

Microsoft* Access 2.x Knowledge Base

Prepared 01/26/95



Categories and Keywords for All Knowledge Base Articles



Microsoft Access KB Categories and Subcategories



Knowledge Base Articles



Sources of More Information




User Survey

THE INFORMATION IN THE MICROSOFT KNOWLEDGE BASE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. MICROSOFT DISCLAIMS ALL WARRANTIES EITHER EXPRESSED OR IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL MICROSOFT CORPORATION OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER INCLUDING DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, LOSS OF BUSINESS PROFITS, OR SPECIAL DAMAGES, EVEN IF MICROSOFT CORPORATION OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES SO THE FORGOING EXCLUSION OR LIMITATION MAY NOT APPLY. Copyright (c) 1994 Microsoft Corporation. All rights reserved.

Microsoft* Access 2.x Knowledge Base



Categories and Keywords for All Knowledge Base Articles

 Categories and Keywords for All Knowledge Base Articles



Microsoft Access KB Categories and Subcategories



Knowledge Base Articles



Sources of More Information



User Survey


Microsoft* Access 2.x Knowledge Base



Categories and Keywords for All Knowledge Base Articles



Microsoft Access KB Categories and Subcategories

 [INF: Microsoft Access KB Categories and Subcategories](#)



Knowledge Base Articles



Sources of More Information








































User Survey

 **Categories and Keywords for All Knowledge Base Articles**

 **Microsoft Access KB Categories and Subcategories**



























 **Knowledge Base Articles**














































-  [INF: "#Name?" or Control is Blank on Form Based on a Query](#)
-  [INF: "Continued..." Header for Groups That Span Multiple Pages](#)
-  [INF: "Form Is Read Only" Error Message](#)
-  [INF: "Incorrect Syntax Near '-'" Using TransferDatabase Action](#)
-  [INF: "n Parameters Were Expected, but Only 0 Were Supplied"](#)
-  [INF: Ability to Add Users Without Permission to View Design](#)
-  [INF: Access Basic Error-Handling Supplemental Information](#)
-  [INF: Adding a Dynamic Counter to a Query to Count Records](#)
-  [INF: Adding a New Record to a Combo Box with a Double-Click](#)
-  [INF: Administrator Setup Does Not Create SETUP Subdirectory](#)
-  [INF: Allowing an Ampersand to Appear in a Caption](#)
-  [INF: Attach R:BASE Table by Exporting dBASE Format in R:BASE](#)
-  [INF: Attached Table Must Be Opened as Dynaset from Modules](#)
-  [INF: Attaching Clipper Tables to Microsoft Access Databases](#)
-  [INF: Base Combo Box on Parameter Query to Filter Values](#)
-  [INF: Btrieve File Owner Names, Passwords, and Security](#)
-  [INF: Cannot Distribute MS Access 2.0 ODBC Driver with ADT Apps](#)
-  [INF: Case Sensitivity in Queries Differs with Attached Tables](#)
-  [INF: Changes in Security in Version 2.0](#)
-  [INF: Compound Indexes Must Restrict First Indexed Field](#)
-  [INF: Concatenating Fields in a Text box to Remove Blank Lines](#)
-  [INF: Connecting to SYBASE SQL Server from an ODBC Application](#)
-  [INF: Connecting to the First Available Network Drive](#)
-  [INF: Conversion of Oracle Data Types to Microsoft Access](#)
-  [INF: Converting Julian Dates with Access Basic](#)
-  [INF: Correct Syntax for TransferDatabase to dBASE and Paradox](#)
-  [INF: Could not Find SYSTEM.MDA or UTILITY.MDA File](#)
-  [INF: Create and Drop Tables and Relationships Using SQL DDL](#)
-  [INF: CreateControl\(\) and CreateReportControl\(\) Functions](#)
-  [INF: Creating a Crosstab Query with Multiple Value Fields](#)
-  [INF: Creating Additional ADT Application Program Groups, Items](#)
-  [INF: Creating Custom Setup That Does Not Create Program Group](#)
-  [INF: Creating Fixed-Scale Graphs in Reports and Forms](#)
-  [INF: Creating Floating Pop-Up Menus](#)
-  [INF: Creating Mailing Labels for Dot-Matrix Printers](#)
-  [INF: Creating Reports to Mail Merge Microsoft Access Data](#)
-  [INF: Creating Virtual Indexes with SQL Data-Definition Queries](#)
-  [INF: Criteria Parameters Require Concatenated References](#)
-  [INF: Currency Format Causes Report Alignment Problems](#)














































- INF: [Custom Domain Functions Similar to DFirst\(\) and DLast\(\)](#)
 - INF: [Custom Menus Cannot Be Implemented with Pop-up Forms](#)
 - INF: [DAO2016.DLL Not Available from External Applications](#)
 - INF: [Data Type of Value Overrides Data Type of Variable](#)
 - INF: [Database Normalization Basics](#)
 - INF: [DatePart\(\), DateDiff\(\) Functions Ignore Global Settings](#)
 - INF: [Debugging & Troubleshooting Tips for Attached SQL Tables](#)
- INF: [Defragment and Compact Database to Improve Performance](#)
 - INF: [Deleting Duplicate Records from Two Tables](#)
 - INF: [Description of Btrieve](#)
 - INF: [Description of Btrieve Section of WIN.INI File](#)
 - INF: [Description of Btrieve/Xtrieve .DDF files](#)
 - INF: [Description of the Visual Basic Compatibility Layer](#)
 - INF: [Description of Vendor Independent Mail \(VIM\)](#)
- INF: [Determining How Many Instances of Application Are Active](#)
 - INF: [Determining the Version of Catalog Stored Procedures](#)
 - INF: [Differences Between Native and Attached Tables](#)
 - INF: [Directory Contents of Service Pack File ACCSVC.EXE](#)
 - INF: [Displaying Start, Finish, and Elapsed Times on a Form](#)
 - INF: [DLookup\(\) Usage, Examples, and Troubleshooting Tips](#)
- INF: [DoCmd Statement in Transaction Not Affected by Rollback](#)
 - INF: [Dynamically Synchronizing Two Forms Using Access Basic](#)
 - INF: [Eliminating White Space in Reports with CanShrink and AB](#)
 - INF: [Enforcing Transactions on Attached SQL Server Tables](#)
 - INF: [Error Messages When Concatenating Variables or Controls](#)
 - INF: [Error Msg "Operation Not Supported on Paradox Table..."](#)
- INF: [Explanation of "Option Compare Database" Statement](#)
 - INF: [Explanation of Compliance Checking Program \(CCP\)](#)
 - INF: [Explanation of the KeepTogether Property for Sections](#)
 - INF: [Export Btrieve Table Without Existing Xtrieve FILE.DDF](#)
 - INF: [Exporting a Parameter Query to a Spreadsheet or Text File](#)
 - INF: [Exporting Tables from MS Word with AccessExporter Macro](#)
 - INF: [Exporting WordPerfect Secondary Files to Microsoft Access](#)
- INF: [Expressions to Count Yes, No, and Other Responses](#)
 - INF: [Field and Record Validation Rule Enforced at Engine Level](#)
 - INF: [Fill New Record with Data from Prev. Record Automatically](#)
 - INF: [Filling a Multiple-Column List Box Using Access Basic](#)
 - INF: [FindEOM\(\) Sample Function to Find Last Day of Month](#)
 - INF: [Fixes to Btrieve ISAM in MS Access Service Pack](#)
 - INF: [Fixes to Paradox ISAM in MS Access Service Pack](#)
- INF: [Fixes to xBASE ISAM in MS Access Service Pack](#)
 - INF: [Form and Report Modules Have Local Scope](#)
 - INF: [Formatting a Label for Each Axis of a Graph](#)
 - INF: [Formatting Immediate IF \(IIF\) Statements in Reports](#)
 - INF: [Four Ways to Move to a Record from a Combo Box Selection](#)

-  [INF: Function to Determine Retail or Run-Time Version](#)
-  [INF: Function to Show Date/Time as Hours, Minutes, and Seconds](#)
-  [INF: Functions to Left and Right Pad Character Strings](#)
-  [INF: Hiding the Database Window Menu Bar](#)
-  [INF: Hints for Printing Mailing Labels](#)
-  [INF: How First Line of Data is Used to Import Delimited Text](#)
-  [INF: How Microsoft Access Deletes Objects from a Database](#)
-  [INF: How Microsoft Access Uses SHARE.EXE](#)
-  [INF: How to Activate an Application with Class Name](#)
-  [INF: How to Add a User to a Group with CreateUser Method](#)
-  [INF: How to Add an Icon to Program Manager Using DDE](#)
-  [INF: How to Add ToolTips to Form Controls](#)
-  [INF: How to Attach All the User Tables on a SQL Server](#)
-  [INF: How to Automatically Detect If a Form Is Being Edited](#)
-  [INF: How to Avoid Abandoned Group Headers in Reports](#)
-  [INF: How to Base Subforms on SQL Pass-Through Queries](#)
-  [INF: How to Bring a Subtotal from a Subform to a Main Form](#)
-  [INF: How to Calculate a Credit Card Expiration Date](#)
-  [INF: How to Calculate Averages Excluding Zero Values](#)
-  [INF: How to Call Functions Using a String Variable](#)
-  [INF: How to Change Default Template for Forms and Reports](#)
-  [INF: How to Change the "Microsoft Access" Window Caption](#)
-  [INF: How to Change the Caption on the Microsoft Access Window](#)
-  [INF: How to Change the Font for the Zoom Box](#)
-  [INF: How to Change the Order of Columns in a Graph](#)
-  [INF: How to Change the Status Bar Text Using SysCmd\(\)](#)
-  [INF: How to Check a Menu Item Using Access Basic](#)
-  [INF: How to Check for Duplicate Values in Primary Key Fields](#)
-  [INF: How to Communicate with NetDDE](#)
-  [INF: How to Compare Data Among Records in a Query](#)
-  [INF: How to Control the Number of Records Printed Per Page](#)
-  [INF: How to Convert Currency or Numbers into English Words](#)
-  [INF: How to Convert Twips to Pixels](#)
-  [INF: How to Correctly Back Up Multiuser Databases](#)
-  [INF: How to Create a Custom Startup "Splash" Screen](#)
-  [INF: How to Create a Custom Toolbar Using a Form](#)
-  [INF: How to Create a Cyclic Tab Order in a Form](#)
-  [INF: How to Create a Function to Capture Screens of Your Forms](#)
-  [INF: How to Create a Multiple-Selection List Box](#)
-  [INF: How to Create a Multiuser Custom Counter](#)
-  [INF: How to Create a Parameter In\(\) Statement](#)
-  [INF: How to Create a Running Sum on a Form](#)
-  [INF: How to Create a Top 10 Report](#)
-  [INF: How to Create an AfterUndo Form Event](#)
-  [INF: How to Create an SQL Pass-Through Query in Access Basic](#)






































- INF: [How to Create Custom Domain Function Similar to DCount\(\)](#)
 - INF: [How to Create Custom Navigation \(VCR\) Buttons on a Form](#)
 - INF: [How to Create For Next Loops In a Macro](#)
 - INF: [How to Create Synchronized Combo Boxes](#)
 - INF: [How to Create Tasks in MS Project for Windows Using DDE](#)
 - INF: [How to Create, Debug, and Use a Microsoft Access Library](#)
 - INF: [How to Declare C++ Functions Called from Access Basic](#)
- INF: [How to Delete Duplicate Records from a Table](#)
 - INF: [How to Detect When MS Access Is Activated or Deactivated](#)
 - INF: [How to Determine If a Form Exists in a Database](#)
 - INF: [How to Determine if a Specific Windows Program Is Running](#)
 - INF: [How to Determine If a Table or Query Exists](#)
 - INF: [How to Determine If Current Database Is Open Exclusively](#)
 - INF: [How to Determine If Current User Is in a Particular Group](#)
- INF: [How to Determine Number of Working Days Between Two Dates](#)
 - INF: [How to Determine the Current Screen Resolution \(Size\)](#)
 - INF: [How to Determine Version of INSTCAT.SQL File on Server](#)
 - INF: [How to Determine Whether a Menu Item Is Checked](#)
 - INF: [How to Determine Which Snaking Column Is Current](#)
 - INF: [How to Dial a Phone Number in Microsoft Access](#)
 - INF: [How to Dim \(Gray\) Menu Items with Access Basic](#)
- INF: [How to Direct DDE to a Specific Instance of a Server](#)
 - INF: [How to Disable PAGE UP and PAGE DOWN Keys in a Form \(2.0\)](#)
 - INF: [How to Display Intervals Greater Than 24 Hours](#)
 - INF: [How to Display Line Numbers on Subform Records](#)
 - INF: [How to Dynamically Create a Filter](#)
 - INF: [How to Edit Library Databases](#)
 - INF: [How to Fill a List Box with Database Object Names](#)
- INF: [How to Fill an Array Using Access Basic](#)
 - INF: [How to Filter a Subform by Changing the Record Source](#)
 - INF: [How to Find a Record in a Form Using Access Basic](#)
 - INF: [How to Find a Record Using a Bound Control \(2.0\)](#)
 - INF: [How to Find Access Basic Error Codes](#)
 - INF: [How to Find If an Object Exists in the Database](#)
 - INF: [How to Find the Number of Days In a Month](#)
- INF: [How to Force a New Line in a MsgBox Statement](#)
 - INF: [How to Format Data in Combo Boxes and List Boxes](#)
 - INF: [How to Get and Use the Microsoft Access Window Handle](#)
 - INF: [How to Get Form Window's Right and Down Measurements](#)
 - INF: [How To Get Microsoft Works Files into Microsoft Access](#)
 - INF: [How to Get Record Position Number from Record Selector](#)
 - INF: [How to Get the Name and Path of the Current Database](#)
- INF: [How to Give Users the Ability to Change Passwords](#)
 - INF: [How to Group Column Headings in a Crosstab Query](#)
 - INF: [How to Hide the Combo Box Drop-Down Arrow on a Form](#)

-  [INF: How to Hide the Database Window at Startup](#)
-  [INF: How to Import a Microsoft Windows Cardfile Data File](#)
-  [INF: How to Import FoxBase for Macintosh Files](#)
-  [INF: How to Import Several dBASE Databases Simultaneously](#)
-  [INF: How to Index an Existing Field with DAO](#)
 -  [INF: How to Make a Combo Box Default to First Item in List](#)
 -  [INF: How to Make Report Footer Print at Bottom of Last Page](#)
 -  [INF: How to Make the Immediate Window Available at All Times](#)
 -  [INF: How to Minimize, Maximize, and Restore Microsoft Access](#)
 -  [INF: How to Move List Box Selections to a Second List Box](#)
 -  [INF: How to Open UTILITY.MDA](#)
-  [INF: How to Optimize Microsoft Access Version 2.0 Performance](#)
 -  [INF: How to Optimize Microsoft Access Version 2.0 Queries](#)
 -  [INF: How to Optimize MS Access Using ODBC Data Sources](#)
 -  [INF: How to Pass an Array As an Argument to a Procedure](#)
 -  [INF: How to Play .WAV Sounds on Events in Microsoft Access](#)
 -  [INF: How to Position Microsoft Access on the Screen](#)
 -  [INF: How to Prevent Users from Creating New Databases](#)
-  [INF: How to Print a Blank Line Every Nth Line in a Report](#)
 -  [INF: How to Print a Conditional Page Header in a Report](#)
 -  [INF: How to Print a Constant Number of Lines Per Group](#)
 -  [INF: How to Print a Single Record from a Form in a Report](#)
 -  [INF: How to Print Duplicate Mailing Labels for Each Record](#)
 -  [INF: How to Print Group Footer at Specific Location](#)
 -  [INF: How to Print Odd or Even Pages of a Report](#)
-  [INF: How to Print Text to Default Printer with Access Basic](#)
 -  [INF: How to Programmatically Embed or Link an Object in a Form](#)
 -  [INF: How to Query an Oracle Table Without Attaching It](#)
 -  [INF: How to Query on Category Keywords](#)
 -  [INF: How to Query SQL Server Tables Without Attaching](#)
 -  [INF: How to Rank Records Within a Query](#)
 -  [INF: How to Refer to a Control on a Subform or Subreport](#)
-  [INF: How to Reference Controls or Fields on a Subform](#)
 -  [INF: How to Repair a Damaged Database](#)
 -  [INF: How to Repeat Subreport Header at Top of Page](#)
 -  [INF: How to Replace a String with Another String](#)
 -  [INF: How to Replace the Default Input Mask Error Message](#)
 -  [INF: How to Reset Toolbar Options](#)
 -  [INF: How to Retrieve the Path of an Attached MS Access Table](#)
-  [INF: How to Retrieve Windows For Workgroups User Information](#)
 -  [INF: How to Run Access Basic Functions with DDE](#)
 -  [INF: How to Search for Characters or Strings in Records](#)
 -  [INF: How to Select Controls on Subforms with SetFocus Method](#)
 -  [INF: How to Send Information to the Clipboard](#)
 -  [INF: How to Shell to a Windowed MS-DOS Application](#)














































-  [INF: How to Shrink and Grow a Rectangle in a Report](#)
-  [INF: How to Simulate Multiple Record Selections in a Form](#)
-  [INF: How to Skip Used Mailing Labels and Print Duplicates](#)
-  [INF: How to Specify a Custom Starting Page Number for a Report](#)
-  [INF: How to Speed Up Iterative Processes in Access Basic](#)
-  [INF: How to Start the AutoDialer Without the Toolbar Button](#)
-  [INF: How to Start the Mail Merge Wizard Without the Toolbar](#)
-  [INF: How to Store Subform Totals in a Main Form Field](#)
-  [INF: How to Synchronize Two Combo Boxes on a Form](#)
-  [INF: How to Trap Keystrokes that Leave a Control on a Form](#)
-  [INF: How to Trap SQL ODBC Login Cancel Error](#)
-  [INF: How to Uninstall Microsoft Access Version 2.0](#)
-  [INF: How to Update a Table Based on Values in a Second Table](#)
-  [INF: How to Use a Query to Filter Unique Data](#)
-  [INF: How to Use a Secured Database in Word Mail Merge](#)
-  [INF: How to Use an Import/Export Spec. in Another Database](#)
-  [INF: How to Use Command Line Switches](#)
-  [INF: How to Use DAO to Assign or View Permissions](#)
-  [INF: How to Use DDE to Communicate with a Run-Time Application](#)
-  [INF: How to Use DDE to Pass Information to MS Access 2.0](#)
-  [INF: How to Use Get, WritePrivateProfileString](#)
-  [INF: How to Use Graphic Images as Wallpaper for Forms](#)
-  [INF: How to Use In-Place Activation with OLE Objects](#)
-  [INF: How to Use OLE Automation to Modify MS Graph Object](#)
-  [INF: How to Use OLE Automation to Transfer Data to MS Excel](#)
-  [INF: How to Use Seek Method with Multiple-Field Index](#)
-  [INF: How to Use SendKeys in a Macro to Change an OLE Link](#)
-  [INF: How to Use the Attributes Property for TableDef Objects](#)
-  [INF: How to Use the ColumnWidth Property](#)
-  [INF: How to Use the GetSystemMetrics\(\) API Call](#)
-  [INF: How to Use the If\(\) \(Immediate If\) Function](#)
-  [INF: How to Use the LIKE Operator in Parameter Queries](#)
-  [INF: How to Use the Query-by-Form \(QBF\) Technique](#)
-  [INF: How to Use the Status Bar Progress Meter](#)
-  [INF: How to Use the Windows OpenFile Dialog Box](#)
-  [INF: How to Use TimeValue\(\) to Calculate Difference in Times](#)
-  [INF: How to Wait for a Shelled Process to Finish](#)
-  [INF: Implementing Counter Values on Attached SQL Server Tables](#)
-  [INF: Imported Microsoft Excel Date Fields Are Off by 4 Years](#)
-  [INF: Importing an Entire Sheet from a MS Excel 5.0 Workbook](#)
-  [INF: Importing DataEase Data Files](#)
-  [INF: Importing Word Print Merge Data Files to Microsoft Access](#)
-  [INF: Importing, Exporting, and Attaching FoxPro 2.0, 2.5 Data](#)
-  [INF: Incrementing the Numeric Portion of a String](#)
-  [INF: Introduction to Windows Programming for MS-DOS Programmer](#)










-  [INF: Launching Doc with Windows API ShellExecute\(\) Function](#)
-  [INF: Like vs. = with Wildcard Characters in Query Searches](#)
-  [INF: Limit on AND's in SQL Select Statement](#)
-  [INF: Limitations on Setting Report Properties at Run Time](#)
-  [INF: List of Articles Relating to MSACCESS.INI File](#)
-  [INF: List of Categories and Examples from SOLUTION.MDB](#)
-  [INF: List of Constants in Macros and Modules](#)
-  [INF: List of New 2.0 Form/Report Events, Methods, & Properties](#)
-  [INF: List of OLE 2.0 Features Supported in MS Access v. 2.0](#)
-  [INF: Listing Object Names Using Data Access Objects \(DAO\)](#)
-  [INF: Local MS Access Installation Required by ADT](#)
-  [INF: Macro to Check for Duplicate Values in a Field](#)
-  [INF: Macro to Enter Date Record Was Last Modified](#)
-  [INF: Microsoft Access 2.0 Disk Contents \(3.5", 1.44 MB\)](#)
-  [INF: Microsoft Access 2.0 Readme Help File Contents](#)
-  [INF: Microsoft Access and ORACLE ODBC Drivers](#)
-  [INF: Microsoft Access and Untested Networks](#)
-  [INF: Microsoft Access Does Not Use Record Numbers](#)
-  [INF: Microsoft Access KB Categories and Subcategories](#)
-  [INF: Microsoft Access KB Help Files for Versions 1.x and 2.x](#)
-  [INF: Microsoft Access Repair Utility Enhanced](#)
-  [INF: Microsoft Access, Scanners, and Scanning Software](#)
-  [INF: Microsoft ADT Disk Contents \(3.5 Inch, 1.44 MB\)](#)
-  [INF: Microsoft Graph Changes May Not Be Retained](#)
-  [INF: Modified Toolbar Information Stored In SYSTEM.MDA File](#)
-  [INF: MS Access Version 2.0 Service Pack Questions and Answers](#)
-  [INF: Network Users Can Open Their Own SYSTEM.MDA Exclusively](#)
-  [INF: New Events and Properties in Microsoft Access v. 2.0](#)
-  [INF: No Custom Printer Setup for Mailing Labels ReportWizard](#)
-  [INF: Nulls Allowed in Foreign Key with Referential Integrity](#)
-  [INF: Number of Times a Custom Function Executes in a Query](#)
-  [INF: Obtain Name of Current User Database Using Access Basic](#)
-  [INF: ODBC Setup for Microsoft Access and SQL Server](#)
-  [INF: OLE 2.02 Required to Use Custom Controls](#)
-  [INF: OpenDatabase and CloseDatabase Pseudo Actions](#)
-  [INF: Opening Databases as Read-Only Causes Error Message](#)
-  [INF: OpenTable Method on Attached Table Causes Error](#)
-  [INF: Overview of Conversion from Version 1.1 to 2.0 Issues](#)
-  [INF: Overview of MS Access Developer's Toolkit \(ADT\) v. 2.0](#)
-  [INF: Overview of New Query Features in Version 2.0](#)
-  [INF: Overview of Version 2.0 Add-ins, Builders, and Wizards](#)
-  [INF: Page Header Will Span Full Width Of Multi-Column Report](#)
-  [INF: Page Locking, No Cluster Indexing Support](#)
-  [INF: Paradox Tables, Indexes, and Multiple Connections](#)
-  [INF: Part 2 DDE in Visual Basic to Request Data from MS Access](#)








































- INF: Passing Values from a Form to a Parameter Query
 - INF: Placing the User Login Name on a Form or in the Title Bar
 - INF: Print a Sequential Number for Each Record/Group on Report
 - INF: Programmatically Add Counter Field to Table Using DAO
 - INF: Query by Form (QBF) Using Dynamic QueryDef
 - INF: Query with Parameters to Evaluate Complex Criteria
 - INF: Questions and Answers About .LDB Files
- INF: Reading, Storing, & Writing Binary Large Objects (BLOBs)
 - INF: README.TXT File and Novell Locking Issues
 - INF: Referencing Fields in Previous or Next Record in a Form
 - INF: Relationship Between Btrieve and Microsoft Access
 - INF: Repeat Group Name at the Top of New Column or Page
 - INF: Requery, Repaint, and Records Refresh
 - INF: Requirements to Connect to SYBASE SQL Server
- INF: Returning Case-Sensitive Match in Queries
 - INF: Round or Truncate Values to Desired Number of Decimals
 - INF: Row Fix-up in the NWIND.MDB Order Form
 - INF: Running a MS Access Macro from Visual Basic Using DDE
 - INF: Running MS-DOS SHARE with Windows for Workgroups
 - INF: Sample Code for Running Temporary SQL Pass-Through Query
 - INF: Sample DDE Macro Communicates with Microsoft Excel 4.0
- INF: Sample Expressions to Extract Portion of Text String
 - INF: Sample Function CopyFile() to Copy Disk Files
 - INF: Sample Function for Summing Date/Time Field
 - INF: Sample Function to Capitalize Words and Phrases
 - INF: Sample Function to Create a Running Sum on a Form
 - INF: Sample Function to Determine Current Page of a Form
 - INF: Sample Function to Determine the Access Startup Directory
- INF: Sample Function to Determine the Existence of a Query
 - INF: Sample Function to Find File Associations
 - INF: Sample Function to Generate a Random Temporary File Name
 - INF: Sample Function to Make ENTER Add Lines in a Text Box
 - INF: Sample Function to Obtain the Last Word in a String
 - INF: Sample Function to Open a Form to a Particular Record
 - INF: Sample Function to Provide Password for Attached Tables
- INF: Sample Function to Remove Alphas from a Numeric Field
 - INF: Sample Function to Replace Special Characters
 - INF: Sample Function to Retrieve File's Date and Time Stamp
 - INF: Sample Function to Return a Random Record
 - INF: Sample Function to Use the COMMDLG Color Dialog Box
 - INF: Sample Functions to Check User, Group Information (2.0)
 - INF: Sample Functions to Send MCI Commands from Access Basic
- INF: Sample Macro for Repair/Compact Operations
 - INF: Sample Macro to Reset Page Number on Group Level Report
 - INF: Sample MSACC20.INI File for MS Access Version 2.0



-  [INF: Sample OLE Automation for MS Word and MS Excel](#)
-  [INF: Sample Table Design to Support Questionnaire Applications](#)
-  [INF: Sending a MAPI Mail Message Using Access Basic](#)
-  [INF: Set Focus to a Subform Control Using GoToControl Action](#)
-  [INF: Setup Wizard Cannot Create Compact and Repair Icons](#)
-  [INF: Setup Wizard OLE Default Dependent on MSARN200 Choice](#)
-  [INF: Shortcut Method for Creating an SQL Union Query](#)
-  [INF: Showing All Records \(Including Null\) in a Parameter Query](#)
-  [INF: Shrinking Empty or Null OLE Object on Report](#)
-  [INF: Sorting String Values Based on their Numeric Values](#)
-  [INF: Sorting Titles Without Leading Articles \(The, An, etc.\)](#)
-  [INF: Sorting/Grouping a Query Using a Portion of Field's Value](#)
-  [INF: SQL Server Views Read-Only When Attached](#)
-  [INF: Storing Calculated Values Using ControlSource Property](#)
-  [INF: Storing SQL Database Login IDs and Passwords Locally](#)
-  [INF: Summary List of Known Issues for Version 2.0](#)
-  [INF: Three Methods for Testing the VBCL](#)
-  [INF: Tips for Debugging Access Basic Code](#)
-  [INF: Tips for Improving Combo and List Box Performance](#)
-  [INF: Tips for Improving Subform Performance](#)
-  [INF: Tips for Optimizing Queries on Attached SQL Tables](#)
-  [INF: Tips on How to Troubleshoot Microsoft Access Macros](#)
-  [INF: Topics Supported by Microsoft Access as a DDE Server](#)
-  [INF: TraceSQLMode Setting Helps Debug SQL Queries to ODBC](#)
-  [INF: Trapping SQL Server RAISERROR\(\) Function Values](#)
-  [INF: Troubleshooting Setup and Installation Problems](#)
-  [INF: Troubleshooting Tips for Error Values](#)
-  [INF: Two Functions to Calculate Age in Months and Years](#)
-  [INF: Two Methods to Refer to a Field in a Previous Record](#)
-  [INF: Understanding Event Order in MS Access 2.0](#)
-  [INF: Updating One Table with Data from a Second Table](#)
-  [INF: Updating, Troubleshooting Novell NetWare VLM Installation](#)
-  [INF: Upper Limit for QueryTimeout and LoginTimeout Parameters](#)
-  [INF: Use a Query to Parse a Full Name into Two Fields](#)
-  [INF: Use Append Query to Set Initial Value of Counter Field](#)
-  [INF: Use AutoLookup Technique to Look Up Info. Automatically](#)
-  [INF: Using "<", ">", and "=" Operators in Query Parameters](#)
-  [INF: Using a Conditional Macro to Confirm Changes to a Field](#)
-  [INF: Using Access Basic to Derive a Statistical Median](#)
-  [INF: Using Column Property of Combo Box to Update Text Box](#)
-  [INF: Using Data from MS Access in a WordPerfect Mail Merge](#)
-  [INF: Using Date/Time Data Type and Calculating Elapsed Time](#)
-  [INF: Using DLookup\(\) to Look Up Values in Forms and Reports](#)
-  [INF: Using Find Method to Find a Quotation Mark or Apostrophe](#)
-  [INF: Using IIf\(\) in Crosstab Query to Limit Column Headings](#)

