enter CALC.EXE to start Calculator.
2. From the File menu, choose New to create a new [database](#).

3. From the File menu, choose New, then select Form to create a new blank form.
4. When the form is displayed, from the View menu, choose Toolbox.
5. Select the command button tool on the Toolbar, then double-click on the form to place a command button on the form.
6. From the View menu, choose Properties to display the Properties window. Set the following command button property:

Property	Value
On Push	=CalculatorUp()

7. Close and save the form as "FindWindow Form."
8. From within the Database window, open a new module and enter the Access Basic code listed below.

Note: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```

=====
'Declarations section
=====
Option Explicit
Declare Function FindWindow% Lib "user" (_
                                ByVal lpClassName As Any, _
                                ByVal lpCaption As Any)
=====
'End of Declarations section
=====

=====
'Code section
=====

Function CalculatorUp ()

    Const lpClassName = "SciCalc"
    Const lpCaption = "Calculator"

    'This demonstrates three different ways to call FindWindow:
    '1. The ClassName only.
    '2. The Caption only.
    '3. Both the ClassName and the Caption

    MsgBox "Calculator Handle = " & FindWindow(lpClassName, 0&)
    MsgBox "Calculator Handle = " & FindWindow(0&, lpCaption)
    MsgBox "Calculator Handle = " & FindWindow(lpClassName, lpCaption)

    'This function could return the existence of a window.
    CalculatorUp = FindWindow(lpClassName, 0&)

```

End Function

```
'=====
'End of code section
'=====
```

9. Close and save the module as "FindWindow Code."
10. From within the Database window, open FindWindow Form. Choose the FindWindow Form command button with CALC.EXE running and without CALC.EXE running. If CALC.EXE is running, your application will print an arbitrary handle. If CALC.EXE is not running, your application will print the number zero as a handle.

Below are some class names of common applications that are shipped with Windows:

Class Name	Application
OMain	ACCESS.EXE
SciCalc	CALC.EXE
CalWndMain	CALENDAR.EXE
Cardfile	CARDFILE.EXE
<u>Clipboard</u>	CLIPBOARD.EXE
Clock	CLOCK.EXE
CtlPanelClass	CONTROL.EXE
XLMain	EXCEL.EXE
Session	MS-DOS.EXE
Notepad	NOTE.EXE
pbParent	PBRUSH.EXE
Pif	PIFEDIT.EXE
PrintManager	PRINTMAN.EXE
Progman	PROGMAN.EXE (Windows Program Manager)
Recorder	RECORDER.EXE
Reversi	REVERSI.EXE
#32770	SETUP.EXE
Solitaire	SOL.EXE
Terminal	TERMINAL.EXE
WFS_Frame	WINFILE.EXE
MW_WINHELP	WINHELP.EXE
#32770	WINVER.EXE
OpusApp	WINWORD.EXE
MSWRITE_MENU	WRITE.EXE

References:

"Programming Windows: the Microsoft Guide to Writing Applications for Windows 3," by Charles Petzold, Microsoft Press, 1990

"Peter Norton's Windows 3.0 Power Programming Techniques," by Peter Norton and Paul Yao, Bantam Computer Books, 1990

"Microsoft Windows 3.0 Software Development Kit: Reference Volume 1"

INF: Like vs. '=' with Wildcard Characters in Query Searches

Article Number: Q88166
CREATED: 19-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

When searching a string using wildcards (for example, * or ?), you must use Like or Not Like in the Criteria field rather than = or <>. When you use the = (or the <>) operators, Microsoft Access treats them as actual characters to search for.

More Information:

If you have a table with one column called Col_Name and the following four records

```
Col_Name
-----
Foos
Foosball
Foos ball
Foos*
```

and you make a query based on the table with the criteria

```
="Foos*"
```

your query will return:

```
Foos*
```

If your criteria is:

```
Like "Foos*"
```

your query will return the following four records:

```
Foos
Foosball
Foos ball
Foos*
```

The SQL statements built by Microsoft Access for these queries are as follows.

The SQL statement using Like is:

