Microsoft Access Knowledge Base

Welcome to the Microsoft Access Knowledge Base help file.   This help system includes 419 articles that relate to the Microsoft Access Relational Data Base for Windows.   These articles were developed by Microsoft's Product Support Engineers.   This help file is a snapshot of the Knowledge Base as of January 14, 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, 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.

Tables
Queries
Forms
Reports
Macros/Modules
Import/Export/Attach
Multi-user/Network
OLE/DDE
SQL Server/ODBC
Security
Setup
Third Party
Misc

Tables/DB Design

Queries

Forms

Reports/Printing

Macros/Modules

Import/Export

Multi-user/Network

OLE/DDE

SQL Server/ODBC

Security

Setup

Third Party

Misc

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.

Informational articles are Summary/More Information articles. They 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 (that is, the articles have a RAID bug number). 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.

Compuserve

The MS Access Forum is administered and maintained by the Product Support Services (PSS) of Microsoft. The Forum is also monitored by representatives from Product Development, Training, Customer Service, Product Management, Marketing and Administration.

Please leave a message with comments, criticisms, concerns, praise and advice. Your contribution and help in developing this forum is appreciated. If you wish to address a private message to the SYSOP only, use an asterisk:

(To:*SYSOP).

[Membership Announcement](#)
[A Few Simple Tips Suggestions](#)
[CompuserveSysops](#)

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.

A Few Simple Tips Suggestions

While we try to operate as freely as possible, there are a few simple rules we expect all users to follow. Of course, all of the CompuServe rules regarding appropriate language and previously copyrighted materials apply.

1. 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."

2. All Message Subtopics correspond to the same Libraries and Conference Rooms.

3. 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

4. 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.

Compuserve Sysops

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

Steve Alboucq	Support Manager	72350,3370
MariEsther Burnham	Sysop	70363,152
Monte Slichter	Sysop	70621,1344
Kim Hightower	Sysop	70761,655
Kim Abercrombie	Sysop	70761,650
Dan Madoni	Sysop	70761,640
Ryan LaBrie	Sysop	72350,3371
Joe Howard	Sysop	72350,3367
Don Funk	Sysop	72350,3372
Tim Leidig	Sysop	72350,3374
Gary Yukish	Sysop	72420,1356

The Windows Interface: An Application Design Guide

Weve included this free book and disk set. The book covers user interface design techniques for the Windows™ operating system environment.   Its the same book our designers at Microsoft use when theyre developing applications for Microsoft Windows!

The disk includes a sample buttons and cursors library, as well as an online style guide browser.   Use it to create true Windows-based database applications!

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 youre 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 youre interested in becoming a trainer for Microsoft Access, call (800) 227-4679 to request an application.

Periodicals

Access Advisor	
Data Based Advisor	Data Based Solutions, Inc.
DBMS Client/Server Computing	M&T Publishing Inc.
Inside Microsoft Access	
Smart Access	

Books

ABC's of Microsoft Access	SYBEX
Access Consulting	Prima Publishing
Access for Windows Power Programming	
Access from the Ground Up	
Access Programmer's Reference	Que
Access Workshop	Brady Books
Critical Information Network	
Guide to Access	
Inside Microsoft Access	New Riders Publishing
Microsoft Access Developer's Guide	Sams
Microsoft Access Inside & Out	
Microsoft Access Instant Reference	SYBEX
Microsoft Access Programming	
Microsoft Access Running Start	SYBEX
MS Access Handbook	Bantam Books
PC Learning Labs Teaches Access	
PC World Access Handbook	IDG Books
Running Microsoft Access	
teach yourself... Access	
Understanding Microsoft Access	SYBEX
Using Access for Windows, Special Edition	
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 Microsofts leading-edge database program by following best-selling author Miriam Liskins 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 Shammas

Microsofts 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 Shammas latest book is an accessible, hands-on introduction to database programming for anyone who is familiar with the Basic language.   Shammas relaxed, friendly style leads the reader gently but thoroughly through the fundamentals of Access Basic programming to using the host of powerful database extensions.   Its a practical tutorial approachloaded with ready-to-use program code and peppered with professional tips, tricks, and warningsthat has proven successful in the authors 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 softwareis 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 datafrom a small business manager to an enterprise-wide information system administratorwill 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, youll 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).   Dont 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 cant find in books and manualstips 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 Publishings 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 wont have to spend tedious hours digging through the manuals for solutionsand youll save enough time and money to pay for your subscription many times over!

Order Information: Subscribe to *Smart Access* today12 monthly issues for just $79youll 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.

INF: Adding a New Record to a Combo Box with a Double-Click

Article Number: Q88148
CREATED: 18-AUG-1992
MODIFIED: 15-NOV-1992

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

INF: "Continued..." Header for Groups on More than One Page

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.

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)
      'Note: The following three lines are placed on one line:

         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: How to Programmatically Create a Table

Article Number: Q88157
CREATED: 18-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

There is no inherent command in 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.

This information applies to Microsoft Access 1.0.

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, 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 simply 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: Some lines below have an underscore (_) to denote line breaks. These commands should be placed on one line.

```
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: How to Create a Multiuser Custom Counter

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:

To 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 Hide the Database Window at Startup

Article Number: Q88160
CREATED: 18-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

During normal startup of Microsoft Access, the Access 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 an Access database is first loaded into the environment, 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. 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

INF: Importing the Named Range "Database" from Excel
Article Number: Q88161
CREATED: 19-AUG-1992
MODIFIED: 17-NOV-1992

Summary:

When trying to import the named range "Database" from Microsoft Excel
versions 3.x and 4.x files, the following error appears:

    Invalid Range.

More Information:

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 or R5C10:R170C23).

This error is caused by the difference in the Excel BIFF (binary
information file format) structures between versions. BIFF2 has the
range listed in its internally defined named range table, which
Microsoft Access can read. 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.

References

INF: Introduction to Windows Programming for MS-DOS Programmer

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](#)

INF: Like vs. "=" with Wildcard Characters in Query Searches

Article Number: Q88166
CREATED: 19-AUG-1992
MODIFIED: 4-DEC-1992

Summary:

When searching a string using wildcards (for example, * or ?), you
must use Like or Not Like in the Criteria field rather then = or <>.
When you use the = (or the <>) operators, 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 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: How to Determine if a Specific Windows Program Is Running

Article Number: Q88167
CREATED: 19-AUG-1992
MODIFIED: 15-NOV-1992

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 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 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 Access database container, open a new module and
   enter the Microsoft Access Basic code listed below.

```
'======================================================================
'Declarations section
'======================================================================
Option Explicit
'NOTE: ENTER THE DECLARATION FUNCTION ON ONE LINE.
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 container, 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
Clipboard           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
```

Reference(s):

"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"

WINSDK.HLP file shipped with Microsoft Windows 3.0 SDK

References

INF: Error Msg: Can't Hide the Control that Has Focus

Article Number: Q88168
CREATED: 19-AUG-1992
MODIFIED: 22-DEC-1992

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: Incrementing the Number Portion of String

Article Number: Q88169
CREATED: 19-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

This article describes a formula that sets the default value of a
bound text field in a form to the next higher value of its numeric
portion.

The formula finds the highest used number in the table, strips the
leading text characters, adds one to the result, and then concatenates
the text characters back on.

More Information:

Note: This example assumes that the number of leading text characters
is known at the time the form is designed.

1. Open an existing database, or create a new database from the File
   menu.

2. From the File menu, choose New, then select Table to create a new
   table.

3. Add a field called Book ID of type text. From the Edit menu,
   choose Set Primary Key to define the field as the primary key.

4. Add a second field of any type to the table.

5. From the File menu, choose Save As, and name the table
   "Increment."

6. Switch the table to design view and enter the following values in
   three separate rows:

Book ID	Second Field
BO-110	
BO-111	
BO-112	

7. From the File menu, choose Close to close the table.

8. From the File menu, choose New, and select Table to create a new
   blank form. Base the form on the Increment table.

9. From the View menu, choose Field List to display the form's field
   list.

10. The Properties window should now display the form's properties.
    Change the following form property:

    Default View: Single Form.

11. Drag both the field [Book ID] and [Second Field] from the Field List to the form.

12. From the View menu, choose Properties to display the [Book ID] properties. Click on the [Book ID] text box to display the [Book ID] properties in the Properties window.

13. Set the following property of this new bound text box as follows:

    Default Value: ="BO-" & Right(DMax("[Book ID]","Increment"),
                        Len(DMax("[Book ID]","Increment"))-3)+1

    (Note: The formula above must all be on one line.)

14. From the View menu, choose Form to change the form to Browse view.

The formula does the following:

 - DMax("[Book ID","Increment") get the highest value in the table.

 - Len(DMax(...)) finds the length of the value returned by DMax().

 - Right(DMax(...),Len(...)-3) strips the three text characters.

 - +1 increments the result of the Right() function.

 - "BO-" and concatenates the text "BO-" onto the incremented number.

Note: This works when the form is in Single view; it may not work consistently in Continuous view. When you move to a new record and begin to enter data, Access displays the next empty record. The default values for the next empty record are calculated before the record you are currently editing is committed. Because DMax() calculates the maximum value from those stored in the table, the ID of the record you are editing will be repeated.

If you are working in a multiuser environment, it is possible that more than one user will receive the same calculated Book ID. Although a user can change the ID by hand, you may want to track the highest number portion in a separate table and use either a macro or Access Basic code to maintain it. For more information about multiuser counters, query on the following words in the Microsoft Knowledge Base:

    multiuser and counter

References

INF: Determining the Existence of a Query in Access

Article Number: Q88170
CREATED: 19-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

This article describes how to determine the existence of a query with
Access Basic code.

More Information:

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 = False
    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: Using APIs to Make a Modal Microsoft Access Program

Article Number: Q88171
CREATED: 19-AUG-1992
MODIFIED: 12-NOV-1992

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
by calling the Windows API functions SetSysModalWindow and FindWindow.

This article discusses the API functions used to make Microsoft Access
a system-modal window. A sample function is described that displays
how the Windows API functions are used to make Microsoft Access
system-modal.

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

Also query the Knowledge Base on the following word:

    findwindow

More Information:

Microsoft Windows is designed so that users can switch between
applications without terminating one program to run another program.
However, there may be times when a Microsoft Access application needs
to take control of the entire environment and run from only one
window, restricting the user from switching to any other application.
An example of this is a security system, or a time-critical
application that needs to run uninterrupted.

Passing the handle of the window as an argument to SetSysModalWindow
will limit the user to that particular window. This will prevent the
user from moving to other applications or bringing up the Task Manager
with the mouse or by pressing ALT+ESC or CTRL+ESC. Using
SetSysModalWindow in conjunction with the GetSystemMenu can disable
the Control (system) menu and prevent the user from quitting Windows
by pressing ALT+F4.

All child windows that are created by the system-modal window become
system-modal windows. For example, if Microsoft Access is system-
modal and a form is opened, the form becomes system-modal until it is
closed.

When developing applications in Microsoft Access and using the
SetSysModalWindow API function, keep in mind the restrictions that
will be placed upon the user. 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 will not be available to the
user.

When the SetSysModalWindow API function is used, the system-modal
window (in this case, the Microsoft Access window) must be destroyed
before other applications or windows can be accessed. Therefore,
make sure the application has a means to exit Microsoft Access.

To use the SetSysModalWindow API function within Microsoft Access
Basic, do the following:

1. Within the Declarations Section, enter the following, with each
   Declare statement on a single line:

```
Option Explicit

Declare Function SetSysModalWindow% Lib "User" (
            ByVal hwnd%)

Declare Function FindWindow% Lib "User" (
            ByVal lpszClassName As Any,
            ByVal lpszWindow As Any)
```

2. Create a procedure by entering the following:

```
Sub 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&)

  ' The Handle for Microsoft Access is sent then Microsoft Access
  ' becomes modal.
  Success = SetSysModalWindow(AccessHandle)

End Sub
```

3. Save the module by choosing Save from the File menu. Save the
   module as "Modal Access" and 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, explore the possibilities of the functions
by trying to select other windows. Note that the windows cannot be
saved, maximized, or minimized. Prior to using this procedure in an
Microsoft Access application, make sure all situations have been
considered so that the application is designed to be thoroughly modal.

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

[References](#)

INF: Norton Desktop 1.x Is Not Compatible with Access

Article Number: Q88172
CREATED: 19-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

Running Norton Desktop for Windows versions 1.x with Access results in unpredictable behavior in Access (for example, you may lose the background screen or experience a general protection [GP] fault in Access). 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: ODBC Setup for Access and SQL Server

Article Number: Q88173
CREATED: 19-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

Installation and setup of Microsoft ODBC for Microsoft 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 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 ODBC) 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:

    isql /U <sa login name> /n /P <password> /S <SQL server name> /i
    <drive:\path\INSTCAT.SQL> /o <drive:\path\output file name>

(Note: The above two lines should be entered as one line at the
command prompt. Do not include the angle braces <> in the command.)

    /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.

[References](#)

INF: How to Obtain Access Startup Directory with API Calls

Article Number: Q88174
CREATED: 19-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

Access Basic does not have an inherent command to return the Access startup directory. However, this can be accomplished within an Access Basic program by making Windows API calls.

This article describes the steps involved in creating a function that returns the 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 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 Access startup directory, do the following:

1. From the Access File menu, choose New, and select Module.

2. In the Declarations section of the module, enter the following
   Declare statements (each on a single line):

   ```
   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("ACCESS.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 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.

Reference(s):

"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: Creating, Debugging, and Using an Access Library

Article Number: Q88175
CREATED: 19-AUG-1992
MODIFIED: 17-NOV-1992

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: Programming with the Object Analyzer Engine and Utilities

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 have some experience with developing
applications in Microsoft Access using macros and Access Basic.

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 on.
For more information on what the Database Analyzer is and how to
install and use it, choose Contents from the Help and 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. You can use these commands and functions to
add data-analysis capability in your custom applications or to write
your own database-object analysis tool.

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. Load the Database Analyzer library into memory. For information on
   how to install the Database Analyzer library, choose Contents from
   the Help menu, then choose the Product Support topic.

2. Open the NWIND database in Microsoft Access.

3. In the Database window, switch to Modules. Select the Introduction
   To Programming module and choose the Design button.

4. From the View menu, choose Immediate Window.

5. 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>

   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 table to report 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 SQL representation of the
                       query.

    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 macro 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.

If you would like more information on how to use these commands in a program, search in the Microsoft Knowledge Base for the article titled "Practice Using Object Analyzer Commands" (Q88651).

References

INF: How to Force a Cascading Delete

Summary:

The capabilities of referential integrity in Access do not extend to performing cascading deletes (a way to delete all the 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 NWIND Categories table, and you want to delete the category ID "BEVR." If Categories is related to the Products table that contains many child Category ID records with a "BEVR" value, and if referential integrity is enabled for this relationship, Access would not permit you to delete the record in Categories until you delete all of the child "BEVR" records in Products. 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 that attaches to the OnDelete event of a the Categories form as follows:

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 View menu.

3. Add the following actions and their corresponding properties:

```
  Condition                                             Action
  -------------------------------------------------------------

  MsgBox("Delete record & all child records?",33)<>1    CancelEvent
  ...                                                   StopMacro
                                                        SetWarnings
                                                        RunSQL


  SetWarnings Action
  ------------------

  Warnings On:   No

  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.

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](#)

PRB: Irregular Characters in Attached dBASE IV Memo Field

Article Number: Q88647
CREATED: 30-AUG-1992
MODIFIED: 6-DEC-1992

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 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
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. 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: Easy Way to Add Items to the Help Menu

Article Number: Q88648
CREATED: 30-AUG-1992
MODIFIED: 6-DEC-1992

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 undocumented MSACCESS.INI
section.

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](#)

INF: Opening Databases Read Only Causes Information Message

Article Number: Q88649
CREATED: 30-AUG-1992
MODIFIED: 6-DEC-1992

Summary:

When you open a database under the following conditions:

 - As read-only, whether its file attribute is set to read-only
 - The directory it resides in is read-only
 - You use the "/Ro" command line parameter

you will 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 in read-only mode an .LDB
file will never be created.

More Information:

The only current workaround, if somebody does not want to be prompted
with the above message, is to implement security. Here you would not
physically open the file as read-only but would open all the objects
in the database as read-only. The process for implementing security is
outlined in chapter 25 of the "Microsoft Access User's Guide."

References

INF: How to Create a Password Textbox

Article Number: Q88650
CREATED: 30-AUG-1992
MODIFIED: 6-DEC-1992

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 passowrd 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 it has focus.

3. In the toolbar, drop down the list of fonts located directly to the
   right of the Paint Pallette 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: Practice Using Object Analyzer Commands

Summary:

This article provides an example for using the Object Analyzer commands discussed in the article, "Programming with the Object Analyzer Engine and Utilities" (Q88620). For more information on the commands themselves, search on the following words in the Microsoft Knowledge Base:

```
DumpTableInfo or DumpQueryInfo or DumpFormOrReport or
DumpMacroInfo or DumpModuleInfo
```

This article assumes that you have some experience with developing applications in Microsoft Access using macros and Access Basic.

More Information:

As discussed in the article titled "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 gain some practice using these commands, follow the steps below to create your own Database Analyzer in NWIND:

1. Create a new form called "Little Analyzer".

2. Create a text box control on the form and change the Control Name property to "Name".

3. Place an option group control called "Options" on this 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

5. For the AfterUpdate property of the "Options" option group, specify the following:

   ```
   =AnalyzeIt(Name, Options)
   ```

6. Save and close the report.

7. Create a new module and add the following function:

```
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","@RCtrls",InName,False
       Case 5
          DumpMacroInfo "NWIND.MDB","@Macro",InName
       Case 6
          DumpModuleInfo "NWIND.MDB","@Procs","@Vars",InName
    End Select
    DoCmd Hourglass False
End Function
```

 8. Save and close the module.

 9. Open the "Little Analyzer" form.

10. In the Text Box, type "Categories" (without quotation marks).

11. Choose the Tables button.

At this point, the Object Analyzer will analyze the Categories table
and create a table called "@Table", which will contain the information
on the structure of the Categories table.

You might also want to experiment with analyzing a form or other
object in the same way.

If you have the Database Analyzer library loaded, you can use these
commands to build your own object-analysis tool, or you may want to
incorporate these commands into your application to make it self-
documenting.

[References](#)

INF: Creating a Proper Function for Data Entry

Article Number: Q88652
CREATED: 30-AUG-1992
MODIFIED: 11-NOV-1992

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: How to List the Related Tables in a Database

Article Number: Q88653
CREATED: 30-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

Although 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
   Read Data box.

6. Click Assign and then choose the Close button.

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 = SysIndexes.ObjectId,
       Primary
   INNER JOIN MSysIndexes ON Primary.Id = MSysIndexes.ObjectId,
       Foreign INNER JOIN MSysIndexes ON Foreign.Id =
       SysIndexes.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.

[References](#)

INF: Performing an SQL Bulk Action Query from Access Basic

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 an 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 you can add to your program
called PerformSQLAction. This subprocedure allows you to simply 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 simply expect a SQL command to execute on a line by itself. For
example, the following SQL command, although valid in an 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 using 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:

   Set MyQuery = MyDB.CreateQueryDef("TempQuery" ,UPDATE
                 Cust SET Cust.[Phone] = "(206) " & [Phone];

(Make sure the entire command shown above appears on one line.)

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 Access Uses SQL Server Connections

Article Number: Q88655
CREATED: 30-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

This article describes how 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. Access, however, is not the 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 Access to obtain more connections
than the conventional front-end needs.

Although some servers, such as Microsoft SQL Server, are not too
stingy on giving out connections, there are 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 Access uses connections.

Connections are opened by 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 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 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, 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 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 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 doesn't support it or because the semantics provided by the server differ greatly with those provided by 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 isn't executable on the server is somehow involved with the join between the two tables. Note that in 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 an 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, 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 an Access query is executed, it typically requires only one (for a static view or two (for a dynaset) connection. This doesn't represent an extravagant use of server connections, but good applications could involve the execution of many 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 type 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 Access uses connections, and with prudent application design, you should not run out of server connections.

[References](#)

PRB: Scrollbar Thumbs Don't Move Scrolling Thru a Combo Box

Article Number: Q88656
CREATED: 30-AUG-1992
MODIFIED: 11-NOV-1992

Summary:

SYMPTOMS

When you are using a combo box on a form, the scrollbar thumb
does not always move when you scroll toward the bottom of the list by
using the down-arrow button in the scrollbar. When you reach the
bottom of the combo box, the scrollbar thumb jumps immediately to the
bottom of the scrollbar. Once you have reached the bottom of the
list, the scrollbar thumb works properly and moves as you scroll
upward and downward through the combo box.

CAUSE

The reason why this happens has to do with the fact that the combo box
control does not always know how many records are in its ControlSource
table or query when you drop the list down.

At the point of dropping the list, Microsoft Access begins to access
the data in the table or query. This occurs even as you are scrolling
through the rows in the list. In order for the thumb in the scrollbar
to move, the combo box must know how many items are in the list so
that a calculation can be made as to where the thumb visually belongs.
For example, if you have 100 records in your table and you are viewing
the 50th record in the drop down list, then the thumb should appear
halfway down the scrollbar.

Once you have scrolled to the bottom of the list, the combo box knows
the number of items that were included in the list. With this number,
the thumb in the scrollbar can now adjust itself relative to the row
position of the list.

RESOLUTION

This behavior is by design.

References

INF: Using the Date/Time Data Type and Elapsed Time Example

Article Number: Q88657
CREATED: 30-AUG-1992
MODIFIED: 25-NOV-1992

Summary:

This article describes how Microsoft Access stores the date/time data type, gives an example of calculating elapsed time, and lists some important considerations about the date/time data type.

More Information:

Microsoft Access uses double-precision, floating-point numbers (up to 15 decimal places) to store date/time values. Using this double-precision number 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, and 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 what is stored internally for date/time values:

```
2.0 represents 1/1/1900 12:00:00 AM
0.25 represents 6:00:00 AM (0.25 * 24 hours = 6 hours)
33914.125 represents 11/6/1992 3:00:00 AM (0.125 * 24 hours = 3 hours)
24794.84375 represents 11/18/67 8:15:00 PM (0.84375 * 24 hours = 20 25
   or 20 hours and 15 minutes (.25 * 60 minutes = 15 minutes)
```

Null Value Notes

Microsoft Access recognizes when a portion of a date/time field has not been entered. For example, if you enter only a time value, Microsoft Access will not display the 0 date value of 12/30/1899. Calculations that include a null date/time field will return a null.

Formatting Notes

Date/time fields can display either the date, the time, or both (Microsoft Access always stores both values). How you choose to display or format the date/time field will determine the values displayed. For the available formatting options, use the Help menu to search for the "Format Property Date/Time Values" topic.

Comparing Dates Using Functions

Because Microsoft Access uses double-precision, floating-point numbers
when calculating functions that involve dates and times, unexpected
results may be returned. For example, when tested from the Immediate
window in Modules, the following expression will return 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 AM. This is because the Now() function returns
the double-precision number of the current date and time, which may
not be equal to the integer number of only the date, which is returned
by the DateValue() function. The correct formula for comparing today's
date with another date is the following:

    ?Date()=DateValue("1/31/92")

Another acceptable formula is listed below:

    ?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.

Elapsed Time

Microsoft Access includes many commands that allow manipulation of
date/time values as well as many options for displaying desired
formats. Below are two formulas used for returning the amount of
elapsed time between two given date/time values where each formula
displays a different format.

The following formula uses the CVDate() function to convert the string
expression into a date value; it then takes the absolute value of the
difference between the two date/time values and multiplies this by 24
to account for the rollover into a new day:

    Abs(Time1 - Time2) * 24
This new value is then used to return just the integer portion. The
integer now represents the total number of hours elapsed:

    Int(Abs(Time1 - Time2) * 24)

The Format() function is then used to concatenate the integer portion
of the number of hours elapsed with the absolute value of the
difference between the two time values; however, it displays just the
remaining portion or the minutes and seconds:

    Format(Int(Abs(Time1 - Time2) * 24), "###:") + Format(Abs(Time1 -
        Time2), "nn:ss")

Example 1

```
Function HoursTest()
' This function displays the total hours elapsed in the format of:
' "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 next example follows the same format but displays the difference
between the two date values as "Total days, hours, minutes, seconds".

Example 2

```
Function ElapsedTest ()
' This function will display elapsed time as "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
```

Microsoft Excel Notes

Microsoft Excel also uses double-precision numbers to store date/time
values, but Microsoft Excel's values are limited to the range of
1/1/1900 through 12/31/2078. This date and time system begins with the
double-precision number 1.0, which represents 1/1/1900 00:00:00
(12:00:00 AM). Microsoft Access originally supported the same dates as
Microsoft Excel (1900-2078); however, when the date range was extended
(100-9999), Microsoft Access did not include the Lotus/Excel 1900 =
leap year anomaly. This means that between 1/1/1900 and 2/29/1900, the
internal integer portion of the date values for Microsoft Access and
Microsoft Excel is off by one.

[References](#)

INF: Using TransferDatabase Macro to Attach to SQL Server Data

Article Number: Q88658
CREATED: 30-AUG-1992
MODIFIED: 15-NOV-1992

Summary:

This article describes how to use the TransferDatabase macro action to attach to data in SQL Server tables and what information is necessary to execute this process.

This article assumes the reader is fluent with Microsoft Access Macro code and its operation.

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 basically 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 (DBQ): 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;DBQ=Pubs;WSID=wrkstn1;USID=JohnDoe;PWD=MARCH

This string would be placed in the Database Name argument of the TransferDatabase action.

[References](#)

INF: Creating a Duplicate Function for Repetitive Data Entry

Summary:

In a data entry environment, everything is based on how quickly the information can be entered with a high degree of accuracy. Some information is repetitive for the data entry people to enter. That information will change occasionally, and the data entry person needs to be able to modify the data. The function below duplicates the previous value entered, but once the value is changed, the function will duplicate the new value.

This article assumes you have a working knowledge of Access Basic.

More Information:

1. Create a new module and define a global variable:

```
Option Compare Database   'Database order for string comparisons
Global Dupe_Val As Variant
```

2. Create a function that places the value from the previous record into the field:

```
Function Stick_Val (Ctrl as Variant)
   Ctrl = Dupe_Val
End Function
```

3. The next function checks the value to see if it has been changed from the previous value. If it is different, then it changes Dupe_Val to reflect the change.

```
Function New_Val (Ctrl as Variant)
   Dim Check_Val As Variant
   Let Check_Val = Ctrl
   If (Check_Val <> Dupe_Val) Then
      Let Dupe_Val = Check_Val
   End If
End Function
```

4. Set the following form property:

```
OnCurrent:     =Stick_Val([Field1])
```

5. The function New_Val ( ) needs to be called on the control property of the field you want this to occur on. Use something like the following

```
BeforeUpdate:  =New_Val([Field1])
```

where Field1 is the name of the control.

This process only has one save variable in it and can only work for

one control per active form.

INF: Using the LIKE Operator with Parameter Queries

Article Number: Q88671
CREATED: 30-AUG-1992
MODIFIED: 17-NOV-1992

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 parameterized query. The LIKE operator is used
along with the wild card 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.

Reference(s):

For more information, search for "parameter query", then "Creating a
Parameter Query" using the Help menu.

[References](#)

INF: OnCurrent Is Not Triggered on Unbound Forms

Article Number: Q88687
CREATED: 30-AUG-1992
MODIFIED: 15-NOV-1992

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.

Example

This example creates a simple form with a text box attach 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 the text box flashes several times when the form is opened.

Note: You cannot change the Visible property of a control that has the
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, and select Form.

2. Leave the Select A Table/Query box blank and 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 with the following property:

       Control Name: = Text2

5. Save the form as "Flash" by choosing Save from the File menu.

6. Open a new macro and choose Macro Names from the View menu. Add the
   following macro names and actions:

    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 group 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

INF: Importing Several dBASE Databases at Once

Article Number: Q88764
CREATED: 31-AUG-1992
MODIFIED: 6-DEC-1992

Summary:

This article discusses how to import dBASE databases in a batch process using Microsoft Access Basic.

One of the first things that many new Microsoft Access users do when they begin using the product is import data from the system they are currently using. The Import Dialog in Microsoft Access allows you to import one table at a time. This is usually enough to do the job. However, some users may have numerous tables to import. Furthermore, importing a number of tables might be a common task for some.

This article assumes that you have some experience working with the Microsoft Access Basic programming environment and that you can create and use Microsoft Access tables for data entry.

More Information:

Importing tables through a batch process can be done fairly easily using a Microsoft Access Basic procedure and a table. This procedure is designed for importing dBASE databases only, but it can be easily modified to accommodate other file formats.

A "batch" table will be used as a list of which tables you want to import. The Microsoft Access Basic procedure will read the table and import each foreign table listed in the batch table.

The table must be called "Batch Import" and have the following structure:

```
    Field Name            Type    Length
    ------------------------------

    Source Directory    Text    50
    Source Database     Text    50
    Imported Name       Text    50
    Type of Table       Text    50
```

Once you have created the table, fill it with information on what tables you want to import. The list below shows what information you should add to each field in the Batch Import table:

Source Directory – This is the full path reference of 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,

"Customer Table".

Type of Table -   This can be either dBASE III or dBASE IV. Specify
                    "dBASE III" for both dBASE III and dBASE III PLUS
                    databases.