- INF: Using Microsoft Access 2.0 Databases with Visual Basic
 - INF: Using Microsoft Access Macro to Quit Windows
 - INF: Using Microsoft Word to Edit and Spell Check Memo Field
 - INF: Using OLE Automation to Change a Graph's Type
 - INF: Using SetSysModalWindow API Call in Access Basic
 - INF: Using SHARE.EXE and VSHARE.386 with Microsoft Access 2.0
 - INF: Using SQL Server ODBC Driver with Sybase Servers
- INF: Using the DDE Item SQLText to Request the Text of a Query
 - INF: Using the Generic Print Driver with Forms/Reports
 - INF: Using the INSTCAT.SQL Script with an ODBC Application
 - INF: Using the Scroll Bar Control to Move Through Records
 - INF: Using the Windows 3.1 API to Connect to Network Resources
 - INF: Using Undocumented CreateForm(), CreateReport() Functions
- INF: Using Variables in the Criteria Row of the QBE Grid
 - INF: Using Version 1.1 Databases in Version 2.0
 - INF: Using WIN.INI & ODBC.INI When Attaching SQL Server Table
 - INF: Validation Rule to Accept Only Alphabetic Characters
 - INF: Why OLE Objects Cause Databases to Grow
 - INF: Word for Windows Print Merge Using Microsoft Access Data
 - INF: Writing Functions Called from Events or Expressions
- INF: Xtrieve Data Dictionary (.DDF) File Information
 - PRA: "Can't Place Item at This Position" Error Message
 - PRA: "Couldn't Find Object '<Name>' with SQL Server Table
 - PRA: "Invalid Reference" Saving Record on Form with Button
 - PRA: "No Permission for DBSTRUCT.MDB" Error Message
 - PRA: "Unexpected Error from External Database Driver [7]"
 - PRA: 1.x Table/Query Datasheet Properties Not Converted to 2.0
- PRA: ADT Application Setup Always Defaults to Drive C
 - PRA: ADT Application Setup Fails with Error -1809
 - PRA: ADT Splash Screen Does Not Get Repainted
 - PRA: Attached Paradox 4.0 Table Displays "#Deleted"
 - PRA: AutoDialer Saves Only Five Characters of Prefix
 - PRA: Calendar Control Font and Color Properties Unavailable
 - PRA: Can Print Macro Definition Without Read Design Permission
- PRA: Cannot Add Record After Changing DefaultEditing Property
 - PRA: Cannot Close Form from Control's OnExit Property
 - PRA: Cannot Connect to SQL Server w/ NetWare, Workgroups
 - PRA: Cannot Delete Templates Created with the Setup Wizard
 - PRA: Cannot Edit Record from Two Forms in One Session
 - PRA: Cannot Embed OLE Objects Using Access Basic Code
 - PRA: Changes to Print Setup Not Saved with Form or Report
 - PRA: Compile Errors Not Reported When Form Opened with Code
- PRA: Database Documentor Generates Incorrect Form Information
 - PRA: Database Documentor Incorrectly Reports Permissions
 - PRA: Empty Recordset Report Hangs with Group KeepTogether














-  [PRA: Error "Setup Cannot Create SYSTEM.MDA File \(-1809\)"](#)
-  [PRA: Error \(-1310\) Running Crosstab or Totalling Query](#)
-  [PRA: Error Deleting Value from Attached SQL Server Table](#)
-  [PRA: Error In SQL Pass-Through Action Query Not Detected](#)
-  [PRA: Error Message "Couldn't Open SYSTEM.MDA"](#)
 -  [PRA: Error Message "Internal Database Error -1310"](#)
 -  [PRA: Error Message "Undefined Function in Expression"](#)
 -  [PRA: Error Previewing Subreports that Call Filter Macro](#)
 -  [PRA: Error When Import dBASE, FoxPro, or Paradox Table](#)
 -  [PRA: Errors After Reinstalling MS Access to Workstation](#)
 -  [PRA: Export to MS Excel Uses Version 4.0 File Format](#)
-  [PRA: Exporting Data to MS Excel 5.0 Twice Causes Error](#)
 -  [PRA: Expression Builder Creates Incorrect Section Syntax](#)
 -  [PRA: Extra White Space Between Last Section and Report Footer](#)
 -  [PRA: GP Fault Creating Report Based on Query with Subquery](#)
 -  [PRA: GP Fault in STORAGE.DLL with Forms Switchboard Form](#)
 -  [PRA: GP Fault Using Query Wizards and Mono VGA Video Driver](#)
 -  [PRA: GPF Closing Data Outline Control Properties Box](#)
-  [PRA: GPF Using List Box Fill Function with Requery Method](#)
 -  [PRA: Graph 5.0 Legend Moves Each Time Graph Is Refreshed](#)
 -  [PRA: Help Topic "Converting Code" Sample Code Incorrect](#)
 -  [PRA: Horizontal/Vertical Lines Missing from 600 dpi Printout](#)
 -  [PRA: Imported Microsoft Excel Data Contains Wrong Columns](#)
 -  [PRA: Input Mask Wizard Not Available with All DataTypes](#)
-  [PRA: Mailing Label Wizard Creates One-Page Report](#)
 -  [PRA: Navigation Buttons Not Enabled, Disabled Correctly](#)
 -  [PRA: No Serial Number In the About Microsoft Access Dialog Box](#)
 -  [PRA: No Visible Property for Form Page Header/Footer Sections](#)
 -  [PRA: ODBCTimeout Ignores MSACC20.INI ODBC Setting](#)
 -  [PRA: ODBCTimeout Set to Zero for New Pass-Through Queries](#)
 -  [PRA: OLE Objects Clipped Using Fahrenheit Video Driver](#)
-  [PRA: OnError Property Does Not Trap Locking Error in Form](#)
 -  [PRA: Only One Page Created for Legal Size Form or Report](#)
 -  [PRA: Options in Form and Report Wizards Are Not Displayed](#)
 -  [PRA: Oracle Driver Errors on Delimited Column Names](#)
 -  [PRA: Ordinal Number in SQL ORDER BY Clause Ignored](#)
 -  [PRA: Print Definition Report Truncated](#)
 -  [PRA: Printer Fonts Cause Text to Shift Down and Right](#)
-  [PRA: Report Based on Fast Query Seems to Run Slowly](#)
 -  [PRA: Report Output to RTF Format Contains Blank Lines](#)
 -  [PRA: Report Section with Subreport Not Included In Output File](#)
 -  [PRA: Reserved Error -1001 with Linked Crosstab Subreport](#)
 -  [PRA: Resources Lost Printing Reports with OLE Objects](#)
 -  [PRA: RunningSum Text Box in Report Not Accumulating Values](#)
 -  [PRA: SETUP /Q Always Installs Complete Version](#)

-  [PRA: Subreport Placed Low on Main Report Causes Blank Pages](#)
 -  [PRA: Text Moves When Locked and Enabled Properties Set to No](#)
 -  [PRA: Toolbar Not Updated When Switching Window Types](#)
 -  [PRA: TransferDatabase Fails in Code, Not in Immediate Window](#)
 -  [PRA: Update or Delete Query Fails Without Generating Error](#)
 -  [PRA: Words Split When CanGrow=Yes and KeepTogether=No](#)
 -  [PRA: WRKGADM.EXE File Installed Without Being Specified](#)
-  [PRB: GoToRecord Macro Does Not Work on Subform](#)
 -  [PRB: "#Error" Message Referencing Subreport Values \(2.0\)](#)
 -  [PRB: "#ERROR" Message When Referencing Subreport Controls](#)
 -  [PRB: "Can't Bind Name '\[XXX\]'" Error with Crosstab Query](#)
 -  [PRB: "Can't Have More Than 10 Fields In an Index" Error Msg](#)
 -  [PRB: "Can't Restore Field's Previous Value..." Error Message](#)
 -  [PRB: "Cannot Change...SETUP.INI" Installing on Novell Server](#)
-  [PRB: "Couldn't Find File WIZARD.MDA" Error Message](#)
 -  [PRB: "Couldn't Find Installable ISAM" Error Message](#)
 -  [PRB: "Couldn't Find Object <Tablename>" Error with Btrieve](#)
 -  [PRB: "Couldn't Insert or Paste; Data Too Long for Field" Msg](#)
 -  [PRB: "Couldn't Lock File; SHARE.EXE Hasn't Been Loaded" Error](#)
 -  [PRB: "Couldn't Open File" Error Message Using Paradox Table](#)
 -  [PRB: "Couldn't Update; Currently Locked by User" Error Message](#)
-  [PRB: "Disk or Network Error" Err. Msg. Attaching Btrieve Data](#)
 -  [PRB: "Error Reading Table or Query '<Name>'" Error Message](#)
 -  [PRB: "Help Topic Does Not Exist" Invoking Custom Help File](#)
 -  [PRB: "Internal Database Error \(-5001\)" Msg Using Btrieve NLM](#)
 -  [PRB: "Invalid Entries in \[Btrieve\] Section..." Error Message](#)
 -  [PRB: "Invalid Operation" Error Msg. Running QueryDef Example](#)
-  [PRB: "Not Enough Memory to Run SETUP.EXE" Error Message](#)
 -  [PRB: "ODBC Call Failed" When Exporting Table to SQL Server](#)
 -  [PRB: "ORA-00904: Invalid Column Name" Error Message in Query](#)
 -  [PRB: "ORA-9241" Error Deleting Then Attaching ORACLE Table](#)
 -  [PRB: "Out of Memory" Error Message Loading Large Module](#)
 -  [PRB: "Out of Memory" or "Query Too Complex" Running Report](#)
-  [PRB: "Out of Memory" or Low System Resources Printing Lines](#)
 -  [PRB: "Output To" of Print Definition Contains Only Headers](#)
 -  [PRB: "Record Lock Threshold Exceeded" with Large Action Query](#)
 -  [PRB: "Save Changes to Form" Prompt When No Changes Made](#)
 -  [PRB: "Setup Parse Error..." Error Message in Run-Time Setup](#)
 -  [PRB: "Unexpected Error from External Database Driver \[20\]"](#)
 -  [PRB: "Unexpected Error from External Database Driver \[22\]"](#)
-  [PRB: 'Unable to Load Communication Module'](#)
 -  [PRB: AccessWizards Button Unavailable](#)
 -  [PRB: Adding Records to AutoLookup Form Generates Error](#)
 -  [PRB: ADT Repair/Compact Documentation Correction](#)
 -  [PRB: Append Query Results in Duplicate Counter Numbers](#)

-  [PRB: AutoLookup Query Not Performed with 1-to-1 Relationship](#)
-  [PRB: Avery 4xxx-Series Labels Do Not Work on Laser Printer](#)
-  [PRB: BeforeUpdate Macro Running Twice](#)
-  [PRB: Blank Lines in Exported Report Formatted Incorrectly](#)
-  [PRB: Btrieve ISAM Is Not Installed with Typical Setup](#)
-  [PRB: Calendar Control Cannot Format Individual Dates](#)
-  [PRB: Cannot Add Security Items to Custom Menu](#)
-  [PRB: Cannot Attach Paradox Table on Read-Only Drive](#)
-  [PRB: Cannot Choose Custom Bitmap for Toolbar Button](#)
-  [PRB: Cannot Clear Selection in Combo Box](#)
-  [PRB: Cannot Connect DDE from Word 6.0 to Microsoft Access](#)
-  [PRB: Cannot Delete Admin User](#)
-  [PRB: Cannot Delete Attached Table with Relationship](#)
-  [PRB: Cannot Display More Than Four Graph Objects at Once](#)
-  [PRB: Cannot Enforce Referential Integrity in Relationship](#)
-  [PRB: Cannot Open Multiple Instances of the Same Form](#)
-  [PRB: Cannot Open Paradox 4.x Table](#)
-  [PRB: Cannot Perform OpenTable Method on an Attached Table](#)
-  [PRB: Cannot Process Close Action in Report's OnOpen Property](#)
-  [PRB: Cannot Run Setup for Solutions Pack on Workstation](#)
-  [PRB: Cannot Search for Name with Apostrophe in Find Customers](#)
-  [PRB: Cannot Share Some Microsoft Access Wizards](#)
-  [PRB: Cannot Sum Calculated Controls in Forms or Reports](#)
-  [PRB: Cannot Use Counter to Determine If Record Is New Record](#)
-  [PRB: Cannot Use SQL Pass-Through Command for Subreport](#)
-  [PRB: Change Option Affects Only One Instance of MS Access](#)
-  [PRB: Changing Control Properties Affects All Records in a Form](#)
-  [PRB: Changing Printer Port Resets Printer Setup Information](#)
-  [PRB: Choosing a New Printer Resets Some Printer Settings](#)
-  [PRB: Closing Forms, Tables, or Queries Is Slow](#)
-  [PRB: Compile Error When Running ADT Application](#)
-  [PRB: Concatenation of Memo Fields Creates Text Field](#)
-  [PRB: Creating Multiple New Import/Export Specifications](#)
-  [PRB: Crosstab Query Returns "<>" as a Column Heading](#)
-  [PRB: Data Not Sorted In Index Field Order](#)
-  [PRB: Data Outline Control Events Not Triggered](#)
-  [PRB: Data Type Error Inserting Data Outline Control](#)
-  [PRB: DateAdd\(\) "w" Interval Does Not Work As Expected](#)
-  [PRB: DateDiff "W" Option Does Not Work](#)
-  [PRB: DateDiff\(\) Function Returns Incorrect Value](#)
-  [PRB: DeleteObject Action Causes "Illegal Function Call" Msg](#)
-  [PRB: Detail Section Blank When Form Is Printed or Previewed](#)
-  [PRB: DFirst\(\) and DLast\(\) Functions Return Unexpected Records](#)
-  [PRB: Editing Allowed Although AllowEditing Set to Unavailable](#)
-  [PRB: Empty Table List when Attaching Btrieve Table](#)

-  [PRB: Err Msg "Can't Set Value" Adding Record to Subform](#)
-  [PRB: Err: "Data Source Not Found, No Default Driver Specified"](#)
-  [PRB: Error In Help Files for Custom Control Refresh Method](#)
-  [PRB: Error Message "Couldn't Open File" in Shared Environment](#)
-  [PRB: Error Message "Unable to Connect to Data Source"](#)
-  [PRB: Error Message When Importing Informix Delimited Text File](#)
-  [PRB: Error Message When Opening Second Instance of .MDB File](#)
 -  [PRB: Error Message: "This Action Cannot Be Completed..."](#)
 -  [PRB: Error Msg. "Error Occurred in WriteSTF" in Setup Wizard](#)
 -  [PRB: Error Msg: "Can't Open File 'C:\ACCESS\WZLIB.MDA'"](#)
 -  [PRB: Error Occurs in Print Preview Without Driver](#)
 -  [PRB: Errors Using Recordset Methods on Data Outline Control](#)
 -  [PRB: Every Other Page Blank in Printed Reports](#)
-  [PRB: Export Table to Other Database Type Does Not Create Index](#)
 -  [PRB: Exporting Date Fields to Text Includes Time Format](#)
 -  [PRB: Exporting to Fixed-Width Text File Left Justifies Numbers](#)
 -  [PRB: Field Object Type Property Returns "Invalid Operation"](#)
 -  [PRB: Field Properties Not Inherited with Make-Table Query](#)
 -  [PRB: Form Name Error Message Using Data Outline Control](#)
 -  [PRB: Form Opened in Form View Is Completely Blank](#)
-  [PRB: Form Size Changes Not Retained](#)
 -  [PRB: Forms and Reports Print Slowly, Large Temporary Files](#)
 -  [PRB: GP Fault Attaching Tables with Old CTL3D.DLL File](#)
 -  [PRB: GPF with Diamond Viper VESA Local Bus Video Driver V1.01](#)
 -  [PRB: Imported MS Excel Carriage Returns Become Vertical Bars](#)
 -  [PRB: Incorrect Catalog Stored Procedures Error Message](#)
 -  [PRB: Incorrect Results Received with DSum\(\) Function](#)
-  [PRB: Index Missing for Attached FoxPro or dBASE Table](#)
 -  [PRB: Input Mask Character \(!\) Does Not Work Correctly](#)
 -  [PRB: Invalid Path Error Message Using UNC with 10+ Characters](#)
 -  [PRB: Invalid Reference to Field '<Value>' Using SetValue](#)
 -  [PRB: Irregular Characters in Attached dBASE IV Memo Field](#)
 -  [PRB: LimitToList Enforced Even When Set to No](#)
-  [PRB: Locking First Record Also Locks Last Record](#)
 -  [PRB: Mail Label Wizard Cuts Off First Field at Top of Page](#)
 -  [PRB: Make-Table Query Generates "Invalid" Error Message](#)
 -  [PRB: Maximizing One Form Maximizes All Forms](#)
 -  [PRB: Memo and OLE Fields in a Snapshot Are Not Static](#)
 -  [PRB: Memory or Disk Space Problems Using MS-DOS 6.0](#)
 -  [PRB: MS Access Converts Extended Chars in Attached FoxPro DBF](#)
-  [PRB: Network Problems May Cause "Segment Load Failure" Error](#)
 -  [PRB: New Label Controls Inherit FormWizards' Label Text Align](#)
 -  [PRB: No AutoRepeat with GoToRecord/Next and DoMenuItem](#)
 -  [PRB: No Password Prompt Using Microsoft Access .MDB with ODBC](#)
 -  [PRB: No Results When Calling ORACLE 7.0 Stored Procedure](#)

-  [PRB: No Way to Specify Page Range Using Output To Command](#)
-  [PRB: NotInList Event Is Not Triggered](#)
 -  [PRB: Office Manager 4.2 Will Not Load Microsoft Access 2.0](#)
 -  [PRB: OLE MS Excel, Word Do Not Close When Form Closed](#)
 -  [PRB: OnExit Macro Runs with AfterUpdate Macro in Option Group](#)
 -  [PRB: Only Admins Group Members Can List Groups They Belong To](#)
 -  [PRB: OpenRecordset on Recordset Causes "Invalid Operation"](#)
 -  [PRB: Operation Stops when Editing Attached SQL Tables](#)
-  [PRB: ORA-00942 Error Opening Attached ORACLE Table](#)
 -  [PRB: Outer Join with Where Clause Returns Unexpected Records](#)
 -  [PRB: Parameter Limit for Access Basic Functions in Validation](#)
 -  [PRB: Paste Unavailable After Record Is Copied to Clipboard](#)
 -  [PRB: Permissions Command Unavailable When Logged in as Guest](#)
 -  [PRB: Printing on Avery 5267 labels](#)
 -  [PRB: Prompt to Save Libraries After Replace Operation](#)
-  [PRB: Query by Form Returns No Rows When All Rows Expected](#)
 -  [PRB: Record Sort Order Is Different Than Expected](#)
 -  [PRB: RecordCount Property Incorrect Using OpenTable Method](#)
 -  [PRB: RecordCount Property Returns Incorrect Number of Records](#)
 -  [PRB: Removing ADT Does Not Unregister Custom Controls](#)
 -  [PRB: Replace Command Does Not Find NULL Values](#)
 -  [PRB: Report or Form Bound to a Table Locks the Table](#)
-  [PRB: Rounding Errors Importing/Exporting dBASE IV Values](#)
 -  [PRB: Rounding Errors Using Floating-Point Numbers](#)
 -  [PRB: SCSI Drives Can Generate Corrupt Error Loading Database](#)
 -  [PRB: Seek Method Is Faster Than Find Method](#)
 -  [PRB: SendKeys Macro Action Does Not Work on Toggle Keys](#)
 -  [PRB: Setting KeyCode=0 for KeyDown Does Not Ignore All Keys](#)
-  [PRB: Setup Wizard Cannot Find VSHARE.386 with NT](#)
 -  [PRB: Setup Wizard Templates Lost Installing Service Pack](#)
 -  [PRB: Shortcut Menus Cannot Be Disabled In Microsoft Access](#)
 -  [PRB: Some Labels Skipped in Mailing Label Report](#)
 -  [PRB: Stored procedure 'PROCEDURE' not found -- Error Message](#)
 -  [PRB: Strange Characters Appear in Imported dBASE IV Database](#)
 -  [PRB: Subform Not Updated](#)
-  [PRB: Subreport or Subform Data Missing from "Output To" Result](#)
 -  [PRB: Syntax Error on Page 196 of "Advanced Topics" Manual](#)
 -  [PRB: Syntax Errors in "Advanced Topics" Sample Functions](#)
 -  [PRB: System Error When Sharing Database on LAN Man/Workgroups](#)
 -  [PRB: Type Conversion Failure when Importing Excel Data](#)
 -  [PRB: Unable to Change Permissions \(Security\)](#)
 -  [PRB: Unable to Import Portion of Fixed-Width Text Files](#)
-  [PRB: UndoCurrentRecord In BeforeUpdate Property Causes Error](#)
 -  [PRB: Unexpected Cursor Behavior in Text Box with Input Mask](#)
 -  [PRB: Unexpected Results with Make-Table or Append Query](#)

-  [PRB: Unknown Reference or Function in Validation Expression](#)
-  [PRB: Update Query Based on Totals Query Fails](#)
-  [PRB: User Needs Read/Write Privileges for WORKDIR Directory](#)
-  [PRB: Validation Rules Evaluate Expressions in Unexpected Ways](#)
-  [PRB: White Lines or Missing Data When Printing Reports](#)
-  [WX0994: Form and Report Questions and Answers](#)
-  [WX0995: Table and Query Questions and Answers](#)
-  [WX0996: Setup and Conversion Questions and Answers](#)
-  [WX0997: Interoperability Questions and Answers](#)
-  [WX0998: Macro and Module Questions and Answers](#)
-  [WX0999: MS Access Dev. Toolkit Questions & Answers](#)
-  [WX1051: Security Wizard and White Paper App. Note 2.0](#)
-  [WX1124: Microsoft Access Version 2.0 Service Pack](#)



Sources of More Information



User Survey

 Access 2.x Knowledge Base



Categories and Keywords for All Knowledge Base Articles



Microsoft Access KB Categories and Subcategories



Knowledge Base Articles



Sources of More Information

 **More Information About Knowledge Base (KB) Help**



User Survey

 Access 2.x Knowledge Base



Categories and Keywords for All Knowledge Base Articles



Microsoft Access KB Categories and Subcategories



Knowledge Base Articles



Sources of More Information



User Survey

 **Microsoft Knowledge Base Help File Survey**

Categories and Keywords for All Knowledge Base Articles

Article ID: Q94671

CATEGORIES AND KEYWORDS FOR ALL KNOWLEDGE BASE ARTICLES

To categorize articles within the Microsoft Knowledge Base and make finding information easier, a common set of keywords has been adapted for use throughout the KB. These keywords are used to categorize the subject of each article. Each article has one SUBJECT keyword. A secondary keyword indicating the TYPE of article can also be used.

To query on one category, enter the category code and a wildcard. For example:

kb3rdparty*

To query on more than one category, enter each category code with a wildcard. For example:

kb3rdparty* or kbinterop* or kbusage*

To query on one subcategory, enter the complete code. For example:

kbuglist

To query on more than one subcategory, enter all the complete codes. For example:

kbuglist or kbfixlist

You can also mix categories and subcategories in one query. For example:

kbshoot or kbnetwork*

NOTE: At the end of this article is a table that lists formerly used keywords and their replacements.

INFORMATION CATEGORIES AND KEYWORDS

Article subject	Keyword
Interactions with third-party products	kb3rdparty
Interoperability with other MS products	kbinterop
How to use product features/functionality	kbusage
Printing	kbprint
Setup and installation issues	kbsetup
Networking	kbnetwork
Environment and configuration information	kbenv
OLE technology	kbole
Tools, utilities, applets, such as MS Draw or Write	kbtool
Programming	kbprg
User interface	kbui
Graphics	kbgraphic

Multimedia	kbmm
Hardware	kbhw
Microsoft At Work	kbatwork
Sound (audio) issues	kbsound
Display (video, monitor, resolution) issues	kbdisplay
Readme files	kbreadme
Lists of vendor phone numbers, disk directory listings, other lists or references	kbref
Support boundaries, policies, processes & procedures	kbpolicy
Other: Any subject not covered in other categories	kbother

Article types Keyword

Application Notes	kbappnote
Bug info for a particular version of a product	kbbuglist
Sample code	kbcode
Documentation errors	kbdocerr
Error message followup information	kberrmsg
FastTip scripts or maps	kbfasttip
Binary file information located in the Microsoft Software Library	kbfile
Fixed bug info for a particular version of a product	kbfixlist
Articles listing product-specific keywords	kbkeyword
Sample macro with article	kbmacro
Problems not classified as bugs	kbprb
FastTip Technical Library Catalog	kbtlc
Troubleshooting information	kbtshoot

The following table lists formerly used keywords and their replacements. For explanations of what these keywords represent, see the tables above.

Old keyword(s)	New keyword
3rdparty, isv, 3rd party	kb3rdparty
phoneref, diskdir	kbref
errmsg, err msg	kberrmsg
docerr, doc err	kbdocerr
buglist	kbbuglist
fixlist	kbfixlist
softlib	kbfile
tshoot	kbtshoot
appnote	kbappnote
fasttip, ivr	kbfasttip
tlc	kbtlc

MORE INFORMATION

=====

Future querying tools may allow for use of friendly names for information categories. With current searching tools, "printing" cannot be used as a keyword because that word is also used in many articles whose main subject is not printing. Until our tools allow searching on keywords separately from article text, we need to create unique spellings for Knowledge Base

keywords. Therefore, all standard Knowledge Base keywords will begin with the letters "kb."

Additional reference words: kbkeyword key word kbcdg dskbguide

KBCategory: kbref

KBSubcategory:

INF: Microsoft Access KB Categories and Subcategories

Article ID: Q119526

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Every Knowledge Base (KB) article contains a category and a subcategory. You can query on the categories and subcategories to help you find specific groups of articles. The categories and subcategories are abbreviated and concatenated to form unique words that you can query on. For example, FmsCmbo stands for Forms, Combo/List Box, where Forms is the category and Combo/List Box is the subcategory. Note that the first letter of the category and subcategory are capitalized; this does not affect your queries, but makes it easier to recognize the subcategory.

See the "References" section later in this article for information about how to enter queries based on the categories and subcategories listed in this article.