```
SELECT DISTINCTROW tablename.Col_Name
FROM [tablename]
WHERE ((tablename.Col_Name Like "foos*"));
```

The SQL statement using = is:

```
SELECT DISTINCTROW tablename.Col_Name
FROM [tablename]
WHERE ((tablename.Col_Name="foos*"));
```

[References](#)

INF: Introduction to Windows Programming for MS-DOS Programmer

Article Number: Q88164
CREATED: 19-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

This article discusses some differences between programming in the MS-DOS environment and programming in the [event](#)-driven Windows environment.

More Information:

For the purposes of this discussion, consider the use of the term "traditional programmer" as someone who has not programmed in Windows, but has experience programming in an MS-DOS environment.

As a traditional programmer, you may not only have become comfortable with a particular programming style, but also with certain accepted fundamentals, such as writing an instruction and expecting it to be carried out in a controlled order. [Access Basic](#) makes good use of Windows, making it easy to learn to program.

"One Entry, One Exit" vs. Event-Driven Programming

Consider the following pseudocode of a program designed to get user input, count all the records in a [table](#), and display the result in a box if the user presses 1, or exit if the user presses 2.

```
START PROGRAM

LOOP WHILE TRUE
    GET KEYPRESS INTO X

    IF X IS "1"
        COUNT ALL RECORDS IN THE TABLE INTO Y
        DRAW BOX FROM ROW 10 COLUMN 5 TO ROW 12 COLUMN 7
        DISPLAY Y AT ROW 11 COLUMN 6

    IF X IS "2"
        EXIT LOOP

END LOOP

STOP PROGRAM
```

The purpose of this program is to continuously loop until a key press of a 1 or 2 is detected. At that point, a decision is made to

perform some sort of operation, or to ignore the keypress and continue looping. The programmer has full control over what happens.

The Windows programming model is event-driven and graphic object oriented. In other words, programming in Windows involves creating objects and modifying aspects (or properties) of those objects based on different events. Consider the following sample program that presents two buttons to the user. If the user chooses the Count button, the program counts the records in the database and displays the result in a window. The user can press the Exit button to exit from the program.

First, you create the necessary objects. Most of this phase of Access Basic programming is created graphically with the Access Forms designer. The list of controls and properties below defines a form that will be used to illustrate this.

Form: "MasterForm"

Push Button: "CountButton"
Caption: "Count"
OnPush: "=DisplayCount()"

Push Button: "ExitButton"
Caption: "Exit"
Caption: "=CloseProgram()"

Text Box: "DisplayWindow"

Note: OnPush is a property of command buttons that gives you the ability to invoke an Access Basic procedure or macro.

You can then create the modules that the objects will invoke. In this case, buttons are the only objects that will have the ability to invoke procedures. The procedures shown below are pseudo-code examples. The first procedure defined is the DisplayCount procedure:

```
PROCEDURE DisplayCount()  
  
    COUNT ALL THE RECORDS IN THE TABLE INTO Y  
    CHANGE THE DISPLAYWINDOW TEXT PROPERTY TO Y  
  
END PROCEDURE
```

Notice that the code did not direct the resulting count to display in a box painted on the screen. Instead, the Text property of DisplayWindow was changed to the resulting count value. The next procedure defined is the CloseProgram procedure.