As an example of the data you would add to this table, suppose you
want 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   Type of Table
----

C:\DBASE4   EMPLOYEE.DBF   Employee Table   dBASE IV
D:\DBASE3\DATA   ORDERS.DBF   Orders Table   dBASE III

In addition to creating and filling the table above as instructed, you
must create a new module and add the procedure listed below. Note that
the line that starts with "DoCmd" is broken into three lines to fit in
this article. Make sure that the entire instruction is only on one
line when you type or copy this procedure into your module.

```
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
```

Once you have added and saved this procedure, you can invoke the
process of importing the databases named in the Batch Import table by
following these instructions:

1. Open a module window if none are already open.

2. Bring up the Immediate window by choosing View, Immediate Window.

3. In the Immediate window, enter the following command:

        BatchImport

After you enter the command, the mouse pointer will become an
hourglass and remain that way until all the databases are imported.
This process may take several minutes, depending on the size of the
databases you are importing.

References

INF: How to Modify the Toolbar in Microsoft Access

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.

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 Microsoft
Access does not allow you 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: Running MS-DOS SHARE with Windows for Workgroups

Article Number: Q88914
CREATED: 03-SEP-1992
MODIFIED: 05-JAN-1993

Summary:

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 the Microsoft Windows for Workgroups operating
system, it is recommended that this SHARE command be removed.

More Information:

Windows for Workgroups has its own sharing mechanism (VSHARE.386),
which is loaded with Windows (there is an entry for it in the
SYSTEM.INI file). VSHARE.386 is intended as a drop-in replacement for
SHARE.EXE.

SHARE.EXE is dependent on the number of locks specified with the /L:
parameter (with a default of 20). VSHARE.386, on the other hand,
dynamically allocates the number of locks.

Because VSHARE.386 is an enhanced mode virtual device driver (VxD), it has
features which improve file sharing when running in enhanced mode. Because
it was designed to work with Windows for Workgroups, VSHARE.386 has
features that make it "friendlier" than SHARE.EXE, especially in access
control. If SHARE.EXE is loaded and you start Windows for Workgroups in
enhanced mode, VSHARE.386 takes over as the file sharer until you exit
Windows. If you usually run Windows for Workgroups in enhanced mode, you
might not need to run SHARE.EXE at all, saving you 5K or more of
conventional memory. The number of available locks is very important if you
are using your Windows for Workgroups machine as a network server.

References

INF: Using the GetSystemMetrics API Call in Access Basic

Article Number: Q88922
CREATED: 3-SEP-1992
MODIFIED: 15-NOV-1992

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 API function GetSystemMetrics. Incorporating this function
into an 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 example 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
                                  '  horizontal scroll bar
   Const SM_CYHSCROLL = 3         ' 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 example call will return the height of the caption bar
of a form:

```
    HeightY% = GetSystemMetrics(SM_CYCAPTION)
```

[References](#)

INF: The Immediate Window for Something Other Than Access Basi

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 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 it's 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: Cannot Group on Memo or OLE Object in Query

Article Number: Q88926
CREATED: 3-SEP-1992
MODIFIED: 15-NOV-1992

Summary:

If you attempt to create a Select, Update, or Append query that groups
on a Memo or OLE object, you will receive the following error message:

    Can't group on Memo or OLE object (<clause>).

Access cannot group by a Memo 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. For example, you can enter the following expression to
Group By the Left 25 bytes of a Memo field called Description:

    Expr:Left([Description],25)

Choose the Totals button on the tool bar, and in the Total row select
Group By as the option for the expression. Expr is the name of the
column header in Datasheet view.

Reference(s):

Online Help topic titled "Calculating Totals in a Query"

References

INF: Sorting/Grouping a Query Using a Portion of a Field Value

Summary:

You can sort or group a query using a portion of a field's value. For example, in the NorthWind sample database, you can create a query based on the Employees table that sorts by six characters starting with the fourth character position in the Last Name field using this expression:

```
Sort String:Mid([Last Name],4,6)
```

Click the Sort cell in the QBE grid for the Last Name field. Click the down arrow and select an option. (In order to group by the expression, choose the Totals button on the tool bar and select Group By in the Totals row as the option for the expression.)

Sort String is the name of the column header in Datasheet view. You can name the column any way you like.

Use the Left and Right functions to return the leftmost and rightmost characters of a string argument.

More Information:

This technique is useful in a situation where a field contains values that have a set number of leading or trailing characters that are insignificant to your sort. An example of this might be a part number field called PartNo that contains values such as the following:

```
PartNo
------

A-453-34567
A-123-45675
B-234-75658
B-645-65758
```

and so on.

In order to sort using only the third through the fifth characters, use an expression like this:

```
Expr:MID([PartNo]),3,3)
```

See online Help in Access for more information on String functions and related functions such as Left, Right, and Len.

[References](#)

INF: Generating a Random Temp File Name

Article Number: Q88929
CREATED: 3-SEP-1992
MODIFIED: 11-NOV-1992

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: How to Dim (Gray) Menu Items with Access Basic

Article Number: Q88940
CREATED: 3-SEP-1992
MODIFIED: 15-NOV-1992

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 API
functions.

For more information about the structure of the Access menu system,
query on the following words in the Microsoft Knowledge Base:

    MENU and SYSTEM and FINDWINDOW and GETMENU and GETSUBMENU

More Information:

The following 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)

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 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.

```
'********************************************************************
'Declarations section of the module.
'********************************************************************
Option Explicit

'Note: Each Declaration must be placed on a single line.
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%)

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 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%)

            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

INF: Deciding When to Use Macros and When to use Access Basic

Article Number: Q88999
CREATED: 6-SEP-1992
MODIFIED: 12-NOV-1992

Summary:

Because 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:

Access has a very effective user interface that is powerful enough to accommodate most needs. However, like many other database products, Access provides the user with the ability to develop his or her own way to manipulate the power of the product.

One distinct advantage of 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.

Wizards are an example of a program you can write that is beyond the capability of a macro. Wizards 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 modules instead of macros is easier when you understand how Access works. If you understand and are familiar with Access, you will find that it is easier to determine when modules are required to accomplish a specific task.

New users of Access will likely be surprised at the ease, power, and versatility that Access macros can provide.

References

PRB: Arrange Icons Doesn't Work on More Than 40 Windows.

Article Number: Q89546
CREATED: 24-SEP-1992
MODIFIED: 6-DEC-1992

Summary:

SYMPTOMS

When Microsoft Access has more than 40 Windows open, selecting
Window, Tile or Cascade does not work.

CAUSE:

This is functionality totally 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. Close the number of
windows open.

More Information:

Steps to reproduce behavior:

1. Open and minimize 40 document windows
2. Move the icons around a bit
3. Windows Arrange Icons
4. Result: No change

References

PRB: Floating Objects in Access Cause Unusual Effects

Article Number: Q89548
CREATED: 24-SEP-1992
MODIFIED: 6-DEC-1992

Summary:

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 Access finds the information which is obscured by the immediate window, the dialog box which asks you to verify that you want to change that information is displayed but you can't see the code.

Reference(s):

"Microsoft Access User's Guide" version 1.0, page 195.

"Microsoft Access Basic: Introduction to Programming" version 1.0, Chapter 1 pages 2-3, Chapter 4, pages 9-10.

References

PRB: Table Design Doesn't Check Validity of Default Value

Article Number: Q89549
CREATED: 24-SEP-1992
MODIFIED: 6-DEC-1992

Summary:

SYMPTOMS

The default value is not checked according to the validation of a
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: Reserved Words won't Format (Capitalize) in Module Window

Article Number: Q89576
CREATED: 27-SEP-1992
MODIFIED: 6-DEC-1992

Summary:

```
SYMPTOM
   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](#)

INF: Commenting a Macro Line to Temporarily Disable the Action

Article Number: Q89577
CREATED: 27-SEP-1992
MODIFIED: 6-DEC-1992

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 front of a line of code so that it is not executed. This is useful when you want to determine which line of code is causing an error by disabling one line at a time until the error is no longer reproduced.

This method is also useful for debugging macros. This article describes how you can temporarily disable a macro line so that it is ignored.

More Information:

You can disable a macro action line without deleting it by doing the following:

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, enter "False" if there is nothing in the column for that line, or "False And" if there is already a Condition 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

INF: Topics Supported by Access as a DDE Server

Article Number: Q89586
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

Summary:

Access supports dynamic data exchange (DDE) both as a client and as a server. This article lists the five topics that 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 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 on 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: 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>.

Examples: NWIND;SQL Select * from Employees;
          c:\ACCESS\NWIND.MDB;SQL Select [First Name] from Employees;

Note: All SQL statements must end with a semicolon (;).

The Items supported on the Topic SQL are the same as those for

Name> or <Query Name>.

[References](#)

INF: DoCmd and Transaction Processing

Article Number: Q89587
CREATED: 27-SEP-1992
MODIFIED: 6-DEC-1992

Summary:

In Microsoft Access, the DoCmd command will not be affected by a
Rollback command if the DoCmd is within a transaction.

More Information:

Transaction processing in Microsoft Access is only valid on virtual
table (VT) objects, such as dynasets. The DoCmd action will start
another session of Jet to process the arguments of the DoCmd. Since
Jet manages transaction processing, a new session of Jet will have its
own transaction management. The two are mutually exclusive, meaning
transactions in one are not managed or affected by the other.
Therefore, any action done by DoCmd within a transaction will not be
affected by a Rollback command.

The simplest workaround for this is to use a QueryDef.Execute action,
which uses a dynaset. NOTE: The QueryDef.Execute property works only
when a QueryDef is based on an action query (update, insert, delete)
that does not return a record set.

References

INF: Making ENTER Add Lines in a Text Box

Article Number: Q89590
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

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: Error() Function Does Not Return Detailed Information

Article Number: Q89591
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

Summary:

When errors are trapped in Microsoft 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
Microsoft 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: DoEvents, SendKeys May Not Be Supported in UDFs

Article Number: Q89592
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

Summary:

Under certain circumstances, DoEvents and SendKeys are not supported
when they are called from user-defined functions (UDFs) with Microsoft
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

PRB: Autokeys Macros with the Same Name Are Still Executed

Article Number: Q89593
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

Summary:

You will receive an error if you save two Autokeys macros with the
same name; however, the duplicate macros will still be executed.

More Information:

```
Steps to Reproduce Problem
--------------------------
```

1. Create a new macro and add two macro names with the same name; for
   example:

```
    Macro Name          Action
    ------------------------

    ^z                  MsgBox
    ^z                  MsgBox
```

2. Save the macro as "Autokeys".

   The following error message appears:

      The key combination "Z" in 'Autokeys' is also assigned to
      another macro; only the first one will be used.

3. Choose the OK button, then press CTRL+Z (^Z).

   Both macros are executed, and two message boxes appear. However, if
   you put another macro between the two MsgBox macros, as follows

```
    Macro Name    Action
    ------------------

    ^z            MsgBox
    ^a            Beep
    ^z            MsgBox
```

   only the first MsgBox macro will be executed.

References

INF: How to Display Immediate Window Without Module Window

Article Number: Q89594
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

Summary:

Access version 1.0 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 browse mode of a form. By
calling two Windows 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 will 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 SendKeys 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:
   (You must include each declaration on a separate line, and each
   declaration must be entered on one line.)

   ```
   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)

       ' Note that the MsgBox function must be on one line.
       If IhWnd% = 0 Then MsgBox _
                   ("You need to open the IW once for this to work.")

       ApiResults% = ShowWindow(IhWnd%, SW_SHOW)
   End Function
   ```

4. Attach the code to a command button of a form.

5. When you first start Access, the Immediate window is not displayed. When you open a module and display the Immediate window then close it, Access sets the window's Visible property to False. Calling ShowWindow will reset the Visible property to True. If you call this function without first displaying the Immediate window at least once, you will receive the error message, because Access has not created the Immediate window yet, and thus 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 browse mode will display the Immediate window. Any Debug.Print statements will now be visible.

Note: This is unsupported code, and there may be instances when this example will not work.

References

INF: Using Scroll Bars Doesn't Change Record Pointer

Article Number: Q89595
CREATED: 27-SEP-1992
MODIFIED: 13-NOV-1992

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: Using Microsoft Access Macro to Quit Windows

Article Number: Q89596
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

Summary:

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(). To do this, you must first
create a module that declares the ExitWindows() function. You can then
call the Microsoft Access function in a macro using the RunCode
action.

More Information:

The call to the Windows dynamic-link library (DLL) acts the same way
that using the Exit command on the File menu in Program Manager does.
Each application must agree to be closed (for example, if you choose
Cancel when an application asks you if you want to save a file, the
exit request is also canceled).

To create a Microsoft Access function for quitting Windows:

1. From the File menu, choose New, and select Module. In the
   Declaration section, type the following statements. Note that the
   Declarations statement is on one line.

   ```
   Option Explicit
   Declare Function ExitWindows% Lib "user" (ByVal dwReserved&,
                                        ByVal wReturnCode%)
   ```

2. Create a Microsoft Access function that calls the ExitWindows()
   function. Enter it in the following format:

   ```
   Function ExitNow ()
      RetVal% = ExitWindows(0, 0)
   End Function
   ```

   Save the module.

3. From the File menu, choose New, and select Macro. Add the following
   action and its properties:

   ```
   Action        Function Name
   --------------------------

   RunCode       ExitNow()
   ```

   Save the macro.

You can now add the macro to a form and use it like any other
Microsoft Access macro.

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: Maximizing, Minimizing, and Restoring Access in Modules

Article Number: Q89597
CREATED: 27-SEP-1992
MODIFIED: 12-NOV-1992

Summary:

Use 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. Type
   each declaration statement on one line.

   ```
   Option Explicit
   Declare Function GetActiveWindow% Lib "User" ()
   Declare Function ShowWindow% Lib "User" (ByVal hWnd%,
                                           ByVal nCmdShow%)
   ```

2. Create the function MaximizeAccess().

   ```
   Function MaximizeAccess ()
      ActiveWnd% = GetActiveWindow()
      MaxIt% = ShowWindow(ActiveWnd%, 3)
   End Function
   ```

3. Create the function MinimizeAccess().

   ```
   Function MinimizeAccess ()
      ActiveWnd% = GetActiveWindow()
      MinIt% = ShowWindow(ActiveWnd%, 2)
   End Function
   ```

4. Create the function RestoreAccess().

   ```
   Function RestoreAccess ()
      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()
   ```

Reference(s):

"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: ENTER in Text Box Does Not Move to Next Line

Article Number: Q89598
CREATED: 27-SEP-1992
MODIFIED: 19-OCT-1992

Summary:

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.

To move the cursor to the next line of a multiline text box, press CTRL+ENTER.

This behavior is by design.

[References](#)

INF: END Key Behavior in Edit Boxes

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 "edit" mode has been invoked, pressing the TAB key moves the cursor to the next control and exits "edit" mode.

References
combobox editbox

INF: AutoExec Macro Invoked Only When Database Is Opened

Article Number: Q89602
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

Summary:

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 Microsoft Access Basic or another macro.

References

PRB: Close Action from AB Does Not Trigger OnClose Errors

Summary:

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

PRB: New Label Controls Inherit FormWizards' Label Text Align

Article Number: Q89604
CREATED: 27-SEP-1992
MODIFIED: 12-NOV-1992

Summary:

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](#)

INF: OLE: New Graph Object Frames are Blank

Article Number: Q89607
CREATED: 27-SEP-1992
MODIFIED: 18-NOV-1992

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 will 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

INF: Tips on How to Debug Microsoft Access Macros

Article Number: Q89610
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

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

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 523-524

[References](#)

INF: Delete Command on Form Open if Counter has Focus

Article Number: Q89617
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

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 is posted. One work around 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: Import/Export Accepts Only 24-Hour Time Format

Article Number: Q89618
CREATED: 27-SEP-1992
MODIFIED: 15-NOV-1992

Summary:

When importing or exporting time values, Microsoft Access works only
with 24-hour time values. This behavior is by design.

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 a 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

PRB: Avery Labels 4143, 4160, and 5096 Don't Work on Laser Ptr

Article Number: Q89621
CREATED: 27-SEP-1992
MODIFIED: 12-NOV-1992

Summary:

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.
```

RESOLUTION

```
This behavior is by design.
```

[References](#)

INF: How to Use a Toggle Button to Hide and Display a Subform

Article Number: Q89622
CREATED: 27-SEP-1992
MODIFIED: 11-JAN-1993

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. Display the Toolbox.

11. From the Toolbox, choose the Toggle button.

12. Place the mouse somewhere on the Quarterly Orders in an open area to position the Toggle button.

13. Display the Object Properties box for the Toggle button.

14. 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
    Default Editing:   Allow Edits
```

When the Quarterly Orders form is shown in browse mode the Toggle
Subform toggle button hides and displays the Quarterly Orders
subform.

Reference(s):

"Microsoft Access User's Guide" Chapter 22, "Using Macros with Forms" and
Chapter 9, "Designing Forms;" Appendix C

[References](#)

INF: Cut/Paste for Immediate Window and SQL Window

Article Number: Q89680
CREATED: 28-SEP-1992
MODIFIED: 12-NOV-1992

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: Obtain Name of Current User Database with Access Basic

Summary:

Access does not provide any inherent way for you to determine the name
of the current open user database. This article discusses a function
that will allow you to do so.

This article assumes that you have some experience with writing
procedures in Access Basic.

More Information:

Access does not provide any inherent way for you to determine the name
of the current open user database. Although you can obtain the Object
Value of the current user database using the CurrentDB() function, you
are not able to obtain the text representation of the database name
using such a function. This article discusses a user-defined function
called GetUserDBName(), which obtains a text representation of the
user database name by using a DDE request.

GetUserDBName() returns the name of the database with no path or
extension. Because of this, you must add your own path or extension
string to the result of GetUserDBName() when necessary.

Example
----

1. Copy the GetUserDBName() function below into a new module in the
   NWIND database.

```
Function GetUserDBName () As String
   DDECHAN% = DDEInitiate("MSACCESS", "System")
   DDESTRN$ = DDERequest(DDECHAN%, "Topics")
   DDETerminate DDECHAN%

   DBNAME$ = ""
   CNTR = Len(DDESTRN$) - 1
   Do Until Mid$(DDESTRN$, CNTR, 1) = Chr(9) Or CNTR = 0
      If Asc(Mid$(DDESTRN$, CNTR, 1)) > 47 Then DBNAME$ = _
                                   Mid$(DDESTRN$, CNTR, 1) & DBNAME$
      CNTR = CNTR - 1
   Loop
   GetUserDBName = Trim(DBNAME$)
End Function
```

2. Open the Immediate window.

3. Enter the following commands in the Immediate window:

```
    CurrentPath$ = "C:\ACCESS\"
    DBExtension$ = ".MDB"
    CurrentDBPath$ = CurrentPath$ & GetUserDBName() & DBExtension$
```

```
      ? CurrentDBPath$
```

Note: When you enter the last command, the text string

```
   C:\ACCESS\NWIND.MDB
```

is displayed.

References

INF: Selecting Text in a Text Box from the Left or Middle

Article Number: Q89683
CREATED: 28-SEP-1992
MODIFIED: 4-DEC-1992

Summary:

This article discusses the editing characteristic of Microsoft Access that disallows selecting text without selecting all text on the right.

This is usually perceived as an editing bug.

More information:

You are unable to highlight a specific area in a text box, or any area in Microsoft Access where you can select several columns or characters by highlighting one, then dragging either way to select others.

Instead, anytime you place the mouse in a text box or other such area, a highlight selection appears from the point of the mouse all the way to the right of the text box or other such area - preventing you from highlighting anything specifically in the middle or left side.

Editing in Microsoft Access is currently set to EXT mode as indicated on the status bar as a result of pressing F8 at some point.

Press Escape to turn EXT mode off. This is standard with Microsoft Excel and Microsoft Word.

For more information, search Help for 'Extend'.

References

PRB: 'Compile All' is Enabled After Saving a Module

Article Number: Q89684
CREATED: 28-SEP-1992
MODIFIED: 4-DEC-1992

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 Access startup performance. For more information on compiling a module in Microsoft Access, search the Knowledge Base for

    COMPILE ALL and PARSER.

References

INF: Flexible Formatting of Phone Numbers

Article Number: Q89685
CREATED: 28-SEP-1992
MODIFIED: 15-NOV-1992

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 that 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$)
   '* Exit the function if there is no information passed
   If IsNull(PhoneNumber$) 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 the two above functions, bind the control (Text Box) to the function by setting the following property:

```
    AfterUpdate:    =Format_Phone([<control name>])
```

Example for a field with the control name [Text1]:

```
    AfterUpdate:    =Format_Phone([Text1])
```

References

PRB: "Field0" Appears in a Query But Not in the Source Table

SYMPTOM
    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: Allowing an Ampersand to Appear in a Caption

Article Number: Q90090
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

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

Reference(s):

"Microsoft Access User's Guide," version 1.0, page 598.

[References](#)

PRB: Arrange Icons do Not Locate Icons in Correct Place

Article Number: Q90091
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access Getting Started," version 1.0, pages 154, 158.

References

INF: Close Action has Blank Line in Object Type Argument

Article Number: Q90092
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide." Version 1.0, Chapter 21 "Macro Basics"

References

PRB: ESC While Creating Control Doesn't Deselect Control

Article Number: Q90094
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 214-230

References

PRB: Margins Do Not Relate to Printable Area

Article Number: Q90095
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

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
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: Update Queries Do Not Check Validation Rules

Article Number: Q90096
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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.

Reference(s):

"User's Guide," version 1.0, pages 174-178

References

INF: AccessWizards Button Grayed Out

Article Number: Q90097
CREATED: 8-OCT-1992
MODIFIED: 23-DEC-1992

Summary:

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
```

[References](#)

INF: Creating a Snapshot Is Slower Than Creating a Dynaset

Article Number: Q90098
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

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:

On a machine in Microsoft Product Support, the query "Order Review" in
the NWIND.MDB took 13 seconds to create a dynaset, whereas it took 160
seconds to create a snapshot. Both virtual tables (VTs) are populated.

[References](#)

INF: Hiding Rows and Columns in Tables

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.

Reference(s):

"Microsoft User's Guide," version 1.0, chapter 14, pages 355-357.

References

PRB: SQL Tables Are Read Only Without a Unique Index

Summary:

SYMPTOMS

When you try to edit an attached SQL table that does not have a
unique index defined for it, Microsoft Access will beep and the
status bar will display "Form is read only." This will happen
whether you are trying to edit the table in datasheet view or a
form.

CAUSE

Microsoft Access requires that a unique index be defined for each
table in order to reference the records in the table for editing
purposes.

RESOLUTION

Define a unique index for each table by using the SQL command
CREATE UNIQUE INDEX. The basic syntax of that command is:

```
CREATE UNIQUE INDEX <index_name>
  ON [[<database.>]<owner.>]<table_name.> (<column_name>
    [, <column_name>]...)
```

This procedure cannot be done from inside Microsoft Access; it must
be done in the System Administration Facility (SAF) on the SQL
server machine or by using a program such as Q+E (which comes with
Microsoft Excel) that can execute SQL commands directly on the SQL
server machine.

STATUS

This behavior is by design.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 66-68.

References

PRB: QBE Grid Disappears When Selected and Deleted

Article Number: Q90101
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

SYMPTOMS

The Query By Example (QBE) grid, which appears when you enter
design mode for a query in Microsoft Access, disappears if you
highlight the grid (by clicking the query grid's column header),
drag to the end of the query grid, and choose Delete from the Edit
menu.

CAUSE

When you highlight an entire column or grid and choose Delete from
the Edit menu, Microsoft Access deletes the grid columns instead of
clearing the contents of the columns.

RESOLUTION

To display a blank query grid column after the QBE grid has
disappeared, choose Insert Column from the Edit menu.

To display a column that is already filled in with field
information, click a field name in one of the field lists that is
displayed in the top portion of the query design window, and drag
the field to the query grid portion of the query design window.

STATUS

This behavior is by design.

Reference(s):

"Microsoft Access User's Guide" version 1.0, pages 92-100.

References

INF: How to Get SQL Pass-Through Functionality Using Q+E

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-though SQL. This allows you to use pass though 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 SQLPassThru (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 PassThru"
         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 SQLPassThru function attempts to initiate a DDE channel with Q+E. If Q+E is not available, SQLPassThru attempts to start Q+E. If Q+E cannot be started, SQLPassThru 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

For additional information on using Q+E to access Oracle and Rdb, see
the "Q+E for Microsoft Excel User's Guide."

Reference(s):

"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
"Q+E for Microsoft Excel User's Guide," version 4.0, pages 101-102

References

INF: How to Abort Creating a Field in Table Design

Article Number: Q90103
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

To recreate this problem do the following :

1. Open a new table.

2. For the data type of the first row, select any data type.

3. With the delete key or by using the backspace, remove the data
   type that was selected.

4. Try to move off the row.

When trying to move off the row, an alert will result as follows:

    The text you enter must match an entry in the list.

More information:

It does not matter if you have text in the field name column. The
only way to move off the row is either:

1. Fill in the row by selecting a data type.

2. Canceling the entry by pressing the ESC key.

3. Select "delete row" from the edit menu.

This is by design due to validation on exit of the field.

Reference(s):

"Microsoft Access User's Guide." Version 1.0, page 27

[References](#)

INF: OpenTable Action Ignores Data Mode If Table Open

Article Number: Q90104
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 22, "Using Macros with Forms"

[References](#)

INF: Couldn't Find SYSTEM.MDA or UTILITY.MDA File

Article Number: Q90105
CREATED: 8-OCT-1992
MODIFIED: 6-DEC-1992

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.

For more information, search for "MSACCESS.INI" using the Help menu.

[References](#)

PRB: Criteria Sections Combined into Same Column in QBE Grid

Article Number: Q90106
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 5, "Query
Basics."

[References](#)

PRB: Minimizing, Maximizing the Desktop Flashes Slightly

Article Number: Q90107
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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

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 with an example and 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."

[References](#)

PRB: Pressing SHIFT While Dragging Not Working on New Control

Article Number: Q90109
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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](#)

PRB: Database Window Loses its Focus

Article Number: Q90110
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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: "Couldn't Find Installable ISAM" Error

Article Number: Q90111
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide" version 1.0, Chapter 4, "Importing,
Exporting, and Attaching," pages 60-69.

References

PRB: Make-Table Query Causes Existing Table to Be Deleted

Article Number: Q90112
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide" version 1.0, Chapter 7, "Designing
Action Queries and Parameter Queries," page 164

References

PRB: Nulls Not Concatenated with "+" Operator

Article Number: Q90114
CREATED: 8-OCT-1992
MODIFIED: 6-DEC-1992

Summary:

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

Reference(s):

"Microsoft Access User's Guide," version 1.0, Appendix C
```

References

PRB: OnDblClick Macro Requires CancelEvent to Change Focus

SYMPTOMS

The macro action GotoControl does not successfully change the focus
to another control if the macro is attached to the OnDblClick
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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 21, page 541.

References

PRB: LimitToList Combo Box Won't Let You Leave After Delete

Article Number: Q90116
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

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.

Reference(s):

"Microsoft Access Getting Started," pages 83-87
"Microsoft Access User's Guide," pages 233-245

References

PRB: List Box Not Dimmed When Disabled

Article Number: Q90117
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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: Report Prints with Blank Page Between Every Printed Page

Article Number: Q90118
CREATED: 8-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

SYMPTOMS

When a report generated in Microsoft Access is printed, a blank
page appears between every printed page of the report.

CAUSE

This behavior occurs when the total of the form width plus the left
and right margin widths exceed the page width (usually 8.5
inches).

RESOLUTION

Reduce the width of the form, reduce the left or right margins, or
reduce all three so that the total width of the form and margins
does not exceed the page width. In other words:

    (FormWidth) + (LeftMargin) + (RightMargin) <= (PageWidth)

STATUS

This behavior is by design.

[References](#)

PRB: Cannot Undo Property Settings in Table Design Mode

Article Number: Q90119
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

The field name will be removed, but the data type will remain.

Reference(s):

"Microsoft Access Getting Started," version 1.0, chapter 3, "Creating
a Table," pages 36 - 45
```

[References](#)

PRB: CTRL+BREAK Makes Mouse Disappear With a Module Active

Article Number: Q90120
CREATED: 8-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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: Microsoft Access GP Fault with ATI Ultra Video Card

Article Number: Q90121
CREATED: 8-OCT-1992
MODIFIED: 4-DEC-1992

Summary:

If you are using an ATI Graphics Ultra video card in 8514 video mode,
you will get a general protection GP-fault in Microsoft Access when
you create a new form and choose Align from the Layout menu.

To work around this problem, select a control on the form before
choosing Align from the Layout menu.

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](#)

INF: How Access Deletes Objects from a Database

Article Number: Q90129
CREATED: 11-OCT-1992
MODIFIED: 6-DEC-1992

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 isn't 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. (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 shouldn't 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.

PRACC9210: Transaction Limit Can Cause "Out of Memory" Error

Article Number: Q90130
CREATED: 11-OCT-1992
MODIFIED: 21-DEC-1992

Summary:

PROBLEM ID: PRACC9210001

SYMPTOMS

When you attempt to make a change to a database, an "Out of memory"
error message occurs, even if there are ample system resources.

CAUSE

Microsoft Access has a 4-MB-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
version 1.0. We are researching this problem and will post new
information here as it becomes available.

References

PRB: Grid Disappears When Measurement Changed to Metric

Article Number: Q90131
CREATED: 11-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Microsoft has confirmed that this behavior is by design.

References

PRB: Combo Boxes Open When Window Sized to Vertical Minimum

Article Number: Q90132
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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

INF: Access Databases Truncated w/ LAN Manager 2.1a & Earlier

Article Number: Q90133
CREATED: 11-OCT-1992
MODIFIED: 23-NOV-1992

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=X0XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

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
Corruption 2.10a 2.10

PRACC9211: Cannot Cascade or Tile with 50 Windows in Access

Article Number: Q90134
CREATED: 11-OCT-1992
MODIFIED: 17-NOV-1992

Summary:

PROBLEM ID: PRACC9211001

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
version 1.0. 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

PRB: Form Displays Data from Deleted Field in Query

Article Number: Q90135
CREATED: 11-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

SYMPTOMS

```
If you have a form based on a query and you remove a field from the
QBE grid, you may still see the data from the deleted field in Form
browse mode.