MORE INFORMATION

=====

The following list contains all the categories and subcategories used in Microsoft Access KB articles:

Conversion/Compatibility	2.0 Behaves Differently Than 1.x	CnvDif20
Conversion/Compatibility	Convert fails on a db	CnvNodb
Conversion/Compatibility	ConvertErrors table	CnvErtbl
Conversion/Compatibility	Errors while running Convert	CnvErrs
Conversion/Compatibility	General	CnvGnrl
Conversion/Compatibility	Other	CnvOthr
Conversion/Compatibility	Performance of Convert Utility	CnvPrfm
Conversion/Compatibility	Problems with conversion	CnvProb
Conversion/Compatibility	Rewriting/Optimizing for 2.0	CnvOpt20
Conversion/Compatibility	Running 1.x databases from 2.0	CnvRun1x
Conversion/Compatibility	Secured Databases	CnvScrdb
Documentation/Sample Apps	Cue Cards	DcmCucd
Documentation/Sample Apps	Help	DcmHlp
Documentation/Sample Apps	Manuals	DcmMnl
Documentation/Sample Apps	NWIND	DcmNwd
Documentation/Sample Apps	Orders	DcmOrds
Documentation/Sample Apps	Other	DcmOthr
Documentation/Sample Apps	Solution	DcmSltn
Environ/Config	GP Fault	EvnGpf
Environ/Config	Hardware	EvnHdwr
Environ/Config	Networks	EvnNtw
Environ/Config	Operating System	EvnOs
Environ/Config	Other	EvnOthr
Environ/Config	Other Drivers	EvnOdr
Environ/Config	System Performance	EvnPrfm
Environ/Config	TEMP Directory	EvnTemp

Environ/Config	TSR Problems	Evntsr
Environ/Config	Video Drivers	EvnVido
Expressions	Expression Builder	ExrBlDr
Expressions	Functions	ExrFnc
Expressions	General	ExrGnrl
Expressions	Nulls and Zero-length strings	ExrNls
Expressions	Other	ExrOthr
Expressions	Referencing Access Objects	ExrObj
Expressions	String manipulation	ExrStrg
Forms (Specific)	Button Design Builder	FmsButb
Forms (Specific)	Combo/List Box	FmsCmbo
Forms (Specific)	Combo/List Box Builder	FmsCmbob
Forms (Specific)	Custom Help/Help Compiler	FmsHlp
Forms (Specific)	Events (Calling Macro/Mods)	FmsEvt
Forms (Specific)	Form Properties	FmsProp
Forms (Specific)	How To	FmsHowto
Forms (Specific)	Option Group Builder	FmsOptB
Forms (Specific)	Option Groups	FmsOpt
Forms (Specific)	Other	FmsOthr
Forms (Specific)	Problems	FmsProb
Forms (Specific)	Requery/Refresh	FmsRqry
Forms (Specific)	SubForm	FmsSubf
Forms (Specific)	Wizard: AutoForm/Quick Form	FmsWiza
Forms (Specific)	Wizard: Main/Subform	FmsWizm
Forms/Reports (Common)	Code Behind Forms/Reports	FmrCdbeh
Forms/Reports (Common)	Code Builder	FmrCodeb
Forms/Reports (Common)	Color Builder	FmrColrb
Forms/Reports (Common)	Colors/Color Palette	FmrColr
Forms/Reports (Common)	Default Properties/Template	FmrProp
Forms/Reports (Common)	Field List	FmrFld
Forms/Reports (Common)	How To	FmrHowto
Forms/Reports (Common)	Other	FmrOthr
Forms/Reports (Common)	Problems	FmrProb
Forms/Reports (Common)	Toolbox	FmrTlxb
FastTips Q&A scripts		Fstqa
General Functionality	Application Design	GnlApp
General Functionality	Compact/Repair	GnlCmp
General Functionality	Corrupted Database	GnlBaddb
General Functionality	Data Entry/Editing	GnlDe
General Functionality	Database Documentor/Print Def	GnlDbdoc
General Functionality	Encrypt/Decrypt	GnlEncr
General Functionality	File Manipulation	GnlFilem
General Functionality	Find/Replace	GnlFnd
General Functionality	Formatting	GnlFrmt
General Functionality	Global Settings (View Options)	GnlGlobl
General Functionality	Input Masks	GnlInpt
General Functionality	MSACC20.INI	GnlAcini
General Functionality	MultiUser/Locking	GnlMu
General Functionality	Other	GnlOthr
General Functionality	Performance Overall	GnlPrfm
General Functionality	Quick Sort	GnlQksrt
General Functionality	Reserved Words	GnlRsvd
General Functionality	Validation	GnlValid
Interoperability	Custom Controls	IntpCstm
Interoperability	DDE	IntpDde
Interoperability	Graph	IntpGrph
Interoperability	Mail (Send from File menu)	IntpMail

Interoperability	OLE	IntpOle
Interoperability	OLE Automation	IntpOlea
Interoperability	Other	IntpOthr
Interoperability	Print Merge/Mail Merge	IntpPrtm
Interoperability	VB Mapping Layer	IntpVb
Interoperability	Word 6.0/Merge It	IntpWrd6
ISAM	Btrieve	IsmBtrv
ISAM	Excel 2.0-4.0	IsmExl4
ISAM	Excel 5.0	IsmExl5
ISAM	Fox/xBase	IsmXbase
ISAM	General	IsmGnrl
ISAM	How To	IsmHowto
ISAM	Imp/Exp/Attach other Access MDBs	IsmIea
ISAM	Import/Export Setup (specs)	IsmSetup
ISAM	Import/Export Text (delimited)	IsmTxd
ISAM	Import/Export Text (fixed width)	IsmTxfx
ISAM	Lotus	IsmLotus
ISAM	Other	IsmOthr
ISAM	Paradox	IsmPdox
ISAM	Problems	IsmProb
ISAM	Works	IsmWorks
Knowledge Base	Category Codes	KbCodes
Macros	Action Arguments	McrArg
Macros	Actions	McrActn
Macros	Condition Column	McrCond
Macros	General	McrGnrl
Macros	How To	McrHowto
Macros	Macro Builder	McrBldr
Macros	Macro Name	McrName
Macros	Other	McrOthr
Macros	Problems	McrProb
Modules	Adding/deleting records	MdlAdrec
Modules	DAO	MdlDao
Modules	Declarations/DIM	MdlDecl
Modules	Editing Records	MdlEdit
Modules	Find Methods	MdlFind
Modules	General	MdlGnrl
Modules	Immediate Window	MdlImmed
Modules	Libraries (MDA)	MdlLib
Modules	Navigating	MdlNav
Modules	Other	MdlOthr
Modules	Performance	MdlPrfm
Modules	Problems	MdlProb
Modules	Querydefs	MdlQry
Modules	Recordsets	MdlRcd
Modules	Transactions	MdlTran
Modules	Using Bookmarks	MdlBkmk
Modules	VB Compatibility	MdlVb
Non Product	Chatter	NpdChat
Non Product	Compliment/complaint	NpdCmnt
Non Product	CompuServe	NpdCis
Non Product	Legal/Licensing issues	NpdLegal
Non Product	Manufacturing Issues	NpdMfg
Non Product	Non MS Product	NpdNotms
Non Product	Other	NpdOthr
Non Product	Post-Sales Request	NpdPstsl
Non Product	PSS Support	NpdPss

Non Product	Sales Support	NpdSales
ODBC	Access 2.0 ODBC Driver	ObcAc20
ODBC	Driver Pack	ObcDrvpk
ODBC	How To	ObcHowto
ODBC	MSysConf	ObcMsys
ODBC	Non-MS Drivers	ObcNonms
ODBC	Oracle	ObcOracl
ODBC	Other	ObcOthr
ODBC	Problems	ObcProb
ODBC	SQL Server: MS	ObcSqlms
ODBC	SQL Server: Sybase	ObcSqlsy
Output To	AutoStart option	OtpAuto
Output To	General	OtpGnrl
Output To	Microsoft Excel/Analyze It	OtpExl
Output To	MS-DOS Text (*.txt)	OtpTxt
Output To	Other	OtpOthr
Output To	Problems	OtpProb
Output To	Rich Text Format (*.rtf)	OtpRtf
Printing	Controlling Printing Options	PtrOpt
Printing	Other	PtrOthr
Printing	P. Preview doesn't match output	PtrPrev
Printing	Print Dialog Box	PtrDialog
Printing	Print Fixed Font Output	PtrFxfnt
Printing	Print Setup	PtrSetup
Printing	Printing Graphs/Lines	PtrGraph
Printing	Printing Limitations	PtrLimit
Printing	Problems/Performance	PtrProb
Programming/Logic	API/DLL Calls	PgmApi
Programming/Logic	Error Handling/Debugging	PgmErr
Programming/Logic	How To	PgmHowto
Programming/Logic	Object Manipulation	PgmObj
Programming/Logic	Other	PgmOthr
Programming/Logic	Parsing Strings and Files	PgmParse
Programming/Logic	Performance/Optimization	PgmPrfm
Programming/Logic	Process Control	PgmPrs
Queries	Append	QryAppnd
Queries	Criteria	QryCrit
Queries	Crosstab	QryCross
Queries	Delete	QryDel
Queries	Filters	QryFiltr
Queries	General	QryGnrl
Queries	How To	QryHowto
Queries	Joins	QryJoin
Queries	Make Table	QryMktbl
Queries	Other	QryOthr
Queries	Parameters/Referencing	QryParm
Queries	Performance	QryPrfm
Queries	Problems	QryProb
Queries	Properties	QryProp
Queries	Rushmore	QryRush
Queries	SQL View/Syntax questions	QrySqlvw
Queries	SQL: Data Definition query	QrySqldd
Queries	SQL: Pass-Through query	QryPass
Queries	SQL: Union query	QryUnion
Queries	Top N	QryTopn
Queries	Totals/Aggregate Functions	QryTotal
Queries	Updatability/Row Fix-Up	QryFixup

Queries	Update	QryUpdat
Queries	Wizard: Archive	QryWiza
Queries	Wizard: Crosstab	QryWizc
Queries	Wizard: Find Duplicates	QryWizd
Queries	Wizard: Find Unmatched	QryWizu
Relationships	Can't save multiple layouts	RltNosav
Relationships	How to create relationships	RltCreat
Relationships	How to delete relationships	RltDel
Relationships	How to view relationships	RltView
Relationships	Need mouse to join tables	RltMous
Relationships	On attached Access tables	RltAttac
Relationships	Other	RltOthr
Relationships	Referential Integrity	RltRef
Reports (Specific)	Envelopes	RptEnv
Reports (Specific)	Events	RptEvent
Reports (Specific)	Layout	RptLayou
Reports (Specific)	Mailing Labels	RptLabel
Reports (Specific)	Other	RptOthr
Reports (Specific)	Page Numbers	RptPagen
Reports (Specific)	Problems	RptProb
Reports (Specific)	Report Properties	RptProp
Reports (Specific)	Sorting and Grouping	RptSort
Reports (Specific)	SubReports	RptSub
Reports (Specific)	Totaling	RptTotal
Reports (Specific)	Wizard: AutoReport/Quick Report	RptWiza
Reports (Specific)	Wizard: Groups/Totals	RptWizg
Reports (Specific)	Wizard: Single-Column	RptWizsi
Reports (Specific)	Wizard: Snaking Columns	RptWizsn
Reports (Specific)	Wizard: Summary	RptWizsu
Reports (Specific)	Wizard: Tabular	RptWizt
Security	Changing ownership	ScrtOwnr
Security	Default accounts	ScrtDef
Security	Implementing	ScrtImpl
Security	Other	ScrtOthr
Security	Permissions	ScrtPerm
Security	Users/Groups	ScrtUsr
Security	Workgroup Administrator	ScrtAdm
Setup	Compliance Checking	StpComp
Setup	Errors starting Access	StpErr
Setup	How To	StpHowto
Setup	Install Options (Typical, etc.)	StpOptio
Setup	Install: General Issues	StpGnrl
Setup	Install: Side by side (1.x&2.0)	StpSide
Setup	Install: Update (2.0 over 1.x)	StpUpdat
Setup	Maintenance Install	StpMaint
Setup	Network Install	StpNet
Setup	ODBC Setup	StpOdbc
Setup	Other	StpOthr
Setup	Registration errors (REG.DAT)	StpReg
Setup	Setup fails	StpFail
Tables	Counters	TblCount
Tables	Data Types	TblDatyp
Tables	Database Design	TblDsign
Tables	Field Builder	TblFldb
Tables	Field Properties	TblFldp
Tables	How To	TblHowto
Tables	Modify	TblModfy

Tables	Other	TblOthr
Tables	Primary Keys/Indexes	TblPriky
Tables	Problems	TblProb
Tables	Wizard: Table	TblWiz
User Interface	Database window (container)	UifDbwin
User Interface	Datasheet view	UifDatash
User Interface	Dialog Boxes	UifDialog
User Interface	Keyboard	UifKeybd
User Interface	Menu Builder	UifMenub
User Interface	Menus	UifMenu
User Interface	Other	UifOthr
User Interface	Problems	UifProb
User Interface	Right mouse menus	UifRtmous
User Interface	Toolbars	UifToolbr
Wizards	Add-in Manager	WzAdmgr
Wizards	Attachment Manager	WzAtmgr
Wizards	Auto Dialer Builder	WzAutod
Wizards	General	WzGnrl
Wizards	How To	WzHowto
Wizards	Import Database	WzImport
Wizards	Other	WzOthr
Wizards	Problems	WzProb

REFERENCES

=====

For more information about how to query the KB using categories and subcategories, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q119525

TITLE : INF: How to Query on Category Keywords

For more information about how to query using the Find button, click the Find button, then click the Hints button in the Search dialog box.

Additional reference words: 1.00 1.10 2.00 kbkeyword

KBCategory: kbref

KBSubcategory: KbCodes

INF: "#Name?" or Control is Blank on Form Based on a Query
Article ID: Q92683

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

If you have a form based on a query, and the query contains multiple fields with the same name (such as FieldA from Table1 and FieldA from Table2), the control(s) on the form that are bound to that field name will be blank or may display the error "#Name?".

MORE INFORMATION

=====

Because the same field name appears multiple times in the query that the form is based on, Microsoft Access does not know which field to link to the control(s). Therefore, instead of making an assumption about which field to bind to, it leaves the control(s) linked to duplicate field names blank or displays the error "#Name?" in the control.

To correct this problem, rename the field in the query.

To change a field name in a query:

1. In a query's Design view, place the insertion point in front of the first letter of the field name in the QBE grid.
2. Type the new name followed by a colon. Delete any names assigned by Microsoft Access, but do not 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 does not change.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, page 103

Additional reference words: 1.00 1.10 2.00 poundname kbtshoot

KBCategory: kbusage

KBSubcategory: QryOthr

INF: "Continued..." Header for Groups That Span Multiple Pages
Article ID: Q88156

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to create a label that will print at the top of each report page when a data group spans multiple pages.

The events used in this technique capture the value of the group in the page footer. If this value is the same as the page header value on the next page, a label containing "Continued from previous page..." is printed at the top of the page.

MORE INFORMATION

=====

The steps below use the sample database NWIND.MDB to demonstrate how to create a label that will print at the top of each report page when a data group spans multiple pages:

1. Open the NWIND database and create a new module.
2. Enter the following global Declarations and functions in the module.

NOTE: In the following sample code, underscores (_) are used as line-continuation characters. Remove the underscores when re-creating this code in Access Basic.

```
'-----  
'Global Declarations  
'-----  
Global CurrentGroupVal as String  
  
'-----  
'Code Section  
'-----  
  
Function SetGlobalVar (InReport as Report)  
    CurrentGroupVal = InReport!SetGroupVal  
End Function  
  
Function SetContinuedLabel (InReport as Report)  
    If InReport.Page <> 1 then  
        InReport!ContinuedLabel.Visible = _  
            IIf (Trim(InReport!CheckGroupVal) = _  
                Trim(CurrentGroupVal)), True, False)  
    End If  
End Function
```


3. Open the List Of Products By Category report in Design view.
4. Increase the height of the detail section to approximately 1 inch.
5. Increase the height of the page header section by about 0.5 inch. Add a label to the page header section. Position the label below the existing fields. Set the label's Caption property to "Continued from previous page..." and set its ControlName property to ContinuedLabel. (NOTE: In Microsoft Access version 2.0, the ControlName property is called the Name property.) Set the label's Visible property to No.
6. From the View menu, choose Field List.
7. Create a text box bound to the Category Name field by dragging the Category Name field to the page header section. Change the text box's ControlName property to CheckGroupVal, and set its Visible property to No.
8. Click anywhere in the page header section not occupied by a control to display the page header properties in the property sheet. Set the page header's OnPrint property to:

 =SetContinuedLabel (Report)
9. Create a text box bound to the Category Name field in the page footer section. Change the text box's ControlName property to SetGroupVal, and set its Visible property to No.
10. Click anywhere in the page footer section not occupied by a control. Set the page footer's OnPrint property to:

 =SetGlobalVar (Report)
11. Preview the report. Note the "Continued from previous page..." message at the top of any page whose category details span more than one page.

Note that the steps above are for illustrative purposes only; your report may require adjustments other than those above. You may want to set the section KeepTogether properties to Yes to keep sections from being split across pages. In the example above, you may also want to set the Category Name header section's ForceNewPage property to Before Section to help control the page layout.

In Microsoft Access version 2.0, there is a new GroupKeepTogether property that you can use to keep groups of records together, or to keep the group header with the first record in the group. You may want to experiment with this property to see how it affects your report's layout.

Additional reference words: 1.00 1.10 2.00 container
KBCategory: kbusage
KBSubcategory: RptSort

INF: "Form Is Read Only" Error Message

Article ID: Q98666

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SYMPTOMS

=====

The message "Form is read only" in Microsoft Access version 1.x or "This Recordset is not updatable" in Microsoft Access version 2.0 is displayed when you are trying to update data, but the form's DefaultEditing property is not set to Read Only.

CAUSE

=====

If your form is based on a query with a property set to Read Only, you will not be able to modify data in the form.

MORE INFORMATION

=====

Steps to Reproduce Behavior

1. Create a new query based on the Order Details table in the sample database NWIND.MDB.

Query: Orders

Type: Select Query

Join: None

Field: Quantity

Total: Group By

Field: Unit Price

Total: Group By

Field: Expr1: [Unit Price]*[Quantity]

Total: Expression

2. Run the query, and then try to change the Unit Price in any record. The computer beeps, and the status bar displays "Form is read only."

You will get the same result when you try to update a form based on this query. If you remove the Total line in the query grid you will be able to modify the records.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.x, Chapter 6, "Designing Select Queries"

Microsoft Access "User's Guide," version 2.0, Chapter 11, "Designing Select

Queries"

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsOthr

INF: "Incorrect Syntax Near '-'" Using TransferDatabase Action
Article ID: Q106244

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

When you are running a TransferDatabase macro action to attach a SQL Server table, the following error message may be generated:

Connection Failed. SQL State 37000. SQL Server Error 102. Incorrect Syntax Near '-'.

MORE INFORMATION

=====

This error is generated because there is an error in the database name argument, or connect string, of the TransferDatabase macro action.

The Microsoft Access Help system suggests that the connect string can be copied from the table properties of the attached table. When you copy the connect string into the database name argument, a string similar to the following example is produced:

NOTE: In the following example, an underscore () is used as a line-continuation character. Remove the underscore when re-creating this example.

```
ODBC;DSN=MyServer;UID=User2;APP=Microsoft Access;WSID=User2; _  
DATABASE=pubs-->-->dbo.authors
```

The syntax error is caused by the text after the "pubs" database name. All text after the database name should be removed from the string, and the table name should be moved to the source argument.

The following example demonstrates correct arguments for the TransferDatabase action:

```
Transfer Type: ATTACH.  
Database Type: <SQL DATABASE>.  
Database Name: ODBC;DSN=MyServer;UID=User2;APP=Microsoft Access; _  
                WSID=User2;DATABASE=pubs  
Object Type: TABLE  
Source: DBO.AUTHORS  
Destination: DBO_ AUTHORS  
Structure Only: NO
```

REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 472-474

Additional reference words: 1.00 1.10 2.00 transfer
KBCategory: kbusage kberrmsg
KBSubcategory: ObcOthr

INF: "n Parameters Were Expected, but Only 0 Were Supplied"
Article ID: Q105522

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Before you can run a parameter query in Access Basic, you must set the values of all the parameters. If the parameters are not set, you will receive the error message:

```
# parameters were expected, but only 0 were supplied.
```

Below is a description of the syntax for setting the value of a parameter that is a form reference. An example is also included.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

To set the value of a parameter that is a form reference, use the following syntax:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Dim MyDB As Database
Dim MyQDef As QueryDef

Set MyDB = CurrentDB()
Set MyQDef = MyDB.OpenQueryDef("Parameter Query")
MyQDef![Forms!Form Name!ControlName] = Forms![Form _
    Name]![ControlName]
```

In the example above, the definition variable, the exclamation point, and the parameter, which is enclosed in brackets, are to the left of the equal sign. Note that if the form name or control name in a form reference contains spaces, it is usually enclosed in brackets. Do not include the brackets if you are setting the value of the form reference parameter. However, do include the brackets if you are referencing the form listed to the right of the equal sign.

The next example demonstrates how to create a query that will prompt you to enter the date when you run the query:

```
Dim MyDB As Database
```

```
Dim MyQDef As QueryDef

Set MyDB = CurrentDB()
Set MyQDef = MyDB.OpenQueryDef("Parameter Query")
MyQDef![Please enter date:] = "#12/12/93#"
```

NOTE: If you have more than one parameter in the query, add a line similar to the preceding line for each parameter.

The following example uses the Orders table from the sample database NWIND.MDB:

1. Create the following query in the NWIND.MDB database:

```
Query: Customer Orders Parameter Query
-----
Type: Select Query
Field: Customer ID
      Table: Orders
      Criteria: [Forms]![Search Form]![Customer To Find]
Field: Order ID
      Table: Orders
Field: Order Date
      Table: Orders
```

2. Create the following form bound to the Orders table:

```
Form: Search Form
-----
Text box:
  ControlName: Customer To Find
  ControlSource: Customer ID
Command button:
  ControlName: Button0
  Caption: ParamQD
  OnPush (or On Click): =ParamQD()
```

3. Create a new module and add the following code:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****
' Declarations Section
'*****
Option Compare Database
Option Explicit

'*****
' Function ParamQD()
'
' Purpose: To demonstrate how to set the value of a parameter that
'         is a form reference.
'*****

Function ParamQD()
```

```

Dim MyDB As Database
Dim MyQDef As QueryDef
Dim MyDyna As Dynaset

Set MyDB = CurrentDB()
Set MyQDef = MyDB.OpenQueryDef("Customer Orders Parameter _
    Query")

'Set the Value of the Parameter.
MyQDef![Forms!Search Form!Customer To Find] = Forms![Search _
    Form]![Customer To Find]

'Create the dynaset.
Set MyDyna = MyQDef.CreateDynaset()
MyDyna.MoveLast
MsgBox MyDyna.RecordCount
MyDyna.Close
MyQDef.Close
End Function

```

When you enter a Customer ID on the Orders form and choose the ParamQD button, the ParamQD() function will run and a message box will display how many orders that customer has.

REFERENCES

=====

Microsoft Access "Introduction to Programming," versions 1.0 and 1.1, pages 137-138

Microsoft Access "User's Guide," version 2.0, pages 274-277

For more information about parameter queries and Microsoft Access 1.x documentation errors, please see the following article in the Microsoft Knowledge Base:

Q100147

PRB: Error When Using OpenQueryDef Method on Parameter Query

Additional reference words: 1.00 1.10 2.00 parameter query basic expected supplied

KBCategory: kbprg

KBSubcategory: PgmObj

INF: Ability to Add Users Without Permission to View Design
Article ID: Q120911

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

You may need to create a database in which some users have the ability to add other users, but do not have the ability to view the design of objects in the database. You can accomplish this by making certain users members of the Admins group, but not shipping the system database (SYSTEM.MDA file) that was used to create the database.

MORE INFORMATION

=====

The following example demonstrates how to make a user named Mary able to create new user accounts, but unable to view the design of existing objects in the database:

1. Quit Microsoft Access if it is running, and then make a backup copy of your SYSTEM.MDA file.
2. Using the Workgroup Administrator program, create and join a new workgroup. Make note of the Name, Organization, and Workgroup ID values used to create the new workgroup. These values are combined to create the group Security ID (SID) for the Admins group.
3. Start Microsoft Access, open any database, and choose Change Password from the Security menu. Assign a password for the Admin user.
4. From the Security menu, choose Users. Choose the New button, and then create a new user called Developer. Make note of the Personal ID that you assign for the new user. Make the Developer user a member of the Admins group.
5. Quit and then restart Microsoft Access. Log on to Microsoft Access as the Developer user.
6. Create a new database called MYDB.MDB. Create a new module in the database and then enter the following function in the module:

```
Function TestSecurity ()  
    MsgBox "This is a test of Security"  
End Function
```

Save the module as TestModule.

7. Remove all permissions on the TestModule module from the Users and Admins groups.
8. Quit Microsoft Access and then use the Workgroup Administrator program

to create and join a new workgroup. Make sure to use a different Workgroup ID than you did when you created the workgroup in step 2. If you do not, the module you created in step 6 will not be secure.

9. Start Microsoft Access and open the MYDB database.
10. Re-create the user named Developer that you created in step 4. Make sure to use the same Personal ID that you used in step 4. Make the Developer user a member of the Admins group.
11. Create a new user called Mary. Make Mary a member of the Admins group. Since Mary was not a member of the Admins group in the system database in use when the database was created, Mary will not have privileges on objects that already exist in the database.
12. Assign a password for the Admin user.
13. Quit and then restart Microsoft Access. Log on to Microsoft Access as Mary and then open the MYDB database. Note that although Mary is a member of the Admins group, and can create new users, Mary has no permissions on the TestModule module.
14. To test the security, quit and then restart Microsoft Access. Log on to Microsoft Access as Developer, and note that you have permissions on the TestModule module.

REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 14,
"Securing Your Application," pages 313-344

Additional reference words: 2.00 secure permission

KBCategory: kbusage

KBSubcategory: ScrtPerm

INF: Access Basic Error-Handling Supplemental Information
Article ID: Q101324

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article supplements the information available in the Microsoft Access "Language Reference" manual about how to handle errors in an Access Basic application.

MORE INFORMATION

=====

There are times when your code becomes susceptible to user error problems, even though the code itself is proven. The larger the program, the more often situations arise that were not anticipated at the time the code was written. For this reason, always include error-handling routines in your code to prevent your application from crashing when unexpected errors occur. Error-trapping routines offer significant help in the debugging process.

On Error Statement

Access Basic includes the On Error statement for error trapping, as well as several other commands for reporting and processing errors. The On Error statement is followed by either a GoTo clause, which causes program control to branch to a predefined label when an error occurs, or by a Resume Next clause, which tells Access Basic to ignore the error condition and continue to the next statement.

To create a label for an On Error GoTo statement, type the name of the label, followed by a colon (:). The commands that follow the label are executed when control is passed to the label by the On Error GoTo statement.

Generally, you should place the label and its accompanying error-handling routine at the bottom of the procedure, preceded by an End Sub or End Function statement, to ensure that the routine executes only when branched to and not by mistake.

NOTE: A label does not necessarily indicate a break in program flow.

Shown below is a Sub procedure that performs a SQL action based on a SQL string value. The On Error Resume Next statement is used to handle any errors generated by the first DeleteQueryDef method (for example, if "TempQuery" does not exist). On Error GoTo SQL_Error means that if any other errors occur, program control passes to the SQL_Error error-handling routine. The On Error GoTo 0 statement is used at the end of the procedure to disable error trapping.

```

Sub PerformSQLAction (InDB As String, SQLStmt As String)
    Dim SQLDb As Database, SQLQuery As QueryDef

    On Error Resume Next
        Set SQLDb = OpenDatabase(InDB)
        SQLDb.DeleteQueryDef ("TempQuery")
    On Error GoTo SQLError:
        Set SQLQuery = SQLDb.CreateQueryDef("TempQuery", SQLStmt)
        SQLQuery.Execute
        SQLQuery.Close
        SQLDb.DeleteQueryDef ("TempQuery")
    On Error GoTo 0
    Exit Sub

SQLError:
    MsgBox "An error occurred while executing the SQL statement."
    Exit Sub
End Sub

```

The Exit Sub statement is required at the end of the error-handling routine, even though the End Sub statement follows. In place of the Exit Sub statement, you could use a Resume Next statement to return control to the line following the line of code that caused the error.

Err(), Erl(), and Error\$() Functions

Access Basic includes the Err(), Erl(), and Error\$() functions, all of which return information about the error that occurred. The Error\$() function returns the error message as a string, the Err() function returns a number representing the error message, and the Erl() function returns the number of the line in which the error occurred. Using these functions, you can code your error-trapping routine to display meaningful error messages and trap for specific errors.

A generic error-trapping routine that uses both Error\$() and Erl() functions is shown below. This type of routine tells you what error occurred and where; it also allows you to either ignore the error and continue or cancel execution and further investigate the problem.

```

On Error GoTo ErrorHandler

ErrorHandler:
    If MsgBox("The following error has occurred at line " & _
        Trim(Str(Erl)) & ":" & Chr(13) & Chr(10) & Chr(13) & _
        Chr(10) & Error$, 17) = 1 Then Resume Next Else Stop

```

The following Sub procedure illustrates how the Error\$() function can create informative error messages to help you and your users debug your program:

```

Sub MyError ()
    On Error GoTo ErrorHandler
    INTEGERVAL% = 99999 'Generates Numeric Overflow error
    Debug.Print "Error was ignored"
    Exit Sub

```

```

ErrorHandler:
  If MsgBox("The following error has occurred at line " &
    Trim(Str(Erl)) & ":" & Chr(13) & Chr(10) & Chr(13) &
    Chr(10) & Error$, 17) = 1 Then Resume Next Else Stop
  Exit Sub
End Sub

```

You can use the Err() function to build a SQL SELECT statement that traps for specific errors and takes a different action depending on the error number. You can use the Erl() function to pinpoint exactly which line is causing the problem in your program, to anticipate error conditions in your program, and to handle errors more smoothly.

If you use the Erl() function, you do not need to number every line of your code to find the offending line number. Instead, number only those lines of code most likely to cause problems. (The line numbers represent labels, rather than classic line numbers, as in other Basic languages.) Then, if an error occurs in one of the numbered lines, the Erl() function reports only that line number.

You can also use this method if you have two lines of code that may generate the same error and you want to handle each case separately, as in the following example:

```

Function ErlTest()
  On Error Goto ErlTest_Err
  10: Open "AUTOEXEC.BAT" For Input As #1 'causes an error.
  20: Open "CONFIG.SYS" For Input As #2
  Close
Exit Sub

ErlTest_Err:
  If Erl = 10 Then
    MsgBox "Could not open AUTOEXEC.BAT file."
  ElseIf Erl = 20 Then
    MsgBox "Could not open CONFIG.SYS file."
  End If
  Exit Sub
End Sub

```

Since there is no possibility that a file called "AUTOEXEC.BAT" exists, this line of code errors out and program control passes to the error-handling routine "ErlTest_Err". The Erl() function detects the number of the line in which the error occurred and displays the "Could not open AUTOEXEC.BAT file" error message.

REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 331-333

For more information about error-handling routines, search for "Error handling" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 debugging
 KBCategory: kbprg
 KSubcategory: PgmErr

INF: Adding a Dynamic Counter to a Query to Count Records
Article ID: Q94397

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to use a dynamic counter value in a query field for the following purposes:

- Returning the top x number of records in a Microsoft Access version 1.x query. In Microsoft Access version 2.0, you can use the TopValues query property to accomplish this.
- Displaying the record's position number on a form. For more information about this topic, query on the following words here in the Microsoft Knowledge base:

record and position and number and form

- Printing the top x number of records on a report. For more information about this topic, query on the following words here in the Microsoft Knowledge Base:

top and 10 and records and report

NOTE: If the report includes totals, totals will be displayed for all the records, not just the top x number of records. To total only the top x number of records, limit the records included in the report using the method described later in this article.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

Sample code to implement a dynamic counter is described later in this article. There are several limitations to this method, including the following:

- You must reset the global counter to 0 before running the query each time. You can use the function SetToZero() to do this.
- Any scrolling, minimizing, maximizing, switching focus away and back, and so on, causes a repaint to occur, which in turn causes the query to call the function again. For this reason, you should not use this function in a select query. Instead, you should base a report on the select query or run a make-table query and view the resulting table.

- When basing a report on a select query with this function, scrolling back through the pages will cause the function to be called multiple times, and the counter expression will be off by the total number of records in the dynaset.
- When you place criteria on the counter expression, the function will be called twice per record. For example, if you specify criteria of <=10 to return the first 10 records, the dynaset will contain 10 records, but the counter expression will be off by the total number of records in the dynaset.
- Do not place the counter expression in the first column of the query; the number will be off by one.
- Do not combine criteria and sorting in a query with the counter expression. The criteria will be applied before the sort. Instead, you must create a make-table query with the sort, and then base a second query containing the counter expression on the new table with sorted data.