```
PROCEDURE CloseProgram  
  
    CLOSE MASTERFORM  
  
END PROCEDURE
```

Notice that this procedure does not provide an exit from some kind of

loop or other program structure. Instead, it closes the object that contains the buttons and window.

At this point, you have a master form object containing two buttons, a window, and a couple of coded procedures. They are in no special order, they simply exist as part of the form. So, where is the loop that checks for button activity? Where is the command to invoke the program?

The answer is that these do not exist as you might expect them to. You "run" the program by opening MasterForm. When you open the form, all the control objects (that is, the buttons and so on) exist on the form waiting for something to happen. In this example, there is no flow of control (no looping to check activity).

While the form is active, Windows constantly checks for events. When an event occurs, the user's input is put in a queue and "waits in line" until it is processed. For example, when you push the "Count" button, Windows detects that the button object you placed on the form has been affected. Windows sends a "Mouse Click" message to Access. Access then translates the message and determines that the DisplayCount() function should be called based on the "On Push" field of the command button.

Advantages

The traditional programmer will find this new approach to programming a bit challenging. There are a few things to learn and "unlearn," but there are many advantages.

Windows Interface

The Windows interface is one that has been regarded throughout the industry as being very user-friendly. Familiar objects such as push buttons, radio buttons, list boxes, and a wide variety of colors and screen fonts are generally more appealing than standard ASCII text characters.

The Windows Standard

Because Access Basic forces you to some extent into the Windows standard, others who are familiar with Windows applications can immediately recognize the "look and feel" of your application. This reduces the learning time because the user does not have to learn entirely new interface controls and prompts.

Advantages Offered by the Windows Environment

You do not have to worry too much about different devices such as monitors, printer drivers, and so on. The Windows operating environment takes care of most device compatibility and user preference issues. In addition, because Windows handles and processes events, you will find it much easier to create and manage many aspects

of an application.

[References](#)

PRB: "Invalid Range" Error When Importing from MS Excel

Article Number: Q88161
CREATED: 19-AUG-1992
MODIFIED: 25-AUG-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access version 1.0
-

SYMPTOMS

When you import the named range "Database" from Microsoft Excel versions 3.x and 4.x files, the following error appears:

Invalid Range.

This error does not occur when you import from Microsoft Excel version 2.0.

CAUSE

This error is caused by the difference in the Microsoft Excel binary information file format (BIFF) structures between versions:

- BIFF2 has the range listed in its internally defined named range [table](#). Microsoft Access can read this table.
- BIFF3 and BIFF4 use an internal construct, rather than listing "Database" in the internal named range table. Microsoft Access cannot read the internal construct of BIFF3 and BIFF4.

RESOLUTION

There are three recommended workarounds that enable you to import data successfully from Microsoft Excel:

- Save the Excel file in 2.x format (in the File Save As dialog) before importing the file into Microsoft Access.
- Give the range "Database" a second name (such as "My_Database_Range"). Microsoft Excel allows users to assign multiple names to the same range.
- Use the range specification rather than a named range (for example, A1:D50).

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0. This problem does not occur in later versions of Microsoft Access.

MORE INFORMATION

Steps to Reproduce Behavior

1. In Microsoft Excel version 3.0 or higher, create a named range called "Database" (without the quotation marks).
2. Save the file as TEST.XLS
3. In a Microsoft Access [database](#), choose Import from the File menu
4. Choose Microsoft Excel
5. Choose TEST.XLS and choose Database for the named range

[References](#)

INF: How to Hide the Database Window at Startup

Article Number: Q88160
CREATED: 18-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

Summary:

During normal startup of Microsoft Access, the Database window is visible. If you do not want to give a user access to the Database window, you can create an AUTOEXEC macro that will hide the Database window from the user. This article describes how to create a macro to do this.

More Information:

The ability to access the Database window allows the user to modify database objects, which could lead to corruption of the database and database objects. Limiting the user to specific forms and windows can minimize the potential loss of objects and data.

When a Microsoft Access database is first loaded into the environment, Microsoft Access checks for an AUTOEXEC macro and runs it if it exists. The following macro will hide the Database window from the user:

1. From the File menu, choose New to create a new database. Give the database a unique name.
2. From the File menu, choose New to create a new macro.
3. Add the following actions:

Action	Comment
SelectObject	Select a macro in the Database window so the focus is set to the Database window.
DoMenuItem	Hide the object that has the current focus.

4. Set the following properties for the actions entered above:

SelectObject Action

Object Type: Macro
Object Name: Autoexec
In Database Window: Yes

DoMenuItem Action

Menu Bar: Form
Menu Name: Window
Command: Hide

5. From the File menu, choose Save As, and save the macro as "Autoexec."

The macro will hide the Database window from the user; however, menu options still exist to show the Database window. To hide these options from the user, you can create a custom menu for your application, or disable the Show command on the File menu.

References:

For more information about using Windows application programming interface (API) [function](#) calls to disable menu commands, [query](#) on the following words in the Microsoft Knowledge Base:

GetMenu and GetSubMenu

[References](#)

container [security](#)

INF: How to Create a Multiuser Custom Counter

Article Number: Q88159
CREATED: 18-AUG-1992
MODIFIED: 27-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

Microsoft Access allows you to define a field in a table as a counter. When defined as a counter, Microsoft Access manages the unique numbering of fields in the table. If you add a record, Access gives this counter field the next unique value available. As a user, you have no control over this value.

There may be times when you need unique counters that are not immediately sequential in nature. Examples are counters that decrement, or counters that step according to some formula (such as in steps of 10).

This article describes how to implement a custom counter field with Access Basic commands. This will give you more flexibility over the assignment of the actual value you use in this field.

MORE INFORMATION

The following steps outline in general how to create a custom counter. Further below is a more detailed description on how to create a custom counter.

1. Create a separate table that will maintain the next available custom counter. This table will have one field and one record, with the value of the next available counter in this one record.
2. From within Access Basic, open this counter table and retrieve the value stored there.
3. Increment the value retrieved and store the number back into the database.
4. Close the table and use the value in the appropriate table as the next available counter.

In a single-user environment, the steps described above can be done with macro actions. In a multiuser environment, Access Basic is needed to handle the event where this counter table is locked. Macro actions do not provide the locking control needed in the multiuser environment.

Example

The following example describes how to create a custom counter field with positive number values divisible by 10 (that is, 10, 20, 30, and so on) in sequential order.

The Table

1. From the File menu, choose New, then select Table.
2. Add a field to the table called "Next Available Counter." Set the data type of "Next Available Counter" to number.
3. From the Edit menu, choose Primary Key to make "Next Available Counter" the primary key.
4. From the View menu, choose Datasheet View.
5. A dialog box will prompt you to save the table. Save it as "Counter Table."
6. Once in Datasheet view, enter a value of 10 into the "Next Available Counter" field.
7. From the File menu, choose Close to close the table.

The Module

8. From the File menu, choose New, then select Module to create a new module.
9. Add the following function to the module:

```
Function Next_Custom_Counter ()
    Const TABLE_LOCKED = 3000

    On Error GoTo Next_Custom_Counter_Err

    Dim MyDB As Database
    Dim MyTable As Table

    Dim NextCounter As Integer

    '=====
    'Open table and get the next available number, increment value
    'by 10 and save the number back into the table.
    '=====

    Set MyDB = CurrentDB()
    Set MyTable = MyDB.OpenTable("Counter Table")

    MyTable.Edit
    NextCounter = MyTable("Next Available Counter")

    '=====
    'The next line can be changed to conform to your custom counter
    'preferences. This example only increments the value by +10
```

```

each time.
=====

MyTable("Next Available Counter") = NextCounter + 10
MyTable.Update

MsgBox "Next available counter value is " & Str$(NextCounter)
Next_Custom_Counter = NextCounter

Exit Function

Next_Custom_Counter_Err:
    If Err = TABLE_LOCKED Then Resume
    MsgBox "An unexpected error has occurred:" & Err
End

End Function

```

10. From the File menu, choose Save. Save the module as "Custom Counter Demo."

The Form

11. From the File menu, choose New, then select Form. Choose the Blank Form button to start with a blank [form](#).
12. Add a [command button](#) to the form from the [toolbox](#). If the toolbox is not displayed, from the View menu, choose Toolbox.
13. If the Properties window is not displayed, choose Properties from the View menu.
14. Select the command button with the mouse so that the Properties window displays "Command Button" on the [title bar](#).
15. Set the OnPush property to =Next_Custom_Counter().
16. Save the form as "Custom Counter Demo" by selecting the Save As command from the File menu.

Execution

17. From the View menu, choose [Form view](#) to move to Form view. Click on the command button; a message box will display the value of the next counter.

[References](#)

INF: How to Programmatically Create a Table

Article Number: Q88157
CREATED: 18-AUG-1992
MODIFIED: 13-JUL-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1

Summary:

There is no inherent command in Microsoft Access to programmatically create a [table](#). However, the following is a procedure that needs only a special template table to programmatically create any kind of table you want.

This article discusses a user-defined [function](#) for [Access Basic](#) that you can use to accomplish this task. This user-defined function also requires a "template table" that the user can create. The template table is discussed further below.

More Information:

Before you create the function, you must create a template table. You must use a template table to define your own data types. For example, you may want to create a [data type](#) called "Long Text" that is a text [field](#) with a 255-character length. In order to represent this in the template table structure, "Long Text" is defined as a text [column](#) with a length of 255.

For example, in the template table below, the user has created a data type called "Long Text," which is a 255-character Text field. The user has also created a "Short Text," "Long Int," and so on.

Table Name: MyTemplate

Field Name	Field Type	Length
Long Text	Text	255
Short Text	Text	50
Long Int	Number	Long Integer
Integer	Number	Integer
Memo	Memo	N/A
Date/Time	Date/Time	N/A
OLE Object	OLE Object	N/A

This template can be used as a generic template for using the CreateTable procedure. Once you have defined your data types in a template table, you can use the CreateTable procedure in the following manner:

Definition

CreateTable <target table>, <template table>, <structure definition>

Where:

<target table> is the name of the table to create and MUST NOT exist in the [database](#). You may want to code a routine to check for the existence of such a table before invoking this procedure.

<template table> contains the user-defined data types that will be used in creating the target table.

<structure definition> defines the structure of the columns in the table.

Where:

<structure definition> = "<field name> As <user-defined data type>,..."

Note: There are quotation marks around the entire <structure definition> parameter.

Below is an example of how you would invoke the CreateTable procedure to create a table based on the template table discussed earlier:

```
CreateTable "NewTable",  
            "MyTemplate",  
            "First Name As Short Text,  
            Last Name As Short Text,  
            Description As Long Text,  
            Amount As Integer,  
            Notes As Memo"
```

How CreateTable Works

As mentioned earlier, there is no Access Basic command or function to create a table. Fortunately, Microsoft Access supports the [SQL](#) SELECT INTO command that can be used to create a table. However, SELECT INTO requires a table for its FROM clause, which explains why you need to have a template table.

CreateTable parses out the structure definition, then builds a SELECT INTO statement based on that definition. Once the SQL statement is built, the [QueryDef](#) object is employed to invoke the command, resulting in a new table.

Procedure Listing

Note: In the following sample code, an underscore _ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Sub CreateTable (TargetTbl As String, TemplateTbl As String,  
                StructureDef As String)
```

```

Dim LineChunk As String, SelectStmt As String, TempChunk As String
Dim CharPos As Integer, HomePos As Integer, BuildLoop As Integer
Dim CrtTblDB As Database, CrtTblQry As QueryDef

BuildLoop = True
HomePos = 1
SelectStmt = "Select "

Do While BuildLoop
    If InStr(HomePos, StructureDef, ",") <> 0 Then
        LineChunk = Trim(Mid$(StructureDef, HomePos, (InStr(HomePos, _
            StructureDef, ",") - HomePos)))
    Else
        LineChunk = Trim(Mid$(StructureDef, HomePos))
        BuildLoop = False
    End If

    TempChunk = Trim$(Mid$(LineChunk, InStr(UCase$(LineChunk), _
        " AS ") + 3))
    SelectStmt = SelectStmt & "[" & Trim(Mid$(TempChunk, 1)) _
        & "]" & " As [" & Trim(Mid$(LineChunk, 1, _
            InStr(UCase$(LineChunk), " AS "))) & "]"

    HomePos = InStr(HomePos, StructureDef, ",") + 1
Loop

SelectStmt = Left$(Trim(SelectStmt), Len(Trim(SelectStmt)) - 1) & "_
    Into [" & TargetTbl & "] From [" & TemplateTbl & "];"

Set CrtTblDB = CurrentDB()
Set CrtTblQry = CrtTblDB.CreateQueryDef("TempQuery", SelectStmt)

CrtTblQry.Execute
CrtTblQry.Close
CrtTblDB.DeleteQueryDef ("TempQuery")
End Sub