You will not see the data from the deleted field if you save and
reopen the form. The field will have the "#Name?" error in the text
box.
```

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.
There is another way to have the form recompile the query. While in
Design view for the form, place your cursor in the text box for the
newly deleted field. Press the ENTER key. From the View menu,
choose Form and note there is an error in the TITLE text box.
```

STATUS

```
This behavior is by design.
```

More Information:

```
Steps to Reproduce Behavior
---------------------------

1. Load the NWIND database.

2. Create a new query using the EMPLOYEES table.

3. Drag the LASTNAME and TITLE fields to the QBE 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. View the form in Browse mode by choosing Form from the View menu.

8. Go back to the query and delete the TITLE column from the grid.
   Save the query.

9. Go back to the Form window. View the form in Browse mode again.
```

Reference(s):

"Microsoft Access User's Guide," version 1.0, page 100

[References](#)

PRB: Limited to 65535 Pages in Print Dialog Box

Article Number: Q90136
CREATED: 11-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide." Version 1.0, pages 31-32, 209, 410.

References

PRB: Report Can't Be Opened or Previewed

Article Number: Q90137
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 522, 566.
```

References

PRB: Pressing CTRL+ALT+SPACEBAR Enters Control's Default Value

Article Number: Q90138
CREATED: 11-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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, the underlying table's default value is entered in the
field.

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 229-230

[References](#)

PRB: #Error Appears Using TRIM() in Forms or Reports

Article Number: Q90139
CREATED: 11-OCT-1992
MODIFIED: 6-DEC-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, page 485-486

[References](#)

INF: Err Msg: Operation Not Supported on Paradox Table...

Article Number: Q90140
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

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 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.

For more information, search on "Paradox" using the Help menu.

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 62-63.

[References](#)

PRB: DoCmd OpenReport Opens Rpt in 3 Views or Prints to File

Article Number: Q90141
CREATED: 11-OCT-1992
MODIFIED: 23-DEC-1992

Summary:

SYMPTOMS

You can enter a Microsoft Access Basic command in the Immediate
window to open a report in a view that is not one of the available
views you see when you create a macro using the OpenReport action.
The Output As dialog box appears when the following command is
issued in the Immediate window:

        DoCmd OpenReport "My Report", 3

The following error message will be displayed if you try to save
the report using the displayed 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:

        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 appears and waits for user input to specify
the name of the text file that output should be directed to.

Reference(s):

"Microsoft Access Language Reference," version 1.0, pages 355-356
"Microsoft Access User's Guide," version 1.0, page 562

References

PRB: Toggle Button with CancelEvent on BeforeUpdate Property

Article Number: Q90142
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, page 529

References

PRB: Memory Block Not Freed If Cue Cards Not Installed Right

Article Number: Q90143
CREATED: 11-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

SYMPTOMS

```
If you do not install Cue Cards during a custom installation of
Microsoft Access, and later you copy the CUECARDS.DLL file to the
directory where you installed MSACCESS.EXE, you will receive an error
stating that Cue Cards has not been installed correctly when you
push one of the Cue card buttons. When you click the OK button to
cancel out of Cue Cards the following error message is displayed:

    1 Global memory block is not freed.
```

RESOLUTION

```
When you end the current session of Microsoft Access this block of
memory will be released. To correctly install Cue Cards, re-run the
Setup program and choose Custom Installation. Select only the
Cue Card section to install.
```

STATUS

```
This behavior is by design.
```

[References](#)

INF: Form OnOpen Event May Not Set Values of Unbound Controls

Article Number: Q90146
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

If you have a form with unbound controls that should have values set
by a macro called from the form's OnOpen event, the macro should
include a GoToControl or GoToRecord action before the SetValue
action(s). The reason you cannot immediately refer to controls' values
in an OnOpen macro is that the OnOpen event occurs before the form is
opened.

If the first action defined in the OnOpen macro is SetValue, Microsoft
Access cannot refer to any controls on the object because it has not
yet been opened.

More Information:

The OnOpen event for forms and reports occur before the object is
opened. When a macro is designed for use as an OnOpen event, actions
such as GoToControl or GoToRecord, by default, refer to the object the
macro was called from. To force the focus to a specific control, insert
the GoToControl action before the SetValue action in the macro. Then
you can set the values of any control(s) on the form.

If the control's values should come from a record, insert the
GoToRecord action.

For more information, search Help for "OnOpen" and select the topic
"OnOpen, OnClose Properties."

Reference(s):

"Microsoft User's Guide," version 1.0, chapter 22, "Using Macros with
Forms," pages 528, 538-539

References

PRACC9110: Cannot Unhide Windows After Out of Memory Message

Article Number: Q90147
CREATED: 11-OCT-1992
MODIFIED: 17-NOV-1992

Summary:

PROBLEM ID: PRACC9210002

SYMPTOMS

If you receive an "Out Of Memory" error message after you have hidden
many Access windows you may be unable to unhide, or Show, any of the
hidden windows. You will still be able to continue to hide additional
windows.

CAUSE

In order for Access to show hidden windows, the Windows
"Unhide Windows" dialog box must be brought up. This requires memory.
If Windows has previously detected insufficient memory and generated
an "Out of Memory" error message this dialog box cannot be called.
Therefore, you will not be able to choose Show from the Access Window
menu to unhide your windows.

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](#)

INF: Access Not Prompting for Second Parameter in Query

Article Number: Q90148
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

Microsoft Access allows the use of user-entered parameters in queries.
If you construct a query that requires more than one prompt, each
prompt must have different syntax.

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: Exporting to SQL Server Does Not Create Indexes

Article Number: Q90149
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

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.

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

INF: Custom Menus Cannot Be Implemented with Pop-up Forms

Article Number: Q90150
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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.

Reference(s):

"The Windows Interface - An Application Design Guide," page 57,
Microsoft Press, 1992.

References

INF: Windows for Workgroups, Novell, SQL Server, and Access

Article Number: Q90151
CREATED: 11-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

If you are running SQL Server on a NetWare network and try to add
Access and Windows for Workgroups, you will not be able to talk to
SQL Server from 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) are
serviced from NETAPI.DLL. The Windows for Workgroups NETAPI.DLL
overrides the Novell NETAPI.DLL, so communication with SQL Server is
broken.

References

INF: Paradox Tables, Indexes, and Multiple Connections

Article Number: Q90152
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

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: Using Access Push Button Controls with Bitmaps

Article Number: Q90153
CREATED: 11-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

This article discusses the limitations of using Access push button
controls with bitmaps.

More Information:

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.

Reference(s):

"Microsoft Access Users Guide," version 1.0, chapter 8, pages 188-211.

[References](#)

INF: Form Must Close for Modal & Popup Settings to Take Effect

Article Number: Q90154
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

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.

Reference(s):

"Microsoft Access Users Guide," version 1.0, chapter 9, pages 213-268

References

INF: Horizontal Sizing Controls Availability on Controls

Article Number: Q90155
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

After placing a control on a form, 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 grip 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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, chapter 9, pages 213-268.

References

INF: Border Widths Not Supported with Raised or Sunken Effects

Article Number: Q90156
CREATED: 11-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

Border width properties are not applicable to controls with raised or
sunken properties set. You can effectively mimic these features of
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 workaround 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 Reference(s):

"Microsoft Access User's Guide," version 1.0, chapter 9.

References

PRACC9210: Find Toolbar Button Not Grayed After Pressing

Article Number: Q90158
CREATED: 11-OCT-1992
MODIFIED: 17-NOV-1992

Summary:

PROBLEM ID: PRACC9210003

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
version 1.0. We are researching this problem and will post new
information here as it becomes available.

Additional Reference(s):

"Microsoft Access Getting Started Guide," version 1.0, page 117.

References

INF: Minimized Popup Forms Cannot be Easily Restored

Article Number: Q90160
CREATED: 11-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

If you use the Minimize action in a macro to minimize a Popup form in
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 Access do not have a
Minimize Box and therefore they work differently.

Reference(s):

"Microsoft Access User's Guide" version 1.0, Chapter 9, Designing
Forms.

References

PRB: Opening Screen Size Is Different from Earlier Session

Article Number: Q90161
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

When starting Access, the window position and size may not be the same
as it was when Access was last closed. This behavior is by design.
Access does not currently have any support to maintain the window state
between sessions.

More Information:

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. Access
remembers if it was closed while maximized, and will restart that way
if it was.

References

PRB: SCROLL LOCK Has No Effect in Form Browse

Article Number: Q90162
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

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](#)

INF: Temporarily Disabling a Macro Action

Article Number: Q90167
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

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

INF: How to Make Access Opening Screen Display

Article Number: Q90170
CREATED: 11-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

To have the initial Access opening screen appear when starting 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

PRB: Reason Val() Function Returns Invalid Use of Null Message

Article Number: Q90722
CREATED: 21-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Microsoft has confirmed that this behavior is by design.

Reference(s):

"Microsoft Access Language Reference," version 1.0, pg. 489.

[References](#)

PRB: Load From Query Dialog Title Does Not Match Menu

Article Number: Q90723
CREATED: 21-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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

```
Microsoft has confirmed this behavior is by design.
```

[References](#)

PRB: Focus Doesn't Shift When Find Accelerators Are Depressed

Article Number: Q90724
CREATED: 21-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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: Giving Controls Duplicate Name Doesn't Generate Error

Article Number: Q90725
CREATED: 21-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

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.

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](#)
identical

INF: Control Names >40 Chars in Access Basic Require Brackets

Article Number: Q90801
CREATED: 22-OCT-1992
MODIFIED: 15-NOV-1992

Summary:

If controls greater than 40 characters are referenced in Microsoft
Access Basic, the following error message appears:

    Identifier too long

More Information:

There is a limit of 40 characters in a "standard" Access Basic
identifier. If you want to use an identifier longer than 40
characters, use the "nonconforming" identifier syntax by surrounding
it with square brackets ([]).

Steps to Reproduce Behavior
----

1. Create a form (Form1) with a text box (Field0).

2. Create a new module.

3. Open the Immediate window.

4. In the Immediate window, type:

       Forms!form1.field0.controlname = _
          "aaaaaaaaaabbbbbbbbbbccccccccccddddddddddeeeeeeeeee"

   The control is set correctly.

5. Try to reference this control by typing:

       ?Forms!form1.aaaaaaaaaabbbbbbbbbbcccccccccc_
                         ddddddddddeeeeeeeeee.TextColor

   The "Identifier too long" message appears.

Reference(s):

Microsoft Access online Help file, "Standard Naming Conventions" topic

[References](#)

PRB: Invalid Use of Null Alert Assigning Null Control

Article Number: Q90802
CREATED: 22-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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: Object Buttons in Database Window Do Not Change Colors

Article Number: Q90803
CREATED: 22-OCT-1992
MODIFIED: 4-DEC-1992

Summary:

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](#)

INF: Can't Change Borders of Up or Pressed Down Controls

Article Number: Q90804
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

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: GoToControl to Self in AfterUpdate Ignored

Article Number: Q90806
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

Summary:

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.

3. View the Categories form in Browse mode.

4. 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

PRB: HelpContextID Does Not Validate Negative Values

Article Number: Q90808
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

Summary:

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 don't 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](#)

INF: How to Use a Query to Filter Unique Data

Article Number: Q90809
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 139-144

[References](#)

INF: How to Force a New Line in a MsgBox Statement

Article Number: Q90810
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

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 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:

    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!"

Note: The formulas above should be entered on one line.

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.

Reference(s):

"Microsoft Access Language Reference," version 1.0, pages 19, 66, 316-
320.

References

INF: How to Check a Menu Item Using Access Basic

Article Number: Q90811
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

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
Microsoft Access Basic code to call Microsoft Windows application
programming interface (API) functions.

More Information:

The following 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.

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 functions 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 functions 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 Microsoft 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, and the command will be checked. If you choose the UnCheck
   command, the Check command will be unchecked.

```
'*********************************************************************
'Declarations section of the module.
'*********************************************************************
Option Explicit

' NOTE: EACH DECLARATION MUST BE PLACED ON A SEPARATE LINE.

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)


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 will display a check mark next to the menu item.
'====================================================================
Function Check_Menu_Item% (TopLevel%, SubLevel%)
   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 will initialize:
'    - 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
```

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 596-602

[References](#)

PRB: Can't Refer to Record in Function Called by OnClose Event

Article Number: Q90812
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

Summary:

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: MoveSize Arguments in Wrong Order in Language Reference

Article Number: Q90813
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

Summary:

The documentation on the MoveSize action in the "Microsoft Access
Language Reference" on pages 315-316 incorrectly states the order of
the arguments for that action as right, down, height, and width. The
correct order of the arguments, as stated in the Microsoft Access Help
topic for the MoveSize action, is right, down, width, and height.

Reference(s):

"Microsoft Access Language Reference," version 1.0, pages 315-316

References

INF: How to Create Do While/Do Until Loops in an Access Macro

Article Number: Q90814
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

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 Microsoft
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.

Reference(s):

"Microsoft Access Language Reference," version 1.0, pages 316-317,
412-414

References

INF: How to Create For Next Loops in a Microsoft Access Macro

Article Number: Q90815
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

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 a Microsoft 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 Microsoft 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    RunMacro
   Loop1             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() Microsoft Access Basic function prior to using this
example; however, after you create the function it is available
anywhere in Microsoft Access.

The Microsoft 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
```

Reference(s):

"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: Freestanding Labels Do Not Appear in Datasheet View

Article Number: Q90816
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

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: Attached Table Must Be Opened as Dynaset from Modules

Article Number: Q90817
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

In Microsoft 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.

Reference(s):

"Microsoft Access Language Reference," version 1.0, Chapter 8

References

INF: Setting Form's Dynaset Property Puts Bookmark at EOF

Article Number: Q90850
CREATED: 22-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

In Microsoft Access Basic, you can make a copy of the dynaset that a
form may currently be using for data manipulation.

The syntax for accomplishing 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.

Reference(s):

"Microsoft Access Language Reference," version 1.0, Chapter 8,

References

INF: Network Users Can Open Their Own SYSTEM.MDA Exclusively

Article Number: Q90851
CREATED: 22-OCT-1992
MODIFIED: 15-NOV-1992

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: ComboBox On Toolbar Do Not Use Window Background Color

Article Number: Q90852
CREATED: 22-OCT-1992
MODIFIED: 13-NOV-1992

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 Access and open the NWIND.MDB database.

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: Double-click Also Executes OnPush Twice

Article Number: Q90853
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

On a form using the OnPush and OnDblClick events, the double-click runs the OnPush event before and after the OnDblClick event.

More Information:

If you want the OnDblClick event to execute without the OnPush event, you must put an explicit CancelEvent action at the end of the OnPush 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 OnDblClick 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 OnDblClick event.

[References](#)

INF: Shrinking Empty or Null OLE Object on Report

Article Number: Q90855
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

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 NWIND.MDB file that is
shipped with Microsoft Access:

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 Null       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
erase can shrink

PRB: CancelEvent Action in OnClose Macro Prevents Exit

Article Number: Q90856
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 549-552
```

References

INF: Cut and Paste Removes Blank Lines in Macro's Design View

Article Number: Q90857
CREATED: 22-OCT-1992
MODIFIED: 13-NOV-1992

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 21

References

PRB: Mail Label Wizard Cuts Off First Field at Top of Page

Article Number: Q90858
CREATED: 22-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.00, page 401.

References

PRB: Fonts May Change Zooming Out in Print Preview

Article Number: Q90859
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 406-413.

References

PRB: #Error When ControlName Used in ControlSource Expression

Article Number: Q90860
CREATED: 22-OCT-1992
MODIFIED: 23-DEC-1992

Summary:

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

INF: Connecting to the First Available Network Drive

Article Number: Q90861
CREATED: 22-OCT-1992
MODIFIED: 17-NOV-1992

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.

1. Declare the API functions.

```
   '-----------------------------------------------------------------
   'NOTE: ENTER THE DECLARATION FUNCTIONS ON ONE LINE EACH.
   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) + ":"

         'NOTE: ENTER THE FOLLOWING STATEMENT ON ONE LINE.
         'Also substitute the appropriate server share and password.
         Results% = WNetAddConnection("\\server\share", "password", _
                                   FirstFreeDrive)
      End Function
```

Reference(s):

"Microsoft Access Introduction to Programming," version 1.0, page 8

[References](#)

INF: Using the Windows 3.1 API to Connect to Network Resources

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.

1. Declare the functions required to add and remove network
   connections. Type the lines in a module in the Global Declarations
   section.

   ```
   '------------------------------------------------------------------
   ' NOTE: ENTER THE DECLARATION FUNCTIONS ON ONE LINE EACH.
   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. Note that each command
   should be on one line. The line-continuation character ("_")
   improves the readability of the code.

   ```
       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 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.

Reference(s):

"Microsoft Access Introduction to Programming," version 1.0, Chapter 2, "Writing a New Function," pages 8-14

[References](#)

INF: Older Versions of Shared DLLs Cause Problems with Access

Article Number: Q90863
CREATED: 22-OCT-1992
MODIFIED: 23-DEC-1992

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 operating system
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   97984 bytes   10/1/92   Windows for Workgroups

OLECLI.DLL    83456 bytes             Same for all

OLESRV.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     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     41520 bytes   10/1/92   Windows for Workgroups

VER.DLL        9008 bytes
```

(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 doesn't 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: Access Saves Printer Information with Each Form/Report

Article Number: Q90864
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

Microsoft Access version 1.0 saves printer driver information with
each form or report that it generates. This means you don't 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
true type

PRB: GP Fault If MSACCESS.INI Installable ISAM Path Uses "\.."

Article Number: Q90865
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

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: Commonly Asked Questions About 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 README.TXT file that you installed with Access during setup.

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: Select All Not Grayed Even When No Items Are Available

Article Number: Q90867
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

Summary:

SYMPTOMS

In Form View, the Select All option in the Edit menu is never
grayed out, even if there are no items to select.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access
version 1.00. We are researching this problem and will post new
information here as it becomes available.

More Information:

```
Steps to Reproduce Behavior
---------------------------
```

1. Choose the Form control button, then choose New and Blank Form to
   create a new form with no controls.

2. Select the Edit menu. You will notice that Select All is available,
   even though there is nothing on the form to select.

References

PRB: Sizing Rows and Columns to Zero Works Differently

Article Number: Q90868
CREATED: 22-OCT-1992
MODIFIED: 6-DEC-1992

Summary:

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: ReportWizard Select No Fields for Form Still Shows All

Article Number: Q90869
CREATED: 22-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

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

INF: Page Header Will Span Full Width Of MultiColumn Report

Article Number: Q90870
CREATED: 22-OCT-1992
MODIFIED: 13-NOV-1992

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 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 Textbox bound to the group field. Place
   the Textbox in the Group Header. Size this control as 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, 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 spacing or
   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 will create 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, select 'Before Section'. This
    will cause the group header to print at the top the column once
    for each new group.

15) Print or Preview the report.

For another article 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

For additional information on columnar reports search Help for
"Multiple-column reports" and "NewRowOrCol".

Reference(s):

"Microsoft User's Guide," version 1.0, chapter 18, "Designing
Reports," and chapter 19, "Sorting and Grouping Data."

[References](#)

INF: Mouse Click Not Turning Off Extend (Selection) Mode

Article Number: Q90873
CREATED: 25-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

When you are in the QBE grid and enter extend (EXT) mode by pressing
F8, choosing another column with the mouse will not turn the extended
selection off (this is reflected in the status bar).

More Information:

This is the correct function of extend mode. To turn off extend mode,
press the ESC key.

[References](#)

INF: Correct Procedure for Inserting New Field Above Existing

Article Number: Q90874
CREATED: 25-OCT-1992
MODIFIED: 6-DEC-1992

Summary:

On page 51 of "Getting Started," 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: Headers On Non-Continuous Forms Disable Scroll Bar

Article Number: Q90875
CREATED: 25-OCT-1992
MODIFIED: 13-NOV-1992

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: Displaying List or Combo Boxes with More Than One Field

Article Number: Q90876
CREATED: 25-OCT-1992
MODIFIED: 12-NOV-1992

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 9

[References](#)

INF: Copy/Paste Operations from Forms to Queries

Article Number: Q90877
CREATED: 25-OCT-1992
MODIFIED: 12-NOV-1992

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

PRB: Tables Are Not Browsed in Primary Key Order

Article Number: Q90878
CREATED: 25-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 21-33 and 87-114

References
source

INF: Empty Table Created by SELECT INTO with No Matching Rows

Article Number: Q90879
CREATED: 25-OCT-1992
MODIFIED: 12-NOV-1992

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 6

References

PRB: Long WHERE Clause Is Truncated After View SQL Command

Article Number: Q90880
CREATED: 25-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

SYMPTOMS

In Microsoft Access' Query By Example (QBE) grid, having a WHERE
clause greater than 255 characters causes that clause to be
truncated after the SQL command on View menu is used to see the SQL
statement. 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 combines queries
that are entered on separate lines. The limit for each QBE grid 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

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 6

References
remove erase missing delete doesn't show appear buglist1.00

INF: DDE Link Can't Find Data; Opens Second Copy of Server App

Article Number: Q90881
CREATED: 25-OCT-1992
MODIFIED: 6-DEC-1992

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.

Reference(s):

"Microsoft Access Basic: An Introduction to Programming," version 1.0,
Chapter 9, "Dynamic Data Exchange"

References
msg source container document worksheet

PRB: Two Users Can Open the Same dBASE Database Exclusive

Article Number: Q90882
CREATED: 25-OCT-1992
MODIFIED: 19-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 4

References

PRB: Alert With Wrong Filename Given When Opening File Copied

Article Number: Q90985
CREATED: 26-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

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 at DOS during the
copy procedure, Microsoft Access has no way of knowing what is
happening to the file being opened.
```

RESOLUTION

```
Microsoft has confirmed this behavior in Microsoft Access version
1.0 for Windows and that this is proper behavior according to the
design specification.
```

References

PRB: Change Option Affects Only One Instance of Many of Access

Article Number: Q90986
CREATED: 26-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Reference(s):

"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

INF: Common Dialog Boxes Allow Only Lowercase Typing

Article Number: Q90987
CREATED: 26-OCT-1992
MODIFIED: 12-NOV-1992

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
letters

INF: How To Use Get and Write PrivateProfileString

Article Number: Q90988
CREATED: 26-OCT-1992
MODIFIED: 13-NOV-1992

Summary:

Microsoft Access version 1.0 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:

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.

   Enter each declaration statement below on one line.

```
        Option Explicit
        Declare Function GetPrivateProfileString% Lib "Kernel"
                                (ByVal lpapplicationname$,
                                 ByVal lpkeyname$,
                                 ByVal lpDefault$,
                                 ByVal lpreturnedstring$,
                                 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"
           lpreturnedstring$ = Space$(255)
           nSize% = Len(lpreturnedstring$)

           'NOTE: The following statement is entered on one line.

           GetIni = GetPrivateProfileString(lpapplicationname$,
                                            lpkeyname$, lpDefault$,
                                            lpreturnedstring$, nSize%,
                                            lpfilename$)
           GotInfo = lpreturnedstring$
        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"

           'NOTE: The following statement is entered on one line.

           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.

Reference(s):

"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: How To Tell If An Object Exists in the Database

Article Number: Q90989
CREATED: 26-OCT-1992
MODIFIED: 15-NOV-1992

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 As String - 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 declaration statements that follow in the Declarations page in the new module.

   ```
   Global ObjectFound%
   Global WhatObjectType As String
   ```

4. 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 Then      'Object was found in the DBC
            ObjectFound% = 0
            MsgBox "Error #" & Err & ": " & Error
        Else
            ObjectFound% = -1
        End If
    End Function
```

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.

Reference(s):

"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

PRB: Cannot Scroll Left in Form's Design View with Rulers

Article Number: Q90990
CREATED: 26-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 8, "Form Basics," and Chapter 9, "Designing Forms"

[References](#)

INF: Transparent Text Boxes Not Supported in Access

Article Number: Q91045
CREATED: 27-OCT-1992
MODIFIED: 23-DEC-1992

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](#)

INF: SQL Server Triggers and Key Values to and from Access

Article Number: Q91179
CREATED: 29-OCT-1992
MODIFIED: 12-NOV-1992

Summary:

If an 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 Access procedure will lose reference to the record.

More Information:

When information is sent to a SQL Server table, 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, Access will lose
reference to that record because the unique key stored by 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, Access views the record just sent
to SQL Server as having been deleted.

[References](#)

PRB: Expression in Query Grid Doesn't Show All Text

Article Number: Q91705
CREATED: 3-NOV-1992
MODIFIED: 6-DEC-1992

Summary:

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 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 column:

   [Unit Price]*[Quantity]

4. Press tab or enter to move to the next column.

The expression in the first column appears truncated.

Reference(s):

Microsoft Access On-Line Help, search on "Zoom Box"
```

References

INF: Use CTRL+M to Open Objects in Design Mode

Article Number: Q91707
CREATED: 3-NOV-1992
MODIFIED: 12-NOV-1992

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: Query Design Status Bar Does not Have Help Message

Article Number: Q91708
CREATED: 3-NOV-1992
MODIFIED: 13-NOV-1992

Summary:

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.

Reference(s):

On-Line Help, search on "F1" and "Keyboard: Navigation Keys".

References

PRB: Changing Caption on Title Label Leaves Shadow Unmodified

Article Number: Q91709
CREATED: 3-NOV-1992
MODIFIED: 6-DEC-1992

Summary:

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: No Implicit Parameters on CrossTab Queries

Article Number: Q91710
CREATED: 3-NOV-1992
MODIFIED: 6-DEC-1992

Summary:

SYMPTOMS

If you create a CrossTab (XTAB) Query using an implicit parameter
[XXX] in a WHERE clause, Microsoft Access will return the following
error:

    "Cannot Bind '[XXX]'"

If you use an explicit parameter, the query works as
expected.

CAUSE

A CrossTab query dynamically creates column names. Therefore,
Microsoft Access cannot tell whether [XXX] is referring to a parameter
or a column name until after the query is bound. This may result in a
Catch-22 situation, since sometimes it is necessary to run the query
before binding it.

RESOLUTION

To avoid this situation, define [XXX] as an explicit parameter.

STATUS

This behavior is by design.

Reference(s):

Search On-Line Help for "Creating a Parameter Query."

[References](#)

INF: 'Couldn't Open File' Error Message Using Paradox Table

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: File open (I104) - c:\pdoxdata\yourfile.db
   (Code 5) -- Leaving Paradox
```

This may be caused by Microsoft Access having the table open.

In both situations the possibilty 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](#)

INF: Height/Width Settings Don't Affect Size of Objects

Article Number: Q91712
CREATED: 3-NOV-1992
MODIFIED: 11-NOV-1992

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.

For additional information, search Help for 'Option Button Properties' and select the topic 'Option Button Control Properties.'

Reference(s):

"Microsoft User's Guide" version 1.0, chapter 9, "Designing Forms", pages 247-252.

References

PRB: Grid Lines Not Visible in Access When Using Wizards
Article Number: Q91713
CREATED: 3-NOV-1992
MODIFIED: 13-NOV-1992

Summary:

SYMPTOMS

The grid is not visible in design mode when using Wizards for Forms or
Reports.

CAUSE

Report Wizards and Form Wizards 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 the Wizard, but will not be visible if the Wizards are
used. This is true only in design mode. The grid can be made visible
when you use the Wizards 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

PRB: Toolbars Do Not Use Control Panel Button Background

Article Number: Q91714
CREATED: 3-NOV-1992
MODIFIED: 15-NOV-1992

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 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: Text May be Hard to Read on an EGA Screen

Article Number: Q91715
CREATED: 3-NOV-1992
MODIFIED: 11-NOV-1992

Summary:

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: Two Labels on Top of Each Other Both Print

Article Number: Q91716
CREATED: 3-NOV-1992
MODIFIED: 11-NOV-1992

Summary:

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

INF: Form Sizing Grid Accepts Non-Integer Values

Article Number: Q91720
CREATED: 3-NOV-1992
MODIFIED: 11-NOV-1992

Summary:

When you use 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: FormWizard Labels Placed on Left Margin Grow Down

Article Number: Q91721
CREATED: 3-NOV-1992
MODIFIED: 11-NOV-1992

Summary:

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

Summary:

```
=======================================================================
  Microsoft(R) Product Support Services Application Note (Text File)
          WX0635: DATABASE STRUCTURE QUESTIONS & ANSWERS
=======================================================================
                                              Revision Date: 10/92
                                                 No Disk Included


  -------------------------------------------------------------------
| 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 Microsoft Corporation. All Rights Reserved.        |
| Microsoft and MS-DOS are registered trademarks and Microsoft      |
| Access and Windows are trademarks of Microsoft Corporation.       |
| 1-2-3 and Lotus are registered trademarks of Lotus Development    |
| 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.                    |
  -------------------------------------------------------------------

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), 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 for Windows versions 2.x
      and later, 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 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.

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.