The Dynamic Counter Function

The following Access Basic functions, QcNtr() and SetToZero(), can be used to implement a dynamic counter in a query.

```
'Global Declarations Section
Option Explicit
Global Cntr
```

```
'*****
'Function:  QcNtr()
'
'Purpose:  This function will increment and return a dynamic
'counter.  This function should be called from a query.
'*****
```

```
Function QcNtr(x) As Long
    Cntr = Cntr + 1
    QcNtr = Cntr
End Function
```

```
'*****
'Function:  SetToZero()
'
'Purpose:  This function will reset the global Cntr to 0. This
'function should be called each time before running a query
'containing the QcNtr() function.
'*****
```

```
Function SetToZero()
    Cntr = 0
End Function
```

Returning the Top X Records

The following example uses the Orders table in the sample database NWIND.MDB to demonstrate how to return the top 10 orders:

1. Create a new query based on the Orders table. Include the following fields:

```
Field: Order ID
Field: Customer ID
Field: Order Date
Field: Order Amount
Sort: Descending
Field: Expr1: Qcntr([order id])
```

2. From the Query menu, choose Make Table. When you are prompted for a table name, type "Sorted Orders" (without quotation marks).
3. Save the query as Make Sorted Orders.
4. Create a new query that will be based on the Sorted Orders table once it is created. For now, close the Add Table dialog box without adding a table to the query. From the view menu, choose SQL, and then enter the following SQL statement:

```
Select * from [Sorted Orders] where [Expr1] <= 10
```

5. Save this query as Top 10 Orders and then close it.
6. Create the following new macro:

```
Actions
-----
RunCode
SetWarnings
OpenQuery
OpenQuery

Macro Action Arguments
-----
RunCode
Function Name: SetToZero()
SetWarnings
Warnings On: No
OpenQuery
Query Name: [Make Sorted Orders]
OpenQuery
Query Name: [Top 10 Orders]
```

7. Run the macro. The table Sorted Orders will be created, and then the Top 10 Orders query will be displayed.

NOTE: Since the Top 10 Orders query is based on a table, you can scroll through the records and use criteria on the counter expression without causing the function to be called multiple times per record.

Creating a Top 10 Report That Includes Totals

1. Follow steps 1-5 in the above section.
2. Create a new report called Top 10 Orders based on the Top 10 Orders query.
3. Create the following new macro:

Actions

RunCode
SetWarnings
OpenQuery
OpenReport

Macro Action Arguments

RunCode
 Function Name: SetToZero()
SetWarnings
 Warnings On: No
OpenQuery
 Query Name: [Make Sorted Orders]
OpenReport
 Report Name: [Top 10 Orders]

4. Run the macro.

NOTE: The counter expression will not renumber if you change the sort order or add grouping in the report, because it is calculated in the make-table query. The counter expression is being used here to limit the dynaset to the top 10 orders.

REFERENCES

=====

For information about when functions are executed in queries, query on the following words here in the Microsoft Knowledge Base:

 custom and function and executes

Additional reference words: 1.00 1.10 2.00 rank top 10 custom

KBCategory: kbusage

KBSubcategory: QryOthr

INF: Adding a New Record to a Combo Box with a Double-Click
Article ID: Q88148

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to set up a macro so that you can add new values to a combo box, using the Orders form in the sample database NWIND.MDB as an example.

MORE INFORMATION

=====

Follow these steps to modify the Orders form so that you can add new values to the Salesperson combo box:

1. Open the NWIND database.
2. Create the following new macro, and save it as Macro1:

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. Change the properties of the Employee ID combo box on the Orders form as follows:

OnDblClick: Macro1.NewRecord
StatusBarText: Double-click to Add a New Employee

4. Change the OnClose property of the Employees form as follows:

On Close: Macro1.OnClose

To see how this process works, open the Orders form in Form view, then double-click the Salesperson 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.

Additional reference words: 1.00 1.10 2.00 doubleclick update record
KBCategory: kbusage
KBSubcategory: FmsEvnt

INF: Administrator Setup Does Not Create SETUP Subdirectory
Article ID: Q115916

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

This article lists the directories created by Microsoft Access during an Administrator Setup (SETUP /A) to a network server, and describes how to run a workstation installation of Microsoft Access.

MORE INFORMATION

=====

Setting up Microsoft Access with the Administrator Setup lets you run Microsoft Access from a network server. This reduces the amount of disk space needed by workstations running Microsoft Access by storing most of the Microsoft Access files on the server.

When you run an Administrator Setup, the following subdirectories are created on the server below the Microsoft Access directory:

- SAMPAPPS
- SHARED
- SYSTEM
- WORKDIR

Note that no SETUP subdirectory is created. The SETUP.EXE file is installed in the Microsoft Access directory on the server.

To run a workstation installation, run SETUP.EXE from the Microsoft Access directory on the server, not from the SETUP subdirectory as stated in the Microsoft Access "User's Guide." Use the full path to the SETUP.EXE file to run a workstation installation, as in the following example:

```
N:\ACCESS\SETUP.EXE
```

REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 1, "Setting Up Microsoft Access," pages 4-6

For more information about installing to a network server, see the topic "Network Installation" in the ACREADME.HLP file.

For more information about Setup, query on the following words here in the Microsoft Knowledge Base:

setup and conversions and questions

Additional reference words: 2.00

KBCategory: kbsetup kbdocerr
KBSubcategory: StpNet

INF: Allowing an Ampersand to Appear in a Caption

Article ID: Q90090

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

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 in a caption, you must specify two ampersand characters in a row. For example, entering the following caption in the property sheet

Jack && Jill

will result in the following caption in the resulting form or control:

Jack & Jill

REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00 title

KBCategory: kbusage

KBSubcategory: FmsHowto

INF: Attach R:BASE Table by Exporting dBASE Format in R:BASE
Article ID: Q94037

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Microsoft Access does not ship with an ISAM driver that supports attaching R:BASE tables to Microsoft Access databases. However, R:BASE can export tables into several different formats that Microsoft Access can import.

To retain column names and data type definitions, export R:BASE tables from R:BASE using the dBASE format.

MORE INFORMATION

=====

R:BASE can export the following data types:

- ASCII delimited
- ASCII fixed width
- DIF (VisiCalc or other program)
- SYLK (Multiplan or other program)
- dBASE II, dBASE III, or dBASE III Plus (DBF)
- Lotus 1-2-3 (version 1.0 and 2.0) or Symphony (WKS, WK1, or WRK)

R:BASE and dBASE are manufactured by vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 1.00 1.10 2.00 rbase

KBCategory: kbinterop

KBSubcategory: IntpOthr

INF: Attached Table Must Be Opened as Dynaset from Modules
Article ID: Q90817

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

In Access Basic modules, a table can be opened as the Table data type in the following manner:

```
Dim MyDB as Database
Dim MyTable as Table
Set MyDB = OpenDatabase("NWIND.MDB")
Set MyTable = MyDB.OpenTable("Categories")
```

However, if you have an attached table that is in the database, it cannot be opened as the Table data type. An attached table must be opened as the Dynaset data type in the following manner:

```
DIM MyDB as Database
DIM MyDyn as Dynaset
Set MyDB = OpenDatabase("NWIND.MDB")
Set MyDyn = MyDB.CreateDynaset("AttachedTable")
```

MORE INFORMATION

=====

If you try to execute the OpenTable action on an attached table, you will receive the following error message in Microsoft Access version 1.x (error # 3219):

```
Can't perform operation; it is illegal
```

In Microsoft Access version 2.0, you will receive error # 3219:

```
Invalid Operation
```

When you refer to a Table data type, the table must reside in the .MDB file. This is why an attached database must be opened as a dynaset. An attached table only has a pointer to where the table resides.

REFERENCES

=====

For more information on table objects, search for "OpenTable" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 datatype
KBCategory: kbprg kberrmsg
KSubcategory: PgmObj

INF: Attaching Clipper Tables to Microsoft Access Databases
Article ID: Q95044

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

It is possible to attach Clipper .DBF files to a Microsoft Access database. To do so, attach it as a dBASE IV file. Note, however, the index may or may not be updateable.

Clipper provides an alternative indexing method (.NTX) to the dBASE standard (.NDX). The Clipper index format is not recognized by Microsoft Access, so Microsoft Access cannot update it correctly.

However, if the Clipper application was developed using the dBASE .NDX format, Microsoft Access updates the index correctly.

Additional reference words: 1.00 1.10 2.00 indexes

KBCategory: kbinterop

KBSubcategory: IntpOthr

INF: Base Combo Box on Parameter Query to Filter Values
Article ID: Q99937

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to filter values that appear in a combo box by basing the combo box on a parameter query.

MORE INFORMATION

=====

There may be situations in which you want to limit the values that appear in a combo box. For example, you may want to show only suppliers whose names begin with a certain letter. Although Microsoft Access does not allow filters on combo boxes, basing your combo box on a parameter query filters the values using a criteria clause.

This article describes the steps to create a combo box that displays only the suppliers that meet the criteria you specify. The sample database NWIND.MDB is used:

1. Create the following query to fill the combo box:

Query: Filtered Supplier List

Type: Select

Field: Company Name

Table: Suppliers

Sort: Ascending

Criteria: Like [My Criteria Box] & "*"

2. Create the following macro to update the combo box with new criteria:

MacroName: CB Test Macro

Action: Requery

ControlName: My combo box

3. Create the following unbound form to test your controls:

Form: TestForm

Record Source: <None>

TestForm Controls

Combo Box: My combo box

ControlSource: <None>

RowSourceType: Table/Query

RowSource: Filtered Supplier List
Text Box: My Criteria Box
After Update: CB Test Macro

4. Open the form in Form view. Choose the arrow button. (Do not enter any value in the Criteria box.) Note that all suppliers are listed because you have not yet entered a filter criteria.
5. Type a single letter in the Criteria box (for example, "E").
6. Choose the arrow button. Note that only suppliers whose names begin with the letter "E" are displayed.
7. Give the Criteria box the focus. Type a new letter (for example, "P").
8. Choose the arrow button. Note that now only suppliers whose names begin with the letter "P" are displayed.

Additional reference words: 1.00 1.10 2.00 limit drop down list box ddb
KBCategory: kbusage
KBSubcategory: FmsCmbo

INF: Btrieve File Owner Names, Passwords, and Security

Article ID: Q115917

The information in this article applies to:

- Microsoft Access version 2.0

SUMMARY

=====

Btrieve files have owner names embedded in them. These owner names can be used as passwords, but many applications that use Btrieve use other passwords to control user access.

Owner names are frequently set to read-only access. As a result, you can get the error message "Drive or network error" when you attempt to update the database file. If the owner name allows read-only access, Microsoft Access does not prompt you for a password until you attempt to update the file.

The DDFPassword line in the [Btrieve ISAM] section of the MSACC20.INI file can be used to set the owner name of .DDF files, but cannot be used to set the owner names of other Btrieve files. The owner names of other Btrieve files can be set programmatically when they are attached to Microsoft Access.

Only the user who created the Btrieve files and originally set the owner name can provide the owner name to you. It may be necessary for you to contact the original file creator to obtain the owner name, or have the files' creator provide files that do not have owner names embedded in them.

MORE INFORMATION

=====

Btrieve database files contain only data. They do not contain other database information (unlike Microsoft Access database files, which do contain other information or objects). Each Btrieve file contains only one table. You must use a separate application to access the data contained in a Btrieve database file, and this application must also provide all security control. When attached to a Microsoft Access database, each Btrieve file is considered a separate table.

Applications that use Btrieve files can either use a password to control entry to the application (possibly including multiple levels of user privilege), or use file owner names.

File owner names are both embedded in the database file and hard-coded in the application using that file. Since an application may draw information from numerous Btrieve files, and each file can have a unique owner name, the owner name for each file is generally hard-coded in the application. When the application opens a Btrieve database file, the Btrieve database engine compares the database file's embedded owner name with the owner name sent by the application to determine user privileges.

There is no connection between an application's password and the embedded

file owner name. You may have an application password that allows you full file-access privileges. This does not mean that you can directly access the files without the application, since the owner name hard-coded in the application is different than your application password.

Owner names assigned to Btrieve files can have one of four security options:

1. Require an owner name for any access--no data encryption.
2. Permit read-only access without an owner name--no data encryption.
3. Require an owner name for any access--with data encryption.
4. Permit read-only access without an owner name--with data encryption.

If a read-only option is set on the Btrieve database file (with or without an owner name), Microsoft Access can open the file and display its contents. However, if you attempt to update the file, you will receive the error message "Drive or network error."

Microsoft Access Prompts for Password

When you are using any of the installable ISAMs (such as Btrieve, Paradox, or dBASE), if you attempt to open an attached table that is password protected, Microsoft Access will prompt you for a password after the open attempt fails.

In Btrieve, if you set the owner name option to read-only access, the open does not fail and Microsoft Access does not prompt you for a password. No error or warning occurs until you attempt to update the file.

How to Specify an Owner Name

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

You can specify an owner name for the .DDF files using the DDFPassword line in the [Btrieve ISAM] section of the MSACC20.INI file. However, this is of little benefit since it only sets the password for the .DDF files, not for the Btrieve database file.

You can use Access Basic to attach the Btrieve database file and specify the owner name so that full read-write privileges are available. Once you attach the table, it is not necessary to specify the owner name again.

The following sample Access Basic code demonstrates how to attach a Btrieve file called MYFILE and specify the file owner name as a Microsoft Access password:

```
Sub TablePwd ()
```

```
Dim MyDB As Database, MyTD As TableDef
Set MyDB = DBEngine.Workspaces(0).Databases(0)
Set MyTD = MyDB.CreateTableDef("MYFILE")
MyTD.connect = "btrieve;PWD=OwnrNm;Database=C:\MYDIR\FILE.DDF"
MyTD.sourcetableName = "MYFILE"
MyDB.tableDefs.Append MyTD
End Sub
```

Note that this example uses the FILE.DDF file located in the C:\MYDIR directory to search for a Btrieve table named MYFILE, and then attempts to attach the file to the current database using the password "OwnrNm."

The products discussed here are manufactured by vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 2.00 Attaching ODBC
KBCategory: kb3rdparty kberrmsg
KBSubcategory: IsmBtrv

INF: Cannot Distribute MS Access 2.0 ODBC Driver with ADT Apps
Article ID: Q119567

The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
-

SUMMARY

=====

You cannot legally ship the Microsoft Access version 2.0 ODBC driver (ODBCJT16.DLL) shipped with Microsoft Access version 2.0 and Microsoft Office Professional version 4.3 with your application developed with the Microsoft Access Developer's Toolkit (ADT).

MORE INFORMATION

=====

The Microsoft Access 2.0 ODBC driver is not legally distributable in any form. This fact is not mentioned in the ADT documentation, nor is it listed as one of the files prohibited from distribution.

Users who require the Microsoft Access 2.0 ODBC driver (such as Microsoft Word version 6.0 for Windows or Microsoft Excel version 5.0 users) should purchase either Microsoft Access version 2.0 or Microsoft Office Professional version 4.3 and install Microsoft Access and the ODBC driver.

The Microsoft Access version 2.0 ODBC driver works only with Microsoft Word 6.0 and Microsoft Excel 5.0. It does not work with other ODBC-compliant applications such as Microsoft Visual C++.

Additional reference words: 2.00 interoperability app

KBCategory: kbinterop

KBSubcategory: IntpWrd6

INF: Case Sensitivity in Queries Differs with Attached Tables
Article ID: Q100921

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

When you query on an attached table, the criteria are case sensitive. However, when you query on a native Microsoft Access table, the criteria are not case sensitive.

MORE INFORMATION

=====

There are several ways to prevent case sensitivity in criteria when you query on attached tables:

- Import the attached table, and then base your query on the imported table. Criteria values will not be case sensitive.
- Include a native Microsoft Access table joined to the attached table in the query. Criteria values will not be case sensitive.
- Use the UCase() function in your query to ensure that the field in the query dynaset is returned in uppercase, and specify the criteria in uppercase. The following sample query demonstrates how to use the UCase() function in a query:

Query: Query1

FieldName: UCase(CITY)

Show: True

First Criteria Line: Like "S*" (Note that the "S" is uppercase)

- Another option is to be aware that you are using an attached table and make sure that the values you use for your criteria match the case of the values in the attached table.

Steps to Reproduce Behavior

1. Create a new Microsoft Access database and then attach the sample dBASE file NEWCUST.DBF. (This sample dBASE file is supplied with Microsoft Access.)
2. Create a new query based on the attached NEWCUST table. This query will be called QueryTest.
3. Drag all the NEWCUST table fields to the query grid.
4. Make the following change to the CITY column:

Query: QueryTest

FieldName: CITY

Show: True

First Criteria Line: Like "s*"

Note that the "s" is lowercase.

5. Run the query. The result will be an empty set.
6. Import the same dBASE table. Select the Database window, and then choose Import from the File menu. In the Data Source box, select dBASE III. Choose OK.
7. Recreate the query using steps 1-4. Make sure to base the query on the imported table.
8. Run the query. The results return all the records with Seattle as the city.

REFERENCES

=====

For more information about criteria in queries, search for "Criteria: queries" using the Microsoft Access Help menu.

For more information about the UCase() function, search for "UCase, UCase\$ Function" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 isam paradox btrieve fox pro

KBCategory: kbusage

KBSubcategory: QryOthr

INF: Changes in Security in Version 2.0

Article ID: Q112121

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

This article describes the changes in the Microsoft Access security model from version 1.x to version 2.0.

MORE INFORMATION

=====

The security model is essentially unchanged from version 1.x to version 2.0. The following internal items are changed in version 2.0 security:

- The SID column of the MSysACEs table is no longer directly readable through the Microsoft Access user interface. This prevents users from directly reading the user SIDs and using them to breach security. To view and update permissions, you must use Access Basic.
- In Microsoft Access version 1.x, users had permission to open any database that was not protected by the file system. In Microsoft Access version 2.0, there are two new database-level permissions to control this: Open/Run and Open Exclusive. You can use the Open/Run permission to lock unauthorized users out of your database entirely. You can use the Open Exclusive permission to prevent users from inadvertently locking each other out of multiuser applications by opening the database exclusively.

As in Microsoft Access 1.x, once a user has opened a database there is no way, using the user interface, to prevent that user from creating new objects in the database. However, you can use Access Basic code to prevent users from creating new tables or queries in a database by revoking permissions on the Tables Container object.

- The Microsoft Access 2.0 Setup program uses only the user and company names as seeds for the Admins group's SID. The version 1.x Setup program uses the user and company names, as well as the serial number from the installation disks.
- Query permissions can cross database boundaries. This means you can use owner's permissions security with attached tables, as well as with local tables. The Run With Owner's Permissions query property in version 1.x has been replaced in version 2.0 by a query property called RunPermissions that can be set to User's or Owner's.
- There was a known and documented hole in Microsoft Access version 1.x security whereby an unauthorized user could read a SID from a database and paste it over a SID in the MSysAccounts table in the SystemDB, thereby masquerading as a different user. The following changes were made to version 2.0 security to correct this problem:

- System tables in version 2.0 are not updatable.
- The SID columns in the MSysACEs and MSysObjects tables are unreadable in version 2.0.
- The file encryption algorithm is enhanced. SIDs are readable by members of the Admins group only, and are not writable at all in version 2.0.

Microsoft Access version 2.0 can run version 1.x databases, but they are still vulnerable to the version 1.x security hole. This is so that sites can run a mixed environment, with some users running Microsoft Access version 2.0 and others running Microsoft Access version 1.x. Microsoft Access version 2.0 cannot make changes to security in a version 1.x database unless the database is converted to version 2.0 format.

Administrators of version 1.x sites that are upgrading to version 2.0 should be aware that in order to prevent unauthorized users from exploiting the version 1.x security hole to break into a secure version 2.0 system, it is necessary to re-create user and group accounts with the new longer PIDs, and to reassign permissions. If this is not done, it is possible for an unauthorized user to read a user's SID from an old version 1.x copy of the database and paste it over their own account's SID using Microsoft Access version 1.x and a version 1.x SystemDB. If the users' SIDs are re-created using Microsoft Access version 2.0, there is no way for an unauthorized user to ever read a user's SID.

If you are not concerned about the possibility of an unauthorized user exploiting the 1.x security hole to break into a secure version 2.0 system (perhaps because you are using security only to protect well-meaning users from inadvertently destroying data or applications, rather than protecting yourself from unauthorized intrusion), then you do not need to re-create your user and group accounts. Version 2.0 security will work properly with the old SIDs. Although the security in Microsoft Access version 2.0 is enhanced to protect your databases from unwanted intrusion, Microsoft recommends that you convert your databases to use version 2.0 security.

When you are using the Microsoft Access 2.0 Upgrade disks to upgrade an existing version 1.0 or 1.1 installation to version 2.0, a new SystemDB is created. The old SystemDB is not changed in any way. If you want to use your old SystemDB and your old SIDs, use the Workgroup Administrator to join your old workgroup.

- In Microsoft Access version 1.x, the Guests group is given read permissions on all objects by default. In Microsoft Access 2.0, the Guests group has no default permissions on newly created objects.

Additional reference words: 2.00 Security

KBCategory: kbusage

KBSubcategory: ScrtOthr

INF: Compound Indexes Must Restrict First Indexed Field

Article ID: Q98793

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Before an index can be used, queries with restrictions on multiple-field (compound) indexes must restrict at least the first field of the index. You can, however, also use adjacent fields in the index (up to 10 fields). If the restriction is placed on a field other than the first field, the query optimizer scans the table rather than using the index. This is not always desirable, since table scans are slower than index searches for most queries.

MORE INFORMATION

=====

By default, when a compound index is created in Microsoft Access, no individual indexes are assigned to the fields included in the compound index. This behavior is by design.

For the query optimizer to use an index, you must use a comparison of either the first field in the compound index or the first field and any number of adjacent fields (up to 10) that make up the compound index. You must query the indexed fields in the order they appear in the Indexes window, beginning with the first indexed field and continuing with adjacent fields.

NOTE: This principle also applies to using criteria with the Find method in Access Basic.

For example, consider a table (T1) containing three fields: key_part1, key_part2, and key_part3. If there is a composite index created on these three fields and all fields are the primary key, then:

```
SELECT * FROM T1 WHERE key_part2 = <value>
```

does not use the index because the first field, key_part1, is not used.

```
SELECT * FROM T1 WHERE key_part1 = <value> AND key_part3 = <value>
```

also does not use the index, because, although key_part1 is referred to, key_part1 and key_part3 are not adjacent fields.

However, each of the following three SQL statements do use the index because they each include the first field, or the first field and one or more adjacent fields of the composite index:

```
a. SELECT * FROM T1 WHERE key_part1 = <value>
```

```
b. SELECT * FROM T1 WHERE key_part1 = <value> AND key_part2 =
```

<value>

```
c. SELECT * FROM T1 WHERE key_part1 = <value> AND key_part2 =  
   <value> AND key_part2 = <value>
```

The above fields are not prohibited from having indexes on them; individual indexes can be built for each field, allowing comparisons on those fields with index searches. Be aware, however, that indexes can take up as much (or more) space than the data.

REFERENCES

=====

For more information on creating indexes, search for "indexes: creating" then "Creating an Index" using the Microsoft Access Help menu.

For more information about optimizing queries with Rushmore technology, search for "Rushmore technology" then "Optimizing Queries with Rushmore Technology" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 index query performance multi
KBCategory: kbusage
KBSubcategory: TblPriky

INF: Concatenating Fields in a Text box to Remove Blank Lines
Article ID: Q95917

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to concatenate multiple fields for use in a form or report. This is especially useful in a mailing label report when some fields are null or empty. Null fields can cause blank lines to be printed on a report. Additionally, from a form, you may want to be able to copy and paste an entire name and address into another application like Microsoft Word for Windows.

Using the IIF() and the IsNull() functions, you can determine if a field is blank. If a field is blank, it returns an "empty" value. Chr(13) and Chr(10) are used to add a return and a line feed to the text box.

MORE INFORMATION

=====

The following steps can be used to create a concatenated field eliminating blank lines. In this example, the Employees Table from the sample database NWIND.MDB is used.

1. Create a new form or a report. Create your report based on the table or query where the information resides.
2. Add a text box control to the form/report.

NOTE: In the following example, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this expression.

Object: Text Box

ControlName: Full Address

ControlSource:

```
=IIf (IsNull([FIRST NAME]),"",[FIRST NAME] & " ") &_  
  IIf (IsNull([LAST NAME]),"",[LAST NAME]&Chr(13)&Chr(10)) &_  
  IIf (IsNull([ADDRESS]),"",[ADDRESS] & Chr(13) & Chr(10)) &_  
  IIf (IsNull([CITY]),"",[CITY] & ", ") &_  
  IIf (IsNull([REGION]) ,"",[REGION] & " ") &_  
  IIf (IsNull([POSTAL CODE]),"",[POSTAL CODE])
```

Can Grow: Yes

Can Shrink: Yes

When the form or report is printed, there will be no blank lines.

For more information, search for "IIF," then "IsNull," then "Chr," then "Concatenation" using the Help menu.

Additional reference words: 1.00 1.10 2.00
KBCategory: kbusage
KSubcategory: FmrHowto

INF: Connecting to SYBASE SQL Server from an ODBC Application
Article ID: Q101073

The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
 - Microsoft Visual Basic version 3.0 for Windows
 - Microsoft FoxPro version 2.5 for Windows
-

SUMMARY

=====

You can connect to a SYBASE SQL Server computer from an ODBC application (such as Microsoft Access, Microsoft Visual Basic, or Microsoft FoxPro for Windows) using the ODBC SQL Server driver. This article describes how to set up the ODBC data sources.

If you are already connected to SYBASE SQL Server from Microsoft Windows DB-Library applications, no other components are necessary.

MORE INFORMATION

=====

The key to both Microsoft SQL Server and SYBASE SQL Server connectivity is a module called the Net-Library. This module consists of two interfaces: one with the network and one with the application. The network interface is customized to support a particular network; however, the application interface remains the same.

Since ODBC, DB-Library, APT-SQL, APT-Library, and so forth communicate with the Net-Library, these utilities can be written independent of the network. The ODBC SQL Server driver, in particular, can be used to connect either to a Microsoft SQL Server computer or to a SYBASE SQL Server computer (on a UNIX or VMS system), as long as there is a Net-Library for whatever network you are using.

The following paragraphs discuss how to connect to a SYBASE SQL Server from an ODBC application. The remainder of the article assumes that you are connected to a SQL Server called "Mysqlsvr." This server should be set up as an ODBC data source, using the ODBC Control Panel utility, as follows:

1. Open ODBC Control Panel and choose Add.
2. Select SQL Server from the list of installed drivers and choose OK.
3. In the ODBC SQL Server Setup dialog box, type a data source name and a description (optional).

Note that a single server can function as multiple ODBC data sources, since each database in the server can be a data source. The data source name need not be the same as the server name.

4. Enter the true name of the server, then enter the network address.

Since a SYBASE SQL Server on UNIX uses sockets and not named pipes, the network address must contain the IP address and the port identification number. This string is in the form "ip_address,port" (for example, "11.1.14.40,3180").

For a SYBASE SQL Server on a VAX system, this string should be in the form "<node address>,<process_id>", where <node address> is the DECnet node address of the server and <process_id> is the process identification number to use for the connection (for example, "1.997,141").

The Network Library field should contain the name of the Net-Library you need to use. This is dependent on what network you are using. For example, if you are using the FTP Software product PCTCP, the Net-Library name is "wdbftptc".

5. Choose Options to specify a database name or a language name.
6. Choose OK to go back to the first screen.
7. Run the INSTCAT.SQL script file.

NOTE: This is a file containing SQL statements that will create certain stored procedures needed to process ODBC calls. Without these procedures, Microsoft Access cannot attach a SQL Server table and Visual Basic is not able to function correctly.

At the UNIX command line or the MS-DOS prompt, type the following:

```
isql -S<servername> -Usa -P<sa-password> -i<path>\instcat.sql
```

Finally, note that if the above procedure is followed correctly, the following changes appear in the ODBC.INI and WIN.INI files. The [Data Sources] section of ODBC.INI will contain the following entry:

```
<data-source-name>=SQL Server
```

where <data-source-name> is the name of the data source that you added. There will be a new section called [data-source-name] containing the location of the SQL Server driver and a description of the data source. The [SQLSERVER] section of the WIN.INI will contain the following entry:

```
data-source-name=<net-library-name>,<network address>
```

The following products are manufactured by vendors independent of Microsoft: DECnet, VMS, and MicroVAX by Digital Equipment Corporation; FTP Software by FTP Software, Inc.; DB-Library, Net-Library, and SYBASE SQL Server by Sybase, Inc.; and UNIX by UNIX System Laboratories. We make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 1.10 2.00 third-party
KBCategory: kbusage
KBSubcategory: ObcOthr

INF: Connecting to the First Available Network Drive

Article ID: Q90861

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

To make a network connection with the Windows API, you can use an unused logical drive letter or a UNC name (\\SERVER\SHARE). You can call the Windows API function GetDriveType() from an Access Basic module to find the first available drive letter and then make a network connection to this available drive.

MORE INFORMATION

=====

Use the following steps to find the first free drive and make a network connection:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

1. Declare the API functions:

```
'-----  
Option Explicit  
  
Declare Function GetDriveType% Lib "kernel" (ByVal nDrive%)  
Declare Function WNetAddConnection% Lib "User" (ByVal lpszNetPath$, _  
                                                ByVal lpszPassword$, _  
                                                ByVal lpszLocalName$)  
'-----
```

2. Write a function to return the first free drive and make the new connection:

```
Function FreeDrive ()  
    Dim DriveNum, FirstFreeDrive  
    Dim FirstDrive%, Results%  
  
    DriveNum = -1  
  
    Do  
        DriveNum = DriveNum + 1  
        FirstDrive% = GetDriveType(DriveNum)  
    Loop Until FirstDrive% = 0  
  
    FirstFreeDrive = Chr$(DriveNum + 65) + ":"  
  
    'Substitute the appropriate server share and password.  
    Results% = WNetAddConnection("\\server\share", "password", _
```

FirstFreeDrive)

End Function

REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

INF: Conversion of Oracle Data Types to Microsoft Access
Article ID: Q104977

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY
=====

Column, or field, data types are different for each database management system. This means that when you are accessing table structures from other database management systems, a conversion, or data type mapping, must occur. This article describes the data type mapping between Oracle and Microsoft Access data types.