```

[References](#)

INF: 'Continued...' Header for Groups on More than One Page

Article Number: Q88156
CREATED: 18-AUG-1992
MODIFIED: 16-SEP-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to create a [label](#) that will print at the top of each [report](#) page when a [group](#) of data spans more than one page.

The events used capture the value of the group in the page [footer](#), and if this value is the same as the page [header](#) value on the next page, the label containing "Continued from previous page..." is displayed.

MORE INFORMATION

Below is a step-by-step example of how to create this report by modifying the sample [database](#) NWIND report "List of Products by Category."

Open NWIND, the sample database that is included with Microsoft Access.

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in [Access Basic](#).

1. From the File menu, choose Module to create a new [module](#).
2. Enter the following global declaration and functions into the module:

```
'-----  
'Global Declarations  
'-----  
Global CurrentGroupVal as String  
  
'-----  
'Code Section  
'-----  
  
Function SetGlobalVar (InReport as Report)  
    CurrentGroupVal = InReport!SetGroupVal  
End Function  
  
Function SetContinuedLabel (InReport as Report)  
    InReport!ContinuedLabel.Visible = _  
        IIf(Trim(InReport!CheckGroupVal) = _  
            Trim(CurrentGroupVal)), True, False)  
End Function
```

3. In the Database window, choose the Report button to open the "List of Products by Category" report. Select "List of Products by Category," then choose the Design button.
4. Increase the height of the Details sections to approximately 1 inch. This will make more than one category split between two pages so that "Continued from previous page..." will be displayed.
5. From the View menu, choose Toolbox so that the toolbox is displayed.
6. Using the toolbox, create a label in the page header section, below the existing field, with a caption that reads "Continued from previous page...". Change the Control Name property to "ContinuedLabel". Set the Visible property to "No". You will have to increase the height of the page header to accommodate the label. For additional help on the toolbox and labels, search the online Help file for "toolbox" and "label."
7. From the View menu, choose Field List to display the field list.
8. Create a hidden text box bound to the group field [Category Name]. Place this text box in the page header section, and change the Control Name property to "CheckGroupVal." For additional help binding controls on a form, search the online Help for "Controls: Adding" and select "Binding a control to a field."
9. From the View menu, choose Properties to display the property sheet.
10. Click on any portion of the page header not covered by a control to display the page header properties in the property sheet.
11. Set one of the following properties of the page header:

```
OnFormat:   =SetContinuedLabel(Report)
```

-or-

```
OnPrint:    =SetContinuedLabel(Report)
```
12. Create a hidden text box bound to the group field [Category Name] in the page footer section, and change the control name property to SetGroupVal.
13. Click on any portion of the page footer not covered by a control to display the page footer properties in the property sheet.
14. Set one of the following properties of the page footer:

```
OnFormat:   =SetGlobalVar(Report)
```

-or-

```
OnPrint:    =SetGlobalVar(Report)
```

15. From the File menu, choose Print Preview. The report "List of Products by Category" will have a "Continued from previous page..." message at the top of the pages where the category details span more than one page.

[References](#)

INF: Adding a New Record to a Combo Box with a Double-Click

Article Number: Q88148
CREATED: 18-AUG-1992
MODIFIED: 30-AUG-1993
VERSION(S): 1.00 1.10

The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1
-

SUMMARY

This article describes how to set up a [macro](#) that will allow users to add a new value to a [combo box](#). The steps to add this functionality to the Order [form](#) in the NWIND [database](#) are also explained.