[References](#)

FastTips: Access Basic and Macros Questions & Answers

Article Number: Q92472
CREATED: 8-NOV-1992
MODIFIED: 12-NOV-1992

Summary:

```
========================================================================
  Microsoft(R) Product Support Services Application Note (Text File)
        WX0636: ACCESS BASIC AND MACROS QUESTIONS & ANSWERS
========================================================================
                                              Revision Date: 10/92
                                                 No Disk Included
```

```
1. Q. How can I run macros from Microsoft Access Basic?

   A. The following are three suggested ways to run macros from Access
      Basic:

      a. 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>"

      b. 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>"

      c. 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.

[References](#)

FastTips: Forms Questions & Answers

Article Number: Q92475
CREATED: 8-NOV-1992
MODIFIED: 12-NOV-1992

Summary:

```
========================================================================
  Microsoft(R) Product Support Services Application Note (Text File)
                  WX0637: FORMS QUESTIONS & ANSWERS
========================================================================
                                                  Revision Date: 10/92
                                                     No Disk Included


  --------------------------------------------------------------------
| 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 Microsoft Corporation. All Rights Reserved.         |
| Microsoft and MS-DOS are registered trademarks and Microsoft       |
| Access and Windows are trademarks of Microsoft Corporation.        |
  --------------------------------------------------------------------
```

1. Q. How do I invoke my Microsoft Access Basic code from within a
      form?

   A. To call a function from a property, type the following:

        =function name()

      The equal sign and parentheses are required. You can call an
      Access Basic function from form or control properties, such as
      After Update or Default Value. 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![master form name]![subform control name].form![control name]

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."

References

FastTips: Querying Questions & Answers

Article Number: Q92476
CREATED: 8-NOV-1992
MODIFIED: 12-NOV-1992

Summary:

```
========================================================================
  Microsoft(R) Product Support Services Application Note (Text File)
                WX0639: QUERYING QUESTIONS & ANSWERS
========================================================================
                                               Revision Date: 10/92
                                                  No Disk Included
```

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. 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. In Microsoft Access queries, you specify DISTINCT by
      choosing Query Properties from the View menu and 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 major 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-side table). DISTINCTROW is specified by default in
      Microsoft Access queries and is ignored in queries where it has
      no effect.

References

FastTips: Reporting Questions & Answers

Article Number: Q92477
CREATED: 8-NOV-1992
MODIFIED: 24-NOV-1992

Summary:

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 Record Source 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

FastTips: General Information Questions & Answers

Article Number: Q92496
CREATED: 8-NOV-1992
MODIFIED: 12-NOV-1992

Summary:

```
========================================================================
   Microsoft(R) Product Support Services Application Note (Text File)
           WX0638: GENERAL INFORMATION QUESTIONS & ANSWERS
========================================================================
                                              Revision Date: 11/92
                                                 No Disk Included


  -------------------------------------------------------------------
| 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 Microsoft Corporation. All Rights Reserved.        |
| Microsoft and MS-DOS are registered trademarks and Microsoft      |
| Access and Windows are trademarks of Microsoft Corporation.       |
  -------------------------------------------------------------------
```

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, GEnie, 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" (without

the quotation marks) at the command prompt. To locate the WDL.TXT file, search on the word S13354.

If you do not have a modem, you can order the WDL on disk 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. There is a $20 charge to order the WDL on disk.

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.

References

PRB: Cannot Install Cue Cards Without Installing Help

Article Number: Q92639
CREATED: 12-NOV-1992
MODIFIED: 13-NOV-1992

Summary:

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 Cards" files increases from 1688 K to
5112 K. However, the total disk "Space required" does not increase and
the size of "Help" does not decrease from 3424 K to 0 K.

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 Cards
files.

STATUS

This behavior is by design.

[References](#)

INF: How to Take a Button Out of Tab Order

Article Number: Q92640
CREATED: 12-NOV-1992
MODIFIED: 6-DEC-1992

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

INF: Differences Between Native and Attached Tables

Article Number: Q92644
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

Attached tables (ie. Paradox, dBase IV, etc.) are not treated
differently from Microsoft Access tables, except for:

1. In the table list, attached table icons look slightly different
   than Access table icons. For attached tables, a black arrow is
   added to the table icon.

2. When attempting to take an attached table into design mode, you get
   a warning message, "Table 'xxxxx' is an attached table. Some
   properties can't be modified. Open it anyway?"

3. When you delete an attached table, its contents are not deleted
   (Which is what happens when you delete a native Access table.) Only
   the attached table entry is deleted from the list of tables.

   However, the alert message is the same for both types of deletions,
   which may lead you to believe that the attached table has been
   erased when it has not.

References

INF: How to Remove Relationships

Article Number: Q92649
CREATED: 12-NOV-1992
MODIFIED: 23-DEC-1992

Summary:

This article explains how to delete a relationship in Microsoft Access once
it has been created.

More Information:

The sample database NWIND 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

PRB: OLE Objects Format From Excel Changes After Being Updated

Article Number: Q92650
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

New OLE Objects created in Microsoft Excel 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:

This article contains the steps to reproduce this problem.

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

PRB: Excel Custom Date Format Not Detected by Import

Article Number: Q92651
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

SYMPTOMS

When importing an Excel spreadsheet containing custom date formats
into a new table, Microsoft Access imports the formatted fields with
the data type of Double instead of Date/Time.

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 Excel file into an existing
table where that field has been formatted for Date/Time. Access will not
recognize the custom format but will display the date in the standard
mm/dd/yy Date/Time format.

Another option is to go into the design view of the newly imported
table 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 Excel spreadsheet.

2. Enter a date into Cell A1.

3. Create a New Date format such as mmmmm/dd/yyyy.

4. Apply the format to the A column.

5. Import the spreadsheet with Access.

Results: Date comes in as a Number (Double) not Date/Time.

Reference(s):

"Microsoft Access User's Guide," version 1.0, chapter 4, pages 72-75.

References

PRB: Can't Attach to Paradox Table on Read-Only Drive

Article Number: Q92653
CREATED: 12-NOV-1992
MODIFIED: 17-NOV-1992

Summary:

SYMPTOMS

You cannot attach to a Paradox table (*.DB) that is on a read-only
drive. If you attempt to attach you receive the error message:

Couldn't open file 'FILENAME#DB'

CAUSE

Neither Paradox or the Paradox ISAM shipped with Microsoft Access
support databases on read-only shares.

STATUS

This behavior is by design.

More Information:

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

PRB: How to Create a Chart Without a Title

Article Number: Q92655
CREATED: 12-NOV-1992
MODIFIED: 6-DEC-1992

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: CTRL+HOME Navigation Problem in Macro Design Mode

Article Number: Q92665
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

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: No Permission to Modify Data in a Table

Article Number: Q92667
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

When you attempt to modify data in a table (while in datasheet view),
you may receive the following error message:

    No permissions for '<name of table>'

This is confusing, because you were able to open the table without any
warning messages.

More Information:

This message indicates that you only have Read Definitions and Read
Data permissions for the table. Because you have rights to view the
data, there is no error message when you first open the table in the
default datasheet view.

A more descriptive message is generated when you first open the table
in design view:

    You don't have permission to modify '<name of table>'.
    Open as read-only?

This message appears when opening the table in design view because you
have no security permissions that would provide you design view
usability.

Reference(s):

"Microsoft Access User's Guide", version 1.0, page 621.
For more information, search in Help on "permissions command".

References

PRB: Can't Exit Extend Mode in Parameters Dialog Box

Article Number: Q92669
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

SYMPTOMS

When in the Parameters dialog box of a query, you cannot get out of
Extend mode by pressing SHIFT+F8.

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.

At this point pressing SHIFT+F8 does not take you out of Extend mode.

[References](#)

PRB: Default Width in Report Design Does Not Reflect Margins

Article Number: Q92670
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

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.
```

More Information:

```
Search Help for "Creating a New Template for Forms and Reports" to change
the desired layout of reports or forms.
```

References

INF: Error Message "Unable to Recover"

Article Number: Q92671
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

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 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 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: TransferDatabase Macro Accepts Invalid File Names

Article Number: Q92674
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

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](#)

PRB: Insert Object Dirties Record

Article Number: Q92675
CREATED: 12-NOV-1992
MODIFIED: 19-NOV-1992

Summary:

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: Subform not Updated

Article Number: Q92676
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

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.

STATUS

This behavior is by design.

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).

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.

Reference(s):

In the online Help, search on "Link: main form and subform" and
"function keys".

References

PRB: Scrolling to Bottom of Module Window Goes Past Typed Info

Article Number: Q92677
CREATED: 11-NOV-1992
MODIFIED: 15-NOV-1992

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: Access Trims Spaces Off Database Names

Article Number: Q92678
CREATED: 12-NOV-1992
MODIFIED: 15-NOV-1992

Summary:

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: Out Of Memory When Pasting Text Into Module

Article Number: Q92679
CREATED: 12-NOV-1992
MODIFIED: 19-NOV-1992

Summary:

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](#)

INF: Currency is Significant to Four Digits

Article Number: Q92680
CREATED: 12-NOV-1992
MODIFIED: 6-DEC-1992

Summary:

When using the currency data type within Microsoft 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 degree of 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](#)

INF: Defragment and Compact Database to Improve Performance

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: Parameter Queries, Wildcards, and SQL Server

Article Number: Q92682
CREATED: 12-NOV-1992
MODIFIED: 18-NOV-1992

Summary:

"*" and "?" are not valid wildcard characters in parameterized SQL
Server queries, and thus are interpreted by SQL Server as text. This
means that unless "*" and/or "?" happen to be present in the field
restricted by the parameter, the query will return an empty set.

Instead of using "*" and "?" in parameterized queries, use the SQL
Server wildcard characters ("%" and "_").

More Information:

SQL Server uses the ANSI standard wildcards "%" and "_", whereas
Microsoft Access uses "*" and "?". When using a parameterized query,
Microsoft Access does not translate the "*" or "?" to the SQL Server
wildcard equivalents "%" or "_". The string that is typed in the
parameter dialog box is passed verbatim to SQL Server. Microsoft
Access only translates strings that are part of the QBE Grid.

Reference(s):

For more information, search for "Parameter Query" using the Help
menu.

References

INF: '#Name?' or Control is Blank on Form Based on a Query

Article Number: Q92683
CREATED: 12-NOV-1992
MODIFIED: 18-NOV-1992

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: How to Change the Caption on the Access Window

Article Number: Q92684
CREATED: 12-NOV-1992
MODIFIED: 23-DEC-1992

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 optionally 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. On the next line in the module, type the following and press enter:

```
   Function ChangeAccessCaption(Caption as String)
```

6. 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
```

7. Save the module as "Module1".

8. 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](#)

PRB: GoToRecord Macro Doesn't Work on Subform

Article Number: Q92685
CREATED: 12-NOV-1992
MODIFIED: 19-NOV-1992

Summary:

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 up 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

        Macro1   GoToControl
                                        GoToRecord
        Macro1 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](#)

PRB: Opening a Database Read Only Forces Others to Read Only

Article Number: Q92686
CREATED: 12-NOV-1992
MODIFIED: 18-NOV-1992

Summary:

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.
```

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.

Work Around
-----------

To override the database opening in Read Only mode, do not select
Exclusive when opening the second instance.
```

[References](#)

PRB: Report or Form Bound to a Table Locks the Table

Article Number: Q92687
CREATED: 12-NOV-1992
MODIFIED: 17-NOV-1992

Summary:

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

INF: Setting a Default Value to The Last Record Entered

Article Number: Q92688
CREATED: 12-NOV-1992
MODIFIED: 18-NOV-1992

Summary:

Below is a technique you can use to set a control's default value to
the last value entered into it.

More Information:

This technique is a two-step process:

1. Create a hidden control on the form.

2. In the control whose default value you want to change, use the
   SetValue macro in its OnEnter and OnExit Properties.

Example

For example, if you have a form (Form1) and you want to set a control
on that form (Control1) to the last value that was entered into it:

1. Create another control, HiddenControl2, and set its Visible
   property to no (hidden).

2. Create two Macros, Macro1 and Macro2:

```
    MacroName        Action
    ---------------------------------------------------
    Macro1           SetValue
    Macro2           SetValue

    Macro1 Actions
    ---------------------------------------------------
    SetValue
      Item:  Forms![Form1]![Control1]
      Expression:  Forms![HiddenForm2]![HiddenControl2]

    Macro2 Actions
    ---------------------------------------------------
    SetValue
      Item:  Forms![HiddenForm2]![HiddenControl2]
      Expression:  Forms![Form1]![Control1]
```

3. Set the following properties:

```
    Control1
    ---------------------------------------------------
    OnEntry: Macro1
    OnExit: Macro2
```

Reference(s):

"Microsoft Access User's Guide," version 1.0, chapter 9, "Designing Forms," pages 224-227.

"Microsoft Access User's Guide," version 1.0, chapter 22, "Using Macros with Forms," pages 538-541.

[References](#)

INF: Queries Sort by Order of IN() Parameters

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 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

INF: Limit of AND's in SQL Select Statement

Article Number: Q92690
CREATED: 12-NOV-1992
MODIFIED: 6-DEC-1992

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](#)

INF: Unsupported SQL Keywords Not Listed in Access

Article Number: Q92691
CREATED: 12-NOV-1992
MODIFIED: 17-NOV-1992

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 1.0. However, there is no list of SQL keywords that are not supported in Microsoft Access version 1.0.

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Appendix C, "Expressions", pages 649-650, 655-656.

References

INF: Pasting Columns to the Clipboard in Access

Summary:

It is not possible in Table Datasheet Browse/Edit mode to select a
column of data and copy it to the clipboard, as you can in Microsoft
Excel.

There are, however, other ways to change or rearrange data without
copying to the clipboard. This article gives three likely scenarios
where you might want to change or rearrange data, and techniques to do
it.

More Information:

```
Scenario A
----------
```

If your purpose is to rearrange the order of the columns, there is no
need to cut-and-paste. To rearrange columns:

```
   1. Select the column using the column indicator.
   2. Then, drag the column to the desired position, and release it.
```

For more information, search Help on "Column Width (datasheets)" and
read "Changing a Datasheet's Appearance." Also, read "To Move a
Column" in the Microsoft Access User's Guide, Chapter 14: "Adding and
Editing Data."

```
Scenario B
----------
```

If your purpose is to add an additional column to an Access table, you
must run an Update Query. If you are also adding new records, you must
run an Append Query (step 3 below) before running the Update Query.

The steps below assume two tables, Master and ExtraInfo (data shown in
the example below.) To copy information from ExtraInfo to Master:

1. Create ExtraInfo and Master to have the same primary key.

2. Manually add the new column names from ExtraInfo to Master. Make
   sure they are the same data type. They don't have to be the same name
   as the corresponding fields in ExtraInfo, but it saves a lot of typing
   if the names are the same.

3. (Skip this step if ExtraInfo contains only records that exist in
   Master.) Use ExtraInfo to create an Append Query. Specify Master as
   the destination table. You must make sure that the Unique Values
   query property is checked. Add all the field names from ExtraInfo
   to the QBE grid. If the field names are spelled correctly,
   Microsoft Access will automatically fill in the Append To field
   with the name of the correct field from Master. If not, you must

select the Master fieldname manually in the Append To field. Run
   the Query. New records will be appended to Master from ExtraInfo.
   Other records in ExtraInfo will be processed by the Update Query
   in the next step.

4. Create an Update Query that contains both Master and ExtraInfo.
   Join the two tables by dragging the key field from Master to
   ExtraInfo. Drag the new fields from Master and drop them onto
   the QBE grid. You must type the [ExtraInfo]![<fieldname>]
   manually into the Update From field. Run the query to update
   the new fields in existing Master records.

EXAMPLE for Scenario B
----

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. Now, 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 a new query. Add ExtraInfo to the query & close the Add
   Table dialog box. Select Append... from the Query menu. Choose
   Master as the table to append to. Also make sure that the Unique
   Values setting is checked. Drag the KEY field and the new field,
   ITEM2, into the QBE grid. Microsoft Access will fill in the Append
   To field names. Run the query. Microsoft Access will respond "2
   rows will be appended." Press OK. Microsoft Access will respond
   with a message stating that "1 record(s) were lost due to key
   violations" (This is OK. The record with Key=1 couldn't be
   appended. This will be taken care of by the Update Query). Press
   OK. Master should now look like this:

        MASTER: KEY   ITEM1   ITEM2
                  1   AAAAA
                  2   BBBBB
                  3   33333

4. Create a new query. Add both ExtraInfo and Master to the
   query & close the Add Table dialog box. Join the tables by
   dragging the KEY field from ExtraInfo to Master. A line will
   connect the two tables. Select Update from the Query Menu.
   Drag the new field, ITEM2 from Master into the QBE grid.
   Type [ExtraInfo]![ITEM2] in the Update From position. Run the
   query. Microsoft Access will respond "2 rows will be updated."
   Press OK. 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 Help on
"Appending Records to a Table", "Update Queries, Updating Records as a
Group", and "Joins, Joining Tables in a Query." Also, look in the
Microsoft Access User's Guide under "Appending Records" and "Updating
Tables" in chapter 7, and "Creating Multiple Table Queries" in chapter
6.

Scenario C

If your purpose is to copy a column of data from Access to another
application, such as Microsoft Excel, try one of the following
methods:

1. Use Copy & Paste to transfer the records you desire. Then delete
   the unwanted columns in Excel.

2. 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 desire. You must Unhide each column
   individually.

3. Create a Query that contains only the columns that you want. Then
   use Copy & Paste to transfer the records that you desire. This last
   method is useful if you plan to do this on a regular basis.

For more information, search Help on "Copy Command."

References

INF: Inconsistent Save and Save As Menus with Forms

Article Number: Q92693
CREATED: 12-NOV-1992
MODIFIED: 16-NOV-1992

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: Using TransferDatabase to Export to Btrieve Format

Article Number: Q92694
CREATED: 12-NOV-1992
MODIFIED: 05-JAN-1993

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

PRB: Fields on Popup not Editable when Access is Minimized

Article Number: Q92695
CREATED: 12-NOV-1992
MODIFIED: 16-NOV-1992

Summary:

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.

Reference(s):

"Microsoft Access Language Reference" version 1.0, pages 348-351, and 432-435.

References

PRB: Disabled Buttons Not Visible with ATI 1280x1024

Article Number: Q92701
CREATED: 12-NOV-1992
MODIFIED: 13-NOV-1992

Summary:

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

INF: Status Bar Text Too Long for Auto Repeat Property

Article Number: Q92702
CREATED: 12-NOV-1992
MODIFIED: 13-NOV-1992

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: Alt+ Any Key Combination Does Not Work With Active Popup

Article Number: Q92703
CREATED: 12-NOV-1992
MODIFIED: 13-NOV-1992

Summary:

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: CueCards Started Outside Access Does Not Work

Article Number: Q92706
CREATED: 12-NOV-1992
MODIFIED: 13-NOV-1992

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

INF: Form View Allowed When ViewsAllowed Set to Datasheet

Article Number: Q92709
CREATED: 12-NOV-1992
MODIFIED: 17-NOV-1992

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

PRB: Out of Disk Space Error Makes Paradox Table Unusable

Article Number: Q92739
CREATED: 15-NOV-1992
MODIFIED: 19-NOV-1992

Summary:

SYMPTOMS

When working with a Paradox table in Microsoft Access you receive the
following error message if you run out of disk space:

    Out of Disk Space

You continue to get the same error message even after you remedy the
problem by freeing up disk space.

RESOLUTION

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 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.

[References](#)

PRB: Help Gives Error Importing File From Floppy Disk

Article Number: Q92740
CREATED: 15-NOV-1992
MODIFIED: 19-NOV-1992

Summary:

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: Choosing a New Printer Resets Some Printer Settings

Article Number: Q92741
CREATED: 15-NOV-1992
MODIFIED: 20-NOV-1992

Summary:

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](#)

INF: Planning Database Security

Article Number: Q92742
CREATED: 15-NOV-1992
MODIFIED: 20-NOV-1992

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: No Error Given When Linked OLE Object Can't be Found

Article Number: Q92745
CREATED: 15-NOV-1992
MODIFIED: 6-DEC-1992

Summary:

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: Exporting Table to Excel Does not Set 'Database' Name

Article Number: Q92746
CREATED: 15-NOV-1992
MODIFIED: 6-DEC-1992

Summary:

When you export a table from Microsoft Access to Microsoft Excel,
the defined name "Database" is not set in Excel.

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 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](#)

INF: Count() Ignores Null Values

Article Number: Q92747
CREATED: 15-NOV-1992
MODIFIED: 4-DEC-1992

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;
```

Reference(s):

On-Line Help, search on "Count" and select the topic "Calculating
Totals in a Query" and "Count Function" for more information.

[References](#)

INF: Hints for Printing Mailing Labels

Article Number: Q92807
CREATED: 16-NOV-1992
MODIFIED: 11-DEC-1992

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

PRB: QBE Grid Disappears on Highlight and Delete

Article Number: Q92808
CREATED: 16-NOV-1992
MODIFIED: 11-DEC-1992

Summary:

SYMPTOMS

The Query By Example (QBE) grid, which displays when you enter
design mode for a query in Microsoft Access, disappears if you
highlight the grid by clicking on the query grid's column header and
drag across to the end of the query grid, then select Delete from the
Edit menu.

CAUSE

Microsoft Access deletes the grid columns instead of clearing the
contents of the columns when you highlight a whole column, or an
entire grid, and then select Delete from the Edit menu.

RESOLUTION

To get a blank query grid column to display after the grid has
disappeared, choose Insert Column from the Edit menu.

To get a column already filled with field information to display,
click on a field name in one of the field lists displayed in the top
portion of the query design window, then drag the field to the query
grid portion of the query design window, or double-click on the field.

STATUS

This behavior is by design.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 92-100

References

INF: RunApp Action Command has 255 Character Argument Limit

Article Number: Q92809
CREATED: 16-NOV-1992
MODIFIED: 11-DEC-1992

Summary:

The RunApp action for macros allows 255 characters.

More Information:

MS-DOS has a 127 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

PRB: BeforeUpdate Macro Executing Twice

Article Number: Q92810
CREATED: 16-NOV-1992
MODIFIED: 16-NOV-1992

Summary:

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 Paste Append records, 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 the user
is 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 OK for the Message box (that you created in Macro1).

10. Close the form.

11. Choose OK for the Message box (that you created in Marcro1)

12. Choose OK for the following error message:

    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?

Notice that the message box you created in Macro1 re-appeared in step
#11. The macro that you assigned to the BeforeUpdate property is

executing (1) when the record that you Paste Appended is saved, and (2) when when the edited record is saved.

Reference(s):

Microsoft Access On-Line Help, search on "BeforeUpdate, AfterUpdate Properties"

[References](#)

PRB: Can't db.OpenTable With Only Read Definition Permission

Summary:

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: Cannot Use DoMenuItem for Open Database or New Database

Article Number: Q92812
CREATED: 16-NOV-1992
MODIFIED: 11-DEC-1992

Summary:

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.
```

STATUS

```
This behavior is by design.
```

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.
```

```
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: Column Widths in Table Design not Saved

Article Number: Q92813
CREATED: 16-NOV-1992
MODIFIED: 11-DEC-1992

Summary:

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: No Error Msg When BoundColumn Greater Than ColumnCount

Article Number: Q92814
CREATED: 16-NOV-1992
MODIFIED: 11-DEC-1992

Summary:

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.
```

STATUS

```
This behavior is by design.
```

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.

For more information, seach for "ColumnCount" and "BoundColumn" using
the Help menu.
```

```
Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 239-241
```

[References](#)
ColumnCount, BoundColumn

INF: Using the Generic Print Driver with Forms/Reports

Article Number: Q92815
CREATED: 16-NOV-1992
MODIFIED: 3-DEC-1992

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.

- Don't 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

INF: Converting Julian Dates with Access Basic

Article Number: Q92816
CREATED: 16-NOV-1992
MODIFIED: 11-DEC-1992

Summary:

Most government agencies and contractors require the use of Julian
dates. (A Julian date is the number of days from the first of the
year and the number of days till the end of the year, as printed on
most desktop calendars.)

Microsoft Access does not have any built-in functionality 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.

More Information:

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: Initial Size of Every MDB File is 64K

Article Number: Q92817
CREATED: 16-NOV-1992
MODIFIED: 11-DEC-1992

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

PRB: "No Default Printer" Error Message When Printing

Article Number: Q92818
CREATED: 16-NOV-1992
MODIFIED: 14-DEC-1992

Summary:

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

PRB: Pressing Home Key in TextBox Does Not Move to Beginning

Article Number: Q92819
CREATED: 16-NOV-1992
MODIFIED: 16-NOV-1992

Summary:

SYMPTOMS

When you are in a textbox control and enter an extremely long amount
of data, pressing the Home key does not move you to the beginning of the data
but does position the cursor at the beginning of the textbox.

CAUSE

Microsoft Access looks at a single line as being up to 20 inches.
If your data excedes this length, or if you have entered a carriage return
(CTRL+ENTER) into the text box, pressing the Home key will not move you
to the beginning of the data. In these cases you can press CTRL+HOME
instead. Both text boxes bound to text fields or memo fields act the same
with regards to Home/End.

References

PRB: A '#' Appears in Place of a '.' When Referring to a Filen

Article Number: Q92820
CREATED: 16-NOV-1992
MODIFIED: 16-NOV-1992

Summary:

SYMPTOMS

   In some error messages which refer to file names, a '#' appears in the
file name in the error message in place of the period such as:

     Can't access file TEST#DBF

RESULT:

   You get the error message 'Can't access file TEST#DBF' instead of 'Can't
access file TEST.DBF'.

CAUSE

   A '.' is not a valid entry in the JET table, and as a result the
translated string that JET provides to Microsoft Access contains a valid
replacement character for the '.', namely the '#'.

RESOLUTION

   This is a known bug in Microsoft Access version 1.0.

References

PRB: Unable to Specify Label Order in Graph Form Wizard

Article Number: Q92821
CREATED: 16-NOV-1992
MODIFIED: 16-DEC-1992

Summary:

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 FormWizard
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
FormWizard 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
   Mar       2
```

3. Use the Graph FormWizard to create a new form based on Table1

The labels on your chart are in alphabetical, not chronological order.

Reference(s):

For more information, search for "Parameter Query," then "GROUP BY clause (SQL)," then "graph labels," then "ColumnHeads" using the Help menu.

[References](#)

PRB: One Side of Join Line Not Attached to Table

Article Number: Q92822
CREATED: 16-NOV-1992
MODIFIED: 16-NOV-1992

Summary:

SYMPTOMS

Join line in the query grid is only attached to one table.

CAUSE

The table is joined to another table that is currently positioned out of view on the query window.

RESOLUTION

Try maximizing query window or using the scroll bars to find the other table, and then move the table closer to the first table.

STATUS

This behavior is by design.

[References](#)

INF: Access Table with Time Field Exports Strangely to Excel

Article Number: Q92823
CREATED: 16-NOV-1992
MODIFIED: 16-NOV-1992

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.

More Information:

Microsoft Access and Excel both use serial numbers to display dates and
time. A Microsoft Access table that is exported to Excel automatically
formats all serial numbers as a date. 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. Select
NUMBER from the FORMAT menu and choose one of the time formats (i.e.:
hh:mm:ss AM/PM). Now the time will display correctly.

Reference(s):

Excel Help, "Serial Numbers", Date and Time Numbers

[References](#)

INF: Custom Setup May Show Zero or Negative Disk Space Needed

Article Number: Q92824
CREATED: 16-NOV-1992
MODIFIED: 19-NOV-1992

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Introduction, "Setting Up and Starting Microsoft Access," pages vi-vii.

References

INF: Using Access Basic to Find Out if a Form Exists in a Data

Article Number: Q92825
CREATED: 16-NOV-1992
MODIFIED: 16-NOV-1992

Summary:

This article discusses how to obtain a dynaset containing the names of all of 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 the VT Object.

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 of 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 which obtains the names of all the objects in MSysObjects that are forms:

```
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 of the forms in the current user database. You can then proceed to 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 = '" & FormTopOpen$ & "'"
   If S.NoMatch Then
      MsgBox FormToOpen$ & " does not exist in this database"
   Else
      DoCmd OpenForm FormToOpen$
   End If
End Sub
```

PRB: MS Access Short and Medium Time Formats Have Leading Zero

Article Number: Q92826
CREATED: 16-NOV-1992
MODIFIED: 16-NOV-1992

Summary:

SYMPTOMS

```
    When using "Short Time" and "Medium Time" Date/Time formats there
will be a leading zero in your data.
```

STATUS

```
    This behavior is by design.
```

More Information:

```
Steps to Reproduce
------------------
1) Create a Date/Time field.

2) Format it as Medium Time.

3) Enter 4:57pm.

4) Tab out of the field.

The value you entered will be displayed as "04:57 PM".

Note: Custom formats are available (i.e. "h:mm am/pm") and can
eliminate the leading zero.
```

[References](#)

PRB: Removing User Admin from Admins Group Can't be Reversed

Article Number: Q92827
CREATED: 16-NOV-1992
MODIFIED: 16-NOV-1992

Summary:

SYMPTOMS

If you start Microsoft Access as the user 'Admin' (which happens 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.

CAUSE

As soon as you remove the user 'Admin' from the Admins group, this change is written to the Microsoft Access system tables. After this the 'Admin' user no longer has permission to add anyone to the Admins group, including itself.

*MSINTERNAL

RESOLUTION

To add the user 'Admin' back to the group Admins, you must logon 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 non-blank password. (NOTE: this is to bring up the logon dialog 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'

5. Exit out of Microsoft Access and restart.

6. You will get the 'Logon' dialog box because of step #1. Logon as the user that you created in step #2.

7. Add the user 'Admin' back to the Admins group.

Note: To disable the 'Logon' dialog box, you will need to change the Admin password back to an empty string (do not enter any information in the New Password or Verify fields).

For more information on Admin

- Search on "Admin account" and "password" using the Help menu.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 25:
'Administering a Database System'.