MORE INFORMATION
=====

The following table demonstrates data type mapping between Oracle and Microsoft Access data types:

Oracle data type	Maps to Microsoft Access data type
Char(n)	Text(n)
VarChar(n)	Text(n)
Long	Memo
Date	Date/Time
Long Raw	Ole Object
Float	Double
RowID	Text(18)
Smallint	Text(255)

Note that the Smallint data type is the same as an Oracle data type of Number(38,0).

In Oracle, the creator of the table can define the precision (the total number of digits) and scale (the number of digits to the right of the decimal point) for columns with Number data types. If a Number data type is specified without a precision or scale, the column defaults to a data type of Number(9,2). The scale limit is 38, and a scale without a precision is allowed. Values with different precisions and scales are mapped as below:

Oracle number	Microsoft Access data type
Scale = 0 and	
precision <= 4	Integer
precision <= 9	Long Integer
precision <= 15	Double
Scale > 0 and <= 4	
precision <= 15	Double
Scale > 4 and/or	
precision > 15	Text

If an Oracle data type is defined with a precision and scale that does not

map to any data type in Microsoft Access, it will be mapped to a text data type. For example, an Oracle Smallint data type, which is actually a Number(38,0) data type, will map into Microsoft Access as a text data type since Microsoft Access does not have a number value of 38 precision.

Additional reference words: 1.00 1.10 2.00 ADK

KBCategory: kbusage

KBSubcategory: TblDatyp

INF: Converting Julian Dates with Access Basic

Article ID: Q92816

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Most government agencies and contractors require the use of Julian dates. (A Julian date starts with a two-digit year, then counts the number of days from January 1, as printed on many desktop calendars.)

Microsoft Access does not have any built-in functions for using or converting Julian dates to standard dates. However, you can write a function in Access Basic to do this. Below is an Access Basic function which converts a Julian date to a standard serial date in Microsoft Access. Note that the following sample code assumes the year is in the 20th century. The code must be changed to process dates after the year 2000.

MORE INFORMATION

=====

The following sample function, ConvertJulian(), can be used to convert Julian dates to standard dates:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Option Explicit
Function ConvertJulian (JulianDate as Long)
    ConvertJulian = DateSerial(1900 + Int(JulianDate / 1000), _
        1, JulianDate Mod 1000)
End Function
```

Examples of How to Use the ConvertJulian() Function

Example 1:

1. Create a new unbound form and add two unbound text boxes (called field0 and field2) to the form.
2. In the ControlSource property of field2 type:

```
=ConvertJulian([field0])
```
3. Switch to Form view. Enter "90001" (without quotation marks) in the first text box. The Julian date in the first text box will be converted and displayed as a standard date in the second text box.

Example 2:

1. Create a new table called Table1 with one field. Name the field Julian, with a data type of Number. Save the table.
2. View the table in Datasheet view. Add the following values to the table:

90001
90010
92100
3. Create a new query based on Table1 and drag the Julian field onto the query grid.
4. In the Field row of another column on the query grid, enter the following:

Normal: ConvertJulian([Julian])
5. Run the query. Note that the Julian dates are converted to standard dates.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

INF: Correct Syntax for TransferDatabase to dBASE and Paradox
Article ID: Q97519

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

When you attempt to export a Microsoft Access table to dBASE or Paradox, you may encounter errors. This article explains the correct argument syntax for exporting the Categories table from the sample database NWIND.MDB to dBASE III using the TransferDatabase macro action. The equivalent Access Basic syntax is also explained.

MORE INFORMATION

=====

NOTE: Both of the following examples export the Categories table to a dBASE file in the C:\DBASE directory. You need to enter a valid directory in your macro or you may receive the following error message when you run the macro or function:

isn't a valid path

Example 1 - Using the TransferDatabase Macro Action

Create a macro named Export1. This macro will export the Categories table to a dBASE file.

Macro Name	Action
Export1	TransferDatabase

Export1 Actions

TransferDatabase
Transfer Type: Export
Database Type: dBASE III
Database Name: C:\dbase
Object Type: Table
Source: Categories
Destination: Test1.dbf
Structure Only: No

NOTE: To use the TransferDatabase macro action to export to a Paradox table, you must choose "Paradox" for the Database Type argument and enter the Destination argument as "Test1.db".

Example 2 - Using an Access Basic Function

Create a new module and enter the following function:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function ExportdBASE ()  
    DoCmd TransferDatabase a_export, "dBASE III", "c:\dbase", _  
        a_table, "Categories", "Test1.dbf"  
End Function
```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage kbdocerr kberrmsg

KBSubcategory: DcmOthr

INF: Could not Find SYSTEM.MDA or UTILITY.MDA File

Article ID: Q90105

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

When Microsoft Access is started, one or both of the following error messages may be displayed:

 Couldn't find file 'SYSTEM.MDA' (version 1.x)
 Couldn't find file '<path>\SYSTEM.MDA' (version 2.0)

 Couldn't find file 'UTILITY.MDA' (version 1.x)
 Couldn't find file '<path>\UTILITY.MDA' (version 2.0)

These errors occur if the SystemDB and UtilityDB settings in the [Options] section of the MSACCESS.INI file in version 1.x, or the MSACC20.INI file in version 2.0, 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 to which there is no connection.

If the directory does not exist, the following error message is displayed:

 Invalid Path '<pathname>' (version 1.0)
 '<pathname>' isn't a valid path (versions 1.1 and 2.0)

MORE INFORMATION

=====

A good way to manage the MSACCESS.INI or MSACC20.INI file is to keep backup copies that pertain to particular setups. Have one set up 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 or MSACC20.INI file.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbsetup kberrmsg

KBSubcategory: StpOthr

INF: Create and Drop Tables and Relationships Using SQL DDL

Article ID: Q116145

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

Microsoft Access version 2.0 includes a Data Definition Language (DDL) that you can use to create and delete (drop) tables and relationships. These same operations can also be performed with Access Basic using data access objects (DAO). This article describes some of the operations available in DDL.

Note that all the examples in this article assume that the statements are added to a Microsoft Access query and are run by choosing Run from the Query menu.

MORE INFORMATION

=====

To use data definition statements, create a new query. Then, choose SQL Specific then Data Definition from the Query menu. Enter your data definition statement in the Data Definition Query window, then run the query by choosing Run from the Query menu.

To create a table with a PrimaryKey field with a Counter data type, and a field called MyText with a Text data type with a length of 10, enter the following statement in the Data Definition Query window and then run the query.

NOTE: In the following sample queries, an underscore (_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating these queries.

```
CREATE TABLE Table1 (Id COUNTER CONSTRAINT PrimaryKey PRIMARY KEY, _  
MyText TEXT (10))
```

The Foreign Key side of a relationship does not require a PrimaryKey and can be created by running the following query. This query creates a table with one field with a Long data type, and one field with a Text data type with the default size of 255:

```
CREATE TABLE Table2 (Id LONG, MyText TEXT)
```

After creating both tables, running the following query will create a one-to-many relationship between Table1 and Table2, with Table1 as the primary side of the relationship:

```
ALTER TABLE Table2 ADD CONSTRAINT Relation1 FOREIGN KEY ([Id]) _  
REFERENCES Table1 ([Id])
```

To delete the relationship between the tables, run the following query:

```
ALTER TABLE Table2 DROP CONSTRAINT Relation1
```

To delete Table1, run the following query:

```
DROP TABLE Table1
```

To delete Table2, run the following query:

```
DROP TABLE Table2
```

You can also run DDL statements using the RunSQL action. To run a DDL statement with Access Basic, use a procedure similar to the following example:

```
Sub ExecutesQLDDL (SQLString As String)
    Dim db As Database, qd As QueryDef
    Set db = DBEngine.Workspaces(0).Databases(0)
    Set qd = db.CreateQueryDef("")
    qd.SQL = SQLString
    qd.Execute
    db.Close
End Sub
```

To call this Sub procedure, use the following syntax:

```
ExecutesQLDDL "DROP TABLE Table1"
```

REFERENCES

=====

For more information about DDL queries, search for these topics using the Microsoft Access Help menu: "CREATE TABLE (SQL)," "ALTER TABLE (SQL)," "CONSTRAINT (SQL)," "CREATE INDEX (SQL)," and "DROP (SQL)"

Microsoft Access "User's Guide," version 2.0, Chapter 9, "Importing, Exporting, and Attaching," page 169, Chapter 10, "Query Basics," page 199, and Chapter 12, "Advanced Queries," pages 285 and 288

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: QrySqlddl

INF: CreateControl() and CreateReportControl() Functions

Article ID: Q93095

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article discusses the Microsoft Access functions CreateControl() and CreateReportControl(). If you intend to write your own Form Wizards or Report Wizards, you can use these functions to create controls on a form or report that are available in Design view.

This article is for those who are already familiar with Access Basic and with designing forms and reports.

MORE INFORMATION

=====

You can use the CreateControl() and CreateReportControl() functions to create new controls for a form or report opened in Design view. These functions require two arguments: the name of the form or report as a string value and a numeric code that represents the control type.

Here are the control types and their associated numeric codes:

Integer	Control Type
100	label
101	rectangle
102	line
103	(not used)
104	command button
105	option button
106	check box
107	option group
108	bound object frame
109	text box
110	list box
111	text box with drop-down grid
112	subform/subreport
113	graph
114	unbound object frame
118	page break

CreateControl() and CreateReportControl() both return a control object value. Therefore, you need first to define a control variable, and then set the control variable equal to the function name.

For example, the following procedure creates a form and then adds a button to the form:

```
Dim MyForm As Form, MyControl As Control
```

```
Set MyForm = CreateForm()
Set MyControl = CreateControl(MyForm.FormName, 104)
```

When the procedure is finished, you can modify the properties of the new control by using the control variable you defined. For example, you can change the control's Width and Caption properties with these statements:

```
MyControl.Width = 2000
MyControl.Caption = "&Sum All Records"
```

For controls that are frequently associated with a field in a table or query, you can modify the ControlSource property to bind the control to the field.

By default, some controls are created with their Height and Width properties set to zero to make them invisible. Also by default, controls appear in the upper-left corner of the form. You can adjust the size and position of a control immediately after you create it by changing the control's properties. For example, the following code creates, sizes, moves, and then binds a text box to a field by changing the properties:

```
Set MyControl = CreateControl(MyForm.FormName, 109)
MyControl.Width = 1500
MyControl.Height = 200
MyControl.Top = 440
MyControl.Left = 200
MyControl.ControlSource = "[Category ID]"
```

In addition to the form name and the code for the type of control, you can also specify the form or report Section where you want Microsoft Access to place the control.

The following is the CreateControl() syntax. The CreateReportControl() syntax is identical.

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function CreateControl (FormName As String, _
                       ControlType As Integer _
                       [,SectionNumber As Integer])
```

SectionNumber is the Section property setting number that identifies a particular section as follows:

```
Detail section ..... 0
Form or report header section ..... 1
Form or report footer section ..... 2
Form or report page header section ..... 3
Form or report page footer section ..... 4
Group-level 1 header section (reports only) ... 5
Group-level 1 footer section (reports only) ... 6
Group-level 2 header section (reports only) ... 7
Group-level 2 footer section (reports only) ... 8
```

If a report has additional group-level sections, the header/footer pairs

are numbered consecutively beginning with 9.

REFERENCES

=====

For more information on Microsoft Access libraries, query on the following words here in the Microsoft Knowledge Base:

libraries and debugging and creating

For more information on other AccessWizard functions, query on the following words here in the Microsoft Knowledge Base:

CreateControl and CreateReportControl and CreateForm and
CreateReport and "Section Property" and DeleteControl and
DeleteReportControl and CreateGroupLevel

For more information on the CreateForm(), CreateReport(), CreateControl(), and CreateReportControl() functions, search for each function name and then view its topic using the Microsoft Access version 2.0 Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

INF: Creating a Crosstab Query with Multiple Value Fields
Article ID: Q109939

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY
=====

In a Microsoft Access crosstab query, you can specify only one field or calculation as the value. Often it is desirable to show more than one value in the query.

For example, the following shows two columns of information per company, a count of the number of orders, and the order totals for each year:

Company Name	1989 Orders	1989 Total	1990 Orders	1990 Total
ABC Company	12	\$855.00	15	\$1010.25
XYZ Company	1017	\$22045.57	1050	\$25345.29

This type of query is sometimes called a Multiple Value Field query.

MORE INFORMATION
=====

In order to create a Multiple Value Fields crosstab query, you must create two separate crosstab queries with fixed column headings for each value you want to calculate. These two queries can then be joined together using a Select query to display the desired results.

The following example uses the sample database NWIND.MDB to create a query that displays results similar to the example above, showing the number of sales, and total, for each year for each company.

Step One: Create the "Order Total" Crosstab Query

1. Open the sample database NWIND.MDB and create the following Crosstab query based on the Orders, Order Details, and the Customers tables.

Query: Order Total

Type: Crosstab Query

Join: Customers.[Customer ID] <-> Orders.[Customer ID]

Join: Orders.[Order ID] <-> Order Details.[Order ID]

Field: Company Name

 Table Name: Customers

Total: Group By

Crosstab: Row Heading

Field: Expr1: Year([Order Date]) & " " & "Order Total"

Table Name:

Total: Group By

Crosstab: Column Heading
Field: Expr2: Sum(CCur([Unit Price]*[Quantity]*(1.-[Discount])))
Table Name:
Total: Expression
Crosstab: Value

2. Choose Query Properties from the View menu, select the Fixed Column Headings check box and type the following entries in the Fixed Column Headings box:

"1989 Order Total";"1990 Order Total";"1991 Order Total";"1992 Order Total"

NOTE: In Microsoft Access version 2.0, the Fixed Column Headings property is called Property Headings.

Step Two: Create the "Order Count" Crosstab Query

1. Create the following Crosstab query based on the Orders and the Customers tables:

Query: Order Count

Type: Crosstab Query
Join: Customers.[Customer ID] <-> Orders.[Customer ID]
Field: Company Name
Table Name: Customers
Total: Group By
Crosstab: Row Heading
Field: Year([Order Date]) & " " & "Order Count"
Table Name:
Total: Group By
Crosstab: Column Heading
Field: Order ID
Table Name: Orders
Total: Count
Crosstab: Value

2. Choose Query Properties from the View menu, select the Fixed Column Headings check box and type the following entries in the Fixed Column Headings box:

"1989 Order Count";"1990 Order Count";"1991 Order Count";"1992 Order Count"

Step Three: Create the "Multiple Values" Query

1. Create the following Select query based on the Order Totals and Order Count Crosstab Queries:

Query: Multiple Values

Join: Order Total.[Company Name] <-> Order Count.[Company Name]
Field: Company Name
Table Name: Order Count

Show: Yes
 Field: 1989 Order Count
 Table Name: Order Count
 Show: Yes
 Field: 1989 Order Total
 Table Name: Order Total
 Show: Yes
 Field: 1990 Order Count
 Table Name: Order Count
 Show: Yes
 Field: 1990 Order Total
 Table Name: Order Total
 Show: Yes

Running the Multiple Values query will result in a table that looks like the following:

Company Name	1989 Order Count	1989 Order Total	1990 Order Count ...
Always Open	1	\$855.38	4
Andre's	0	0	3
Anthony's	4	\$3,364.71	6

REFERENCES
 =====

Microsoft Access "User's Guide," version 1.0, Chapter 6, "Designing Select Queries," pages 150-155

Microsoft Access "User's Guide," version 1.1, Chapter 6, "Designing Select Queries," pages 154-159

Additional reference words: 1.00 1.10 2.00 crosstab query select
 KBCategory: kbusage
 KBSubcategory: QryCross

INF: Creating Additional ADT Application Program Groups, Items
Article ID: Q119724

The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
-

SUMMARY

=====

This article describes how to create a sample database (ADDICONS.MDB) that your custom application's Setup can run as an executable file to create multiple Program Manager groups and icons. The sample function CreateProgmanGroupIcons() in the ADDICONS.MDB database uses dynamic data exchange (DDE) to create the additional Program Manager groups and icons.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The following steps demonstrate how to set up and use the ADDICONS.MDB database:

1. In Microsoft Access, create a new database called ADDICONS.MDB.
2. Create a new macro with the following actions and arguments, and then save the macro as Autoexec:

Macro Name	Macro Actions
Autoexec	RunCode Quit

Autoexec Actions

RunCode
 Function Name: CreateProgmanGroupIcons()
Quit
 Options: Exit

3. Create the following new table, and then save the table as tbl_Progman:

Field Name: mGroupName
 Data Type: Text
 Field Size: 30
 Required: Yes
Field Name: mIconText
 Data Type: Text
 Field Size: 40
 Required: Yes
Field Name: mIconFile

Data Type: Text
 Field Size: 50
 Field Name: mIconNum
 Data Type: Text
 Field Size: 5
 Field Name: mCommandLine
 Data Type: Text
 Field Size: 120
 Required: Yes
 Field Name: mIconDefault
 Data Type: Text
 Field Size: 120

tbl_Progman Notes:

- The mGroupName field specifies the name of the Program Manager group in which the icons should be created. This is a required entry. This entry should match the entry in the Application Name text box in the Microsoft Access Developer's Toolkit (ADT) Setup Wizard, so that the additional icons are created in the application's program group.
- mIconText specifies the icon's title in the program group. This is a required entry.
- mIconFile specifies the icon's filename. If no filename is specified, a default icon will be used.
- mIconNum specifies the icon's index number in the file specified in mIconFile. If no index is specified, a default icon will be used.
- mCommandLine specifies the full command line required to run the application. This is a required entry.
- mIconDefault specifies the name of the default (or working) directory. If no default directory is specified, the current directory (containing ADDICONS.MDB) will be used.

4. Create a new module and enter the following line in the Declarations section:

Option Explicit

5. Enter the following code in the module.

NOTE: In the following sample code, an underscore (_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```

'*****
' FUNCTION: CreateProgmanGroupIcons()
'
' PURPOSE:
'   To create icons in Program Manager after
'   Setup is run from your ADT distribution disks.
'*****

```

Function CreateProgmanGroupIcons ()

```

Dim Mydb As Database, Rs As Recordset, Rcount As Integer, Xyz _
As Integer

Dim chan As Integer, Exe As String
Dim GroupName$, IconText$, IconFile$, IconNum$, CommandLine$, _
IconDefault$

On Error GoTo Creat_Errors
DoCmd Hourglass False

Set Mydb = DBEngine.Workspaces(0).Databases(0)
Set Rs = mydb.OpenRecordset("tbl_Progman", DB_OPEN_DYNASET)
Rs.MoveLast
Rcount = Rs.recordcount
Rs.MoveFirst

'Begin a DDE conversation with Program Manager.
chan = DDEInitiate("PROGMAN", "PROGMAN")

For Xyz = 1 To Rcount
    GroupName$ = Rs![mGroupName]
    IconText$ = Rs![mIconText]
    IconFile$ = IIf(IsNull(Rs![mIconFile]), "", Rs![mIconFile])
    IconNum$ = IIf(IsNull(Rs![mIconNum]), "", Rs![mIconNum])
    CommandLine$ = Rs![mCommandLine]
    IconDefault$ = IIf(IsNull(Rs![mIconDefault]), CurDir, _
Rs![mIconDefault])

    'The GroupName string names the group to be created or
    'activated.
    'Executing the CreateGroup DDE command to PROGMAN.
    DDEExecute chan, "[CreateGroup(" & GroupName$ & ")]"

    On Error Resume Next
    'If duplicate icons exist they will be replaced.
    DDEExecute chan, "[ReplaceItem(" & IconText$ & ")]"

    'Instructs Program Manager to create a new program icon in
    'the active group specified by the variable GroupName.
    'Executing the AddItem DDE command to PROGMAN.
    Exe = "[AddItem(" & CommandLine$ & ", " & IconText$ & ", " _
    & IconFile$_
    & ", " & IconNum$ & ",,, " & IconDefault$ & ")]"
    DDEExecute chan, Exe
    Rs.MoveNext
Next Xyz
DDETerminate chan
DoCmd Hourglass False
Exit Function
Creat_Errors:
MsgBox Error$
Exit Function
End Function

```

6. Enter the appropriate values in the tbl_Progman table and then close the ADDICONS.MDB database.

7. Run the ADT Setup Wizard to create the distribution disks for your custom application. Make sure to include the ADDICONS.MDB file in the Setup Wizard's Files To Include box.
8. In the "Do you want to run another application after Setup is finished?" screen, enter the following information:
 - a. In the Executable File Name box, enter:

MSARN200.EXE
 - b. In the Command Line box, enter:

MSARN200.EXE ADDICONS.MDB /ini <your app's ini name>

The following are examples of Program Manager items you can create using the CreateProgmanGroupIcons() function:

- To repair and compact the application's database:

```
Field: mGroupName
      Value: NWIND Utilities
Field: mIconText
      Value: Repair/Compact NWIND
Field: mIconFile
      Value: PROGMAN.EXE
Field: mIconNum
      Value: 22
Field: mCommandLine
      Value: MSARN200.EXE NWIND.MDB /INI myapp.ini /Repair /Compact
Field: mIconDefault
      Value: <No value unless you want a specific directory>
```

- To create icons to run additional database command switches:

```
Field: mGroupName
      Value: NWIND Utilities
Field: mIconText
      Value: Solutions Database
Field: mIconFile
      Value: PROGMAN.EXE
Field: mIconNum
      Value: 9
Field: mCommandLine
      Value: MSARN200.EXE ADDICONS.MDB /ini <your app's ini name>
```

- To display your own README.TXT file:

```
Field: mGroupName
      Value: NWIND Utilities
Field: mIconText
      Value: My README.TXT
Field: mCommandLine
      Value: NOTEPAD.EXE NWIND.TXT
```

REFERENCES

=====

For additional information about DDE and Program Manager, please see the following articles in the Microsoft Knowledge Base:

ARTICLE-ID: Q109394

TITLE : How to Add an Icon to Program Manager Using DDE

ARTICLE-ID: Q104943

TITLE : How to Manipulate Groups & Icons in Program Manager
Using DDE

Windows Software Development Kit "Guide to Programming," pages 19-22

Additional reference words: 2.00 install finish

KBCategory: kbusage

KBSubcategory: WzHowto

INF: Creating Custom Setup That Does Not Create Program Group
Article ID: Q123085

The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
-

SUMMARY

=====

This article describes how to create a custom Setup program with the Microsoft Access Developer's Toolkit (ADT) that will not create a program group for your custom application.

MORE INFORMATION

=====

The following steps demonstrate how to create a custom Setup program that will not create a program group for your custom application.

NOTE: These steps demonstrate making changes to the SETUP.STF file. Modifications to this file are not supported by Microsoft. Microsoft provides this information "as is" without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and/or fitness for a particular purpose.

1. Run the Setup Wizard as usual to create Setup disks for your application. Make sure to select the Show In Program Manager Group and Run With MSARN200 check boxes when you select your application.
2. Once the Setup disks are created, make a backup copy of the SETUP.STF file on Disk 1.
3. Open the SETUP.STF file in a spreadsheet program such as Microsoft Excel. SETUP.STF is a tab-delimited file.
4. Search column C for the following entry:

---- User Non File Work ----

This entry should be near the bottom of the column.

5. Remove the first Object ID number in column F of the same row. That number should correspond to the "---- User ProgMan Items ----" entry. Note that there may be multiple Object ID numbers in the cell. Remove only the first number.
6. Save the file. Make sure to save it in tab-delimited format.
7. Run the custom Setup program. Note that no program group is created.

REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0,

Chapter 2, "Creating a Custom Setup Program," pages 19-37

Additional reference words: 2.00 progman icon

KBCategory: kbsetup

KBSubcategory: StpHowto

INF: Creating Fixed-Scale Graphs in Reports and Forms

Article ID: Q109737

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
 - Microsoft Graph versions 3.0 and 5.0
-

SUMMARY

=====

When you graph records with Microsoft Graph in Microsoft Access, Microsoft Graph automatically adjusts the value axis scale for each record. This article describes how to create a graph with a fixed scale for all records manually.

MORE INFORMATION

=====

By default, Microsoft Graph adjusts the value axis scale for each record in the graph. If you want a fixed-value axis scale for all records, you must modify settings in the Microsoft Graph Format Scale dialog box. The following settings can be adjusted in the Format Scale Dialog box:

- **Auto:** Indicates that Microsoft Graph will automatically adjust chart scale values. Clear the Auto check boxes to set fixed scale values.
- **Minimum:** The minimum value of the value axis scale. This value is typically set to zero, but can be set to other values.
- **Maximum:** The maximum value of the value axis scale. This value should be greater than or equal to the largest value in the records you are charting.
- **Major Unit:** The primary unit of measure on the value axis. This value represents the increment value for tick marks on the scale.
- **Minor Unit:** The secondary unit of measure on the value axis. This setting applies to tick marks falling between the major units. This value must be greater than zero and less than or equal to the major unit value. This setting is typically set to the value of the major unit.
- **Category (X) Axis Crosses At:** This value determines where on the value axis the category axis will fall. Typically, this value is set to zero.
- **Values In Reversed Order:** This setting inverts the chart and displays the lowest value on the scale at the top of the chart, and the highest value at the bottom of the chart.

The following example demonstrates how to create a graph, and how to adjust the scale settings for a fixed-scale value axis:

1. Open the sample database NWIND.MDB.

2. Create the following new query based on the Orders table, and then save the query as Percentage Test Query1:

Query: Percentage Test Query1

Type: Select Query

Join: None

Field: Order ID

Criteria: None

Field: Order Amount

Criteria: > 0

Field: Freight

Criteria: > 0

Field: PercentValue: [Freight] / [Order Amount] * 100

Criteria: <= 1.0

3. Create the following new report based on the Percentage Test Query1 query, and then save the report as Fixed Scale Graph Report:

Report: Fixed Scale Graph Report

Text box:

ControlSource: Order Id

Text Box:

ControlSource: PercentValue

Graph:

Data Source: Percentage Query1

Totals: Sum (default)

Type: Horizontal Bar (top row, 2nd from left)

Data to graph: PercentValue

Link to Report: Yes

Link Fields: Order Id

4. Preview the report. Note that each graph has a different horizontal scale.
5. View the report in Design view. Double-click the graph object to start Microsoft Graph.
6. Double-click the bottom horizontal scale in the graph, then click the Scale tab in the Format Axis dialog box. Modify the settings in the Format Axis dialog box to match the following, and then choose OK:

Auto: clear all Auto check boxes

Minimum: 0

Maximum: 1

Major Unit: 0.1

Minor Unit: 0.1

7. From the File menu, choose Exit And Return To Microsoft Access. Choose Yes when you are prompted to update the graph.
8. Preview the report. Note that the value scales are consistent for all the records.

REFERENCES

=====

Microsoft Graph "User's Guide," version 3.0, Chapter 3, "Formatting the Axis Scale," pages 87-92

Additional reference words: 1.00 1.10 2.00

KBCategory: kbtool

KBSubcategory: IntpGrph

INF: Creating Floating Pop-Up Menus

Article ID: Q95444

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to create "floating" pop-up menus using a combination of Microsoft Windows application programming interface (API) calls and Access Basic code.

MORE INFORMATION

=====

To create a floating pop-up menu, perform the following eight steps.

1. Create a new form called "Pop-up Menu Form."
2. Add a List Box control called "Menu" to the form. Make the control large enough to display each of the strings it contains without displaying a scroll bar. Give the control the following properties:

List Box Properties

ControlName: Menu
RowSourceType: Value List
RowSource: String1;String2;String3;String4
AfterUpdate: =ItemSelected([Menu])
Left: 0 in
Top: 0 in
FontName: System
FontSize: 8
FontWeight: Bold

NOTE: Specify the menu strings in the RowSource property. Separate succeeding string values with a semicolon (;) character.

3. Change the properties of the "Pop-up Menu Form" as indicated below.

Form Properties

ScrollBars: Neither
Popup: Yes
RecordSelectors: No
Width: <Width of the list box control>

Section Properties

Height: <Height of the list box control>

4. Save the form and close it.

5. Create a new module and place the following code into the Declarations section for the module.

```
'NOTE: Some of the following Windows API functions may be
'defined in an existing Microsoft Access library. If so, the new
'declarations would cause a duplication procedure name error. If
'this error occurs, remove the offending declare statement from
'your code or convert the declaration to a comment.
```

```
'NOTE: In the following sample code, an underscore (_) is used
'as a line continuation character. Remove the underscore when
're-creating this code in Access Basic.
```

```
Option Explicit
Type POINTAPI
    x As Integer
    y As Integer
End Type
```

```
Global Const GWL_STYLE = (-16)
Global Const WS_DLGFAME = &H400000
```

```
Declare Sub GetCursorPos Lib "User" (lpPoint As POINTAPI)
Declare Function GetWindowLong& Lib "User" (ByVal hWnd%, _
    ByVal nIndex%)
Declare Function SetWindowLong& Lib "User" (ByVal hWnd%, _
    ByVal nIndex%, ByVal dwNewLong&)
```

6. Add the following functions to the module.

```
Function ShowPopup ()
    Dim coord As POINTAPI
    Dim attr&

    GetCursorPos coord
    DoCmd OpenForm "Pop-up Menu Form"

    attr& = GetWindowLong(Forms![Pop-up Menu Form].hWnd, GWL_STYLE)
    attr& = SetWindowLong(Forms![Pop-up Menu Form].hWnd, GWL_STYLE, _
        attr& And Not WS_DLGFAME)

    DoCmd MoveSize (coord.x * 14), (coord.y * 14), , 1100
End Function

Function ItemSelected (WhichItem As String)
    DoCmd Close
    MsgBox "The selected item was " & Trim(WhichItem)
End Function
```

7. Choose the form in which you want the pop-up menu to appear. Open the form in Design view.
8. Select the event property that you want to use to activate the pop-up menu and specify the following function call:

```
=ShowPopup()
```

When you open your form and cause the event assigned in Step 8 above, the pop-up menu appears at the current mouse position. The menu remains on the screen until you select an item from the pop-up menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

INF: Creating Mailing Labels for Dot-Matrix Printers

Article ID: Q103433

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Creating labels that print on dot-matrix or tractor-feed printers usually requires you to adjust the page size. Before you create a mailing label report for these printers, you must first set the default printer and paper size in Microsoft Windows Control Panel.