MORE INFORMATION

To generate an example, do the following:

1. Open NWIND, the sample database included with Access.
2. Create a new macro called Macro1.

Below is an example of how Macro1 should look, and below is an explanation of each action:

Macro Name	Condition	Action
NewRecord		OpenForm
OnClose	[Employee ID] Is Not Null	DoMenuItem
	...	SelectObject
	...	ReQuery
	...	SetValue

NewRecord Actions

OpenForm: Opens the Employees form to add a new employee.

Form Name: Employees
View: Form
Data Mode: Add
Window Mode: Normal

OnClose Actions

DoMenuItem: Saves the [record](#) entered into the employees form and prevents errors if no record is entered.

Menu Bar: Form
Menu Name: File
Command: Save Record

SelectObject: Selects the open Orders form

Object Type: Form
Object Name: Orders
In Database Window: No

ReQuery: Requery the combo box in the Orders form.
ControlName: Employee ID

SetValue: Takes the value from the newly entered Employee ID in
the Employees form and puts it in the Employee ID in the
Orders form.

Item: Forms!Orders![Employee ID]
Expression: Forms!Employees![Employee ID]

3. Modify the Orders form [Employee ID] combo box property by changing the following properties:

On Double Click: Macro1.NewRecord
Status Bar Text: Double-click to Add a New Employee

4. Modify the Employees form by changing the Form property:

On Close: Macro1.OnClose

To see how this process works, open the Orders form in browse mode, then double-click on the Employee ID combo box; the Employees form will open. After filling in the pertinent data, close the form. When the form is closed, the newly entered data will be transferred to the Orders form.

References

combobox listbox [list box](#)

INF: OnCurrent Is Not Triggered on Unbound Forms

Article Number: Q88131
CREATED: 18-AUG-1992
MODIFIED: 25-JUN-1993
VERSION(S): 1.00

The information in this article applies to:

- Microsoft Access versions 1.0

Summary:

If a form is unbound, macros or modules bound to the OnCurrent property will not be executed. In order to modify properties that can be changed at run time, the macro or module must be bound to either the OnOpen property of the form or the OnEnter property of the first control in the tab order of the form.

More Information:

You can use the OnOpen property to trigger a macro or module each time a form is opened. Changes made to properties such as Visible will occur BEFORE the form is displayed. This allows you to make changes that the user does not see.

If, however, you want the user to see the changes taking place, the macro must be run after the OnOpen property.

This example creates a simple form with a text box attached to the form. The macro that is triggered automatically sets the Visible property of the text box to on and off alternately so that it appears to the user that it flashes several times when the form is opened.

Note: You cannot change the Visible property of a control that has focus, so there must be at least one other object on your form that can get the focus.

1. From the File menu, choose New, then select Form (ALT+F, W, F).
2. Leave the Select a Table/Query box blank, and click choose the Blank Form button.
3. Select the text box tool from the toolbox and click on the form to add a new unbound text box. Set the following properties:

```
Control name: Text0  
Control source: ="Hello World"
```

4. Add a second text box.

```
Control name: Text2
```

5. Save the form as Flash by choosing Save from the File menu.

6. Open a new macro and from the View menu, choose Macro Names. Add the following Macro Names and Actions and then save this macro group as "Flashing Macros."

Macro Name	Action
Repeat It	GoToControl RunMacro
Flash	SetValue RepaintObject SetValue RepaintObject

Repeat It Actions

GoToControl
Control Name: Text2

RunMacro
MacroName: Flashing Macros.Flash
Repeat Count: 2

Flash Actions

SetValue
Item: Forms![Flash]![Text0].Visible
Expression: No

RepaintObject
Object Type: Form
Object Name: Flash

SetValue
Item: Forms![Flash]![Text0].Visible
Expression: Yes

RepaintObject
Object Type: Form
Object Name: Flash

7. Close and save the macro as "Flashing Macros."
8. Return to the form Flash. Display the property sheet, select the control Text0, and set the following properties:

OnEnter: Flashing Macros.Repeat.

[References](#)