[References](#)

INF: How to "Freeze" a Record on a Form

Article Number: Q92889
CREATED: 17-NOV-1992
MODIFIED: 4-DEC-1992

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](#)

INF: Mail Label Wizard Shows Blank Line in Label Size ListBox

Article Number: Q92890
CREATED: 17-NOV-1992
MODIFIED: 17-NOV-1992

Summary:

In the Mailing Label Wizard, the LabelSize ListBox 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: Invalid Spreadsheet Range Returns Error in Access

Summary:

SYMPTOMS

When you try to import an invalid range of cells from a spreadsheet,
you will receive the following error message:

    Invalid Range.

Then the 'Select File' dialog box appears.

CAUSE

You specified an invalid range of cells, either improperly referenced
(i.e. R0C0:R5C3) or a non-existing name, to import.

STATUS

This behavior is by design.

RESOLUTION

Re-select the file that you wish to import, and then select Import.
When the 'Import Spreadsheet Options' dialog is displayed, be careful to
enter a valid range reference into the Spreadsheet Range box. This range
can be a specific range of cells (i.e.: A5:B12 or R3C3:R50C9) or a
defined name that refers to a range of cells on the spreadsheet(i.e.:
MYRANGE).

Reference(s):

"Microsoft Access User's Guide," version 1.0, chapter 4, pages 72-75

[References](#)
spreadsheet

PRB: Concatenation of Memo Fields Creates Text Field

Article Number: Q92892
CREATED: 17-NOV-1992
MODIFIED: 4-DEC-1992

Summary:

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](#)

INF: Errors If User Is Disconnected From Server Via NetWatcher

Article Number: Q92893
CREATED: 17-NOV-1992
MODIFIED: 17-NOV-1992

Summary:

If you are connected to a Microsoft Windows for Workgroups server and
you have a database from the server open on your workstation in
Microsoft Access, you may become disconnected by the server and receive
error messages from Microsoft Access. The server can disconnect any
client by using NetWatcher which comes with Windows for Workgroups. If
you have been disconnected you will get two separate error messages when
you attempt to browse a table from within Microsoft Access. The first
message is:

    Internal database error (-1106).

followed by the error message:

    Disk or Network error.

More Information:

The second message is more informative in this case. The client must
reconnect to the server on which the database is stored in order to use
the database.

Steps to Reproduce:
----
1. Connect your machine (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 NetWatcher on the server. (NetWatcher is in the Accessories
   group.) Disconnect the client from the server using NetWatcher
   (Connection-Disconnect).

4. From the client machine, the database still appears to be open.
   Select a table in the database window and attempt to browse the
   table.

You will receive the error message:

    Internal database error (-1106).

After you select OK, you will get the error:

    Disk or Network error.

References

INF: Setup Options Dialog Box Lists Disk Space Info Twice

Article Number: Q92894
CREATED: 17-NOV-1992
MODIFIED: 23-DEC-1992

Summary:

You will get the Setup Options dialog box during a Custom Setup, or
an Express Setup with insufficient disk space for a full
installation. The bottom half of this dialog box provides disk
information (Space Required, Space Available).

The bottom left-hand side provides disk information about the drive that
you specified for Microsoft Access. The bottom right-hand 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-hand side is also displayed on
the right-hand 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 (1) Microsoft Access and Windows on the
same drive, and (2) Microsoft Access and Windows on different drives.
In case (1) above, 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 opens. The bottom
   half of the dialog box lists the disk "Space Required" and "Space
   Available" information twice.

5. Cancel Setup.

*** Steps 6-10 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 opens. 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: CopyObject Appears to Copy Wrong Object

Article Number: Q92895
CREATED: 17-NOV-1992
MODIFIED: 4-DEC-1992

```
----------------------------------------------------------------------
This information in this article applies to:

 - Microsoft Access version 1.0
----------------------------------------------------------------------
```

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.

Reference:

"Microsoft Access Language Reference: Part 2," version 1.0, page 431.

[References](#)

INF: Hourglass in Setup Changes to Pointer During Copy

Article Number: Q92896
CREATED: 17-NOV-1992
MODIFIED: 17-NOV-1992

Summary:

During the Microsoft Access setup when files are being copied the cursor
will change from an hourglass to a pointer if the mouse is moved.

More Information:

If the mouse is moved, causing the hourglass to change to a pointer, it
will not change back to an hourglass until the setup process changes to
another disk.

Note: This is by design and reflects the ability to switch the setup
program to background processing and enabling the installer to continue
working on other things inside the Windows environment.

```
Steps to Reproduce
------------------
```

1. Run SETUP.EXE.

2. Choose Complete Installation. When Microsoft Access begins copying
   the files, move the mouse. The hourglass will change to a pointer.

References

INF: Changing the RowSource of a ListBox/ComboBox at Runtime

Article Number: Q92897
CREATED: 17-NOV-1992
MODIFIED: 4-DEC-1992

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 "Field1", and a
   combo box called "Field2".

2. Add two check box controls to Field1 and specify the following
   property:

       AfterUpdate: =NewOrder(Field1)

3. For Field2, specify the following properties:

       RowSourceType: Table/Query
       RowSource: DeleteMe
       ColumnCount: 2

4. Add the following function to a module in the database:

```
Option Explicit
Function NewOrder (WhichSort%)
   DoCmd Hourglass True
   Dim D As Database, Q As QueryDef
   Set D = CurrentDB()

   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 "Field0"
        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: Double-Clicking on a Control Moves the Control Unintentio

Article Number: Q92898
CREATED: 17-NOV-1992
MODIFIED: 17-NOV-1992

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...

* Rather than double-click on the control to get the property sheet, use
  the TAB key to move to it. 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: Requery Action Parameter Requires No Brackets

Article Number: Q92899
CREATED: 17-NOV-1992
MODIFIED: 17-NOV-1992

Summary:

When trying 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. Help informs the user that they should not use the full syntax, but nothing is mentioned about the requirement of omitting brackets even if the control name consists of more than one word. Because other scenarios require the use of brackets, this is an inconsistency which leads to confusion once the error message is encountered.

Note the following example which reproduces the problem and discusses the solution:

1. Create an unbound form called Form1

2. Place a ComboBox bound to NWIND.MDB 'Sales Totals Query' on the form and call the ComboBox 'Field1'

3. Create a macro called Macro1 with the following Action/Parameter...

        Action   : Requery
        Control Name : [Field1]

        ...note that [Field1] is surrounded with brackets

4. Save and close Macro1 and drag it on to Form1 in design mode

5. Switch to Browse Mode on the form and push the button

   You get the following error message:

        There is no control called '[Field1]'

6. Remove the brackets from the Control Name parameter in Macro1 and try again. Notice that it now works properly.

References

INF: Unexpected Behavior with Design Button on Toolbar

Article Number: Q92949
CREATED: 19-NOV-1992
MODIFIED: 19-NOV-1992

Summary:

While in Form View, pressing the 'Design' button on the
toolbar will not return you to Design Mode if the
form's OnClose property is set to open another form.

More Information:

Steps To Reproduce Behavior
----

1. Create 2 new forms named Form1 and Form2.

2. Create a macro named TestMacro with two actions. These actions will
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 TextMacro you created in step 2.

```
   Object: Form
   --------------------------
   OnClose: TestMacro
```

4. Close and save Form1.

5. Select Form1 from the Database window and press the Design
button.

6. Select the View Form button on the toolbar. (The form view is
displayed.)

7. Now press the Design button on the toolbar.

You would expect to be returned to design mode, instead Microsoft

Access loads Form2.

[References](#)

INF: Bound OLE Object is Blank for Invalid Source in Form View

Article Number: Q92950
CREATED: 19-NOV-1992
MODIFIED: 19-NOV-1992

Summary:

SYMPTOMS

   If a Bound Object Frame on a form has the property Control Source
set to a field that is not of type OLE Object, this control will be
blank when the form is in Form view. In Datasheet view the cells for
this control contains 'Invalid Object'.

CAUSE

   A Bound Object Frame must refer to a field of type OLE Object in a
table/query in order to display anything in Form view. To display an
object in this control, the Control Source property must be set to a
an OLE Object Field. For example, Microsoft Access cannot display the
contents of a Text field as an OLE Object. Object frames will show
blank unless there is something to show.

STATUS

   This is by design.

[References](#)

PRB: Undo Property Setting Appears In Macro Design

Summary:

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, then select 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'. However,
functionality of the UnDo command is unaffected.

STATUS

   This behavior is by design.

More Information:

```
Steps to Reproduce Problem
--------------------------
```

1. Create a new macro.

2. In the Macro Action column type Close and press ENTER.

3. Select the Edit menu. Notice the 'Undo Property Setting' option.

4. Select the 'Undo Property Setting' option. The Close that you typed
   has been erased.

[References](#)

PRB: Import/Export Setup Cancel Button Cancels Unsaved Changes

Article Number: Q93022
CREATED: 22-NOV-1992
MODIFIED: 23-DEC-1992

Summary:

SYMPTOMS

The Cancel button in the Import/Export Setup dialog, cancels only
those changes which have not been saved. It does not cancel all
changes made since entering the dialog.

CAUSE

If you make a change to the specification and choose the Save As
button, choosing the Cancel button does not invalidate those changes.
The Cancel button cancels only those changes which have not been
saved and returns the user to the Database window.

The OK button does a "save and exit", and the Save As button saves
but does not exit the dialog.

STATUS

This behavior is by design.

[References](#)

INF: No Custom Printer Setup for Mailing Labels Report Wizard

Article Number: Q93023
CREATED: 22-NOV-1992
MODIFIED: 22-NOV-1992

Summary:

The Mailing Labels Report Wizard allows you to choose from 66 different labels. However, the Report Wizard does not verify whether the labels you select will correctly fit onto the paper size that you have specified.

You may need to manually customize your printer setup and/or number of labels per page after using the Report Wizard.

More Information:

The need to customize printer setup is most common with Dot Matrix printers. Label sizes other than 1/2" or 1" may not fit evenly onto the default 11" page.

To manually customize your printer setup and/or number of labels per page:

1. Open your report in Print Preview view.

2. From the File menu, choose Print Setup.

3. Customize your margins to accommodate number and/or width of labels

4. Press the "More>>" button in the Print Setup dialog box to customize the number of rows of labels.

References

INF: Default Auto Label and Add Colon Properties

Article Number: Q93024
CREATED: 22-NOV-1992
MODIFIED: 22-NOV-1992

Summary:

The following is a list of the default values for the Auto Label and
Add Colon properties for the different Form and Report objects.

```
   Toolbox Object          AutoLabel     AddColon
   ---------------         ------------------------
   Textbox                   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
```

Reference(s):
"Microsoft Access User's Guide," version 1.0, chapter 9, "Designing
Forms" and chapter 18, "Designing Reports"

[References](#)

INF: Using DLookup to return values to display in forms or rep

Article Number: Q93025
CREATED: 22-NOV-1992
MODIFIED: 05-JAN-1993

Summary:

Frequently, you will want to display multiple fields in a form or
report which are from a table or query other than the one which
your form or report is bound to.

There are several ways to accomplish this:

1) With Forms you can utilize Row Fix-up. For more information
   on Row Fix-up search on the keywords "Row Fix-up and multiple
   and One to Many"

2) By creating multiple DLookups for Forms or Reports (see below)

3) Using the Column property of a Combo Box in either Forms
   or Reports. For information on using the Column property for
   Forms or Reports search on the keywords "Column and Property
   and Display".

This article contains information on using the function:

    DLookup(expr, domain [, criteria])

More Information:

The DLookup Function returns a value from a field in a specified
set of records (also refered to as the domain). The DLookup
function allows you to specify the criteria for the domain. To
make the domain dependant on values in controls on your Form or
Report, refer to them in the criteria.

The following example returns the Last Name of the Employee ID
specified in another control on the Form.

=DLookup("[Last Name]", "Employees", "[Employee ID] =
          Forms!FormName![ControlName]")

Notes:
    1) This example is split into two lines for readability.
   It should be entered as one formula.

    2) For reports, use Reports!ReportName![ControlName].

Even if more than one record satisfies criteria, DLookup returns
only one field. If no record satisfies the criteria, or if the
domain contains no records, DLookup returns Null.

Reference(s):

Search Help for DLookup, Help Topic title "DLookup function"

[References](#)
Combo Column

INF: Users Can Edit but not Commit Read-Only OLE Objects

Article Number: Q93026
CREATED: 22-NOV-1992
MODIFIED: 22-NOV-1992

Summary:

Users are allowed to edit OLE objects in a database which has been
opened read-only, or in a Form that is read-only. However when the
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.

More Information:

In a read-only database, the user is still allowed to edit and update
OLE objects, but they are not allowed to commit a record with these
changes. At the point the record a committed, an alert 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, copy the object to the clipboard before the undo.

To prevent users from attempting to edit OLE objects in a read-only
form set the Enabled property of the Object Frame to No.

References

INF: Wizards Part III - DeleteControl, DeleteReportControl,

Article Number: Q93094
CREATED: 23-NOV-1992
MODIFIED: 23-NOV-1992

Summary:

This article discusses the Microsoft Access undocumented commands
DeleteControl, DeleteReportControl, and the undocumented function
CreateGroupLevel(). If you intend to write your own Forms and/or
Reports Wizard, you can use these functions to delete controls on a form
or report that is available in Design mode, and create groups on a
Report that is available in Design mode.

For more understanding of Microsoft Access Libraries, search
on the Knowledge Base for the following keywords:

    libraries AND debugging AND creating

For more information about other Wizard functions, query the Knowledge Base
for:

    CreateControl AND CreateReportControl AND CreateForm AND CreateReport AND
       'Section Property' AND DeleteControl AND DeleteReportControl AND
          CreateGroupLevel

More Information:

DeleteControl and DeleteReportControl can be used to delete a control
that exists on a form/report in design mode. Both of these functions
require a string value that represents the name of the form 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, then 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/report
is not iconized. The control still appears as if it were not deleted
even though you could not click it into focus. To make the deleted
control disappear in such a case, you would have to repaint the form
design window by minimizing and restoring it.

If you have a report available in design mode, you can create groups
for it programmatically by using the CreateGroupLevel() function. The

syntactical structure of CreateGroupLevel() is shown below:

```
Function CreateGroupLevel(ReportName$,Expression$,
                          HasHeader$, HasFooter$)
```

ReportName is a string expression that indicates the name of the Report.

Expression is a string expression that represents the field name or calculated field on which the group will be based.

HasHeader and HasFooter indicate whether the group includes a group header or group footer, respectively.

When CreateGroupLevel() is executed, it adds a group at the innermost level. For example, if one group already exists, CreateGroupLevel() will create 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() would return 1 since it is the second group, and the first group was Group 0.

References

INF: Wizards Part II - CreateControl(), CreateReportControl()

Summary:

This article discusses the Microsoft Access undocumented functions
CreateControl() and CreateReportControl(). If you intend to write your
own Forms and/or Reports Wizard, you can use these functions to create
controls on a form or report that is available in Design mode.

For more understanding of Microsoft Access Libraries, search
on the Knowledge Base for the following keywords:

    libraries AND debugging AND creating

For more information about other Wizard functions, query the Knowledge Base
for:

    CreateControl AND CreateReportControl AND CreateForm AND CreateReport AND
      'Section Property' AND DeleteControl AND DeleteReportControl AND
        CreateGroupLevel

This article also assumes you are familiar with Access Basic and with
designing forms and reports.

More Information:

CreateControl() and CreateReportControl() can be used to create new
controls for a form/report opened in Design mode. These functions
require the name of the form/report represented as a string value,
and a numeric code that represents the type of control. The list of
control types and their associated numeric codes are shown below:

    Label 100
    Box 101
    Line 102
    Picture 103
    Button 104
    Radio Button ... 105
    Check Box 106
    Ole Object 108
    Text Box 109

The CreateControl() and CreateReportControl() functions return a
control object value, so you must first define a control variable,
then Set the variable to the function name. Note the code example
below which creates a form, then adds a button to the form:

    Dim MyForm As Form, MyControl As Control
    Set MyForm = CreateForm()
    Set MyControl = CreateControl(MyForm.FormName, 104)

After executing the code example above, you can modify the properies

of the new control via the control variable you defined. For example:

```
MyControl.Width = 2000
MyControl.Caption = "&Sum All Records"
```

For controls that are frequently associated with fields in a table or query, you can modify the ControlSource property to bind the control. Some controls are created with Height and Width properties set to 0, so that they are - in effect - not visible. In addition, controls appear at the uppermost left-hand corner of the form by default. Because of this, you should make it a habit to adjust the size and position of the control immediately after creating it. The example below shows how to create, size, move, and bind a text box.

```
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 specifying the form name and type of control, you can optionally specify the Section in which the control should be created. Note the syntactical structure of CreateControl(), (which is identical to the syntactical structure of CreateReportControl()):

```
Function CreateControl (FormName As String,
                        ControlType As Integer
                        [,SectionNumber As Integer])
```

Use the SectionNumber parameter to indicate what section the control is to be placed into. The section that is associated with the SectionNumber is the same as the index representing a section in a Section property array.

[References](#)

INF: Using Undocumented CreateForm(),CreateReport() Functions

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
```

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 about other ReportWizard and FormWizard
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](#)

PRB: Creating Multiple New Import/Export Specifications

Article Number: Q93136
CREATED: 24-NOV-1992
MODIFIED: 23-DEC-1992

Summary:

SYMPTOMS

Creating multiple new Import/Export specifications using the
Import/Export Setup dialog requires that you either exit the dialog or
deleting 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. Select File, Imp/Exp Setup.

3. Enter information in several fields.

4. Save the specification using Save As option.

Result: If you wanted to create another specification at this point,
you must either exit this dialog or manually delete all of the
irrelevant information.

[References](#)

INF: Field Size of 0 Treated as 255

Article Number: Q93137
CREATED: 24-NOV-1992
MODIFIED: 24-NOV-1992

Summary:

If you set a field size to zero(0) for a text field in table design
mode, the field is treated as if the size is set at 255. When you
close and save the table, then reopen it in design mode, the field
size will have changed to 255.

[References](#)

INF: Using Column() for Multiple Fields in Forms/Reports.

Article Number: Q93138
CREATED: 24-NOV-1992
MODIFIED: 24-NOV-1992

Summary:

Frequently, you will want to display multiple fields in a
form or report which are from a table other than the one the
form or report which it is bound. Or you may want to have several
controls updated with new information based on a combo/list box
selection.

There are several ways to accomplish this:

1. With Forms you can utilize Row Fix-up. For more
   information on Row Fix-up query in the Knowledge Base for the
   following keywords:

       "Row Fix-up" and "multiple" and "One to Many"

2. With multiple DLookup() functions in both Forms and
   Reports. For more information query in the Knowledge Base
   for the following keywords:

       DLookup and display and Forms

3. Using the Column property of a combo box in either Forms
   or Reports. (see below)

This article contains information on using the Column()
property of a Combo Box:

       =[ComboControlName].Column(1)

More Information:

1. Create your form or report based the desired table or
   query.

2. Create a combo or list box which retrieves more than one field.
   Below is an example of a multi-column Select statement
   which can be used as the RowSource for a combo/list box to
   display information from the Categories table.

       Select [Category Id], [Category Name], [Description] from
          Categories Order by [Category Name];

   In this example, you will need to set the RowCount to 3.
   Depending on the size of the combo box or list box that you
   generated, you will need to set the ColumnWidths appropriately.
   Once you have a combo/list box defined to return multiple fields
   you can create text boxes to quickly display information
   which the combo/list box has retrieved by using the Column()
   property.

The Column() property refers to a particular column in a
multiple column combo/list box. Use 0 to refer to
the first column, 1 to refer to the second column, and so
on. The following example uses the Column property to refer
to the second column, [Category Name], in the combo/list box
described above.

3. To display the [Category Name] of the category currently
selected in the combo/list box created in Step #2, create a text
box and place the following formula in it:

    =[ComboControlName].Column(1)

NOTE: The Column property will make the text box control
read only.

Reference(s): For more information on how to set up a combo
box see the "User's Guide", version 1.0, Chapter 9, pages 233-245.

Also search Help for Column Property

References

PRB: Inaccurate Message When Opening MDB Shared vs Exclusive

Article Number: Q93139
CREATED: 24-NOV-1992
MODIFIED: 24-NOV-1992

Summary:

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 since 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](#)

PRB: Cannot Rollback Without Closing Database

Article Number: Q93140
CREATED: 24-NOV-1992
MODIFIED: 24-NOV-1992

Summary:

SYMPTOMS

   A subroutine or function has a transaction process, and inside of this
transaction, a sub or function is called that dimensions and opens a database

with the OpenDatabase method. On exiting the subroutine of function, the
database is implicitly closes (there is no Close method called). When the
CommitTrans or Rollback statements are encountered the following error
message is displayed:

      Commit or Rollback without BeginTrans.

CAUSE

   When the called function or subroutine is exited, the database is
implicitly close. Internally, the initial attempt at this close fails because
the database was opened within a transaction. Microsoft Access sees this
error and does an implicit rollback, then closes the database successfully.

RESOLUTION

   To perform a Rollback in this situation, you must first close the
open database with the Close statement from within the called subroutine of
function.

STATUS

   This behavior is by design.

More Information:

```
Steps to Reproduce Problem
--------------------------
1. Create the following module.

    '***********************************************************
    'Declarations section of the module.
    '***********************************************************

    Option Explicit

    '===========================================================
    'The following Sub procedure will start a transaction, call
    'another sub procedure, then attempt to rollback.
    '===========================================================
    Sub Test1
       BeginTrans
          Call Test2
       Rollback
    End Sub
```

```
        '===========================================================
        'The following Sub procedure will set 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. You will receive the error
   message:

        Commit or Rollback without BeginTrans.

The following is the solution:

1. Follow the above steps with one change. You will add MyDB.Close to
   the second function.

```
        '===========================================================
        'The following Sub procedure will set a Database variable to
        'the current database, and then close the database.
        '===========================================================
        Sub Test2
            Dim MyDB as Database
            Set MyDB = CurrentDB()
            MyDB.Close
        End Sub
```

2. Compile the module.

3. In the Immediate window enter Test1.

Result: The Rollback works because the database has been closed explicitly.

[References](#)

INF: Passing a Single-Byte of Data Using API Calls

Article Number: Q93141
CREATED: 24-NOV-1992
MODIFIED: 24-NOV-1992

Summary:

Microsoft Access does not have a single-byte data type. If you want to
pass a byte to an external function such as a Windows API or DLL, you
should declare the byte as an integer.

If you need to pass a single-byte of information in a data type, like
the RGBQUAD, you should pass the data structure as a string type.

More Information:

The Intel chip will only support a full word on the stack. You cannot
just put 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 and an additional 8-bits that will be 'padded' on). This
is a limitation of the processor, not of Microsoft Access.

To pass a single-bytes of data in a structure, like the RGBQUAD, use a fixed
length string of one. For given the 'C' RGBQUAD data structure:

```
   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)
```

In the above example Blue=10, Red=14, and Green=15.

For more information, search for "Type Statement" using the Help menu.

References

PRB: Option Button Has Strange Display in Data Sheet Mode

Article Number: Q93142
CREATED: 24-NOV-1992
MODIFIED: 24-NOV-1992

Summary:

SYMPTOMS

If you create a form and place an option button on the form, then switch to Datasheet mode, the outline of the option button will appear fuzzy. This happens when the focus is on any option button in Datasheet mode.

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 Tool Box select the Option Button and place it on the form.

3. Switch to Datasheet mode.

4. Make sure the focus is on the button. Notice the outline of the button has a fuzzy appearance.

[References](#)

INF: On-Line Help Incorrect for Sigma Button

Article Number: Q93143
CREATED: 24-NOV-1992
MODIFIED: 24-NOV-1992

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 Query Design window Toolbar 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 Module Design window Toolbar.

References

INF: Microsoft Access is Pen Compatible but not Pen Aware

Article Number: Q93144
CREATED: 24-NOV-1992
MODIFIED: 9-DEC-1992

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: How Access Handles Logins to Attached SQL Tables

Article Number: Q93145
CREATED: 24-NOV-1992
MODIFIED: 9-DEC-1992

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.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 60-62, 66-68

References

PRACC9211xxx: Err Msg: Data has changed operation aborted

Article Number: Q93146
CREATED: 24-NOV-1992
MODIFIED: 24-NOV-1992

Summary:

PROBLEM ID: PRACC9211xxx

SYMPTOMS

Attempting to commit a record to an attached Paradox table
which has been changed by another user results in the
following error message:

   Data has changed; operation aborted.

After the OK button is pressed 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.

Selecting Save Record causes the first error message above
to repeat. At this point the user is in a continuous loop
and will not be able to commit the record.

CAUSE

```
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.

RESOLUTION

To save the changes made to the current record, User1 must
select the option Copy to Clipboard. Then select the
record they were attempting to change, choose Paste from
the Edit menu, and commit the record.

To prevent User2 from changing a record while User1 is
editing it, User1 may change their Multiuser options to
Edited Records.

STATUS

Microsoft has confirmed this to be a problem with version
1.00 of Microsoft Access. Microsoft is researching this
problem and will post new information here when it becomes
available.

[References](#)

PRB: Access Forms May Open And Be Totally Blank

Article Number: Q93261
CREATED: 3-DEC-1992
MODIFIED: 3-DEC-1992

Summary:

SYMPTOMS

A form opened in View Mode displays totally blank. There are no
controls displayed or text, everything is blank.

CAUSE

There are several conditions that can cause a form opened in View
Mode to display totally blank. Basically the reason is that the
dynaset contains no records or privileges have been limited.

RESOLUTION

Although you cannot display any fields in the forms detial section,
you are able to display necessary controls such as command buttons,
and check boxes on the form header or footer. Controls that are
displayed in the form header or footer are always visible. You can
also make sure that there is at least one record in the dynaset.

STATUS

This is by design.

More Information:

Situations in which a completely blank form will be displayed are when
the forms are based on:

1. Query with a one to many join and the query returns no records.

2. The form is not updatable at all.
   a) read-only (AllowUpdates = No Tables)
   b) static (DISTINCT, GROUP BY, etc.)

3. The query is a one-to-one join, and one of the tables is in a read-
   only database.

4. The query is a many-to-one inner join, and you haven't output all
   the joined columns of the many side.

5. You have not output all the columns of the primary key of an
   updateable remote table.

6. When a form is based on an empty table and the form is set to read
   only the form will display as blank.

Steps to Reproduce Situation 1
----
1. Change the Catalog query in Nwind to have IS NULL placed in the
   criteria cell for the field [Category Name].

2. Now create a new form based on this query and place some fields in
   the detail section, along with a command button.

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 this 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 only way a form can display properly is in add mode but,
the setting for the form or the joins in the query prevent the form
from opening in add mode.

To work around this issue place your necessary controls or command buttons in
the headers and footers of the form. These sections of the form will
display even when the detail section is blank.

References

PRB: Can Close Dynaset Based on Form Multiple Times

Article Number: Q93292
CREATED: 3-DEC-1992
MODIFIED: 4-DEC-1992

Summary:

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 Foos()
      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

PRACC9212xxx: Unable to use SQL as a Report's Record Source

Article Number: Q93293
CREATED: 3-DEC-1992
MODIFIED: 3-DEC-1992

Summary:

PROBLEM ID: PRACC9212XXX

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.

More Information:

Steps to Reproduce Behavior
----

1. Open the NWIND.MDB database.

2. Open the Category List query in Design mode.

3. To bring up the SQL statement for this query, select 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 Record Source property (using CTRL+V).

7. Switch the View to Form View.

8. Open a new unbound report, and paste the SQL statement into the
   report's Record Source property (using CTRL+V).

9. Select the Print Preview button on the toolbar.

You will receive the above error message on step #9, but not for step
#7. You can successfully use a valid SQL statement as the Record
Source in a form, but not in a report.

References

INF: All Duplicate Macro Names in Macro Group are Executed

Article Number: Q93294
CREATED: 3-DEC-1992
MODIFIED: 3-DEC-1992

Summary:

When you have two macros in the same macro group with the same name,
both macros will be executed. This will not happen if the macros are
not consecutive in the marco group (ie. 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

4. The MsgBox will appear twice.
```

[References](#)

INF: ERR MSG: Couldn't Open File '<database name>'.

Article Number: Q93295
CREATED: 3-DEC-1992
MODIFIED: 3-DEC-1992

Summary:

SYMPTOMS

Attempting to open a database exclusively when another user already has
it open will result 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." This error and help text is slightly
misleading because the problem is really that the file cannot be opened
exclusively.

To open the file successfully:

1. Press ESC or the OK button to exit the error message.

2. Choose Open Database... from the File menu.

3. Select the database you wish to open.

4. Unselect the Exclusive checkbox.

5. Press OK.

STATUS

Microsoft has confirmed this behavior in Microsoft Access version 1.0.

References

INF: View Procedures Sorts Modules Alphabetically

Article Number: Q93296
CREATED: 3-DEC-1992
MODIFIED: 3-DEC-1992

Summary:

When you select Procedures from the View menu after opening
a Module, the Modules box displays 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 3 modules. (Module1, Module2, Module19)

2. Open Module1 and Select Procedures from the View menu.

3. The modules are listed Module1, Module19, and Module2 since they
   are sorted Alphabetically.

To have the list properly sorted you should rename Module1 and Module2
to Module01 and Module02. The modules will appear in the list sorted
in order as Module01, Module02, and Module19. Please note that the modules
are sorted the same way in the Database container.

[References](#)

INF: Removing Attached Tables in Access Basic Code

Article Number: Q93297
CREATED: 3-DEC-1992
MODIFIED: 3-DEC-1992

Summary:

The TransferDatabase action is the Macro equivilent to the File menu Attach Tables command. However, there is no action to unattach a table. You can remove an attached table from the User Interface by selecting the table from the DataBase window, and then pressing the delete key. You can create a macro to unattach a table or call the action from a Access Basic module using the DoCmd statement.