MORE INFORMATION

=====

Setting Default Printer/Paper Size in Control Panel

To set the default printer and paper size, do the following:

1. In the Printers dialog box in Control Panel, select a dot-matrix printer, then choose the Set As Default Printer button. (If you do not already have a dot matrix printer driver installed, you must first install one.)
2. Choose the Setup button to display the dialog box for that printer.
3. In the Paper Size box, select User-Defined Size.
4. In the User-Defined Size dialog box, select the unit of measurement and then type the width and the length of the mailing label page.

To determine the width, measure from the left edge of the leftmost label to the right edge of the rightmost label. To determine the length, measure from the top of the first label to the top of the second label.

For example, if you use two-up labels that measure 3 inches by 5 inches (with .25 inches between), set Unit to 0.01 inches, Width to 625, and Length to 525.

5. Choose OK twice, then choose Close.

Once you set the printer and paper size, you can create a mailing label report for a dot-matrix printer using the Mailing Label ReportWizard, or using tools provided in the Design view of the report.

Creating a Mailing Label Report Not Using ReportWizard

1. Open your report in Design view and remove the page header and

footer. (You need only the Detail section for this report.)

2. Adjust the size of the Detail section as follows:
 - a. Make the width the same as the width of one label.
 - b. Make the height the same as the height of one label.
3. Add controls to the Detail section, making sure that the controls do not overlap. To create margins for the labels, place the text boxes at least 0.2 inches from all edges of the Detail section.
4. From the File menu, choose Print Setup and verify or set the following values in the Print Setup dialog box:
 - a. In the Paper group, set Size to User-Defined Size and Source to Tractor.
 - b. Set all margins to 0.0 inches.
 - c. In the Item Size group, make sure the Same as Detail check box is selected. (Choose the More button to display the Item Size group.)
 - d. If there is more than one label across the page, you must also set the following items:
 - 1) In the Items Across box, enter the number of labels across.
 - 2) In the Row Spacing box, enter the amount of space between the bottom of one label and the top of the next.
 - 3) In the Column Spacing box, enter the amount of space between the right edge of one label and the left edge of the next.

NOTE: When a dot-matrix printer is the default in Control Panel and you have Paper Size set to User-Defined Size, Microsoft Access uses only that paper size. If you want to create or print a label or report on a page of a different size, remember to change the user-defined page size in Control Panel before you create or print your label or report.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptLabel

INF: Creating Reports to Mail Merge Microsoft Access Data
Article ID: Q98798

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

To use a mail merge with Microsoft Access data, you can export or copy your data to a word processor, or in Microsoft Access 2.0, you can choose the Merge It button on the toolbar. However, you can also create a Microsoft Access report to generate simple mail-merge documents. This article demonstrates how to generate a mail-merge document.

MORE INFORMATION

=====

The following is an example of a business letter that you can create:

NOTE: All values shown in angle brackets (<>) are found in the underlying table.

KB Advertising Agency

<February 5th, 1993>

<John Doe>

<123 E. Main Street>

<Middle Town, OH 44044>

Dear <Mr. Doe>,

We are pleased to announce our new line of office furniture. As a <valued> customer, you are entitled to a <10>% discount off our catalog prices.

Call us today toll-free at (800) 555-1212 and receive a free full-color catalog detailing all our top-of-the-line office products.

Sincerely,

J. Grant

Manager, Office Products

How to Create the Sample Document

1. Create the following new table:

Table: Customer

FieldName: Salutation

DataType: Text

FieldName: First Name
DataType: Text

FieldName: Last Name
DataType: Text

FieldName: Address
DataType: Text

FieldName: City
DataType: Text

FieldName: State
DataType: Text

FieldName: ZIP
DataType: Text

FieldName: Customer Type
DataType: Text

FieldName: Discount Pct
DataType: Number

2. Enter the following records in the new table:

Salu- tation	First Name	Last Name	Address	City	State	ZIP	Cust Type	Disc Pct
Mr.	John	Doe	123 Main	Middleton	OH	44044	Valued	10
Mrs.	Sally	White	52A Elm	Florence	KY	45123	Special	5
Dr.	Fred	Weiss	Box 456	Camden	SC	29332	Super	25

3. Create the following new report based on the Customer table:

NOTE: Do not use a Microsoft Access ReportWizard to create this report.

- From the Layout menu in Microsoft Access 1.x, or the Format menu in Microsoft Access 2.0, turn off Report Hdr/Ftr and Page Hdr/Ftr (so that no check mark appears next to either command).
- Add all the table controls to the report's detail section.
- Delete the default text box labels.
- Set the following properties:

Report: Merge Letter

RecordSource: Customer

Section: Detail

ForceNewPage: After Section

CanGrow: Yes

Control: Label

ControlName: Company Label
Caption: KB Advertising Agency

Control: Text Box

ControlName: Todays Date
ControlSource: =Date()
Format: Long Date

Control: Text Box

ControlName: Name Line
ControlSource: =[First Name] & " " & [Last Name]

Control: Text Box

ControlName: Address Line
ControlSource: =[Address]

Control: Text Box

ControlName: CSZ Line
ControlSource: =[City] & ", " & [State] & " " & [ZIP]

Control: Text Box

ControlName: Salutation Line
ControlSource: ="Dear " & [Salutation] & " " & [Last Name] & ", "

Control: Text Box

ControlName: Para 1
ControlSource: ="We are pleased to announce our new line of office
furniture. As a " & [Customer Type] & "customer,
you are entitled to a" & [Discount Pct] & "%
discount off our catalog prices."

CanGrow: Yes

Control: Label

ControlName: Rest of Letter
Caption: Call us today toll-free at (800) 555-1212 and receive a
free full-color catalog detailing all our
top-of-the-line office products.

4. Add the following label controls:

Sincerely,

J. Grant
Manager, Office Products

5. Size and position the controls, then set the font styles and sizes.

NOTE: Press CTRL+ENTER to force new lines and paragraphs in

controls.

6. Save the report. Print or preview the report to see the results of the merge.

Summary and General Notes

- Fixed paragraphs should be entered in label controls.
- Variable paragraphs should be entered in text box controls.
- Margins are determined by the width of the control. The text will wrap within the width of the control, not the width of the letter.
- The ControlName of a text box must be different from its field name in the table. If they are the same, the ControlName prints as a "#Error?" message.

Limitations

- You cannot change the formatting of individual words. You can change only the formatting of the control.
- You are limited to 255 characters in the ControlSource property of a text box. For long paragraphs, create one or more invisible text boxes to contain the text, then concatenate them in a visible text box.

Additional reference words: 1.00 1.10 2.00 mailmerge page header

KBCategory: kbusage

KBSubcategory: RptOthr

INF: Creating Virtual Indexes with SQL Data-Definition Queries
Article ID: Q112131

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

If your server supports the creation of SQL views, you can achieve a compromise between pass-through and Microsoft Access queries by creating a view on the server and then attaching it. The Microsoft Jet database engine treats an attached view exactly like an attached table (with no indexes). The processing defined in a view is always performed by the server, no matter what the Jet database engine decides to execute locally.

MORE INFORMATION

=====

If your server supports updating through views, you need to create an index specification on the attachment to tell Microsoft Access which fields uniquely specify a record returned by the view. This lets the Jet database engine create an updatable dynaset on the view, and on Microsoft Access queries and forms that use it.

For example, you could run the Microsoft Access data-definition query

```
CREATE UNIQUE INDEX index1 ON SeptemberOrders (OrderID)
```

on an attached view named SeptemberOrders that returns a subset of the remote Orders table and has a unique field named OrderID.

This query does not create anything on the server, nor does it take any extra space in your Microsoft Access database. It simply tells the Jet database engine that the OrderID field is the primary key for the attached view. Note that this is not a "real" index in that it does not improve query performance. It is only used to serve as a reference for the Jet database engine.

REFERENCES

=====

For more information about data-definition queries, search for "Data Definition" then "Data Definition Command" using the Microsoft Access Help menu.

Additional reference words: 2.00 Attaching

KBCategory: kbusage

KBSubcategory: QryOthr

INF: Criteria Parameters Require Concatenated References
Article ID: Q95978

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Microsoft Access has both a SQL interpreter and an Access Basic interpreter. Because some Access Basic commands and functions require SQL syntax as a parameter, it may become confusing how the two work together.

For example, if you need to look up a description in a table called Categories where [Category ID] is equal to whatever value is currently in Forms!Form1!Field1, you cannot use:

NOTE: In the following sample code, an underscore _ is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
DLookup("Description", "Categories",_  
        "[Category ID] = Forms!Form1!Field1")
```

Instead, you must concatenate the value of Forms!Form1!Field1 into the criteria parameter as shown below:

```
DLookup("Description", "Categories",_  
        "[Category ID] = '" & Forms!Form1!Field1 & "'")
```

This means that if the value of Forms!Form1!Field1 is "BEVR," the criteria parameter in the expression above would evaluate to:

```
[Category ID] = 'BEVR'
```

MORE INFORMATION

=====

When you make a function call such as:

```
DLookup("Description", "Categories", "[Category ID] = 'BEVR'")
```

Access Basic internally creates a SQL statement with the parameters you supply. In this case, the resulting SQL statement is:

```
SELECT Description FROM Categories WHERE [Category ID] = 'BEVR'
```

Access Basic sends this statement to the Access SQL engine, which returns the result of the SELECT statement back to Access Basic, and, in turn, back to your DLookup() expression.

Note that that a DLookup() coded as follows


```
DLookup("Description", "Categories", _  
    "[Category ID] = Forms!Form1!Field1")
```

would result in a WHERE clause that looks like:

```
[Category ID] = Forms!Form1!Field1
```

The SQL parser is designed to accept SQL syntax, and the Forms!Form1!Field1 reference is Access Basic syntax, so the form reference cannot be expected to work. Instead, the resulting evaluated criteria parameter must be done in such a way as to accommodate the syntax rules of the SQL parser.

Note that this also applies to other Access Basic methods and functions that require a criteria parameter. For example, Find methods use a criteria parameter as shown below:

```
Dim D As Database, S As Snapshot  
Set D = CurrentDB()  
Set S = D.CreateSnapshot("Categories")  
myvar = "BEVR"  
S.FindFirst "[Category ID] = '" & myvar & '"
```

Additional reference words: 1.00 1.10 2.00 concatenation

KBCategory: kbprg

KBSubcategory: PgmOthr

INF: Currency Format Causes Report Alignment Problems

Article ID: Q95927

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

In a report, if you set the Format property of a number control to Currency, you may have difficulty aligning the formatted numbers with numbers that are not formatted to Currency. For example, if you format totals in a group footer section to Currency but do not format the numbers above or below the total, the unformatted numbers may not align with the formatted numbers.

MORE INFORMATION

=====

Currency amounts in the detail section may not always have a currency prefix (\$), but the total amount in the group footer usually will have the prefix, as in the following example:

(detail section)		
	Doodads	4.50
	Widgets	10.00
(group footer)		
	Total:	\$ 14.50

If the Format property of the text box in the group footer is Currency, the decimal point and numbers may not line up with the decimal point and numbers in the detail section.

To correct this problem, set the Format property of the number controls to user-defined format, rather than Currency, for the totals in your report. You can enter a user-defined format in the Format property of a text box.

You can use "\$#,##0.00" (without quotation marks) for the Format property of a number when you want to display the dollar sign, and use "#,##0.00" (without quotation marks) for the Format property of a number when the dollar sign is not required. These examples format the numbers in your report to use the comma as the thousands separator, and format each number to have two decimal places. Numbers formatted in this manner align correctly in the report.

REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 206 and 215-216

For more information in Microsoft Access version 1.x, search for "Format, Format\$ Functions" using the Help menu.

For more information in Microsoft Access version 2.0, search for "Formatting Numbers" using the Help menu.

Additional reference words: 1.00 1.10 2.00 format

KBCategory: kbusage

KBSubcategory: RptLayou

**INF: Custom Domain Functions Similar to DFirst() and DLast()
Article ID: Q103403**

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

By design, the DFirst() and DLast() functions always return the first and last record that you entered in the underlying table, not the first and last record listed in the sorted query. This article provides examples of how to write custom domain functions that can be used in a way similar to DFirst() and DLast().

The DStart() and DEnd() custom domain examples in this article return the first and last records listed in a sorted query. Also included in this article is the DFix() function, which can be used to overcome a limitation of concatenating variables into criteria strings. Information about the limitations of custom domain functions and how to use DFix() appears at the end of the article.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications with the programming tools provided with Microsoft Access.

MORE INFORMATION

=====

Create a new module within Microsoft Access and add the following functions with the appropriate declaration section. Each command should be entered on one line.

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----  
' GLOBAL DECLARATION  
'-----  
Option Compare Database  
Option Explicit  
  
'-----  
' Use DStart() instead of DFirst() to return  
' the first sorted record in a domain.  
'-----  
Function DStart (FieldName As String, DomainName As String, _  
                Criteria As String)  
  
    Dim MyDB As Database, Myset As Dynaset  
  
    ' ERROR OUT IF THERE IS NO FIELDNAME SENT.  
    If Len(FieldName) = 0 Then
```

```

        MsgBox "You Must Specify a Field name", , "DStart"
    End
End If

' ERROR OUT IF THERE IS NO DOMAIN SENT.
If Len(DomainName) = 0 Then
    MsgBox "You Must Specify a Domain name", , "DStart"
    Exit Function
End If

Set MyDB = CurrentDB()
Set Myset = MyDB.CreateDynaset(DomainName)

'
If Len(Criteria) > 0 Then
    Myset.Filter = Criteria
    Set Myset = Myset.CreateDynaset()
End If

' IF THERE ARE NO RECORDS, RETURN THE NULL, ELSE RETURN THE VALUE
' OF THE FIRST RECORD.
If Myset.EOF Then
    DStart = Null
Else
    Myset.MoveFirst
    DStart = Myset(Fieldname)
End If

Myset.Close
MyDB.Close
End Function

'-----
'Use DEnd()instead of DLast() to return
' the last sorted record in a domain.
'-----
Function DEnd (FieldName As String, DomainName As String, _
              Criteria As String)
    Dim MyDB As Database, Myset As Dynaset

    ' ERROR OUT IF THERE IS NO FIELDNAME SENT.
    If Len(FieldName) = 0 Then
        MsgBox "You Must Specify a Field name", , "DEnd"
        Exit Function
    End If

    ' ERROR OUT IF THERE IS NO DOMAINNAME SENT.
    If Len(DomainName) = 0 Then
        MsgBox "You Must Specify a Domain name", , "DEnd"
        Exit Function
    End If

    Set MyDB = CurrentDB()
    Set Myset = MyDB.CreateDynaset(DomainName)

    If Len(Criteria) > 0 Then
        Myset.Filter = Criteria

```

```

        Set Myset = Myset.CreateDynaset()
    End If
    If Myset.EOF Then
        DEnd = Null
    Else
        Myset.MoveLast
        DEnd = Myset(Fieldname)
    End If
    Myset.Close
    MyDB.Close
End Function

```

```

Function DFix (ByVal T, DQuote As Integer)

```

```

'-----
' Fixes string arguments that are passed
'   to Criteria in domain functions
'
' DQuote should be TRUE or -1 if Double Quotes (") delimit Criteria
' DQuote should be FALSE or 0 if Single Quotes (') delimit Criteria
'
' For example, this gives an error
'   (note the quote (') in the data)
'   X="Mike's Diner"
'   A=DStart("ID","Clients","Name='" & X & "'")
'
' Use either:
'   X=DFix("Mike's Diner",False)
' Or:
'   A=DStart("ID","Clients","Name='" & DFix(X,False) & "'")
'-----

```

```

Dim P As Integer, OldP As Integer, Q As String * 1

```

```

If VarType(T) = 8 Then
    If DQuote = 0 Then
        Q = "'"
    Else
        Q = """"
    End If
    P = InStr(T, Q)
    Do While P > 0
        OldP = P + 2
        T = Left$(T, P) & Q & Mid$(T, P + 1)
        P = InStr(OldP, T, Q)
    Loop
End If
DFix = T
End Function

```

```

General Limitations to Custom Domain Functions

```

-
- You cannot use Forms!FormName!ControlName or Form.ID in quotation marks. For example, instead of

```

DStart("ID","LOG","Name=Form.Name")

```

use:

```
DStart("ID","LOG","Name = '" & DFix(Name,False) & "'")
```

The DFix() function is another custom function.

- You must always specify a criteria, even if it is "" or Null.
- Do not place field names or domain names in square brackets ([]), that is, use "First Name" instead of "[First Name]".

NOTE: DFix() can be used anywhere you concatenate criteria strings, such as the FindFirst method or when building custom SQL criteria.

REFERENCES

=====

For information about other custom domain functions, query on the following words here in the Microsoft Knowledge Base:

custom and domain and function

Additional reference words: 1.00 1.10 2.00 ab dfirst dlast modules

KBCategory: kbprg

KBSubcategory: PgmOthr

INF: Custom Menus Cannot Be Implemented with Pop-up Forms
Article ID: Q90150

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

A form will not process the macro associated with the OnMenu or Menubar properties 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 it. 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.

NOTE: According to "The Windows Interface - An Application Design Guide," dialog boxes (pop-up) generally do not have menu bars.

REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00 modal

KBCategory: kbusage

KBSubcategory: FmsEvt

INF: DAO2016.DLL Not Available from External Applications
Article ID: Q115899

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

You cannot call the DAO2016.DLL file from an application written in C to use DAO code to control Microsoft Access tables.

The only application that can call the DAO2016.DLL file is Microsoft Access version 2.0. Trying to call this dynamic-link library (DLL) from any other program will not work and is not supported by Microsoft.

MORE INFORMATION

=====

DAO2016.DLL (the filename refers to Data Access Objects, version 2.0, 16 bit) is the DLL that Microsoft Access uses from Access Basic to modify database objects in the Microsoft Jet database engine.

If you try to use OLE functions from C/C++ to use this DLL, you will receive the error message "VBAJET.DLL not found." This is a misleading error message in that the file VBAJET.DLL does not exist as part of the Microsoft Access installation at all. A more appropriate error message would be "Feature not available."

Using tools such as OLE2VIEW.EXE from Microsoft Visual C++, it is possible to look at some of the objects and methods that are exposed in the DAO2016.DLL file. This is analogous to looking at the entry points to a DLL, except that the information is in a form that is more useful and easier to decode.

While these methods are listed, there is no documented way to call them, and they are not registered in the Microsoft Windows Registration database as OLE Methods that are available for OLE Automation.

Neither Microsoft Access nor the Microsoft Jet database engine are OLE Servers. They cannot be controlled using OLE Automation techniques.

Additional reference words: 2.00

KBCategory: kbole kberrmsg

KBSubcategory: IntpOlea

INF: Data Type of Value Overrides Data Type of Variable

Article ID: Q98796

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Microsoft Access internally converts values with implicit data type definitions before assigning those values to variables. This may result in unexpected rounding of numbers, or in the following error message:

```
Invalid Number
```

MORE INFORMATION

=====

You can implicitly define the data type of a value by appending a type-declaration character at the end of the value. For example, `x=1234&` implicitly defines 1234 as a Long data type (four-byte integer).

Steps to Reproduce Behavior

1. Open a new module.
2. Add the following line underneath the Option Declare Database line:

```
Dim v as Currency
```

3. Press ENTER.
4. From the View menu, choose Immediate Window.
5. In the Immediate window, type the following:

```
v=123456789123.3456789#
```

6. Press ENTER, then type the following line

```
debug.print v
```

and press ENTER again.

The value of the variable <v> is now 123456789123.3457. The number sign (#) implicitly defined the value as a double-precision value. Microsoft Access internally converted the value to that data type before assigning it to the currency variable.

7. Repeat steps 5-6, replacing # with other type-declaration characters (such as the ampersand (&), exclamation point (!), or at sign (@) symbol).

The at sign defines the currency data type, which is limited to four decimal places. When you try to define a number with more than four decimal places as currency, an "invalid number" error message is displayed.

REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 25-26

Microsoft Access "Introduction to Programming," version 1.0, pages 49-51

Additional reference words: 1.00 1.10 2.00 ui misc

KBCategory: kbprg

KBSubcategory: PgmOthr

INF: Database Normalization Basics

Article ID: Q100139

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article explains database normalization terminology for beginners. A basic understanding of this terminology is helpful when discussing the design of a relational database.

MORE INFORMATION

=====

Description of Normalization

Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables, according to rules designed both to protect the data and to make the database more flexible by eliminating two factors: redundancy and inconsistent dependency.

Redundant data wastes disk space and creates maintenance problems. If data that exists in more than one place must be changed, the data must be changed in exactly the same way in all locations. A customer address change is much easier to implement if that data is stored only in the Customers table and nowhere else in the database.

What is an "inconsistent dependency"? While it is intuitive for a user to look in the Customers table for the address of a particular customer, it may not make sense to look there for the salary of the employee who calls on that customer. The employee's salary is related to, or dependent on, the employee and thus should be moved to the Employees table. Inconsistent dependencies can make data difficult to access; the path to find the data may be missing or broken.

There are a few rules for database normalization. Each rule is called a "normal form." If the first rule is observed, the database is said to be in "first normal form." If the first three rules are observed, the database is considered to be in "third normal form." Although other levels of normalization are possible, third normal form is considered the highest level necessary for most applications.

As with many formal rules and specifications, real world scenarios do not always allow for perfect compliance. In general, normalization requires additional tables and some customers find this cumbersome. If you decide to violate one of the first three rules of normalization, make sure that your application anticipates any problems that could occur, such as redundant data and inconsistent dependencies.

NOTE: The following descriptions include examples.

First Normal Form

- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key.

Do not use multiple fields in a single table to store similar data. For example, to track an inventory item that may come from two possible sources, an inventory record may contain fields for Vendor Code 1 and Vendor Code 2.

But what happens when you add a third vendor? Adding a field is not the answer; it requires program and table modifications and does not smoothly accommodate a dynamic number of vendors. Instead, place all vendor information in a separate table called Vendors, then link inventory to vendors with an item number key, or vendors to inventory with a vendor code key.

Second Normal Form

- Create separate tables for sets of values that apply to multiple records.
- Relate these tables with a foreign key.

Records should not depend on anything other than a table's primary key (a compound key, if necessary). For example, consider a customer's address in an accounting system. The address is needed by the Customers table, but also by the Orders, Shipping, Invoices, Accounts Receivable, and Collections tables. Instead of storing the customer's address as a separate entry in each of these tables, store it in one place, either in the Customers table or in a separate Addresses table.

Third Normal Form

- Eliminate fields that do not depend on the key.

Values in a record that are not part of that record's key do not belong in the table. In general, any time the contents of a group of fields may apply to more than a single record in the table, consider placing those fields in a separate table.

For example, in an Employee Recruitment table, a candidate's university name and address may be included. But you need a complete list of universities for group mailings. If university information is stored in the Candidates table, there is no way to list universities with no current candidates. Create a separate Universities table and link it to the Candidates table with a university code key.

EXCEPTION: Adhering to the third normal form, while theoretically desirable, is not always practical. If you have a Customers table

and you want to eliminate all possible interfield dependencies, you must create separate tables for cities, ZIP codes, sales representatives, customer classes, and any other factor that may be duplicated in multiple records. In theory, normalization is worth pursuing; however, many small tables may degrade performance or exceed open file and memory capacities.

It may be more feasible to apply third normal form only to data that changes frequently. If some dependent fields remain, design your application to require the user to verify all related fields when any one is changed.

Other Normalization Forms

Fourth normal form, also called Boyce Codd Normal Form (BCNF), and fifth normal form do exist, but are rarely considered in practical design. Disregarding these rules may result in less than perfect database design, but should not affect functionality.

 * Examples of Normalized Tables *

Normalization Examples:

Unnormalized table:

Student#	Advisor	Adv-Room	Class1	Class2	Class3
1022	Jones	412	101-07	211-02	214-01

1. First Normal Form: NO REPEATING GROUPS

Tables should have only two dimensions. Since one student has several classes, these classes should be listed in a separate table. Fields Class1, Class2, & Class3 in the above record are indications of design trouble.

Spreadsheets often use the third dimension, but tables should not. Another way to look at this problem: with a one-to-many relationship, do not put the one side and the many side in the same table. Instead, create another table in first normal form by eliminating the repeating group (Class#), as shown below:

Student#	Advisor	Adv-Room	Class#
1022	Jones	412	101-07
1022	Jones	412	143-01
1022	Jones	412	159-02
4123	Smith	216	201-01
4123	Smith	216	211-02
4123	Smith	216	214-01

2. Second Normal Form: ELIMINATE REDUNDANT DATA

Note the multiple Class# values for each Student# value in the

above table. Class# is not functionally dependent on Student# (primary key), so this relationship is not in second normal form.

The following two tables demonstrate second normal form:

Students:	Student#	Advisor	Adv-Room
	1022	Jones	412
	4123	Smith	216

Registration:	Student#	Class#
	1022	101-07
	1022	143-01
	1022	159-02
	4123	201-01
	4123	211-02
	4123	214-01

3. Third Normal Form: ELIMINATE DATA NOT DEPENDENT ON KEY

In the last example, Adv-Room (the advisor's office number) is functionally dependent on the Advisor attribute. The solution is to move that attribute from the Students table to the Faculty table, as shown below:

Students:	Student#	Advisor
	1022	Jones
	4123	Smith

Faculty:	Name	Room	Dept
	Jones	412	42
	Smith	216	42

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, pages 18-19

"FoxPro 2 A Developer's Guide," Hamilton M. Ahlo Jr. et al., pages 220-225, M & T Books, 1991

"Using Access for Windows," Roger Jennings, pages 799-800, Que Corporation, 1993

Additional reference words: 1.00 1.10 2.00 BCNF relational model normal

KBCategory: kbusage

KBSubcategory: TblOthr

INF: DatePart(), DateDiff() Functions Ignore Global Settings

Article ID: Q121012

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

In Microsoft Access version 2.0, you can use the First Week and First Weekday options settings to specify values other than the default values of January 1 for the first week, and Sunday for the first weekday.

If you do use these option settings to change the default values, you must specify the optional setting when you use either the DatePart() or the DateDiff() function.

To change these option settings, choose Options from the View menu, select General in the Category box, and then in the Items box select the option you want to change.

MORE INFORMATION

=====

In Microsoft Access 2.0, two new parameters have been added to the DatePart() and DateDiff() functions. The new parameters are FirstWeek and FirstWeekDay. If you change the default setting for the First Week or the First Weekday option, you must explicitly define the matching parameter when you use the DatePart() or DateDiff() function.

For example, if you set the First Week option setting to First Full Week, and the First Weekday option setting to Monday, use the following syntax for the DateDiff() and DatePart() functions:

- DateDiff("ww","11/1","11/30",2,3)

-or-

DateDiff("ww","11/1","11/30",MONDAY,3)

-or-

DateDiff("ww","11/1","11/30",MONDAY,0)

- DatePart("w","31-Dec",2,3)

-or-

DatePart("w","31-Dec",MONDAY,3)

-or-

DatePart("w","31-Dec",MONDAY,0)

REFERENCES

=====

For more information about the DateDiff() function, search for "DateDiff" then "DateDiff Function" using the Microsoft Access Help menu.

For more information about the DatePart() function, search for "DatePart" then "DatePart Function" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: ExrFnc

INF: Debugging & Troubleshooting Tips for Attached SQL Tables
Article ID: Q102437

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

Listed below are some techniques that can be used to determine indirect causes of error messages when using attached SQL tables:

1. Use Microsoft Windows File Manager to search for multiple, corrupted, or outdated copies of the following files:

ODBC.DLL - dated 10-16-92 or later, with 44736 bytes
DBNMP3.DLL - dated 9-15-92 or later, with 8241 bytes
SQLSRVR.DLL - dated 10-16-92 or later, with 135792 bytes
NETAPI.DLL - depends on the network being used
COMMDLG.DLL - dated 10-25-92 or later, with 89248 bytes

NOTE: The sizes and dates referenced are correct if the files were installed by Microsoft Access version 1.0.

2. Make sure that all the files listed in step 1 are in the appropriate locations. All files except NETAPI.DLL are usually installed in the Windows SYSTEM subdirectory. NETAPI.DLL is a network-specific driver; it is usually located in the network's program directory, which should appear in the path.
3. Use another application, such as the three listed below, on the same computer and try to attach the same data source:
 - Q+E (comes with Microsoft Excel)
 - SQL Server Administration Facility (SAF)
 - PowerBuilder

If you cannot access the data source from another application, you may have a problem with the network, not with Microsoft Access or the Microsoft Open Database Connectivity (ODBC) driver.

4. Try to attach another data source, such as another SQL Server table, from a different database server. If you can attach the other data source, the original server may be unavailable, or you may need to reconfigure the entry for that server using the ODBC Administration utility.
5. Try to attach the same data source from a different computer. If you still cannot gain access to the SQL Server computer, you may have a network-wide problem, or perhaps the server is down or not communicating.
6. Try increasing the ODBC time-out settings in the [ODBC] section of your MSACCESS.INI file.
7. Rename the current ODBC.DLL, DBNMP3.DLL, and SQLSRVR.DLL files, restart the server, and reinstall ODBC.

For more information on this procedure, see the Microsoft Access "User's Guide," version 1.0, Appendix D, "Setting Up Microsoft Access on a Network," pages 660-662.

PowerBuilder is manufactured by Powersoft Corporation, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg kbtshoot

KBSubcategory: PgmOthr

INF: Defragment and Compact Database to Improve Performance
Article ID: Q92681

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

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 multiuser 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.

Additional reference words: 1.00 1.10 2.00 compaction

KBCategory: kbusage

KBSubcategory: GnlCmp

INF: Deleting Duplicate Records from Two Tables

Article ID: Q98664

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to delete duplicate records from two tables and retain the most current information (records). In addition, the article shows how to merge the results of these two tables.

MORE INFORMATION

=====

This article walks you through creating two tables, TEST-1 and TEST-2, with duplicate records. After adding additional nonduplicate records to both these tables, you will delete the duplicate records from the TEST-2 table. Finally, you will update the TEST-1 table to include nonduplicate information from the TEST-2 table.

Steps to Reproduce Behavior

1. Open the sample database NWIND.MDB.
2. In the Database window, select the Shippers table.
3. From the Edit menu, choose Copy.
4. From the Edit menu, choose Paste.
5. In the Paste Table As dialog box, type "TEST-1" (without quotation marks) in the Table Name field. Choose OK.
6. Repeat steps 4 and 5, this time pasting the table as TEST-2.
7. Open the TEST-1 table and add the following two additional company names:

 ABC Shipping
 XYZ Express

Close the TEST-1 table.

8. Open the TEST-2 table and add the following two new company names:

 Jiffy Ship
 Hurry Package

Close the TEST-2 table.

9. Create a new query based on the TEST-1 and TEST-2 tables.

10. In Microsoft Access version 1.x, join the tables on the Company Name field. In Microsoft Access version 2.0, remove the join between the Shipper ID fields, then join the tables on the Company Name field.
11. From the Query menu, choose Delete. For this example, the TEST-1 table will be considered the most current information; therefore, duplicate information will be deleted from the TEST-2 table.
12. From the TEST-1 table, drag Shipper ID and Company Name to the QBE grid. Note that the word "Where" appears on the Delete line in the QBE grid.
13. Drag the * from the TEST-2 table to the QBE grid. Note that the word "From" appears on the Delete line in the QBE grid.
14. From the Query menu, choose Run.
15. You will receive a message stating that three rows will be deleted. Choose OK.
16. Open the TEST-2 table. The only two records that remain are the nonduplicate records that you added in step 8 above. All the duplicate records have been deleted.

The remaining steps append the remaining nonduplicate records from the TEST-2 table to the TEST-1 table.

17. Open the TEST-2 table.
18. From the Edit menu, choose Select All Records.
19. From the Edit menu, choose Copy.
20. Close the TEST-2 table and then open the TEST-1 table.
21. From the Edit menu, choose Paste Append. You will receive a message stating that two rows have been pasted into this table. Choose OK to complete the Paste Append.

Your most current information, without duplicates, is now in one table (TEST-1).

Additional reference words: 1.00 1.10 2.00 queries unique union SQL relational
KBCategory: kbusage
KBSubcategory: TblModify

INF: Description of Btrieve

Article ID: Q114505

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Btrieve is not a full, stand-alone database application like Microsoft Access or Microsoft FoxPro. This article describes how Btrieve works with Microsoft Access and with Novell NetWare and non-NetWare networks.

MORE INFORMATION

=====

Btrieve is not a full, stand-alone database program like Microsoft Access or Microsoft FoxPro. Btrieve is a dynamic-link library (DLL), similar in nature to a terminate-and-stay-resident (TSR) program in MS-DOS, that provides the ability to read from and write to database files on a hard disk. It is a record-managing engine only and does not have a user interface.

You can attach Btrieve databases in Microsoft Access, or import data from or export data to Btrieve databases. To use Btrieve without Microsoft Access, you must first write a program (commonly known as a front end) to access the Btrieve functions. These function calls specify the operation you want the Btrieve engine to perform, and also send any necessary data.

Since Btrieve handles all data as complete and separate records, it does not consider the actual contents of the records. To Btrieve, it is just writing information to the hard disk. The only exception to this behavior is when Btrieve deals with the portion of the record that has been identified as an index. In this case, Btrieve uses this portion of the record to create a binary tree index, which is sorted according to the method applicable for the type of data it contains.

Non-NetWare Usage of Btrieve

In MS-DOS, Btrieve is accessed as a TSR whose functions are called by interrupts made by the front end. The function performs the requested operation on the specified file and then returns control to the front end.

In Microsoft Windows, Btrieve is accessed as a DLL named WBTRCALL.DLL. This DLL contains the same functionality as mentioned above, except that the parameters are passed directly.

NetWare Usage of Btrieve

Btrieve is integrated into Novell NetWare networks and is used internally by NetWare to create a full client/server environment. This is accomplished by loading Btrieve on a server as a NetWare Loadable Module (NLM), and then

making function calls to the NLM over the network. Doing this requires a combination of a TSR and a DLL.

The TSR is BREQUEST.EXE on the client (workstation). BREQUEST.EXE is a network communication package that must be loaded before starting Windows, and is activated by WBTRCALL.DLL. This version of WBTRCALL.DLL differs from the one used in non-NetWare environments, even though they have the same name. This DLL's only function is to activate the TSR, which in turn sends the parameters across the network to the server. Where multiple servers are used, Novell provides routers to connect them together so that only one copy of Btrieve must be loaded on the main server.

Btrieve is manufactured by Novell Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00 Attaching
KBCategory: kb3rdparty
KBSubcategory: IsmBtrv

INF: Description of Btrieve Section of WIN.INI File

Article ID: Q94827

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Microsoft Access uses the Btrieve section of the WIN.INI file when it uses Btrieve files. After you install Microsoft Access, the WIN.INI file contains the following default settings in the [btrieve] section:

```
[btrieve]
options=/m:64 /p:4096 /b:16 /f:20 /l:40 /n:12 /t:C:\ACCESS\BTRIEVE.TRN
```

MORE INFORMATION

=====

Although these are the recommended option settings, you can change these options in your WIN.INI file using the following table as a guide. If the option you want to set is not on the line, add the option and desired setting at the end of the line.

Option	Description	Required Value
/m	Memory Size	At least 38
/p	Page Size	4096
/b	Pre-Image Buffer Size	16
/f	Open Files	At least 4
/l	Multiple Locks	At least double the setting for the Open Files option (/f)
/n	Files in a Transaction	At least 4
/t	Transaction Filename	Valid path to a .TRN file; in a multiuser environment, the network path to the .TRN file is shared by all users
/u	Compression Buffer Size	If your Btrieve data files are compressed, at least the length (in kilobytes) of the longest record in your data files
/i	Pre-image File Drive	
/c	Index Compaction	
/a	Activate Logging	
/s	Discard Unneeded Segments	

REFERENCES

=====

For more information on these options, see Novell's Btrieve for Windows "Installation and Operation" manual.

In Microsoft Access version 2.0, see the Readme Help file (ACREADME.HLP).

Additional reference words: 1.00 1.10 2.00 arguments
KBCategory: kb3rdparty
KBSubcategory: IsmBtrv

INF: Description of Btrieve/Xtrieve .DDF files

Article ID: Q113877

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Btrieve database files (which usually have .DAT extensions) consist of a header page, multiple index pages, and multiple data pages. With the exception of index fields located in the header page, none of these items include any information about the data types or the locations of the fields in the Btrieve database file. Also, while the header page contains information about the number and types of indexes, it does not contain any information about the non-index parts of the record. Since Microsoft Access and Microsoft Visual Basic require specific information about each field, they must use .DDF files, which contain the databases' data dictionary information, to obtain this information.

MORE INFORMATION

=====

There are seven available .DDF files: FILE.DDF, FIELD.DDF, INDEX.DDF, ATTRIB.DDF, VIEW.DDF, USER.DDF, and RIGHTS.DDF. Microsoft Access version 2.0 requires only FILE.DDF and FIELD.DDF. Microsoft Access can maintain the INDEX.DDF file for interoperability with Xtrieve, but the default is not to maintain it.

When you are using .DDF files with Microsoft Access, bear the following in mind:

- Each file must use one of the seven names listed above, without any variation. For example, substituting FILE1.DDF for FILE.DDF will not work.
- All the files must be in the same directory.

If either of these criteria are not met, Microsoft Access will not be able to use the files and errors may occur.

Each Btrieve database file (.DAT file) contains only one table. Therefore, a Btrieve application may have several .DAT files

File Definitions

FILE.DDF:

The FILE.DDF file contains the following elements:

- The file ID number.
- The table name.

- The path to the file (including filename).
- The file-type identifier.

A sample FILE.DDF might look like:

Xf\$ID	Xf\$Name	Xf\$Loc	Xf\$Flags
1	x\$File	C:\MYDIR\FILE.DDF	16
2	x\$Field	C:\MYDIR\FIELD.DDF	16
3	x\$Index	C:\MYDIR\INDEX.DDF	16
4	TestFile	TEST.DAT	0
5	Datafile	Q:\PUBLIC\DATA.DAT	0
6	MyData	MOREDATA\MYDATA.BTV	0

The sample FILE.DDF file above indicates that the FILE.DDF, FIELD.DDF, and INDEX.DDF files (file numbers 1, 2, and 3) are in the directory C:\MYDIR. The entry in the Xf\$Flags column (16) indicates that these files are system files that are used to locate other system files.

NOTE: Xtrieve uses the paths listed in the FILE.DDF file to search for these system files. Microsoft Access, however, only looks for these files in the same directory where the FILE.DDF file was found.

File numbers 4, 5, and 6 contain information on three Btrieve database files that can be used by Microsoft Access as attached tables. The entry in the Xf\$Flags column (0) indicates that these are user-defined files.

File number 4, TestFile, does not have a path specified. Therefore, Microsoft Access will search for this file in the directory in which it found the FILE.DDF file.

File number 5, DataFile, has a drive and path specified. Microsoft Access will search the specified drive and path for the file.

File number 6, MyData, has a path but no drive specified. Microsoft Access will search for this file in the MOREDATA subdirectory of the directory in which it found the FILE.DDF file.

The specifications for the FILE.DDF file are:

FieldName	Description	Data Type	Position	Length
Xf\$ID	File ID	Integer	1	2
Xf\$Name	File Name	String	2	20
Xf\$Loc	File Location	String	23	64
Xf\$Flags	File Flags*	Integer	87	1

*If bit 4 equals 1, the file is a dictionary (.DDF) file. If bit 4 equals 0, the file is a user-defined file.

The "Xf\$" prefixes are part of the Novell definition for these files.

Position is the offset from the beginning of the column, and Length is the length of the field.

FIELD.DDF:

The FIELD.DDF file contains:

- The ID number.
- The file ID number (this is cross-referenced to FILE.DDF).
- The field name (the column name in Microsoft Access).
- The data type.
- The offset.
- The size.
- The number of decimal places.
- A case-sensitive flag.

A sample FIELD.DDF file might look like:

Xe\$Id	Xe\$File	Xe\$Name	Xe\$DataType	Xe\$Offset	Xe\$Size	Xe\$Dec	Xe\$Flags
1	1	Xf\$Id	1	0	2	0	0
2	1	Xf\$Name	0	2	20	0	0
3	1	Xf\$Loc	0	22	64	0	0
4	1	Xf\$Flags	1	86	1	0	0
5	2	Xe\$Id	1	0	2	0	0
6	2	Xe\$File	1	2	2	0	0
7	2	Xe\$Name	0	4	20	0	0
8	2	Xe\$DataType	1	24	1	0	0
9	2	Xe\$Offset	1	25	2	0	0
10	2	Xe\$Size	1	27	2	0	0
11	2	Xe\$Dec	1	29	1	0	0
12	2	Xe\$Flags	1	30	2	0	0
13	3	Xi\$File	1	0	2	0	0
14	3	Xi\$Field	1	2	2	0	0
15	3	Xi\$Number	1	4	2	0	0
16	3	Xi\$Part	1	6	2	0	0
17	3	Xi\$Flags	1	8	2	0	0
18	4	nCounter	1	0	4	0	1
19	4	stUnique	0	4	15	0	1
20	4	stDup	0	19	15	0	1
21	4	rNumb	2	34	4	0	0
22	5	nCustNumb	1	0	4	0	0
23	5	stName	0	4	15	0	1
24	5	stAddress	0	19	20	0	1
25	5	stCity	0	39	20	0	1
26	6	nCustNumb	1	1	4	0	0
27	6	rBalanceDue	1	1	8	0	

This file contains actual field definitions. The first 17 rows contain the description of each field in FILE.DDF, FIELD.DDF (self-describing), and INDEX.DDF. The definitions are related to the FILE.DDF file by the Xe\$File field, which correlates to the Xf\$ID field in FILE.DDF. Thus, the first 4 records are for file number 1, which in FILE.DDF is FILE.DDF, the next 8

records are for file number 2, which in FILE.DDF is FIELD.DDF, and 5 records are for file number 3, INDEX.DDF.

Note that the records do not have to be in this order, or have to contain these file numbers, although they usually do. Some .DDF builders will put them in order, others do not. However, the records in FIELD.DDF must be in the same order as the fields within the file they describe. For example, the first record in FIELD.DDF for file number 1 must describe the first field in file number 1, and so on.

In this example, file number 4, TEST.DAT in FILE.DDF, has 4 fields:

1. nCounter: a Long (4-byte) Integer with an offset of 0.
2. stUnique: a 15-byte String with an offset of 4.
3. stDup: a 15-byte String with an offset of 19.
4. rNumb: a 4-byte Real with an offset of 34.

Note that file number 6 (MyData) contains an 8-byte integer named rBalanceDue. This is a Visual Basic Currency data type and is passed directly through to Microsoft Access unchanged.

If any of the fields were of the Btrieve Numeric type, the Xe\$Dec field would contain the number of decimal points.

The specifications for FIELD.DDF are:

FieldName	Description	DataType	Position	Length
Xe\$ID	Field Id	Integer	1	2
Xe\$File	File Id from FILE.DDF	Integer	3	2
Xe\$Name	FieldName	String	5	20
Xe\$DataType	Btrieve DataType	Integer	25	1
Xe\$Offset	Offset	Integer	26	2
Xe\$Size	Field size	Integer	28	2
Xe\$Dec	Decimal places	Integer	30	1
Xe\$Flags	Bit0=Case flag for strings	Integer	31	2

The "Xe\$" prefix is part of the Novell definition for these files.

Btrieve and Xtrieve are manufactured by Novell Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 1.00 1.10 2.00 Importing Exporting Attaching
KBCategory: kb3rdparty
KBSubcategory: IsmBtrv

INF: Description of the Visual Basic Compatibility Layer
Article ID: Q114834

The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
-

SUMMARY

=====

Microsoft Visual Basic version 3.0 has the ability to communicate with Microsoft Access databases. However, the modules supplied with Visual Basic version 3.0 can communicate only with Microsoft Access version 1.x-created databases.

For Visual Basic to communicate with Microsoft Access version 2.0-created databases, a group of files referred to as the Visual Basic Compatibility Layer (VBCL) is required. Some of these files are new, while others replace or modify existing files.

You can obtain the VBCL from 5 sources:

- The Microsoft Access Developer's Toolkit (ADT) version 2.0
- The Microsoft Office Developer's Kit (ODK) version 1.0
- Download the VBCL by downloading COMLYR.EXE, a self-extracting file, from the Microsoft Software Library (MSL) on the following services:
 - CompuServe
GO MSL
Search for COMLYR.EXE
Display results and download
 - Microsoft Download Service (MSDL)
Dial (206) 936-6735 to connect to MSDL
Download COMLYR.EXE
 - Internet (anonymous FTP)
ftp ftp.microsoft.com
Change to the softlib/mslfiles directory
Get COMLYR.EXE

MORE INFORMATION

=====

The Microsoft Access Developer's Toolkit version 2.0 supplies the VBCL and the Microsoft Jet database engine version 2.0, along with numerous other database development tools. At this writing, the ADT retails for \$495.

The Microsoft Office Developer's Kit version 1.0 contains the VBCL, along with other tools to aid in the development of Microsoft Office applications, but does not include version 2.0 of the Microsoft Jet database engine. Version 1.1 of the Microsoft Jet database engine is included. At this writing, the ODK retails for \$19.95.

To obtain pre-sales information about new or updated Microsoft products, call the Microsoft Sales Information Center at (800) 426-9400. If you are outside the United States, contact the Microsoft subsidiary for your area. To locate your subsidiary, call the Microsoft International Sales Information Center at (206) 936-8661.

VBCL Files

NOTE: The file locations listed below assume default installation directories.

Files Replaced or Modified by the VBCL:

File	Description	Location
VBDB300.DLL	VB to Jet DB engine 1.1 Interface DLL	C:\WINDOWS\SYSTEM
PDCTJET.DLL	Run-time DLL for Crystal Reports	C:\WINDOWS\SYSTEM
PDIRJET.DLL	Run-time DLL for Crystal Reports	C:\WINDOWS\SYSTEM
PDBJET.DLL	Run-time DLL for Crystal Reports	C:\WINDOWS\SYSTEM
CRW.EXE	Crystal Report Designer application	C:\VB\REPORT
VB.INI	Visual Basic initialization file	C:\WINDOWS
SETUPWIZ.INI	Setup Wizard initialization file	C:\WINDOWS
KBINTEROP		

Files Needed to Upgrade Visual Basic Version 3.0:

File	Description	Location
MSAJT200.DLL	Version 2.0 Jet database engine	C:\WINDOWS\SYSTEM
MSAJT112.DLL	Jet DB engine 2.0/VBCL	C:\WINDOWS\SYSTEM
VBDB300.DLL	VB to Jet DB engine Support DLL	C:\WINDOWS\SYSTEM
BTRV200.DLL	Btrieve ISAM driver library	C:\WINDOWS\SYSTEM
PDX200.DLL	Paradox ISAM driver library	C:\WINDOWS\SYSTEM
XBS200.DLL	Microsoft FoxPro ISAM driver library	C:\WINDOWS\SYSTEM
CRWACC20.EXE	Program that modifies Crystal Reports files to work with Microsoft Access version 2.0 databases	C:\VB
ACC2COMP.TXT	Readme file for VBCL	C:\VB

Additional reference words: 2.00 interoperability mapping

KBCategory: kbinterop

KBSubcategory: IntpOthr

INF: Description of Vendor Independent Mail (VIM)

Article ID: Q112133

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

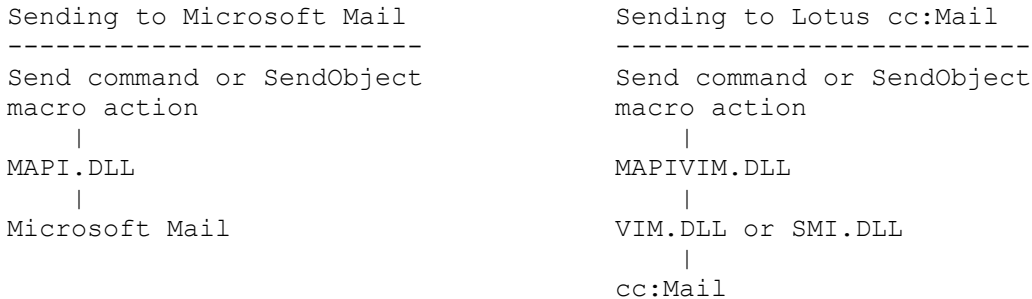
=====

This article describes how the Microsoft Access Setup program installs support for electronic mail (email) applications, and how you can modify your Windows settings to support one of two different electronic mail interface protocols: messaging application programming interface (MAPI) or vendor independent messaging (VIM), which is used by the Lotus cc:Mail electronic mail application.

MORE INFORMATION

=====

All of the Microsoft Office for Windows applications (Microsoft Access, Microsoft Word, Microsoft Excel, Microsoft Mail, and Microsoft PowerPoint) support the MAPI protocol by default. To support VIM electronic mail systems, the applications use a MAPI-to-VIM translation dynamic-link library (DLL) called MAPIVIM.DLL. The following diagrams demonstrate the route Microsoft Access uses when you send electronic mail using either the Send command from the File menu or the SendObject macro action:



How Microsoft Access Installs Electronic Mail Support

1. The Microsoft Access Setup program searches the Windows WIN.INI file for the following setting, which indicates that a VIM-compliant email application (such as cc:Mail) is installed on your computer:

```
[Mail]
MAPIDLL=<WINDOWS>\SYSTEM\MAPIVIM.DLL
```

where <WINDOWS> is the location of your Windows program directory (for example, C:\WINDOWS). If Setup finds this setting, it copies the MAPIVIM.DLL file to your <WINDOWS>\SYSTEM directory and updates the MAPIDLL setting in your WIN.INI file to reflect the location of the new MAPIVIM.DLL file.

If the MAPIDLL setting points to a different .DLL file, Microsoft Office applications use the MAPI functions in the specified .DLL instead of MAPI.DLL.

2. If Setup does not find the above MAPIDLL setting, it searches the WIN.INI file for the following setting, which indicates that a MAPI-compliant email application (such as Microsoft Mail) is installed on your computer:

```
[Mail]
MAPI=1
```

NOTE: If Setup installs MAPI support, it deletes any existing MAPIDLL entry from your WIN.INI file.

3. If Setup detects both MAPI and VIM on your system, it prompts you to select which messaging system to support.
4. If Setup finds either of the above settings, it adds the email commands to the Microsoft Access menus (for example, Setup adds the Send command to the File menu).

NOTE: the cc:Mail program directory must be part of your PATH statement. If it is not, the email commands will not be added to your Microsoft Access menus.

How to Switch from One Mail System To Another

When you access email from Microsoft Access, Microsoft Access searches first for the MAPIDLL setting. If found, Microsoft Access supports VIM-compliant email systems. If not found, Microsoft Access then searches for the MAPI=1 setting and, if found, supports MAPI-compliant email systems.

If you work on a network that uses both MAPI- and VIM-compliant email applications, you may want Microsoft Access to be able to switch back and forth between the two applications. To do this, add both of the above settings to the [Mail] section of your WIN.INI file. When you want to use a MAPI-compliant email application, comment the MAPIDLL line to deactivate it (to comment a line in the WIN.INI file, type a semicolon [;] at the beginning of the line).

The following sample WIN.INI entries demonstrate this procedure:

To use a MAPI-compliant email program (such as Microsoft Mail), comment the MAPIDLL line:

```
[Mail]
;MAPIDLL=<WINDOWS>\SYSTEM\MAPIVIM.DLL
MAPI=1
```

To use a VIM-compliant email program (such as cc:Mail), remove the semicolon (;) from the MAPIDLL line to reactivate it:

```
[Mail]
MAPIDLL=<WINDOWS>\SYSTEM\MAPIVIM.DLL
MAPI=1
```

If you want to use VIM-compliant email systems but Microsoft Access did not install the MAPIVIM.DLL file because there was no MAPIDLL line in the WIN.INI file, you can install the MAPIVIM.DLL file manually. To do this, copy the DECOMP.EXE file from disk 1 of your original Microsoft Access installation disks to your Windows directory. Insert disk 5 (for 3.5-inch disks) or disk 4 (for 5.25-inch disks) in drive A, and then type the following command at a DOS prompt:

```
c:\windows\decomp.exe a:\mapivim.dl_ c:\windows\system\mapivim.dll
```

This command will install the MAPIVIM.DLL file in your WINDOWS\SYSTEM directory.

REFERENCES

=====

For more information about using cc:Mail with Microsoft Access, query on the following words here in the Microsoft Knowledge Base:

cc:mail and mapivim.dll and lotus

cc:Mail is manufactured by Lotus Development Corporation, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 2.00 mail msacc20.ini cc

KBCategory: kbinterop

KBSubcategory: IntpOthr

INF: Determining How Many Instances of Application Are Active
Article ID: Q96591

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

You can use Windows API calls in Access Basic to determine how many instances of an application are running. You can use this to prevent re-entrance of an application that is already running, to determine how many instances of an application are active, or to start a loop in which you remain until the application quits.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

This article outlines three uses of two Windows APIs in Access Basic.

Open a new module or a previously created module and enter the following code:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----  
'Global Declarations Section Of The Module.  
'-----  
Option Explicit  
  
Declare Function GetModuleHandle% Lib "Kernel" (ByVal lpModuleName$)  
Declare Function GetModuleUsage% Lib "Kernel" (ByVal hModule%)
```

1. Prevent re-entrance:

```
'-----  
' The following function will test for re-entrance and return  
' True if there is already an instance of the application active,  
' or it returns False. This function requires an argument that is  
' the name of the application.  
'-----  
Function TestForReentrance% (ApplicationName$)  
    If ApplicationName$ = "" Then Exit Function  
    If GetModuleUsage(GetModuleHandle(ApplicationName$)) Then _  
        TestForReentrance = True
```

End Function

Example of Using this Code

You can use this function to keep from starting an application that is already running

```
Function RunMyApp$ (szAppToRun$)
    Dim x%
    If Not TestForReentrance(szAppToRun) Then
        x = Shell(szAppToRun)
        Exit Function
    End If
    RunMyApp = "AppToRun is presently active."
End Function
```

and a call to this in the Immediate window such as:

```
?RunMyApp$("Clock.exe")
```

2. Count the number of instances:

```
'-----  
' The following function will return the number of instances of an  
' application that are currently active.  
'-----
```

```
Function CountInstance% (szAppName$)
    CountInstance = GetModuleUsage(GetModuleHandle(szAppName))
End Function
```

3. Remain idle until an application quits:

```
'-----  
' The following function will attempt to start an MS-DOS application  
' and remain in a loop until that application quits.  
'-----
```

```
Sub RunDosAppUntil (szAppToRun$)
    Dim hMod%
    hMod% = Shell(szAppToRun, 1)
    If (hMod% > 32) Then
        While (GetModuleUsage(hMod%))
            DoEvents
        Wend
    Else
        MsgBox "Unable to start the Application"
    End If
End Sub
```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

INF: Determining the Version of Catalog Stored Procedures
Article ID: Q95933

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to find the version of catalog stored procedures on a Microsoft SQL Server.

MORE INFORMATION

=====

There are situations in which you want to determine the version of a stored procedure, such as when you are experiencing difficulties attaching to SQL Server. The first method does not use Microsoft Access.

Method 1

Using the System Administration Facility (SAF), or a SQL pass-through query in Microsoft Access 2.0, you need to log onto SQL Server with the user's login ID and password. (Using the SQL administrator's login id (SA) is ideal but is not always readily available.) Once logged on, the user needs to open the query window and issue the command USE MASTER. This tells SQL Server to use the Master database that contains the system tables. To determine the version, enter the following SQL command:

```
USE MASTER
GO
SELECT Attribute_Value FROM MsServer_Info WHERE Attribute_ID = 500
```

NOTE: If you are using SQL Server for Windows NT 4.2 substitute spt_server_info for MsServer_Info.

Method 2

In a situation where you can successfully attach SQL Server data, you can attach the MsServer_Info table from the Master database while using Microsoft Access, then run a query on it for the record where the Attribute_ID is equal to 500.

NOTE: If you are using SQL Server for Windows NT 4.2 substitute spt_server_info for MsServer_Info.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr

INF: Differences Between Native and Attached Tables

Article ID: Q92644

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

Attached tables (such as Microsoft Access, Paradox, Btrieve, FoxPro, or dBase tables) are different from native Microsoft Access tables in four ways:

1. In the Database window, the icons for attached tables contain black arrows.
2. If you try to open an attached table in Design view, you receive the following 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. Only the table entry is deleted. However, the message you receive is the same as for a native table whose contents are actually deleted.
4. You can define relationships between attached tables and native Microsoft Access tables, or between attached non-Microsoft Access tables and native Microsoft Access tables, but you cannot enforce such relationships.

You can create relationships between an attached Microsoft Access table and a native Microsoft Access table.

MORE INFORMATION

=====

Paradox and dBASE IV are manufactured by Borland International Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

REFERENCES

=====

For more information about creating relationships between tables, search for "relationships between tables" using the Microsoft Access Help menu.

For more information about attaching external tables, search for "attaching tables" or "attaching external tables" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr

INF: Directory Contents of Service Pack File ACCSVC.EXE
Article ID: Q123708

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

The Microsoft Access version 2.0 Service Pack is contained in a self-extracting file called ACCSVC.EXE. To extract its contents, run the file.

MORE INFORMATION

=====

When you run the ACCSVC.EXE file, it creates a subdirectory called ACCSVC off the root directory of your hard disk, and then creates two subdirectories called DISK1 and DISK2 below the ACCSVC subdirectory. Following is a directory listing of the files in these subdirectories.

Directory of the ACCSVC Subdirectory

DISK1	<DIR>		12-02-94	4:52p
DISK2	<DIR>		12-02-94	4:52p
LICENSE	TXT	2,583	11-21-94	11:36a

Directory of the ACCSVC\DISK1 Subdirectory

ACMSETUP	EX_	106,497	11-20-94	9:02p
ACMSETUP	HL_	12,618	11-20-94	9:02p
ADMIN	IN_	962	11-20-94	9:05p
BTRV200	DL_	57,148	11-20-94	9:02p
COMPLINC	DLL	21,408	11-21-94	11:49a
COMPOBJ	DL_	50,855	11-20-94	9:03p
CTL3D	DL_	10,085	11-20-94	9:02p
MSACAHBB	DL_	61,904	11-20-94	9:02p
MSAJT200	DL_	504,413	11-20-94	9:03p
MSCPYDIS	DL_	8,805	11-20-94	9:02p
MSJETERR	DL_	6,270	11-20-94	9:03p
MSJETINT	DL_	10,846	11-20-94	9:03p
MSSETUP	DL_	91,812	11-20-94	9:02p
OLE2	RE_	2,901	11-20-94	9:03p
OLE2CONV	DL_	29,914	11-20-94	9:03p
PDX200	DL_	121,341	11-20-94	9:02p
READSRV	TXT	16,551	11-20-94	9:02p
SETUP	EXE	39,856	11-20-94	9:02p
SETUP	INI	149	11-20-94	9:02p
SETUP	IN_	1,020	11-20-94	9:03p
SETUP	LST	620	11-21-94	11:50a
SETUP	STF	38,282	11-21-94	11:50a
STDOLE	TL_	1,849	11-20-94	9:03p
XBS200	DL_	151,014	11-20-94	9:02p

MSSETUP EX 9,147 11-20-94 9:02p

Directory of the ACCSVC\DISK2 subdirectory

DECOMP	EXE	39,044	11-20-94	9:03p
JETVER	TX_	1,622	11-20-94	9:03p
MSAJU200	DL_	48,484	11-20-94	9:03p
OLE2	DL_	136,986	11-20-94	9:03p
OLE2DISP	DL_	68,223	11-20-94	9:03p
OLE2NLS	DL_	42,410	11-20-94	9:03p
OLE2PROX	DL_	17,894	11-20-94	9:03p
SETUPWIZ	MD_	183,653	11-20-94	9:03p
STORAGE	DL_	68,920	11-20-94	9:03p
TYPELIB	DL_	90,645	11-20-94	9:03p

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kbusage

KBSubcategory: DcmOthr

INF: Displaying Start, Finish, and Elapsed Times on a Form
Article ID: Q97523

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article demonstrates how to create command buttons to enter start and finish times on a form, and how to show elapsed time.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic," in version 2.0.

MORE INFORMATION

=====

The following seven steps use macros and Access Basic code to display start and finish times on a form:

1. Create a new module called Module 1. Create the appropriate declaration section and add the following function.

NOTE: In the following sample code, an underscore (_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----  
' GLOBAL DECLARATION  
'-----  
Option Compare Database  
Option Explicit  
  
'=====
```

' The following function calculates elapsed time.
' This function displays the total time elapsed in the
' following format:
' <hours>:<minutes>:<seconds>
' For example:
' 48:57:04
'=====

```
Function TimePassed (Startup, Done)  
    TimePassed = Format(Int(Abs(Startup - Done) * 24), "###:") + _  
                Format(Abs(Startup - Done), "nn:ss")  
End Function
```

2. Create a macro to set the elapsed time, as follows.

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

```

-----
Set Elapsed Time          Set Value

Set Elapsed Time Actions
-----
Set Value
  Item: [Elapsed Time]
  Expression: TimePassed([Start Time],[Finish Time])

```

3. Create a macro to set the start and finish time, as follows.

Macro Name	Condition	Action
SetTime	IsNull(Forms![Time Form]![Start Time])	Set Value
	Not IsNull(Forms![Time Form]![Start Time])	Set Value

```

SetTime Actions
-----
Set Value
  Item: [Start Time]
  Expression: Now()
Set Value
  Item: [Finish Time]
  Expression: Now()

```

4. Create a new blank, unbound form (do not select a table or query).
5. Add three text box controls to the form and assign them the following properties:

```

Object: Text Box
-----
ControlName: Start Time
  Enabled: No
  Locked: Yes

Object: Text Box
-----
ControlName: Finish Time
  Enabled: No
  Locked: Yes

Object: Text Box
-----
ControlName: Elapsed Time
  Enabled: No
  Locked: Yes

```

6. Add three command button controls to the form, and assign them the following properties:

```

Object: Command Button
-----
ControlName: Begin
  Caption: Start
  OnPush: SetTime

```

Object: Command Button

ControlName: Complete
Caption: Finish
OnPush: SetTime

Object: Command Button

ControlName: Time Elapsed
Caption: Time Elapsed
OnPush: Set Elapsed Time

Save the form as Time Form.