More Information:

Create the following macro to unattach a table in Microsoft Access.

```
   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 Microsoft Access Basic you must use DoCmd to run the Macro actions.

[References](#)

INF: Changing Read-Only Properties at Runtime

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:

This is accomlished by creating two controls that are identical except
for the property you wish 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 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
                    SetValue       Item ........ [RedButton].Visible
                                   Expression ... True
                    SetValue       Item ........ [BlueButton].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.

Err Msg: Too Many Expressions in Distinct Output

Article Number: Q93299
CREATED: 3-DEC-1992
MODIFIED: 3-DEC-1992

Summary:

If you create a SQL statement in the SQL dialog or Access Basic that
has a DISTINCT clause with more than ten expressions, the following
error message will appear:

    Too many expressions in DISTINCT output.

The ten field limit for the DISTINCT clause is stated as the reason
for the error in help. However, this is the only place that states
there is such a limit. Help refers the user to Appendix B, in the
Language Reference, where there is no mention of a ten field limit for
the DISTINCT clause.

More Information:

There are the two options to eliminate this error:

1. Remove the reserved word, DISTINCT;

2. Reduce the number of fields in the DISTINCT clause to ten or less.

A possible work around is to concatenate some of the fields together
into an expression when checking for DISTINCT rows. Each concatenation
will equal one DISTINCT expression.

Example: Try using the select statement,

    SELECT DISTINCT
        table1.lname & table1.fname,
        table1.address,...;

instead of

    SELECT DISTINCT
        table1.lname,
        table1.fname,
        table1.address,...;

Utilizing this technique will reduce the number of expressions in the
DISTINCT clause.

References

INF: Using Data From Access In A Word For Windows Print Merge

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 which 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 described 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.

Reference(s):

"Microsoft Word for Windows User's Guide," version 2.0, pages 620-622

[References](#)
delimited

PRB: ErrMsg: Cannot Write Readme.txt

Article Number: Q93590
CREATED: 16-DEC-1992
MODIFIED: 16-DEC-1992

Summary:

SYMPTOMS

There are several possible workarounds that may solve a setup problem when Microsoft Access is writing the README.TXT file. The following error may occur on disk 6 (or disk 7) during setup:

Cannot Write C:\MSACCESS\Readme.txt".

STATUS

Microsoft is researching this problem and will post new information here as it becomes available.

More Information:

In all cases standard troubleshooting techniques should be followed. This includes a clean boot and a standard Windows configuration.

The DOS APPEND command, located in the AUTOEXEC.BAT will reproduce this error everytime if there is a README.TXT file in any of the directories that are specified on the APPEND line of your AUTOEXEC.BAT file.

When SETUP.EXE tries to install README.TXT, the APPEND command can cause DOS to tell SETUP.EXE that there is a file with the same name in the installation directory. When SETUP.EXE is notified by DOS that there is a file with the same name, SETUP will try to overwrite this file. DOS will return an error code that this file is write protected.

Other TSR's such Quicken's BillMind, Logitech's Click and Logimenu have been reported as a problem with installing the README.TXT file.

Workaround 1
----

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 2
----

1. Change to the directory of the path given in the error message.

2. Use edit (or some other text editor) and create the file
   README.TXT. The file does not have to have any contents.

```
4. Reinstall Access from the original disks.


Workaround 3
------------

1. Make a duplicate copy of disk #6 (or disk #7).

2. On the new disk copy README.TX$ file to README.2

3. On the new disk rename README.TX$ to README.1. Then rename README.2
   to README.TX$.

4. Reinstall Microsoft Access using the new disk you made.


Workaround 4
------------

1. Make a duplicate copy of disk #6 (or disk #7).

2. Rename README.TX$ to README.1 on the new disk.

3. Create a new file on the new disk using EDIT and save as
   README.TX$.

4. Reinstall Microsoft Access using the new disk you made.
```

References

INF: Importing Keyed Paradox Tables Requires .PX File

Article Number: Q93640
CREATED: 17-DEC-1992
MODIFIED: 17-DEC-1992

Summary:

When you import a keyed Paradox table into Microsoft Access the associated index file must be in the same directory as the Paradox table. If the index (.PX) file is not found you will receive the following error message:

    'FILENAME#DB' is corrupted or isn't a Microsoft database.

More Information:

If you're importing a Paradox file, this message might be displayed if you don't have an index (.PX) file. Paradox stores important information about a table's primary key in the .PX file. Make sure you have the .PX file before importing.

If a Paradox table and it's associated index file become separated, or the .PX file is corrupt, Access will be unable to import the table. To recreate the index file the table must be Restructured in Paradox.

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: Microsoft Access Performance Enhancement Recommendations

Article Number: Q93684
CREATED: 17-DEC-1992
MODIFIED: 05-JAN-1993

Summary:

If you run Microsoft Access on a 4 MB 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 4 MB 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. Don't run several other applications that require large amounts of
    memory. Running even one other large application (such as a
    spreadsheet, wordprocessor, 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 for 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 don't want to use Access Wizards, you can disable them by
    removing the line "WIZARDS.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 4 MB 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 4 MB 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](#)

INF:"Couldn't find Object <tablename>" with Btrieve files

Article Number: Q93685
CREATED: 17-DEC-1992
MODIFIED: 17-DEC-1992

Summary:

A number of people have reported receiving the error message

    Couldn't find object '<table name.'

when attempting to import or attach to Btrieve files. This
article explains how to identify and fix this problem.

More Information:

Microsoft Access requires the Xtrieve-format data dictionary files,
FILE.DDF, INDEX.DDF, and FIELDS.DDF, to be able to use Btrieve
files. FILE.DDF contains the DOS file name of each data table
in the dictionary. These file names can be in any of the valid four
DOS formats:

    1. File Only ("mydata.dat")

    2. Relative Path ("..\dev\test\mydata.dat")

    3. Fully Qualified ("c:\tools\dev\test\mydata.dat")

    4. Universal Naming Convention, UNC
          ("\\myserver\users\joe\tools\dev\test\mydata.dat")

When importing or attaching to a Btrieve file, Microsoft Access
retrieves the DOS filename from FILE.DDF. It then attempts to open
this file. If the file doesn't exist in the location specified,
you will receive the error

    Couldn't find object

In the case of the first 2 examples, the ability to open the
file is dependant on the current directory being correct. For
example #1, MYDATA.DAT must exist in the current directory.
For example #2, MYDATA.DAT, the "..\DEV\TEST" directory must
exist relative to the current directory, and MYDATA.DAT must
exist in it.

Exactly which directory is the current directory is complicated
by the fact that under Windows, all applications share the same
environment, and therefore the same current directory. If you
change the current directory in Excel, for example by doing a
File Open and changing directories to open a file, the current
directory for Microsoft Access is also changed.

So, the first step to fixing the problem is to find out what the
paths are in FILE.DDF. You can do this by opening FILE.DDF in
WRITE, the word processor provided with Windows. When asked if

you want to convert this file to WRITE format, specify No
Conversion. You will see a bunch of characters that look like a
box. Scroll to the end of the file. There you will see the DOS
filenames of your Btrieve data files. The extension is usually
.DAT, but doesn't have to be.

WARNING: DO NOT save the file after you are finished viewing
it with WRITE. If you save this file, it will become corrupt and
unrepairable.

Now that you have the file names, there are 2 things you can do
to fix the problem

1. Move the files into the correct location(s). This is the
   easiest fix, and the one we recommend.

2. Using Btrieve or one of the third-party and/or shareware
   tools, modify FILE.DDF so that the path to the .DAT file is
   correct. This method is much more complicated, and not
   recommended for anyone not intimately familiar with Btrieve.

Reference(s):

References

INF: Out of Memory Message When Opening A Form

Article Number: Q93686
CREATED: 17-DEC-1992
MODIFIED: 17-DEC-1992

Summary:

Any form or report which performs a calculation in the
Control Source property for any control is limited to
returning 64K of data.

More Information:

One example is using DLookups in the control source for
each of these OLE objects you are limited to displaying
64K of data on the form. When the form is full-screen
it calculates until it gets to this limit and then gives
you the "Out Of Memory" alert for each control it does not
have the memory to complete.

Someone called this non-destructive because you don't hang
and many times your form with display anyway - it just can't
calculate completely.

You should redesign the form and use macros to SetValue
controls which return large objects such as memos or OLE
objects.

Reference(s):

References

INF: Btrieve 6.0 Data Type Does Not Work with Access

Article Number: Q93687
CREATED: 17-DEC-1992
MODIFIED: 18-DEC-1992

Summary:

Btrieve version 6.0 is a specialized version of Btrieve specifically designed for Macola Accounting software on a Novell file server. Microsoft Access was not tested with this custom, specialized version of Btrieve.

More Information:

Btrieve 6.0 introduced a new data type: Negative Sign Trailing Numeric Field. Microsoft Access does not explicitly translate this data type.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 4, page 66

[References](#)

INF: Exporting a Btrieve Table W/O an Existing Xtrieve FILE.DD

Article Number: Q93688
CREATED: 17-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

The Microsoft Access user interface requires that the Xtrieve
dictionary file (FILE.DDF) exists before you can export a Btrieve
table. If FILE.DDF does not exist, the macro action Transfer
Database can be used to export the table.

More Information:

Multiple Btrieve tables can exist in Xtrieve databases just like
Microsoft Access can have multiple Tables. The dictionary FILE.DDF
describes the structure of the Xtrieve database.

The following example Macro can be used to transfer the Customers
Table in the sample database NWIND.MDB to a Btrieve Table named
Customers in an Xtrieve database named c:\access\btv00000.dat. This
Export occurs even if btv00000.dat does not exist as long as c:\access
is a valid directory.

```
    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

    NOTE: THE DATABASE NAME CAN BE ANY COMPLETE FILE SPECIFICATION.
```

Microsoft Access will create the following 4 files if they do not
already exist:

```
    FILE.DDF
    INDEX.DDF
    FIELD.DDF
    BTV00000.DAT
```

Transferring more than one table will add data to the existing files,
but it will not create more files as long as the Database Name is not
changed.

Reference(s):

"Microsoft Access User's Guide," version 1.0, page 86

[References](#)

INF: Troubleshooting GP Faults with ATI Cards

Article Number: Q93689
CREATED: 17-DEC-1992
MODIFIED: 17-DEC-1992

Summary:

We are receiving a high volume of calls from customers who report
GP Faults and Setup problems on systems that have ATI Video Cards,
especially ATI Graphic Ultra Pro Local Bus cards.

We have only been able to reproduce one GP fault and one display
problem consistently. Many of the engineers in the Microsoft Access
group have workstations with ATI cards and are not having problems.
It is VERY important that you do not allow the customer to think that
we do not work with these video cards. In fact, Windows 3.1 is
shipped with a driver for one of the ATI cards and we provide others
via MSDL.

This article includes:
  Troubleshooting Steps
  References to other articles with related information

If you have a customer with one of these graphics cards and you
either solve the problem using the steps described below or have
a customer with a configuration or problem that is not mentioned
please comment this article.

More Information

Troubleshooting Steps
==================
1. Boot Clean.

2. Change the Windows setup to standard VGA

If these two steps do not correct the problem and it is not a known
issue as documented in one of the following articles, have the
customer reinstall Microsoft Access while booted clean. If the problem
persists then it is unlikely that it is related to the ATI Card.
Create a callback for the customer. If the problem is reproducible
send it to Brian Blackman for reporting. If not, escalate it so
that it can be forwarded to program management for further testing.

Reference To Other Articles
====================
Q90121 [B_WACCESS]: Access GP Fault with ATI Ultra Video Card
Q92701 [B_WACCESS]: Disabled Buttons Not Visible w/ATI 1280x1024
Q85166 [W_WIN3]: ATI 8514 Ultra Hangs Windows 3.1 in 800 x 600 Mode
Q87082 [W_WIN3]: Display Problems Solved by the Latest WDL Drivers

Additional References:
Information provided on the MSACCESS forum on Compuserve by a

```
customer:
  "Thanks for the suggestion.  I'm quite sure it's the ATI mach32
   drivers. I have just received a reply from ATI on their forum:
   'We are looking into some issues of MS Access not working well
   with the ATI Mach 32 Windows drivers.  Our Engineering department
   is investigating this, and we will have a resolution soon.'"

Reference(s):
```

[References](#)

INF: How To Update a Table Based on Values in a Second Table

Article Number: Q93690
CREATED: 17-DEC-1992
MODIFIED: 18-DEC-1992

Summary:

To transfer data between two different tables, create an additional
column in one of the tables which contains data that the two tables
have in common, then use an Update query to initiate the transfer.

More Information:

This can be illustrated by an example using the sample database
Northwind Traders (NWIND.MDB). This example adds a product name column
to the order details table, and then copies the product name from the
products table based on the product ID column.

Note: normally you would not duplicate the product name field because it
would result in duplicate data, which is not good database design. It is
done here as a simple illustration of an Update query involving two tables.

1. Modify the order details table:

    a. Open the order details table in design view.

    b. Add a column named product name, using the default data type of Text.

    c. Close the order details table, saving the changes.

2. Create the query:

    a. Create a new update query.

    b. Add the order details and product tables to the query. (The tables are
       automatically joined on product ID because there is a relationship
       already defined. If your tables are not joined, you need to join them
       on the original linking column.)

    c. From the Query menu, choose Update.

    d. (Optional) From the View menu, choose Table Names. This is to ensure
       that the product name field in the Field cell is from the correct table,
       order details, which is critical.

    e. Drag the product name field from the order details table to the first
       field cell.

    f. In the Update To cell, type [Products].[Product Name].

    g. Run the query by clicking the Run button on the tool bar.

    h. Close the query.

A more realistic situation where this might be useful would be where a
database contained a table of products and prices. If the database user

was sent an ascii file containing product ID and new price for some of the products, they could import the file, and then 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 in the Microsoft Knowledge Base:

    UPDATE and COUNTER and QUERY.

Reference(s):

"Microsoft Access User's Guide," version 1.0, Chapter 7

References

INF: Additions to User's Guide Appendix A, MS Access Specifica

Article Number: Q93691
CREATED: 17-DEC-1992
MODIFIED: 23-DEC-1992

Summary:

This article includes additions to Appendix A, "Microsoft Access
Specifications," in the Microsoft Access User's Guide.

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
  Unique Values Only is selected (SELECT DISTINCT)  10
Number of ANDs in a WHERE clause (Criteria)      40
Number of ANDs in a HAVING clause                40
```

References

PRACC9211: Unique Values Only Property Selected May Show Dupli

Article Number: Q93692
CREATED: 17-DEC-1992
MODIFIED: 18-DEC-1992

Summary:

PROBLEM ID: PRACC9211

SYMPTOMS

Duplicate values are displayed in a query's output even though the query
property Unique Values Only (SELECT DISTINCT) is checked in the Query
Properties dialog box.

CAUSE

Duplicate values in a SELECT DISTINCT query's output will only occur when:

1. The query input is greater than approximately 64K with duplicate values
   widely scattered in the input data. (For example, if you were
   performing a SELECT DISTINCT on a text column with an average
   length of 20, you would have to have approximately 3300 values
   selected from the underlying table or query. 65536 / 20 = 3276.8)

   Note: If the query is based on a table that has 10,000 rows, but a
   criteria is specified that would only retrieve 20 rows, this is the set
   of data that counts for determining the size of the query input data
   set - the amount of data after applying the Where clause.

2. The values with duplicates scattered in the input data must be
   very close to what would be the beginning of the query output
   (maybe in the first half dozen values).

   Normally, SELECT DISTINCT displays data implicitly sorted. For example:

       SELECT DISTINCT [Order Details].[Order ID]
       FROM [Order Details];

   would display the data sorted by Order ID. However, if the user was
   doing a SELECT DISTINCT on more than one column, and put a sort on
   a column other than the first one, the duplicates could appear
   anywhere within the query output. The user would need to remove
   the sort to determine if duplicates were being returned.

RESOLUTION

Both SELECT DISTINCT and GROUP BYs in a totals query are limited to 10
columns. Normally, GROUP BY is much faster when the resulting set of data
is much smaller than the set of input data, and SELECT DISTINCT is
faster when there are only a few duplicates.

To ensure the correct results, use a Totals query with GROUP BYs
instead of using the Unique Values Only query property. For example:

    SELECT DISTINCT [Order Details].[Order ID]
    FROM [Order Details];

```
    is equivalent to:

        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.

Reference(s):
Microsoft Access User's Guide, chapter 6, page 128+, page 142+,

For more information, search Microsoft Access Help on DISTINCT and then
"ALL, DISTINCT, DISTINCTROW Predicates (SQL)"

References

PRB: Setup Problem Regarding the COMMDLG.DLL File and MS Acces

Article Number: Q93694
CREATED: 17-DEC-1992
MODIFIED: 23-DEC-1992

Summary:

SYMPTOMS

The following error message dialog may appear when trying to open a
database:

CAUSE

The following error message is displayed:

    Outdated 'COMMDLG.DLL' Please Re-install Microsoft Access.

RESOLUTION

1. You probably have more than one version of the COMMDLG.DLL on
   your computer. Rename the duplicate versions of COMMDLG.DLL.

2. Make sure the COMMDLG.DLL in the windows system sub-directory is
   newer than:

       3/10/92   for Windows 3.1 or
       10/01/92 for Windows for Workgroups or
       10/25/92 for Microsoft Access

3. You are using a third party non-compatible COMMDLG.DLL.

More Information:

To search for all occurrences of COMMDLG.DLL, perform the following
steps:

1. Exit Windows.

2. Search for any other copies of COMMDLG.DLL. The following command
   at the DOS prompt will locate all copies on drive C. You will need
   to do this for each drive that is located on your path statement in
   your AUTOEXEC.BAT file; the drive where your windows directory is
   located; and the drive that you have Microsoft Access installed.

       C:\ cd\
       C:\ DIR COMMDLG.DLL /s

   ONLY one should be on your disk in the windows/system directory.
   If you have one or two copies that have been installed elsewhere by
   other applications, they should be deleted or renamed.

How to install a new COMMDLG.DLL

If the user has deleted all duplicates of COMMDLG.DLL and still gets
the error in Microsoft Access, the user can copy COMMDLG.DL_ from disk
1 of the Microsoft Access retail package to their hard disk, then copy
EXPAND.EXE from disk 3 of their WINDOWS 3.1 diskettes. COMMDLG.DL_

can be expanded using the expand utility. Care should be taken not to
copy COMMDLG.DL$ from the Microsoft Access diskettes. This file can
only be decompressed with the Microsoft Access Setup.

The syntax for using EXPAND.EXE is:

    c:\<path>\EXPAND <path>\COMMDLG.DL_ c:\windows\system\COMMDLG.DLL

Other problems

A situation can arise during setup when it tries to update the
COMMDLG.DLL but returns an error that it can't get access to the file.
In most cases, this means that some other application is running that
is using the COMMDLG.DLL and therefore Microsoft Access setup can not
update it.

The solution is to make sure no applications are running in windows at
the time of setup. To do this, move all icons from the STARTUP group
into a temporary group (in Program Manager). Also, edit the WIN.INI
file and clear both the 'run' and 'load' lines so that they read:

    [windows]
    RUN=
    LOAD=

It might also be necessary to edit the AUTOEXEC.BAT to look for
conflicting terminate-and-stay-resident utilities (TSRs), although
this would be the least likely cause.

Reference(s):

References

INF: Passing Values from a Form to a Parameter Query

Article Number: Q93695
CREATED: 17-DEC-1992
MODIFIED: 17-DEC-1992

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 where you can enter the values for a parameter query.

More Information:

The following example uses a table from sample database, supplied with
Microsoft Access, Northwind Traders (NWIND.MDB) to show you how to
create a form that prompts for a date value. It also shows how to
add a command button to the form that executes 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. (Leave the Select A Table/Query box
   empty and click 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 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 which will run
   the query MyQuery which in turns uses the date entered into the
   MyDate textbox as the parameter to the query.
```

[References](#)

INF: How to Dial a Phone Number in MS Access

Article Number: Q93696
CREATED: 17-DEC-1992
MODIFIED: 18-DEC-1992

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 Microsoft Windows Application Programmer 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. 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

3. 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()

4. Create a new module, enter the code below, and save it as Comm DLL.

```
' Note: Each command should be entered on one line. The line
' continuation character '_' improves the readability of the code.

'********************************************************
'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)
    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: How To Get Microsoft Works Files into Access

Article Number: Q93698
CREATED: 17-DEC-1992
MODIFIED: 18-DEC-1992

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:

- Currency fields are converted to Currency fields.

- 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
progams, see the "Microsoft Works User's Guide".

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 64-65, pages 72-75

[References](#)

INF: Access Cannot Attach or Import Paradox 4.0 Tables

Article Number: Q93699
CREATED: 17-DEC-1992
MODIFIED: 23-DEC-1992

Summary:

Microsoft Access 1.0 only 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. To use data created in
Paradox 4.0 you will need to use Paradox to restructure the table in
version 3.5 format. Paradox provides the capability to create files in
'Compatible' format which will create files in version 3.5 format.

You can save a Paradox 4.0 table to the version 3.5 format within
Paradox 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 a menu.

5. Press F2 to save the file in the restructured format.

For more information on restructuring Paradox tables, refer to the
Paradox 4.0 documentation.

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

PRB: Access Appears to Hang During Compact

Article Number: Q93717
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

SYMPTOMS

When compacting a database in Microsoft Access the status bar indicates the percentage complete. At times the status bar will stay at one value for a very long time, making it seem as if Microsoft Access has hung. In most cases Microsoft Access is still working on the compact.

CAUSE

When Microsoft Access compacts a database it simply copies each object to a new database. Since different objects can be different sizes some take longer to copy. The percentage shown in the status bar is the percentage of the number of objects copied, not the percentage of the overall size of the objects.

RESOLUTION

This behavior is by design.

More Information:

For example, you could have a very large table in a database and the percentage could go up to 90% and then stay there while the large table in being copied, making it look as if Microsoft Access is hung. The 90% should be interpreted as meaning that 90% of the objects have been copied rather than 90% of the processing is complete.

Reference(s):
"Microsoft Access User's Guide", page 627.

References

PRB: Unable to Change Permissions (Security)

Article Number: Q93718
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

SYMPTOMS

```
When trying to open an object or change an account's permissions, 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 group Admins, has full
permissions to all database objects. They are also the only users who
can assign or unassign security permissions.
```

STATUS

```
This behavior is by design.
```

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 group Admins.  If you still have the original user Admin,
make sure it is a member of the group Admins and try again.
```

```
Reference(s):

For more information search for "Error Messages: Reference" using the Help
menu.

"Microsoft Access User's Guide", version 1.0, pages 609-629.
```

References

PRB: Changes to Shared Module Kept Local Until MDB Closed

Article Number: Q93719
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

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 re-opened.

[References](#)

PRB: DDE Statement Are Not Valid In Listbox or Combobox Contro

Article Number: Q93720
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

SYMPTOMS

    If you place a DDE function as the ControlSource of a listbox or
    combobox, you will receive a #Error in the ListBox 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 will cause an
    error message in both Browse and Datasheet modes of the form.

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 which 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 listbox 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.

Work Around

Using Access Basic it is possible to create a list function which
returns to a list box or combo box. You may use any Access Basic routine

to fetch the information used in the list, including DDE commands. The list function is documented on pages 83 through 85 of the Introduction to Programming manual. Coding Dynamic Data Exchange is documented in chapter 9 of the same manual.

[References](#)

PRB: Column in QBE Grid Will Not Lose Focus When Click

Article Number: Q93721
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

SYMPTOMS

   After selecting a column in the QBE grid, the column will not lose
   its focus when you click on a field list in the upper part 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 couple of columns to the query grid.

3. Select a column in the query grid by clicking on the column selector
   above the field box.

4. Click on the field list for the table.

Result: The query grid column does not lose the focus.

Depressing the delete key at this point will remove the table and all the
information in the query grid.

[References](#)

PRB: Access Cannot Paste Entire Column in Query Grid

Article Number: Q93722
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

SYMPTOMS

```
When using the Query Grid, and you are trying to cut and
paste or copy and paste an entire column, you see incorrect
values in the target column.
```

STATUS

```
This behavior is by design.
```

RESOLUTION

```
Select the same size object when pasting information that
you place into the clipboard. In this instance, select the
entire column when you paste the informaiton back to the query
grid.
```

More Information:

```
Steps to Reproduce:
-------------------
1. Create a query.

2. Choose an entire column in the Query Grid (click once at
   the top of the 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 copied column) and choose Paste
   from the Edit menu.

The values in the column you have chosen will not be
correct. Using  these features will not paste correctly into
any object in Microsoft Access.
```

References

PRB: Hourglass May Not Change After Illegal Function Call

Article Number: Q93723
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

SYMPTOMS

In Access Basic, using a user defined function, the user can make
the mouse pointer change from an arrow pointer to an hourglass cursor.

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 things by clicking on them.

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:

Microsoft Access does not reset Echo, Warnings, or Hourglass when run
from Access Basic code. Access only resets them when the last script
runs. Since an illegal function call stopped the execution of the
function prior to the command to reset the mouse pointer from an
hourglass to an arrow pointer, the mouse pointer remains in the
hourglass state.

Work around

You can type DoCmd Hourglass False from the Immediate window to get the
pointer back while in design of the module.

Reference(s):

"Microsoft Access Basic: An Introduction to Programming," chapters 1-5

"Microsoft Access Language Reference," verwsion 1.0, Part 1.

References

PRB: Print Quality List Erroneous With Generic/Text Only

Article Number: Q93724
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

SYMPTOMS

The Print Quality list in the Print dialog will contains the four
options: High, Medium, Low, and Draft. when you have selected
'Generic/Text Only' as your printer.

CAUSE

This happens because all printer drivers, regardless of their
capabilities, have all four options built into their printer driver.
In some cases all four options have the same function as with the
'Generic/Text Only'driver. With other printer drivers the Print
Quality list shows only valid options.

STATUS

This behavior is by design in Microsoft Access.

References

PRACC9212XXX: GPFs Using Non True Types Fonts With 8514.DRV

Article Number: Q93725
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

PROBLEM ID: PRACC9212XXX

SYMPTOMS

General Protection Fualts(GPF) & other problems may occur when
using non True Type fonts with the Windows 8514.DRV video driver.

CAUSE

This can be reproduced under Excel 4.0, including the GPF. This is
clearly a display driver problem.

RESOLUTION

Use only True Type fonts with the Windows 8514 driver.

More information:

Steps to Reproduce Problem

Scenario 1
----
1. Design new form, make it about 8 inches wide.
2. Drop in a label.
3. Set font to MS Sans Serif 8pt if it isn't already.
4. Type in some long text that would span about 5 inches.
5. Size the box to fit the whole line of text on one line.
6. Type in 127 for the point size.
7. Layout/Size to fit
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 with non True
Type fonts. If you use Arial it looks ok.

Scenario 2
----

1. Design new form and make it 22 inches wide.
2. Drop in a label.
3. Set font to Arial 8pt Bold if it isn't already.
4. Type in some long text that would span about 18 inches.
5. Size the box to fit the whole text on one line.
6. Go to the toolbar, select MS Sans-Serif for the font.

What happens varies, but in the end we always end up with a message
saying that MSACCESS caused a GPF in 8514.DRV.

Reference(s):

References

PRB: Overlapped Printing an Access Query Using TTY.DRV
Article Number: Q93726
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

SYMPTOMS

When printing a query using the TTY.DRV (Windows Generic Text Only printer driver) the print overlaps on every sixth or seventh line.

RESOLUTION

The TTY.DRV driver can only print six lines per inch. You need to set your row height to 12 point to get the line spacing to work correctly with this printer driver. To change the row height choose the Row Height option from the Layout menu. Type in 12 and this will allow the TTY.DRV to print your query without overlapping.

STATUS

This behavior is by design.

More Information:

```
Steps to Reproduce:
-------------------
```
1. Open any query.

2. From the File menu select Print Setup.

3. Choose Specific Printer and from the list of printers select
   Generic Text Only. (this driver must be installed on your system for
   this to be a valid option)

4. Run the query.

5. From the File menu select Print.

About every sixth line of print will be overlapped.

[References](#)

PRB: Access Will Not Import YYMMDD Dates in Text Files

Article Number: Q93727
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

SYMPTOMS

Microsoft Access will not import fixed width text files if there
is a date field in the text file formatted as YYMMDD.

CAUSE

Microsoft Access does not recognize the YYMMDD 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

A work around is to format the date as MMDDYY.

STATUS

This behavior is by design.

More Information:

Importing a fixed width text file with a date field formatted as
YYMMDD, Microsoft Access will generate an Import Error log table, but will
not import the data. If you try importing the text file as delimited the
data will import, but Microsoft Access will change the format from date to
number or text depending upon the date delimitor.

References

PRACC9212XXX: GPF With DEBUG Windows Using PPCL5A.DRV

Article Number: Q93728
CREATED: 20-DEC-1992
MODIFIED: 20-DEC-1992

Summary:

PROBLEM ID: PRACC9212XXX

SYMPTOMS

Printing on a HPPCL5A printer under the DEBUG version of Windows
3.1 will cause a General Protection Fault. The message on the debug
terminal states:

    err ACCESS->HPPCL5A GLOBALSIZE+C: Invalid global handle: 0x000

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 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 on the report that overlap.

5. Set printer to HPPLC5A printer and chose File, Print.

Results in an abort on the debug terminal.

[References](#)

INF: Access Does Not Recognize SQL Server Double Indexes

Article Number: Q93925
CREATED: 21-DEC-1992
MODIFIED: 21-DEC-1992

Summary:

Microsoft Access does not recognize a SQL Server table index if the index contains a data type of DOUBLE.

The behavior is by design.

More Information:

Microsoft Access requires a unique index on any SQL Server table for updates (editing). The data type DOUBLE cannot be used by Microsoft Access as a unique index. After reading the DOUBLE index value on the server, the value is converted from binary to decimal and back to binary, causing rounding errors, before it is stored in a Microsoft Access virtual table (VT). The same conversion process happens when the value is sent back to the server for an update. Microsoft Access uses the index value to find the row in the table where it needs to make the update. The aforementioned rounding errors make it impossible for Microsoft Access to find the correct row to make the update, thus failing the update. Since updates cannot be made on this type of index, even though it may be unique, Microsoft Access ignores the index.

Reference(s):

References

INF: Copying and Pasting Text in a Property Sheet

Article Number: Q93926
CREATED: 21-DEC-1992
MODIFIED: 21-DEC-1992

Summary:

```
You cannot use the Edit menu's Copy or Paste commands to edit text in
a Property Sheet.
```

More Information:

```
Use the equivalent keystroke combinations for the Copy and Paste
commands.  CTRL+C will copy information to the clipboard, while CTRL+V
will paste information from the clipboard.
```

References

INF: Print a Group's Name on a Subsequent Column or Page

Article Number: Q93927
CREATED: 21-DEC-1992
MODIFIED: 21-DEC-1992

Summary:

This article will explain how to create a report that prints
"continued" at the top of a page or column, when a group
flows over to another column or page.

More Information:

Microsoft Access does not insert the group name if the group flows
over to another column or page. Although through Macro's, it is
possible to identify the group that the records belong to and print
the group name along with the word "continued" at the top of a new
page or column. For example, if the detail records for the group
"Product Names" starts at the bottom of one page or column and
continues at the top of the next, it is possible to print "Product
Names continued" on the new page or column. This article will explain
in detail how to accomplish this. Listed at the end of this article
are other related articles that may aid in Multi-Column report processing.

To create a report that prints "continued", at the top of a page or
column, when necessary do the following steps.

1. Create a new Macro Group.

2. Put the following macros in the group.

```
    Macro Name      Condition           Action
    -------------------------------------------------
    Page Reset                          SetValue
    Set Top1        [Top1]=0            SetValue
    Print           [Top1]=0            SetValue
                    Top=[Top1]          SetValue
                    ...                 *SetValue
                    ...                 SetValue
                    ...                 StopMacro
                                        SetValue
                                        *SetValue
```

   * Repeat these SetValue actions for each control that is being
   viewed in the detail of the report. Replace [Fieldx] with each
   control that is being viewed on the report.

```
    Page Reset Actions
    ==================
        SetValue
        --------
          Item: [Top1]
          Expression: 0
```

```
        Set Top1 Actions
        ================
            SetValue
            --------
                Item: [Top1]
                Expression: Top

        Print Actions
        =============
            SetValue
            --------
                Item: [Top1]
                Expression: Top
            SetValue
            --------
                Item: [Continued].Visible
                Expression: Yes
            SetValue
            --------
                Item: [Fieldx].Visible
                Expression: No
            SetValue
            --------
                Item: NextRecord
                Expression: False
            StopMacro
            ---------
            SetValue
            --------
                Item: [Continued].Visible
                Expression: No
            SetValue
            --------
                Item: [Fieldx].Visible
                Expression: Yes
```

3. Save the macro as 'Continued'.

4. Open the desired report in Design Mode.

5. 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       On Format     Continued.Page Reset
   *Group Header      On Format     Continued.Set Top1
    Detail            On Format     Continued.Print
                      Can Shrink    Yes
   *Group Footer      On Format     Continued.Set Top1
```

   * Note: - Set the On Format property of all Group Header and Group
   Footer sections to the expression 'Continued.Set Top1'. This will
   set the Top1 control, which tracks the Twip value of the top of
   columns, to the current value of the property Top.

6. Using the Toolbox, create an unbound Text Box and place it in the
   Detail section. Locate the Text Box at the top, in the same
   location vertically to the label of the Group Header. Set the
   following properties of the unbound Text Box:

   ```
   Property          Expression
   --------------------------
   ControlName:      Continued
   ControlSource:    =[Control Name being grouped on] & " continued"
   Visible:          No
   CanShrink:        Yes
   ```

7. Using the Toolbox, create another unbound Text Box and place it in
   the first Group Header section. Set the following properties of the
   unbound Text Box.

   ```
   Property          Expression
   --------------------------
   ControlName:      Top1
   Visible:          No
   ```

8. Save and Run the report.

As explained, this article shows how to identify the group that
records belong to and print the group name along with the word
continued at the top of a new page or column. Although, this approach
will work with a normal single column report, there is another article
written to accomplish this with a single column that is more
simplified. For more information, query on the following words in the
Microsoft Knowledge Base:

     Displaying "Cont.." Groups Displayed Multi-Pages

There is also an article on how to Span the page header to the full
width of a Multi-Column Report. To find this article search on:

     Page Header Span Full Width MultiColumn Report

For additional information on columnar reports search Help
for "Multiple-column reports" and "NewRowOrCol".

** Note - This article only refers to Vertical Multi-Column Reports.

Reference(s):

"Microsoft Access User's Guide" Version 1.0, Chapter 19

References

INF: Export Tables with OLE Object or Memo Fields to Paradox

Article Number: Q93928
CREATED: 21-DEC-1992
MODIFIED: 21-DEC-1992

Summary:

    When you export a table from Microsoft Access that contains a field whose
Data Type is "Ole Object" or "Memo" to Paradox 3.x, the resulting table in
Paradox contains one entry that says "m_i_c_r_o_s_o_f_t". You can see this
result by importing the Paradox database 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 Mode add a field to the copied table with data type of "Ole
   Object" or "Memo".

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 receive an
   error message stating "Invalid Data Type"; however, a file is created
   called "TESTFLD.DB".

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", data type "Number" and 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: How to Open UTILITY.MDA

Article Number: Q93929
CREATED: 21-DEC-1992
MODIFIED: 21-DEC-1992

Summary:

If you try to open UTILITY.MDA, you will receive the following error
message:

    The database 'UTILITY.MDA' is already open as a library database.

To open UTILITY.MDA, you must create a copy of UTILITY.MDA with a
different name and then open this copy within Microsoft Access.

More Information:

To open UTILITY.MDA:

1. Copy UTILITY.MDA to UTILITY.SAV.

2. Within Microsoft Access, open UTILITY.SAV by selecting the Open
   command from the File menu.

3. Select 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

UTILITY.MDA is a collections of Tables, Forms, and Modules used by
Microsoft Access. Note that you should always have a backup copy of
UTILITY.MDA since this is a vital database needed by Microsoft Access,
and any changes could result in unexpected behavior.

References

PRACC9212: GPF When Form Query Contains Financial Functions

Summary:

PROBLEM ID: PRACC9212XXX

SYMPTOMS

Using a Form Wizard to create a form based on a Crosstab query
causes a General Protection Violation if the query contains a
financial function. The following error message is displayed:

    GPF in MSABC100.DLL at 0004:7BBC (retail)

RESOLUTION

Microsoft is researching this problem and will post new
information here as it becomes available.

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 as it becomes available.

More Information:

Steps to Reproduce Problem
----
1. Create the following query:

   Query: Mortgage Analysis

       TRANSFORM First(CCur(PV([Rate]/12,[NPer],-[Pmt],[FV],[Due])))
       AS LoanAmount
       SELECT Format([Rate],"Percent") AS IntRate
       FROM PV
       GROUP BY Format([Rate],"Percent")
       PIVOT Format([Pmt],"$#,#00")
       WITH OWNERACCESS OPTION;

2. Select the query called 'Mortgage Analysis' and push the NewForm
   button on the Toolbar, then choose Form Wizards.

3. Select Single-column and push OK.

Note: If you add Fixed-Column Headings, things work until you press
Open.

References

INF: Using DDE To Fill A ListBox

Article Number: Q94003
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

This article contains an example of an Access Basic list function
which uses DDE to populate a ListBox.

The example assumes that you are familiar with Dynamic Data Exchange,
creating modules in Access Basic, and creating ListBoxes 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 ListBox bound to the
Country field of the Suppliers table in the sample database Northwind
Traders (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 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 sub routine:

        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

11. Verify that Excel is running and the worksheet
    COUNTRY.XLS is open. Switch to browse mode. The
    ListBox should now be populated with the countries
    that you entered into Excel.

Note: This article demonstrates a simple example and is
not a complete 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
   row in the data to Microsoft Access.

 - The form may need to be refreshed to show updated
```

data.

References:
For information on creating ListBoxes on forms see
chapter 9 of the Microsoft Access User's Guide.

For information on creating list functions in Access
Basic see page 82 of the Introduction to Programming
guide.

References

## INF: Installing Access From A Diskless Workstation

**Article Number: Q94010**
**CREATED: 22-DEC-1992**
**MODIFIED: 22-DEC-1992**

## Summary:

According to the third paragraph on page 658 of the Microsoft Access
User's guide version 1.0, a workstation running Microsoft Access from the
network requires 1 megabyte of harddisk space. A diskless workstation will
not have a hard disk. In this case 1 megabyte would be needed in the users
directory on the network not on the local workstation.

You can install and run Microsoft Access from a diskless workstation
as long as your workstation meets the basic requirements for running
Microsoft Access.

Reference(s):

"Microsoft Access User's Guide," version 1.0, page 658.

References

# INF: Converting Null Criteria for a Query by Form

Article Number: Q94025
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

## Summary:

A potential problem you my have when attempting to implement query is that empty criteria fields are likely to include NULL data.  This would return incorrect results upon executing the underlying query.

This article describes how to deal with these NULL fields in such a case.

## More Information:

Queries can reference form fields through implicit parameters, making query by form very feasible.  For example, you might have the following criteria on the 'CustomerName' column:

    Like Forms!CriteriaDialog!CustomerName

where CriteriaDialog is a dialog (form) containing your criteria fields.

If however, the end user doesn't supply a customer name in the criteria dialog (perhaps expecting to see *all* the rows), the query will actually return no rows, since the WHERE clause will be reduced to:

    Like NULL

which is always FALSE (NULL Like ANYTHING is FALSE).  To get around this, add the following function to a module:

The following function converts NULL values to a given value. Essentially, if the first argument to this function is NULL, then the second argument will be returned.  Otherwise, the first argument will be returned (unchanged).

```
    Function CNulls (v As Variant, subs As Variant) As Variant
       If (IsNull(v)) Then
          CNulls = subs
       Else
          CNulls = v
       End If
    End Function
```

Once this function has been added to a module, you can change the query's where clause to read:

    Like CNulls(Forms!CriteriaDialog!CustomerName, "*")

In this case, if the user does not supply a customer name, the CNulls function will return the asterisk, and all of the rows with data in them will be returned.

[References](#)

PRACC9212: Datasheet Form Saved as a Report Causes a GP Fault

Article Number: Q94026
CREATED: 22-DEC-1992
MODIFIED: 11-JAN-1993

Summary:

PROBLEM ID: PRACC9212XXX

SYMPTOMS

A report that contains a subreport created from a form causes a general
protection (GP) fault in Microsoft Windows if the Default View property
of the original form was set to Datasheet.

The following GP fault occurs immediately after you attempt to view the
report in Print Preview mode or to print it:

    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. Microsoft is researching this problem and will post new information
here as it becomes available.

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 will not occur if the main report is not bound.

4. Drag the Categories subform report you created in step 2 to the
   main report you created in step 3.

5. Switch to Print Preview.

Result:
   MSACCESS caused a General Protection Fault in module MSACCESS.EXE at
   006A:0A26

Workaround
----------
```

You can use a form as a subreport by embedding it directly in the main
report. You do not need to use the Save As Report option. A form with
a Default View of Datasheet will not cause a GP Fault when embedded
in a report.

```
1. Create a new report.
```

2. Drag the Categories Subform from the database container to the main
   report you created in step one.

Or

1. Open the original form in Design mode.

2. Change the Default View property to either Single or Continuous Form.

3. Choose Save As Report from the File menu.

4. Embed the new form in your report.

Note: The Main form/subform wizard creates the subform with Datasheet
as the default view. If you save a subform created by this wizard as a
report and then embed the subreport you will encounter this.

Reference(s):

[References](#)

INF: Converting Null Criteria for a Query by Form

Article Number: Q94027
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

A potential problem you my have when attempting to implement query is
that empty criteria fields are likely to include NULL data. This
would return incorrect results upon executing the underlying query.

This article describes how to deal with these NULL fields in such a
case.

More Information:

Queries can reference form fields through implicit parameters, making
query by form very feasible. For example, you might have the
following criteria on the 'CustomerName' column:

    Like Forms!CriteriaDialog!CustomerName

where CriteriaDialog is a dialog (form) containing your criteria fields.

If however, the end user doesn't supply a customer name in the criteria
dialog (perhaps expecting to see *all* the rows), the query will actually
return no rows, since the WHERE clause will be reduced to:

    Like NULL

which is always FALSE (NULL Like ANYTHING is FALSE). To get around this,
add the following function to a module:

The following function converts NULL values to a given value.
Essentially, if the first argument to this function is NULL, then the
second argument will be returned. Otherwise, the first argument will be
returned (unchanged).

```
    Function CNulls (v As Variant, subs As Variant) As Variant
       If (IsNull(v)) Then
          CNulls = subs
       Else
          CNulls = v
       End If
    End Function
```

Once this function has been added to a module, you can change the query's
where clause to read:

    Like CNulls(Forms!CriteriaDialog!CustomerName, "*")

In this case, if the user does not supply a customer name, the CNulls
function will return the asterisk, and all of the rows with data in them
will be returned.

[References](#)

PRB: Can't Check if Current Database Is Opened Exclusively

Article Number: Q94035
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

SYMPTOMS

In a multiuser environment you cannot use the OpenDatabase
command to find 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

JetOpenDatabase ignores the option flags for all opens except the
first for a given Windows's task. This would be one way a system
admin might try to test if a database has been opened exclusively.
This technique will not work since JetOpenDatabase does not check the
option flags in this case.

STATUS

This behavior is by design.

More Information:

The following sub successfully opens the current database from Access
Basic, 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: Each command should be entered on one line. The line
   ' continuation character '_' improves the readability of the code.

   '*******************************************************
   '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. If you run this sub from the Immediate Window by typing 'Test' you
   will not receive an error message.

[References](#)

PRB: Can't Perform OpenTable on an Attached SQL Table

Article Number: Q94036
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

SYMPTOMS

If you try to perform 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 to any valid table. From the File menu select 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. You will get the following message:

    Can't perform operation; it is illegal.

[References](#)

INF: How Access Can Use R:BASE Files

Article Number: Q94037
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

Microsoft Access does not ship with an ISAM driver that will support attaching to R:BASE database files. However R:BASE can export files into several formats that Microsoft Access can import.

To retain column names and data type definintions when tables are exported from R:BASE it would be best to choose 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, of dBASE III Plus (DBF)
Lotus 1-2-3 (version 1.0 and 2.0) or Symphony (WKS,WK1, or WRK)

References

INF: Importing Fixed Width Text Files

Article Number: Q94038
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

In Microsoft Access Fixed-Width Text means that the file contains
records of a fixed, consistent width and that data fields within each
record begin at fixed offsets from the beginning of the record. Files
with variable length records and which have no field separator, such
as commas or tabs, are not importable 'as-is'. They can be imported if
first run through a preprocessor which either inserts field separators
or pads each record with spaces out to a constant width.

More Information:

Microsoft Access imports Fixed-Width files by essentially casting a
record template into the file at calculable offsets. These offsets are
presumed to be the logical beginning of records. The logical record
length is determined by Microsoft Access by scanning the first data
row of the file and locating the first CR/LF. This is assumed to mark
the end of the first record (see point 3 below). The actual data
fields defined in the template come from the info supplied in the
Import/Export Spec that was setup by the user. Once the template is
cast into the file, individual field values are extracted and entered
into the database table. The template is then advanced to the next
record's position.

To be successful importing Fixed-Width files the following must be
observed:

1. The file must be truly Fixed-Width. All records of the same width
   and all fields beginning at consistent offsets from the beginning of
   the record.

2. The Import/Export spec should match the physical file. This spec
   information is used to create the template of fields while the overall
   record size is determined by inspecting the physical file. If the
   field template and actual record size differ you may get undesirable
   results. Make sure that the greatest (Offset + Width - 1) in your
   Import/Export spec is not bigger than the physical record length in
   your file (CR/LF at the end of the record is not counted). In other
   words if you have an Import/Export spec which defines two fields ID
   starting at offset 1 for a width of 8 and DATA starting at offset 11
   for a width of 50, then your physical file must contain records of no
   less than 60 bytes (11+50-1) in length (not counting the CR/LF).
   Longer records are allowed, as long as all records are the same size.
   The extra data beyond the last byte defined in the Import/Export spec
   will simply be ignored.

3. No embedded CR/LF characters in the first row of data. If you have
   embedded CR/LFs in your first row of data, Microsoft Access' record
   size calculation will be wrong and your import will fail to give you
   desirable results.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 75-81

[References](#)

PRB: Access Format Memo Field Truncates to 255

Article Number: Q94039
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

SYMPTOMS

    If you use the Format function on a Memo field the data is truncated
    to 255 characters. The field remains a memo data type. This happens
    when you format a memo field with existing data or enter new data.

STATUS

    Microsoft has confirmed this to be a problem in Microsoft Access
    version 1.00. We are researching this problem and will post new
    information here as it becomes available.

More Information:

```
Steps to Reproduce Behavior
---------------------------

1. Create a Table and specify Memo for the field type.

2. Enter either < or > for Format on the field.

3. Type in more than 255 characters into the field.

4. Commit the record.

5. Examine the record you entered, it will have truncated to 255
   characters while leaving the data type as memo.
```

[References](#)

INF: Using Sendkeys to Change An OLE Link

Article Number: Q94040
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

The only way that you can change the source of a linked OLE
object is by using the option Object, Change Link from the
Edit menu.

This article describes the steps to create a macro which
automatically changes the link based on a file name stored
in a table.

More Information:

```
Step One: Create the Example Table
----------------------------------
```
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. The bitmaps
   used in this example are in the Windows directory. To embed
   the objects open them one at a time in Paintbrush, choose
   Copy from the Edit menu, switch to Microsoft Access, place
   your cursor in the OLE field and select Paste Link from the
   Edit menu.

```
     Key            File Name        Linked Object
     ------------------------------------------------
   (counter)      arcade.bmp       Paste Link arcade.bmp...
   (counter)      arches.bmp       Paste Link arches.bmp...
   (counter)      argyle.bmp       Paste Link argyle.bmp...
```

```
Step Two: Create the Form
-------------------------
```
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 quotes.


```
Step Three: Create the Macro
```

```
----------------------------
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 quotes.

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", without
quotes. The OLE Object should update to display the bitmap
LEAVES.BMP which is also in your Windows directory.

For more information on embedding OLE objects see Chapter
13 of the Microsoft Access User's Guide: Using Pictures,
Graphs, and Other Objects.
```

References

PRB: Grid Lines Not Visible In Table Datasheet View

Article Number: Q94041
CREATED: 22-DEC-1992
MODIFIED: 11-JAN-1993

Summary:

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: How to Delete a Table Programmatically Without DoMenuItem

Article Number: Q94042
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

This article discusses a function you can use to delete a table from
the current database programmatically. This method does not require
the use of DoMenuItem or SendKeys.

More Information:

All of the references to objects in a database - including Tables,
Queries, Forms, Reports, Macros, and Modules - are stored in a system
table called MSysObjects. You can delete an object from MSysObjects
as you would from any other Microsoft Access table.

This is how the function listed at the bottom of this article works.
It simply finds a table object with the name you give it, then deletes
that record from MSysObjects. The function returns True if the table
was successfully deleted, and False if no table with the given name
could be found.

Please make sure you completely understand the ramifications of
modifying data in any system table before doing so.

```
Option Explicit
Function DeleteTable (TableName$)
   Dim D As Database
   S As Dynaset

   Set D = CurrentDB()
   Set S = D.CreateDynaset("MSysObjects")

   S.FindFirst "Name = '" & Trim(TableName$) & "' AND Type = 1"

   If S.NoMatch Then
      DeleteTable = False
   Else
      S.Delete
      DeleteTable = True
      SendKeys "{F11}%VQ%VT" ' Refresh Database Window
   End If
   D.Close
End Function
```

[References](#)

INF: Access NWind Sales by Totals by Amount

Article Number: Q94043
CREATED: 22-DEC-1992
MODIFIED: 22-DEC-1992

Summary:

The report named 'Sales by Totals by Amount' that is included in the
sample database Northwind Traders (NWIND.MDB) supplied with
Microsoft Access version 1.0, does not work correctly. The report uses
the Conditional Page Break macro to print 10 lines on each page of the
report. This only works for the first page because the Counter on the
report is not being reset. To get this to work correctly you must
change the Conditional Page Break macro. In the Macro Name Show 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

PRB: Err Msg: Couldn't Update Currently Locked by User

Article Number: Q94080
CREATED: 23-DEC-1992
MODIFIED: 23-DEC-1992

Summary:

SYMPTOMS

When two users try to simultaneously update a similar object in a
shared database, the following message appears:

    Couldn't update; currently locked by user '<user name>' on machine
    '<machine name>'.

CAUSE

A conflict with the SYSTEM.MDA file exists. By design, Microsoft
Access cannot update the MsysObjects table in the SYSTEM.MDA file for
multiple users at the same time.

RESOLUTION

Try saving the object again. If no one is updating a similar object
at the same time, the Save command should work. This can occur in Access
Basic code also, so you will need to hadle error trapping for this.

More Information:

This 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 file
   name.

4. Simultaneously, User A and User B press Enter to update the object.

5. Microsoft Access presents the error message listed above. Click OK
   to dismiss the message and try again.

References

PRB: Append Query Results in Duplicate Counter Numbers

Article Number: Q94105
CREATED: 28-DEC-1992
MODIFIED: 28-DEC-1992

Summary:

SYMPTOMS

A table that you appended 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.

STATUS

This behavior is by design.

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.

More Information:

You will only get duplicates if there is not a Primary Key or a 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 had values of 1 to 20 in the Counter field, and the table you were appending from had values of 20, 34, and 55, you would end up with one duplicate value. And, the next time you added a new row to the appended to table, the Counter field would be assigned a value of 56.

References

PRB: Access Adds Date Delimiters When Importing Text

Article Number: Q94106
CREATED: 28-DEC-1992
MODIFIED: 28-DEC-1992

Summary:

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 wi
a '/' date delimiter when importing a text file, regardless of whether
the text file is formatted with a date delimiter. This only occurs when yo
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

This behavior is by design.

References

INF: Access and Fixed Width Text File Format

Article Number: Q94107
CREATED: 28-DEC-1992
MODIFIED: 07-JAN-1993

Summary:

This article discusses what Fixed-Width Text file format
is and how to import these files.

More Information:

In Microsoft Access, Fixed-Width Text means that the file
contains records of a fixed, consistent width and data
fields within each record begin at fixed offsets from the
beginning of the record. Files with variable length records
and which have no field separator, such as commas or tabs, are
not importable 'as-is'. They can be imported if first run
through a preprocessor like "Write" to insert field
separators or pad each record with spaces out to a constant
width.

Microsoft Access imports Fixed-Width files by casting a record
template into the file at calculable offsets. These offsets
are presumed to be the logical beginning of records. The
logical record length is determined by Microsoft Access by scanning
the first data row of the file and locating the first
Carriage Return/Line Feed. This is assumed to mark the end
of the first record (see point three below). The actual data
fields defined in the template come from the information
supplied in the Import/Export Spec that was setup by the
user. Once the template is cast into the file individual
field values are extracted and entered into the database
table. The template is then advanced to the next record's
position.

To be successful importing Fixed-Width files the following
must be observed:

1. The file must be truly Fixed-Width. All records of the
same width and all fields beginning at consistent offsets
from the beginning of the record.

2. The Import/Export specification should match the physical
file. This spec information is used to create the template
of fields while the overall record size is determined by
inspecting the physical file. If the field template and
actual record size differ you may get undesirable results.
Make sure that the greatest (Offset+Width-1) in your
Import/Export spec is not bigger than the physical record
length in your file (CR/LF at the end of the record is not
counted). In other words, if you have an Import/Export spec
that defines two fields ID starting at offset 1 for a width
of 8 and DATA starting at offset 11 for a width of 50, then
your physical file must contain records of no less than 60

bytes (11+50-1) in length (not counting the CR/LF). Longer
records are allowed, as long as all records are the same
size. The extra data beyond the last byte defined in the
Import/Export spec will simply be ignored.

3. No embedded CR/LF characters in the first row of data
allowed. If you have embedded CR/LFs in your first row of
data Microsoft Access's record size calculation will be wrong
and your import will fail to give you desirable results.

Reference(s):

Help Topic: "Fixed Width Text Files"

"Microsoft Access User's Guide," version 1.0, pages 72-85.

[References](#)

Parameter Queries Are Not Supported As DDE Topics

Article Number: Q94108
CREATED: 28-DEC-1992
MODIFIED: 28-DEC-1992

Summary:

Microsoft Access does not support Parameter Queries via DDE.
Parameter driven queries require parameters for which Microsoft
Access has no mechanism to receive via DDE.

More Information:

Attempts to initiate a DDE channel with a parameter query
will fail. The resulting error message is dependent upon
the DDE Client. In Microsoft Visual Basic the user will
see:

   "No foreign application responded to DDE Initiate".

In Microsoft Excel the error message is:

   "Remote Data not Accessible. Start 'MSACCESS.EXE'?"

One work around is to link an Microsoft Access form to the DDE
client. This link will transfer the parameter to Microsoft
Access. Then the modify the query to refer to the form for the
parameter. The steps to use this method with Microsoft
Excel follow:

A. Create the DDE Link
----
1. Start Microsoft Excel and open a new Macro sheet.

2. In cell A1 enter "DAIR" without quotation marks.

3. Save the macro sheet as DDETEST.XLM.

4. Select cell A1 and choose Copy from the Edit menu.

5. Start Microsoft Access and open the Northwind Traders
   database, NWIND.MDB.

6. Create a new, unbound form.

7. Choose Paste Special from the Edit menu.

8. Select Text in the Data Type list box and then click
   Paste Link. This creates a Text box with the formula:

       =DDE("Excel","C:\ACCESS\DDETEST.XLM","R1C1")

9. If the Properties box is not available choose Properties
   from the View menu.

10. Select the text box created in step 7 and change the
    Control Name property to: DDE Parameter.

11. Save the form as [DDE Link Form].

B. Modify the Query
----
1. Open the query [Products By Category (Parameter)] in
   design mode.

2. Select Parameters from the Query menu and delete the criteria
   [Enter a Category ID].

3. Under the field [Category ID] in the query grid replace the
   existing criteria with:

        =Forms![DDE Link Form]![DDE Parameter]

4. To test the query verify that the macro sheet DDETEST.XLM
   is open in Microsoft Excel and that the form
   [DDE Link Form] is in Browse mode. Run the query and
   verify that the products are in the category "DAIR".

C. Create a Macro to Open the Form
----
1. Create a new macro.

2. Add the following macro action:

       OpenForm
           Form Name: DDE Link Form
           View: Form

3. Save this macro as "Open Form", without quotes.

D. Create a Macro to Run the Query
----
1. Create a new macro.

2. Add the following macro action:

       OpenQuery
           Query Name: Products by Categories (Parameter)
           View: Datasheet

3. Save this macro group as "Run Query", without quotes.

E. Create the DDE Macro
----
1. In the macro sheet DDETEST.XLM 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()
```

2. Place your cursor in cell A10, choose Define Name from
   the Formula menu, enter Run_Query in the Name: box. Click
   the option Command and then choose OK. This gives the
   name Run_Query to the second macro.

3. Close all forms, macros, and queries in Access.

4. To run the macro place your cursor in cell A3, choose
   Run from the Macro menu, and then press OK. The Excel macro
   will run much more quickly is you set the DDE TimeOut option
   in Microsoft Access to 1 second.

Notes:
   Why are SendKeys used?
       Excel has instructed Microsoft Access to open the
       form[DDE Link Form]. When Microsoft Access opens
       the form it will attempt to update the DDE link in
       the control [DDE Parameter]. However, Excel will
       not process any requests until it receives
       notification that Microsoft Access has successfully
       completed the macro. Microsoft Access will not finish
       executing until it successfully updates the link to Excel.
       Eventually Microsoft Access will time-out and display #Error.
       The SendKeys command sends the key sequence F9 which
       refreshes the form.

   How would you get this information to Excel?
       If you need to transfer the result of the query to
       Excel change the query to a Make Table query. You
       will then be able to use the DDE Request function
       to retrieve the data in the table from Excel.

   Reference:
   The DDE Server Topics supported by Microsoft Access are
       documented in the file README.TXT which in
       installed in your Microsoft Access directory.

   For information on parameter queries refer to the Microsoft
       Access User's Guide, Chapter 7 Designing Action
       Queries and Parameter Queries, pp.178-181

   For information on creating DDE links in forms and reports

refer to the Microsoft Access User's Guide, Chapter
13 Using Picture, Graphs, and Other Objects, pp.337-339

[References](#)

PRB: Trailing CR/LF's Stripped From Text Box When Retrieved

Article Number: Q94109
CREATED: 28-DEC-1992
MODIFIED: 28-DEC-1992

Summary:

SYMPTOMS

```
Trailing CR/LF's are stripped out when retrieving data from a text
box. Although it is possible to stuff CR/LF's into a text box, when
attempting to retrieve the data, the trailing CR/LF's are truncated.
```

STATUS

```
This is by design.
```

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: Full Path Causes Endless Loop In StartApp() Example

Article Number: Q94110
CREATED: 28-DEC-1992
MODIFIED: 28-DEC-1992

Summary:

Passing the full path of a DDE server to the function Startapp() will cause
an endless loop. The function will continue to run, starting up new copies
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 pat
is used.

More Information:

The sample database Northwind Traders (NWIND.MDB) contains a module
called Introduction to Programming. Many of the Access Basic Codes example
the Introduction to Programming manual are included in this module. The
example function StartApp() from page 142 contains a routine which may 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 pass
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() will fail but the Shell() function in the error
routine will be successful.

This is an example of using StartApp() that will cause an endless loop:

```
=StartApp("c:\winword\winword.exe","system")
```

This is an example that will work correctly:

```
=StartApp("winword","system")
```

References

INF: Hewlett-Packard Dashboard GPF with Microsoft Access

Article Number: Q94111
CREATED: 28-DEC-1992
MODIFIED: 28-DEC-1992

Summary:

Several customers have called complaining that Microsoft Access causes
General Protection Faults while being used with Hewlett-Packard
Dashboard 1.0. This is a known problem and Hewlett-Packard has
released a patch fixing this problem. The patch can be obtained on
CompuServe in the HPSYS forum. The file is called DB101U.EXE and is
located in Library 11 of that forum.

More Information:

To obtain the file:

1. Type 'GO HPSYS' at the ! prompt in CompuServe, then press [Enter].
   The user will receive general information about the forum and it will
   ask the user to join.

2. Press [Enter] until the Forum! prompt is reached.

3. Type 'LIB' and press [Enter]. The user will be prompted for a
   library number. Type '11' and press [Enter]

4. Type 'DOWN DB101U.EXE' and press [Enter]. CIS will prompt the user
   for a local file name. Enter 'DB101U.EXE' with a qualified path that
   you would like to store the file and press [Enter], i.e.

       'C:\WINDOWS\DB101U.EXE'

   This will start the download process.

5. When the download is complete, the user will then need to run the
   patch by typing 'DB101U' at the DOS command prompt. The user should
   create a scratch directory and run 'DB101U' in that directory. The
   file will automatically extract and expand the patch and a readme file.
   Hewlett-Packard's recommendation is to follow the instructions in that
   readme file.

Dashboard is a registered trademark of Hewlett-Packard, Inc., a
vendor independent of Microsoft; we make no warranty, implied or
otherwise, regarding this product's performance or reliability.

References

INF: Access 1.0 Disks 1-4 Contents (5.25 inch, 1.2 meg)

Article Number: Q94112
CREATED: 28-DEC-1992
MODIFIED: 28-DEC-1992

Summary:

The following lists the directory contents of diskettes 1-4
shipped with Microsoft Access version 1.0 (5.25 inch, 1.2
meg).

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
```

Reference(s):

References

INF: Access 1.0 Disks 5-9 Contents (5.25 inch, 1.2 meg)

Summary:

The following lists the directory contents of diskettes 5-9 (including
the ODBC Setup Diskette and the Windows for Workgroups Driver
diskette) shipped with Microsoft Access version 1.0 (5.25 inch, 1.2
meg).

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 Diskette

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 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 (3.5 inch, 1.44 meg)

Article Number: Q94114
CREATED: 28-DEC-1992
MODIFIED: 28-DEC-1992

Summary:

The following lists the directory contents of each diskette shipped
with Microsoft Access version 1.0 (3.5 inch, 1.44 meg).

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

Volum 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
MSGRAPH  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
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 Installation Diskette

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 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: How to Retrieve Information From the Clipboard

Article Number: Q94162
CREATED: 29-DEC-1992
MODIFIED: 12-JAN-1993

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:

```
Option Explicit
' Note: Each command should be entered on one line. The line
' continuation character '_' only improves the readability of the code.

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
```

Reference(s):

"Programming Windows by Charles Petzold", Second Edition, pg 780-781.

[References](#)

PRB: Back and Rewind Buttons Disabled in Wizards

Article Number: Q94174
CREATED: 29-DEC-1992
MODIFIED: 29-DEC-1992

Summary:

SYMPTOMS

```
    When you use the Chart Wizard to create a Form the 'FastForward'
    button is disabled after going back one or more screens.
```

STATUS

```
    This behavior is by design.
```

More Information:

```
Steps to Reproduce Behavior
---------------------------

1. Create a new Form using the Form Wizard. Choose the Graph Wizard.

2. Proceed to the very last Graph Wizard screen. This screen gives you
   the option to Open or Design your form.

3. Select the Back button and go back any number of screens.

   The 'FastForward' button will be grayed. This button should take you
   to the last screen of the Wizard.
```

[References](#)

PRB: Date And Time Format Not Used In Listbox Combo box Contro
Article Number: Q94175
CREATED: 29-DEC-1992
MODIFIED: 29-DEC-1992

Summary:

SYMPTOMS
```
When creating a Listbox or ComboBox with a Date/Time field, the date
or Time format will not be used.
```

CAUSE
```
Microsoft Access currently does not provide a 'Format' property for list
boxes even though the underlying table fields are defined with a format.
```

STATUS
```
This behavior is by design.
```

RESOLUTION
```
You can create a query and base the listbox or combo box on the query.
From the Query you can use an expression to format the data as desired.
A query example, on the above:

Change this:

    Query: Querytest
    ----------------------------------
    Field name: Event Date
To:

    Query: Querytest
    ----------------------------------
    Field name: Expr1:Format([event date],"mmmm d"", ""yyyy")

This formatted expression will display correctly for combo or list boxes
```

More Information:

```
Steps to Reproduce Behavior
---------------------------

1. Create a new table called with the following:

    Field Name           Data Type
    --------------------------
    Event ID             Counter
    Event Date           Date/Time

    Event Date           Field Properties
    ---------------------------------
    Format               mmmm d, yyyy

2. Save the table as EVENTS and 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 is correctly displayed as
"August 11, 1992".

3. Now create a new blank form based on the same EVENTS table. Add a
   Listbox. Set the properties for the Listbox as shown below:

```
   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
```

4. Switch to form view.

   At this point, the dates are displayed in the Listbox as '8/11/92' not
   'August 11, 1992', ignoring the format property set on the underlying
   column. The same test can be used on a TIME field with similar
   results.

[References](#)

PRB: From Objects Not Placed on Grid by Wizard.

Article Number: Q94176
CREATED: 29-DEC-1992
MODIFIED: 29-DEC-1992

Summary:

SYMPTOMS

When using the Form Wizard, objects are not aligned to the grid
even though Snap To Grid is on.

CAUSE

Wizards 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 Wizards 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 Form Wizard 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: Can't Create Two Relationships Between Same Tables

Article Number: Q94189
CREATED: 29-DEC-1992
MODIFIED: 29-DEC-1992

Summary:

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. Form 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: Arrow Keys Inconsistent Between Forms and Datasheets

Article Number: Q94190
CREATED: 29-DEC-1992
MODIFIED: 29-DEC-1992

Summary:

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.
In form view however, pressing an arrow key at the end of a control
has no effect. It will not move the focus to the next control.
```

CAUSE

```
This behavior is by design
```

More Information:

```
It is possible to 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 and form view. The difference in behavior depends on
the mode of the Microsoft Access text editor.  For more information on text
editor, search the Knowledge Base for the following key words:

    key AND behavior AND edit AND box

If the editor is in "Enter" mode (the entire field or control is
highlighted), 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 returns to "Enter" mode.  In form view however,
pressing an arrow key at the end of a control has no effect. This
behavior is by design, and is in place to ensure that pressing an
arrow key on the last control in "Edit" mode would not move to the
next record.
```

References

INF: Colon in Field Name Causes Error with Query

Article Number: Q94191
CREATED: 29-DEC-1992
MODIFIED: 29-DEC-1992

Summary:

SYMPTOMS

    You receive 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

    You have a field in a table with a field name that ends in a colon
    (i.e. "FieldName:"), and then do a query that has criteria on that
    field.

RESOLUTION

    If the colon is actually part of the desired field name, rather
    than a label, you must enclose the name in brackets (i.e.
    [FieldName:]).

More Information:

Microsoft Access allows you to rename a field in a query through use
of the colon (:) character. This enables you to describe that field's
data more accurately without having to change the field name in the
underlying table.

You will receive the above error message because Microsoft Access
interprets the colon in the field name as a request to re-name, or
label, that field.

Steps to Reproduce Behavior
----

1. Create a new table named "Test" with one field, named
   FullName:"(without the quotes).

2. Save the table as Table1.

3. Create a Query based on Table1.

4. Drag the field "FullName:" to the query grid.

5. Run the query.

6. You will receive the following error message:

        Query must have at least one output field.

7. Set the criteria for "FullName:" to "a*" (any criteria will work
   for this example).

8. You will receive the following error message:

       Can't set criteria unless you've specified a field.

The error messages in steps #6 and #8 above can be resolved by
enclosing the field name and colon in brackets (i.e. [FullName:]).

Reference(s):

Microsoft Access On-Line Help, search on "queries: renaming fields".

[References](#)

PRB: Incorrect Characters in Crosstab Query Column Headers

Article Number: Q94192
CREATED: 29-DEC-1992
MODIFIED: 29-DEC-1992

Summary:

SYMPTOMS

If the data in a field used as a column header in a crosstab query
contains a decimal, an exclamation mark, or square bracket characters,
these characters are changed to underscore characters in the column
heading. This does not happen if the same data is used as row headers.

CAUSE

Since queries can be used as input to other queries, forms, or
reports, the column headings must be formatted as field names. Since
field names cannot contain ".", "!", "[", or "]", these characters are
converted to underscore characters in the column headings of the
Crosstab query.

STATUS

This behavior is by design.

[References](#)

INF: Attaching to FoxPro Data

Article Number: Q94193
CREATED: 29-DEC-1992
MODIFIED: 29-DEC-1992

Summary:

Microsoft Access Version 1.0 is positioned as not being able to Attach
to FoxPro data, but able to import it. There is an exception to this
rule which would allow a user to attach, rather than import, FoxPro
data.

More Information:

FoxPro uses a DBF file structure, similar to that of dBASE. However,
Microsoft Access cannot attach to most FoxPro files because of differences
memo and index structure. Subsequently, you *can* attach a FoxPro
table as a dBASE IV table if it meets the following conditions:

   * It does not have any memo fields in its structure

   * It is Attached without any FoxPro index files

If one of these conditions is not true, then the FoxPro table must be
imported rather than attached.

References

INF: Converting DataEase 4.2 files to Microsoft Access

Article Number: Q94194
CREATED: 29-DEC-1992
MODIFIED: 29-DEC-1992

Summary:

Microsoft Access does not directly import DataEase data files but you
can export your data to a formatted disk file that Microsoft Access
can import. To export DataEase data you must create a DQL report,
specify the Format as export, and set the Print Style to print to
disk. Choose one of the following export formats to create a file that
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)

The information is this article applies to DataEase versions 4.2, 4.5
and 4.53 a character based relational database system.

More Information:

To create a formatted disk file in DataEase 4.x you will need to
create a DQL (report) following these basic steps. For more detailed
information about creating reports refer to the DataEase
documentation.

In DataEase
----
1. Define the report specifying the fields you wish to include in the
   output.

2. Specify the report Format as Export. The menu option 'Define
   Format' or 'Modify Format' is found under both the 'DQL Advanced
   Processing' and 'Query by Example - Quick Reports' menus.

   Enter the export option by selecting '7. Export' from the list of
   choices. Next you will be presented with a list of 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 one of the formats indicated by the asterisk. Depending on your
   choice, you may be prompted to specify whether field names should be
   included 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.

In Microsoft Access
----
1. Open the database you wish to import the file to.

2. From the File menu select Import.

3. Select the import format that matches the export format you used in
   DataEase.

The following information should be considered when choosing an export
and import format.

Microsoft Access can import Lotus (.wks) files and create a table with
field names if you specify the option to include the field names in
the export process. During the import process you will prompted to
specify if the first row contains field names. Most DataEase data
types should convert to the proper Microsoft Access field types.

If you choose the Variable Length format you will be prompted to enter
the field separator. The comma (,) is commonly used in ASCII text
files and is the default separator Microsoft Access recognizes.

If you choose the Fixed Length format you will need to create an
Import Specification file in Microsoft Access prior to importing the
file.

For additional information registered DataEase users should contact
DataEase International HotLine Support at 1-203-374-2825.

Reference(s):

"Microsoft Access User's Guide," version 1.0, pages 72-81

References

PRB: Unable to use Edit...Paste in Property Sheet

Article Number: Q94218
CREATED: 30-DEC-1992
MODIFIED: 30-DEC-1992

Summary:

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

The user is using the menu commands which do not apply to pop-up windows

STATUS

This behavior is by design.

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.

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 Northwind Traders (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 select
Paste from the Edit command menu, Microsoft Access creates a new label
on the form.

Reference(s):

Microsoft Access On-Line Help, search on "pop-up forms" and "keyboard: editing keys"

[References](#)

PRB: GP Fault Using an Out of Date ODBC Driver

Article Number: Q94219
CREATED: 30-DEC-1992
MODIFIED: 05-JAN-1993

Summary:

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 could cause a problem if the directory where the
older .dll resides is found in the path.

RESOLUTION

Rename or delete the old ODBC.DLL and run the ODBC setup program
from the setup disks that come with Microsfot Access. This will
install a correct version of ODBC.DLL.

References
SQL

INF: BACK Button Disabled After Selecting Wizard Type

Article Number: Q94220
CREATED: 30-DEC-1992
MODIFIED: 30-DEC-1992

Summary:

When you use either the Forms or Reports wizards in Microsoft Access
version 1.0 after you select the type of Wizard to use (Single Column,
Tabular, Graph, etc.), and then select 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 select Cancel and start
the Wizard process all over again.

This behavior is by design.

[References](#)

INF: Access Table Structure is Variable Length

Article Number: Q94221
CREATED: 30-DEC-1992
MODIFIED: 30-DEC-1992

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.

Reference(s):

"Microsoft Access Users Guide," version 1.0, chapter 3, page 38.

References

PRACC9211: Problems with Edit/Find... Dialog Box

Article Number: Q94222
CREATED: 30-DEC-1992
MODIFIED: 30-DEC-1992

Summary:

PROBLEM ID: PRACC9211xxx

SYMPTOMS

There are three known problems with the "Find..." dialog box (under
the Edit menu):

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 checked when using the + and - keys to
   check/uncheck the boxes.

2. There is a slight painting problem when you select the "All Fields"
   option button from the "Search In" group. The default value is
   false for the "Match Case" and "Search Fields as Formatted" check
   boxes, but after clicking the "All Fields" option button, you will
   notice that the "Search Fields as Formatted" check box caption grays
   and also checks the previously unchecked field.

3. After performing the steps above (and seeing the grayed out "X" in
   the "Search Fields as Formatted" check box, if you again select
   "Current Field" from the "Search In" group, you will notice that
   the "Search Fields as Formatted" check box has been incorrectly
   checked. This occurs whether the selections were made by keyboard
   or mouse.

STATUS

   Microsoft has confirmed this to be a problem in Microsoft Windows
   version 3.10. We are researching this problem and will post new
   information here as it becomes available.

References

PRB: Design Can Occur While Machine is Attempting Browse

Article Number: Q94389
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

SYMPTOMS

If you open a table in Microsoft Access in datasheet view and enter
data, you may receive the following error message:

    Control bound to unknown column <controlname>.

CAUSE

The only way that this could occur is if another user opened the
table in design mode at the same time that you opened the table in
datasheet mode. The other user would have to delete the field, or
control, and save the modification to the table before the table is
opened on your machine.

RESOLUTION

This behavior is by design.

More Information:

This behavior is very hard to reproduce. Timing in opening the table
on the two machines is critical.

Steps to Reproduce Behavior
----

1. Simultaneously open a table in design view on one machine and
   datasheet view on a second machine.

2. If timed correctly, the table will open in design view first.
   Delete one of the fields in the table and close the table before the
   table opens in datasheet view on the second machine.

3. On the machine that opens the table in datasheet view, the deleted
   field should be listed. Attempt to enter data in that field. The error
   message should appear.

References

PRB: Press ENTER when Database Window is Minimized Opens a Doc

Article Number: Q94390
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

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 object opened minimized or not, nothing happens when you press ENTER on the Database window icon.

STATUS

Microsoft has confirmed this to be a problem in Microsoft Access version 1.0.We are researching this problem and will post new informatio here as it becomes available.

More Information:

```
Two ways to restore the Database window.
---------------------------------------
```

1. Double click on the icon with the mouse.

2. Use the "Alt" + "-" to bring the Database window's icon menu up and then press ENTER.

References

PRB: Cannot Use "y/n", "t/f" in Yes/No Field

Article Number: Q94391
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

SYMPTOMS

```
You receive the foolowing error message:

    The value you entered isn't approprate for this field.
```

CAUSE

```
You may be entering a small case "y" and "n" into a Yes/No boolean field
```

RESOLUTION

```
You need to enter a capital "Y" or "N".
```

STATUS

```
Microsoft is researching this problem and will post new information here
it becomes available.
```

[References](#)

PRB: Border properties ignored with SpecialEffect

Article Number: Q94392
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

SYMPTOMS

```
The following properties are ignored when you set the SpecialEffects of
control to Sunken or Raised:

    BorderColor
    BorderWidth
    BorderStyle
```

STATUS

```
This is behavior by design.
```

[References](#)

PRB: Welcome to Microsoft Access Screen Disables Status Bar

Article Number: Q94393
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

SYMPTOMS

When you start Microsoft Access with the 'Welcome to Microsoft
Access' screen, the status bar will not show whether the Caps and Num
locks are on or off.

RESOLUTION

If you turn off the 'Welcome to Microsoft Access' screen the status
bar functions normally. Another work around is to close the screen,
then switch the Num and Cap locks off then on. This will reinitialize
the status bar.

More Information:

Turn ON the Caps and Num lock keys. Start Microsoft Access with the
'Welcome to Microsoft Access' screen enabled. Notice that the status
bar doesn't show that the Caps and Num locks are on.

References

PRB: Control Menu Move Command Requires Keyboard Input

Article Number: Q94394
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

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. 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 Northwind Traders (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 successfully moved the Database window.

4. Press ALT+MINUS 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. Now, 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 Reference(s):

"Microsoft Access Getting Started Guide," version 1.0, page 78

References
pallate tool box

PRB: Trailing CR/LF's Stripped From Text Box When Retrieved

Article Number: Q94395
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

SYMPTOMS

Trailing CR/LF's are stripped out when retrieving data from a text box. Although it is possible to stuff CR/LF's into a text box, when attempting to retrieve the data, the trailing CR/LF's are truncated.

STATUS

We are researching this problem to be considered for inclusion in a future release.

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

INF: How to Show Field in Datasheet but not in Form View

Article Number: Q94396
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

There is no inherent command to display a field in datasheet mode, yet make it invisible in form view. With several tricks of changing the color characteristics and the size, you can create a control that will literally be invisible in form view, yet be visible in datasheet view.

More Information:

To create a control that is visible in datasheet view, yet 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 this 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 re-appear.

References

INF: Dynamic Counter in a Query

Article Number: Q94397
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

It is often helpful to have a dynamic counter value in a field of a
query to identify a row position. Microsoft Access has no intrinsic
function to do this, but it can be accomplished easily with a simple
User Defined Function.

More Information:

There are two basic elements to creating dynamic counter values in a
query. A function that increments the counter value and a call to the
function within a query field.

The function should look something like this:

Declarations:

```
   Global Cntr

   Function QCntr(x)
      Cntr = Cntr + 1
      QCntr = Cntr
   End Function
```

In the query grid, the call to the QCntr() function should be:

```
   Field:   Expr: QCntr([Field])
```

Where [Field] is any field from the underlying table or query.
[Field] is passed to the function to force evaluation of the function
for every row in the query dynaset. It does not matter what field is
used as long as it is a valid field name.

Reference(s):

References

PRB: Unable to Import Fixed-Width Text Files

Summary:

SYMPTOMS

You are unable to successfully import any portion of a fixed-width
text file.

CAUSE

One possible cause to investigate is that each record is not
separated by a carriage return or a line feed. Another possibility is
that not all records contain the same number of characters.

STATUS

This behavior is by design.

RESOLUTION

Modify the text file to ensure that all records:

1. Are on different lines, and
2. contain 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 forth. In order to import these records into
Microsoft Access, you will need to ensure that each record is on its
own line with carriage return and line feed at the end of the record.

For example, consider the following string of information:

    lastnamefirstnameagelastnamefirstnameagelastnamefirstnameage

Even selecting fixed-width text, 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 totally unsuccessful and need to repair your
database. To modify the file, you can use Access Basic to:

1. Read in the file in Binary mode with a fixed length string that
   is the same length as the record.

2. Write this string to a new file that is opened for sequential
   OutPut.

3. Continue steps 1 and 2 until you reach the end of file opened
   in step 1.

```
Microsoft Access will successfully import the following records:

    lastnamefirstnameage
    lastnamefirstnameage
    lastnamefirstnameage

To import the records that follow the above structure, you would:

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 that you wish to import, then choose Import.

8. In the Import Text Options dialog box, select the TEST
   specification, then choose OK.


Reference(s):

"Microsoft Access User's Guide", version 1, pages 75-81
```

[References](#)

INF: Printing With the HP Laserjet II in Microsoft Access

Article Number: Q94418
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

If you are printing to an Hewlett-Packard Laserjet II, you will want
to set the option, Print TrueType As Graphics within your Print Setup.
This is necessary because the HP Laserjet II driver does not support
printing True Type graphics.

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:

Some of the problems that may be seen when printing Microsoft Access
reports with this printer when the Print Option has not been set are:

1. A solid black line is printed but the preview displays white text
   on a black line.

2. Reports that have any lines (a graphic) will begin to print with
   text correctly aligned, but subsequent text begins to shift to the far
   right after the line is printed.

For additional information about this printer and Windows 3.1 search the
Knowledge Base for:

    TRUETYPE and GRAHPICS and LASERJET

References

PRB: User is Unable to Attach to a Btrieve Database.

Article Number: Q94419
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

SYMPTOMS

When you try to attach to a Btrieve database you get an error
message when you choose the FILE.DDF:

    Table <tablename> is exclusively locked.
or
    Couldn't open file <filename>.

CAUSE

Whenever Microsoft Access tries to attach to a table, it needs to
open the target table exclusively tempoarily. If the target Btrieve
table is open by any other users at the same time that Microsoft
Access tries to attach to the table, one of the above error messages
will be returned.

RESOLUTION

The only workaround at this time is to find an available time when
the target Btrieve table is not being used by any other users. Once
Microsoft Access has attached to the Btrieve table, you should not
receive the above error messages.

Reference(s):

[References](#)

PRB: Corrupt Error Message with Attached Btreive Table

Article Number: Q94420
CREATED: 06-JAN-1993
MODIFIED: 06-JAN-1993

Summary:

SYMPTOMS

When you try to attach to a Btrieve table you receive the following
error message:

    <tablename> is corrupted or isn't a Microsoft Access database.

CAUSE

When Microsoft Access attaches to a Btrieve table, it compares the
information in the DDF files with the information in the tablename.DAT
file. The most common cause of the above error message is a difference
in the information for the indexes.

RESOLUTION

The user will need to redefine these indexes in a manner that
Microsoft Access will not return the above error message. The most
common solution is to redefine the index to be the exact length of the
field that it is based upon.

More Information:

```
Example #1:
-----------

The tablename.DAT file is defined with 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 which combines the
firstname and lastname fields.

Resolution: This index would need to be redefined into one multiply
segment index, each with a defined length of 10 characters.


Example #2:
-----------

The tablename.DAT file is defined with three columns:

    Column name    Field Length    Index Length
    -------------------------------------------
    Firstname      10              4 characters
    Lastname       10
    Company        25
```

Index2 is defined with a length of 4 characters which is shorter than
the defined length of the firstname field.

Resolution: This index would need to be redefined from 4 characters in
length to 10 characters in length, matching the size of the firstname
field.

Example #3:

The tablename.DAT file is defined with 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 which is slightly
longer than the defined length of the firstname field.

Resolution: This index would need to be redefined from 12 characters
in length to 10 characters in length, matching the size of the
firstname field.

Example #4:

The tablename.DAT file is defined with three columns:

```
Column name    Length
--------------------
Firstname  -   10
Lastname   -   10
Company    -   25
```

Index4 is defined on a numeric byte range and not defined on any
fields in the table.

Resolution: This index would need to be removed completely and then
redefined on an existing field and that has the same defined length as
the field it is based upon.

These types of indexes are valid to the Btrieve file format. Xtrieve
from Novell simply ignores this type of index. Microsoft Access
interrupts these indexes as being invalid and returns the above error
message.

How does a user go about changing the index? The user has several
options:

1. Ask the vendor that created the Btrieve files to change the indexes to
   match the length of the defined fields.

2. Redefine the indexes with Btrieve to match the length of the
   defimed fields.

3. Redefine the indexes with Xtrieve to match the length of the
   defined fields.

4. Use any third party utility that is available to modify the Btrieve
   indexes.

Reference(s):

[References](#)

INF: System Objects Retain System Object Properties

Article Number: Q94597
CREATED: 11-JAN-1993
MODIFIED: 11-JAN-1993

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, and Copy & 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 Object 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 Object to No.

Result: The table Tablefoo disappears from the database container list because it has the attributes of a system object.

References

PRB: Incorrect Action Query Message Box Text

Article Number: Q94598
CREATED: 11-JAN-1993
MODIFIED: 11-JAN-1993

Summary:

SYMPTOMS

```
An error message generated by an action query (i.e. Delete, Make
Table, Update, or Append queries) 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 exact same error message
for every action query instead of a separate error message for each
type of action query.
```

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 as it becomes available.
```

More Information:

```
Steps to Reproduce Problem
--------------------------

1. Create the following two tables ("Table1" and "Table2").

      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.

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. Click OK to delete the records.

8. The following error message will then be 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 incorrectly indicates the loss
of 2 records. In actuality the records were not modified, and
therefore not lost.

[References](#)

INF: How to Place User Login Name in Title Bar

Article Number: Q94599
CREATED: 11-JAN-1993
MODIFIED: 11-JAN-1993

Summary:

In Microsoft Access you can implement security to control user access
to the different objects in your database. You can use the Microsoft
Access Basic 'User()' function to display the current user of these
different objects. This article explains how to use this function to
display the current user name in either a control on a Form or in the
Microsoft Access Window title bar.

More Information:

```
Display user name in a control on a Form
----------------------------------------
```

1. Create a form and place an unbound text box on the form.

2. In the Control Source property enter "=User()" (without the
   quotes).

3. View the form and you will see the current user name displayed in
   the control.

```
Display user name in Microsoft Access Window Title Bar
------------------------------------------------------
```

1. Create a new module.

2. Enter the following Microsoft Access Basic code:

   Note: The '_' is a line continuation character only for this editor.
   You need to combine the lines so that they are entered as one line
   of text.

```
   '====================================================================
   'Declarations section of the module
   '====================================================================
   Option Explicit
   Declare Function SendMessage& Lib "User" (_
                    ByVal hw%, 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 that will have the action item listed below:

```
   Macro Name    Action    FunctionName
   -------------------------------------------
   ShowUser      RunCode   SetDatabase windowWindowTitle()
```

6. From the Run menu, choose Compile All.

7. Restart Microsoft Access and run the macro you created in step 5.
   You should see the window title change from 'Microsoft Access' to
   'User = username'

Reference(s):

Microsoft Access On-Line Help, search on "User" for "user function"

References

PRB: 'Invalid Database Object' Occurs Inexplicably

Summary:

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

1. You should always try to open database objects outside of
   transactions

2. You should always explicitly close database objects.

More Information:

Steps to Reproduce Behavior

The following code fragment will illustrate the problem.

```
Sub Foo()
    Dim MyDB As Database, MyTable As Table

    Set MyDB = CurrentDB()
    BeginTrans
        Set MyTable = MyDB.OpenTable("Table1")
        Call Bar
```

```
        Debug.Print MyTable.RecordCount '"Invalid database object."
        MyTable.Close
    CommitTrans
    MyDB.Close
End Sub

Sub Bar()
    Dim MyDB2 as Database
    Set MyDB2 = CurrentDB()
End Sub
```

MyDB looses scope, implicitly does roll back and close of MyDB with effect of also closing MyDB in the Foo function.

To circumvent the problem, change the procedure Bar to:

```
Sub Bar()
    Dim MyDB2 as Database
    Set MyDB2 = CurrentDB()
    MyDB2.Close
End Sub
```

Reference(s):

References

PRB: Access Microsoft Graph Wizard Does Not Run or Syntax Erro

Article Number: Q94601
CREATED: 11-JAN-1993
MODIFIED: 11-JAN-1993

Summary:

SYMPTOMS

If you choose the Microsoft Graph icon from the toolbox the Graph Wizard
does not run or you get the following error message:

    Syntax Error

CAUSE

This is caused by an incorrect List Separator in your International
section of your WIN.INI file.

RESOLUTION

From your Main group in Windows choose the Control Panel icon.
Choose the International icon from Control Panel. Change the List
Separator to 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:
----

1. Select the Control Panel from your Main group.

2. Choose the International icon.

3. Change the List Separator. Different settings will cause different
   results. For example MINUS(-) will not open graph or give an error
   message. A SPACE will give a Syntax Error.

4. Create a new form.

5. Choose the graph from the toolbox.

6. Select an area on your form and click your mouse to insert the
   graph. The border will briefly appear, but then disappear without
   opening the graph wizard or you will recieve the syntax error message.

[References](#)