7. Choose Form view to test the example as follows:
 - a. Choose the Start button. The current date and time display in both the start time and finish time boxes.
 - b. Choose the Start button again. While the start time should not change, the finish time will update to the current data and time.
 - c. Choose the Finish button to set or reset the finish time.
 - d. Choose the Time Elapsed button to display the elapsed time in days, hours, and minutes.

For additional information on calculating times, query on the following words here in the Microsoft Knowledge Base:

calculating and elapsed and time

REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.0, Chapter 3, "Access Basic Essentials"

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ExrOthr

INF: DLookup() Usage, Examples, and Troubleshooting Tips
Article ID: Q108098

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to use the DLookup() function, and includes examples and tips on usage.

The following topics are addressed in this article:

- The DLookup() function syntax and usage
- DLookup() function examples:
 - A function with no criteria
 - Specifying numeric criteria
 - Specifying numeric criteria that comes from a field on a form
 - Specifying textual criteria
 - Specifying textual criteria that comes from a field on a form
 - Specifying date criteria
 - Specifying multiple fields in the criteria
- Tips for troubleshooting and debugging DLookup() expressions

Note that many difficulties in using the DLookup() function result from failing to set up and supply the criteria expression correctly. The criteria expression is common to many other aggregate functions, such as DSum(), DFirst(), DLast(), DMin(), DMax(), and DCount(). Therefore, the following material can be useful in helping to understand those domain aggregate functions as well as the DLookup() function.

MORE INFORMATION

=====

The DLookup() Function Syntax and Usage

The DLookup() function can be used in an expression or in an Access Basic function to return a field value in a domain, or specified set of records.

The syntax of the DLookup() function is:

DLookup(Expression, Domain [, Criteria])

The function has three arguments: the expression, the domain, and the criteria. (Note that the criteria argument is optional.)

The expression argument is used to identify the field that contains the data in the domain that you want returned, or it can perform calculations using the data in that field.

The domain argument is the name of the record set that identifies the domain. It can be a table or a query name.

The criteria argument is an optional string expression that is used to restrict the range of the data that the DLookup() function is performed on. Note that the criteria argument is identical to the WHERE clause in an SQL expression (except that you do not use the keyword WHERE).

The DLookup() function returns one value from a single field even if more than one record satisfies the criteria. If no record satisfies the criteria, or if the domain contains no records, DLookup() will return null.

DLookup() Function Examples

The following examples demonstrate how to use DLookup() to find or return values from a table or query. These examples refer to the sample database NWIND.MDB and can be entered in the ControlSource property of a text box on a form or report.

NOTE: In the following sample code, an underscore (_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

A Function with No Criteria:

This example demonstrates how to use DLookup() in its simplest form, without any criteria specified. This example, which refers to the Orders form, returns the value contained in the Last Name field of the first record in the Employees table:

```
=DLookup("[Last Name]", "Employees")
```

NOTE: Field names in expressions need to be enclosed in brackets ([]). The table or query name in the domain does not need to be enclosed in brackets.

Specifying Numeric Criteria:

To find the last name of the employee with ID number 7, specify a criteria to limit the range of records used:

```
=DLookup("[Last Name]", "Employees", "[Employee ID] = 7")
```

Specifying Numeric Criteria That Comes from a Field on a Form:

If you do not want to specify a particular value in the expression (as in the example above), use a parameter in the criteria instead. The following examples demonstrate how to specify criteria from another field on the current form:

```
=DLookup("[Last Name]", "Employees", _  
"[Employee ID] = Form![Employee ID]")
```

```
=DLookup("[Last Name]", "Employees", _  
"[Employee ID] = " & [Employee ID])
```

```
=DLookup("[Last Name]", "Employees", _  
        "[Employee ID] = " & Forms!Orders![Employee ID])
```

The three examples above will all have the same results.

In the first example, Form![Employee.ID] appears inside the criteria's quotation marks. "Form" tells Microsoft Access that the field reference "Employee ID" comes from the current form. If you omit it, Microsoft Access will compare Employee ID to itself in the Employees table, and will return the last name from the first record in the Employees table (the same result as if you did not specify any criteria). This is because the first record in the Employees table has a 1 in the Employee ID field, so the argument

```
"[Employee ID] = [Employee ID]"
```

computes to

```
"1 = 1"
```

and would thus be the first record that matched the criteria.

The criteria for the second two examples are made by concatenating two string expressions with an ampersand (&). In the third example, the criteria ends with a form field reference.

When criteria are being evaluated, the individual pieces of the criteria are first evaluated and then appended or concatenated, and then the whole value is computed.

If the current value in the Employee ID field on the Orders form is 7, the original criteria expression

```
"[Employee ID] = " & [Employee ID]
```

would compute to

```
"[Employee ID] = " & 7
```

which, when concatenated, computes to:

```
"[Employee ID] = 7"
```

If you do not specify the full form reference (which the second example above does not), Microsoft Access will assume that you are referring to the current form.

The following example is a derivative of the third example above:

```
=DLookup("[Last Name]", "Employees", _  
        "[Employee ID] = Forms!Orders![Employee ID]")
```

In this example, the full form reference is enclosed inside the criteria's quotation marks. In this case, Microsoft Access will correctly look up the value when the form first opens, but only until the Employee ID field is changed by modifications to a record or the addition of a record. Microsoft Access will not automatically recompute the new employee last name value after such a change. You can manually recompute the expression by placing

the insertion point on the control, and then pressing F9.

If you want the field to automatically update when the criteria changes, make the criteria a variable by using the method of concatenating the expression's parts as described earlier. Note that when you move to a new record, the DLookup() text boxes that update automatically will have "#Error" in them until you enter something in the Employee ID text box.

Specifying Textual Criteria:

All the examples above demonstrate how to use the DLookup() function with numeric criteria. If the criteria fields are textual, enclose the text in single quotation marks, as in the following example:

```
=DLookup("[Last Name]", "Employees", "[Employee ID] = '7'")
```

Note that you can use double quotation marks instead of single quotation marks, although single quotation marks are preferred. Use two double quotation marks to replace one single quotation mark. The following example uses double quotation marks, and is equivalent to the example above:

```
=DLookup("[Last Name]", "Employees", "[Employee ID] = ""7""")
```

Specifying Textual Criteria That Comes from a Field on a Form:

The following example demonstrates how to find the description for a category on the Products form. The Category ID field is a textual key field for the criteria, so the DLookup() statement is

```
=DLookup("[Description]", "Categories", _  
"[Category ID] = '" & [Category ID] & "'")
```

or, equivalently:

```
=DLookup("[Description]", "Categories", _  
"[Category ID] = '" & Forms!Products![Category ID] & "'")
```

In the second example, the criteria is made up by concatenating three string expression pieces. The first is "[Category ID] =", the second is the value contained in the actual Category ID field, and the third is the closing single quotation mark enclosed in double quotation marks.

When this criteria is being evaluated, the individual pieces are first evaluated, their results appended or concatenated, and then the whole value is computed.

If the current value selected in the Category combo box on the Products form is Beverages, the bound column for the combo box will return BEVR as the Category ID. The original criteria expression

```
"[Category ID] = '" & [Category ID] & "'"
```

will compute to

```
"[Category ID] = '" & "BEVR" & "'"
```

which, when concatenated, computes to:

```
"[Category ID] = 'BEVR'"
```

Specifying Date Criteria:

If the criteria fields are date or time values, enclose the date or time value in number signs (#). To find an employee whose birthday is on a given date, use the following sample criteria:

```
=DLookup("[Last Name]", "Employees", "[Birth Date] = #11-15-67#")
```

Specifying Multiple Fields in the Criteria:

The criteria expression can be any valid SQL WHERE clause (without the keyword WHERE). This implies that more than one field can be used to specify criteria for a DLookup() function.

To find the Product ID for one of the products supplied by supplier "Formaggi Fortini", with a Supplier ID of 14 (numeric), and a Category ID of DAIR (textual), use the following sample DLookup() statement:

```
=DLookup("[Product ID]", "Products", _  
"[Category ID] = 'DAIR' And [Supplier ID] = 14")
```

This statement will return Product ID 31, which is the first Product ID that matches the criteria. Product ID 23 also matches the criteria.

The example above uses hard-coded, or specific, Category ID and Supplier ID values. To use variables instead of specifics for the criteria, you could use Access Basic to concatenate multiple string expressions. The following Access Basic example demonstrates this method:

```
' Declare the variables  
Dim CatID As String  
Dim SuppID As Number  
Dim Result  
  
' Assign values to the variables to be used in the criteria  
CatID = "DAIR"  
SuppID = 14  
  
Result = DLookup("[Product ID]", "Products", _  
"[Category ID] = '" & CatID & "' And [Supplier ID] = " & SuppID)
```

If the DLookup() function is unsuccessful, the variable Result will contain null.

Note that the criteria is made up of four pieces that are evaluated individually. The results are appended, and then evaluated as a whole. The original criteria expression

```
"[Category ID] = '" & CatID & "' And [Supplier ID] = " & SuppID
```

computes to

```
"[Category ID] = '" & "DAIR" & "' And [Supplier ID] = " & 14
```

which, when concatenated, computes to:

```
"[Category ID] = 'DAIR' And [Supplier ID] = 14"
```

The next example can be used to find an employee whose birthday falls on today's date:

```
=DLookup("[Last Name]", "Employees", _  
  "Month([Birth Date]) = " & Month(Date) & " And Day([Birth Date]) _  
  = " & Day(Date))
```

If today's date is 12/2/93, the original criteria expression

```
"Month([Birth Date]) = " & Month(Date) & " And Day([Birth Date]) = " _  
& Day(Date)
```

computes to

```
"Month([Birth Date]) = " & 12 & " And Day([Birth Date]) = " & 2
```

which, when concatenated, computes to:

```
"Month([Birth Date]) = 12 And Day([Birth Date]) = 2"
```

Tips for Troubleshooting and Debugging DLookup() Expressions

To troubleshoot expressions, break down the expression into smaller components and test the components individually in the Immediate window to ensure they work correctly. If the smaller components work correctly, they can be put back together, piece by piece, until the final expression works correctly.

The Immediate window is a tool that you can use to help debug Access Basic modules. The window can be used to test and evaluate expressions independently of the form or macro the expression is to be used in. You can set up expressions in the Immediate window, run them, and see the results immediately.

The following example demonstrates a strategy you can use to break down a DLookup() expression into smaller components that you can test in the Immediate window. Assume you are having difficulty with the following statement:

```
=DLookup("[Product ID]", "Products", _  
  "[Category ID] = '" & Forms!MyForm![Category ID] _  
  & "' And [Supplier ID] = " & Forms!MyForm![Supplier ID])
```

Note that the criteria includes multiple fields in the lookup criteria, one of which is numeric and one if which is textual.

To troubleshoot this expression, try the following:

1. Open or create a module. From the View menu, choose Immediate Window.
2. Try the function without any criteria. Type the following in the Immediate window and then press ENTER:

```
? DLookup("[Product ID]", "Products")
```

Microsoft Access will perform the calculation and display the result on the next line in the Immediate window.

3. Make sure the form references correctly reference data. Type each of the following in the Immediate window, and then press ENTER:

```
? Forms!MyForm![Category ID]
```

-and-

```
? Forms!MyForm![Supplier ID]
```

4. Try to get the criteria fields to work independently by hard coding values into the expression. Type each of the following in the Immediate window, and then press ENTER:

```
? DLookup("[Product ID]", "Products", "[Category ID] = 'DAIR'")
```

-and-

```
? DLookup("[Product ID]", "Products", "[Supplier ID] = 14)
```

5. Try to get the criteria fields to work independently with a parameter in the query. Type each of the following in the Immediate window, and then press ENTER:

```
? DLookup("[Product ID]", "Products", _  
  "[Category ID] = '" & Forms!MyForm![Category ID] & "'")
```

-and-

```
? DLookup("[Product ID]", "Products", _  
  "[Supplier ID] = '" & Forms!MyForm![Supplier ID])
```

REFERENCES

=====

Microsoft Access "User's Guide," version 1.1, page 651

Microsoft Access "Language Reference," version 1.1, pages 144-146

Microsoft Access "Introduction to Programming," version 1.1, pages 37-42

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ExrOthr

INF: DoCmd Statement in Transaction Not Affected by Rollback
Article ID: Q89587

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

In Microsoft Access, the DoCmd statement is not affected by a Rollback statement if the DoCmd statement appears in a transaction.

MORE INFORMATION

=====

Transaction processing in Microsoft Access is valid only on virtual table (VT) objects, such as dynasets. The DoCmd statement starts another JET session to process its arguments.

Since JET manages transaction processing, a new session of JET contains its own transaction management. The two are mutually exclusive; transactions in one are not managed or affected by transactions in the other.

The simplest workaround for this problem is the QueryDef.Execute method, which uses a dynaset.

NOTE: The QueryDef.Execute method works only when a QueryDef action is based on an action query (update, insert, delete) that does not return a recordset.

Additional reference words: 1.00 1.10 2.00 commit record set

KBCategory: kbusage

KBSubcategory: McrActn

INF: Dynamically Synchronizing Two Forms Using Access Basic
Article ID: Q119398

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

This article describes how to synchronize a form to the current record in a subform using Access Basic. The method described in this article searches a Recordset returned by the form's RecordsetClone property and synchronizes the form's bookmarks with the Recordset.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The following example demonstrates how to synchronize the Products form to the current record in the Categories Subform form in the sample database NWIND.MDB.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the NWIND database.
2. Open the Categories Subform form in Design view.
3. Set the form's OnCurrent property to the following event procedure:

```
'*****  
'Sub Form_Current()  
'*****  
  
    'If the Product ID is blank, then exit the Sub.  
    If IsNull(Me![Product Id]) Then  
        Exit Sub  
    End If  
  
    'Dimension variables.  
    Dim formname As String, SyncCriteria As String  
    Dim f As Form, rs As Recordset  
  
    'Set the formname to "Products," the form that will be  
    'synchronized.  
    formname = "Products"  
  
    'Check to see if the Products form is open. If it
```

```

'is not open, open it.
If Not SysCmd(SYSCMD_GETOBJECTSTATE, A_FORM, formname) Then
    DoCmd OpenForm formname
End If

'Define the form object and Recordset object for
'the Products form.
Set f = Forms(formname)
Set rs = f.RecordsetClone

'Define the criteria used for the synchronization.
SyncCriteria = "[Product Id]=" & Me![Product Id]

'Find the corresponding record in the Products form to
'the current record in the subform.
rs.FindFirst SyncCriteria

'If a record exists in the Products form, find the
'matching record.
If rs.nomatch Then
    MsgBox "No match exists!", 64, formname
Else
    f.bookmark = rs.bookmark
End If

'*****
'End Sub
'*****

```

4. Save and then close the Categories Subform form.
5. Open the Categories form in Form view. When you open the Categories form, the subform's OnCurrent event procedure will be triggered, causing the Products form to be opened if it is not already open.

You can adapt this method to occur when a command button is clicked by moving the code specified in the subform's OnCurrent property event procedure to a command button on the Categories form. If you do move the code to a command button, make sure to change the references to "Me" in the code to full form reference, using the syntax:

```
Forms!Categories![Categories Subform].Form
```

REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, "Using the OnCurrent Property to Keep Two Open Forms Synchronized," pages 639-641

Microsoft Access "Building Applications," version 2.0, Chapter 4, "Using Forms to Collect, Display, and Filter Information," page 84, and Chapter 11, "Working with Sets of Records," pages 255-259

For more information about the SysCmd() function, search for "SysCmd" then "SysCmd Function" using the Microsoft Access Help menu.

For more information about the Me property, search for "Me" then "Me

Property" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsHowto

INF: Eliminating White Space in Reports with CanShrink and AB
Article ID: Q95390

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Before printing a form or report, you can save space by conditionally reducing the sizes of the controls on the form or report. For example, you might want to reduce the size of text boxes that contain varied-length text or reduce the size of text boxes bound to fields that could be empty.

Note the following information before you try to reduce the size of controls in reports:

- Overlapping controls will not shrink, even when the CanShrink property is set to Yes.
- Controls shrink line by line (vertically). This means, for example, that if there are a group of controls placed on the left side of the page and a large control (for example, an OLE picture) on the right side of the page, the controls on the left side of the page will not shrink, unless the picture is blank and hidden.
- Space between controls is not affected by the CanShrink or CanGrow properties.
- Controls located in the Page Header or Page Footer will not shrink or grow.

MORE INFORMATION

=====

A control containing null information automatically shrinks to nothing and disappears when its CanShrink property is set to Yes, but the space between the controls is not affected by the CanShrink property.

For example, say you have 11 controls in a column of a report, and there is .1 inch of space between each control. Even if all 11 controls contain null information and all 11 have their CanShrink property set to Yes, the space between the controls adds up to 1 inch. When you print the report, even though the 11 controls do not print, the first control that does contain text will print 1 inch lower than you might expect because of all the accumulated space.

A good way to manage the size of your controls is to have as few as possible. This will minimize the white space between the controls.

The following example demonstrates how to print mailing labels by using an Access Basic function to minimize the number of controls and give you nice, even spacing. Say you want to create mailing labels containing a name, business name, address, city, state, and zip code. You could use

several different fields, but it would be better to use a single text box, as in the following step-by-step example:

1. Create a single text box control with the CanGrow property set to Yes, and the Height property set to the height of one line of data then set its ControlSource property to this expression:

```
=AddressBlock([Name],[Business Name],[Address],[City],[State],[ZipCode])
```

2. Create a new Access Basic module, and type the following code into it:

```
Option Explicit
Function AddressBlock$ (AName, Addr1, Addr2, City, State, Zip)
    Dim A1$, A2$, A3$, A4$, CR$

    CR$ = Chr(13) & Chr(10) 'Carriage return and line feed.

    A1$ = IIf(ISB(AName), "", AName & CR$)
    A2$ = IIf(ISB(Addr1), "", Addr1 & CR$)
    A3$ = IIf(ISB(Addr2), "", Addr2 & CR$)
    A4$ = City & ", " & State & " " & Zip

    AddressBlock = A1$ & A2$ & A3$ & A4$ 'Concatenate the strings.
End Function

Function ISB (V) As Integer
    If IsNull(V) or V = "" Then ISB = True Else ISB = False
End Function
```

Now you can print your report and the report will automatically run the function and keep wasted space to a minimum.

Additional reference words: 1.00 1.10 2.00 shrinking functions

KBCategory: kbusage

KBSubcategory: RptLayout

INF: Enforcing Transactions on Attached SQL Server Tables
Article ID: Q95607

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Microsoft Access will automatically support transaction processing on attached SQL Server tables using queries. However, explicit transactions in Access Basic require the use of dynasets on attached tables and further require a specific order in the creation of the dynasets and the implementation of transaction processing.

MORE INFORMATION

=====

Microsoft Access will enforce transaction processing on an attached SQL Server table through the use of a dynaset created on that table. To get more information on how to create a dynaset on an attached SQL Server table, search for "CreateDynaset" using the Help menu. The key to making transaction processing work for attached SQL Server tables is to create and close the dynaset on the attached SQL Server table outside the transaction. Below are pseudo-code examples of the incorrect and correct methods of coding this process:

```
***INCORRECT***  
Dim MyDyna As Dynaset  
BeginTrans  
    MyDyna = CreateDynaset("Table1")  
    <misc.code such as Inserts/Updates/Deletes.>  
    MyDyna.Close  
CommitTrans/Rollback
```

```
***CORRECT***  
Dim MyDyna As Dynaset  
MyDyna = CreateDynaset("Table1")  
BeginTrans  
    <misc.code such as Inserts/Updates/Deletes.>  
CommitTrans/Rollback  
MyDyna.close
```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr

INF: Error Messages When Concatenating Variables or Controls
Article ID: Q96576

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

When you are concatenating variables or controls in a function or CreateDynaset method, you may receive one of the following error messages:

Type Mismatch

-or-

1 parameter expected only 0 supplied

These error messages can result if either of the following is true:

- You are including a variable or control name in the function syntax and Microsoft Access is unable to recognize the data type.
- You are using incorrect syntax to concatenate the variables.

This article presents examples in which these errors occur and explains how to correct them.

MORE INFORMATION

=====

"Type Mismatch" Error Message

You receive the "Type Mismatch" error message when you concatenate a variable or control that has a string data type in a method or function as a numeric data type. For example, the following function produces the "Type Mismatch" error message:

```
Function MyFunction (DataToFind As String)
    Dim MyDB As Database, MySet As Dynaset
    Set MyDB = CurrentDB()
    Set MySet = MyDB.CreateDynaset("Employees")
    MySet.FindFirst "Title = " & DataToFind
```

The correct syntax for the last line of code above is as follows:

```
MySet.FindFirst "Title = '" & DataToFind & "'"
```

Keep the following requirements in mind:

- When the criteria for an Access Basic function or method is a string data type, single quotation marks around the string variable are

required.

- Numeric data types do not require delimiters, as in the following example:

```
MySet.FindFirst "Title = " & NumericDataToFind & "
```

- Dates require the number sign (#).

"1 Parameters Were Expected, But Only 0 Were Supplied" Error Message

You may receive this error message when you use the CreateDynaset method in Access Basic on an existing query. If the query is a parameter query, you need to explicitly declare the parameter and its data type and set the parameter value for that query in the function. The following sample code will generate the error message:

```
Function TestQP ()
    Dim MyDB As Database, MySet As Dynaset
    Set MyDB = CurrentDB()
    Set MySet = MyDB.CreateDynaset("Query1")
    Debug.Print MySet![First Name]; Tab(10); MySet![Last Name]
End Function
```

When you refer to a parameter query, the correct syntax is as follows:

```
Function TestQP ()
    Dim MyDB As Database, MySet As QueryDef, MyDyna As Dynaset
    Set MyDB = CurrentDB()
    Set MySet = MyDB.OpenQueryDef("Query1")
    MySet![Enter a Name] = "Dav*"
    Set MyDyna = MySet.CreateDynaset()
    Debug.Print MyDyna![First Name]; Tab(10); MyDyna![Last Name]
    MyDyna.Close
    MySet.Close
End Function
```

The same error message also appears when you concatenate a variable in the SQL SELECT statement of a CreateDynaset method. The correct syntax is as follows:

NOTE: In the following example, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this example.

```
Set MySet = MyDB.CreateDynaset("SELECT * FROM Employees WHERE _
    [Employee ID] = " & Forms!Form1!Field0 & ";" )
```

This SELECT statement points to a control on a form for the WHERE clause.

Additional reference words: 1.00 1.10 2.00 parameters
KBCategory: kbprg kberrmsg
KBSubcategory: PgmObj

INF: Error Msg "Operation Not Supported on Paradox Table..."
Article ID: Q90140

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

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 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 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 to the query only one time. If you add the table twice, you will receive the error stated above when you run the query.

Report Limitation

You cannot use or design a Microsoft Access report based on a query to which a nonkeyed Paradox table has been added twice.

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 nonkeyed Paradox tables can only be opened once in Microsoft Access. Use of the table in both of these scenarios requires the table to be opened more than one time in order to perform the specific Microsoft Access task.

Paradox is manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

REFERENCES

=====

For more information on using Paradox files, search for "Paradox files" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmPdox

INF: Explanation of "Option Compare Database" Statement

Article ID: Q98227

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

The description of the Option Compare Database statement in the Microsoft Access "User's Guide" is not very clear. The following is the description from the guide:

Comparison is based on the characters' relative position according to the sort order specified when the database was created or compacted.

MORE INFORMATION

=====

The Option Compare statement can be used to make comparisons based on binary value, text value, or values that are defined in the Options dialog box. (To modify the Options dialog box, choose Options from the View menu.)

For example, if you use the Option Compare Binary statement, the expression "a" > "Z" would be true, because the ASCII value of "a" is greater than the ASCII value of "Z". However, if you use the Option Compare Text statement, the expression would be false, because "Z" comes after "a" in the alphabet. If you use the Option Compare Database statement, the expression will be true or false depending on the New Database Sort Order setting you select in the Items box of the Options dialog box.

The Option Compare statement can also affect the results of the Instr() function. For example, the following function finds the first tab character in a string:

```
Function FindTab (mystr$)
    FindTab = Instr(1, mystr$, Chr(9))
End Function
```

If mystr\$="1<space>abc<tab>ABC", the Instr() function will return the following results, depending on which Option Compare statement is used:

```
Option Compare Database: 2
Option Compare Text:     6
Option Compare Binary:   6
```

For another example, the following function returns the position of the first uppercase 'c' in the string:

```
Function FindEm (mystr$)
    FindEm = Instr(1, mystr$, "C")
End Function
```

If mystr\$="1<space>abc<tab>ABC", the function will return the following

results, depending on which Option Compare statement is used:

Option Compare Database: 5
Option Compare Text: 5
Option Compare Binary: 9

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg kbdocerr

KBSubcategory: PgmOthr

INF: Explanation of Compliance Checking Program (CCP)

Article ID: Q112124

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

Microsoft offers the cost-saving option for you to upgrade from previous versions of its products, or to upgrade from a competitive product, for a decreased price. To ensure that the purchase is a true upgrade or a competitive upgrade, several Microsoft products run a program called the Compliance Checking program (CCP) before running Setup.

MORE INFORMATION

=====

CCP is used by Microsoft Access version 2.0, as well as by other Microsoft products such as Microsoft Word for Windows and Microsoft Excel. CCP checks for previously installed software on your system that meets upgrade or competitive upgrade criteria before you can install the upgrade. (The restriction criteria vary according to the package you purchase and the country in which you purchase it. See the product's documentation for details.)

CCP uses a list of acceptable Microsoft or competitors' products for the compliance checking. If the CCP program finds one of these products installed on your system, it allows Setup to continue. If CCP does not find one of the accepted products in the product list, it asks you to list the directory where an acceptable product is installed. This includes any location on a hard disk, CD-ROM, network drive, or floppy disk.

For example, if you purchase and install a Microsoft Access 2.0 upgrade package to upgrade from Microsoft Access version 1.1 to 2.0, CCP checks for the presence of Microsoft Access 1.1 on your system before installation can occur. If you purchase a competitive upgrade package to upgrade from Borland Paradox version 4.5 for Windows to Microsoft Access 2.0, CCP checks for the presence of Borland Paradox version 4.5 for Windows on your system before installation can occur. The back of the upgrade package box describes how compliance is handled, and lists acceptable competitive products. On the box, the term "special software" refers to CCP. Otherwise, the reseller is mentioned.

Different CCP Versions

There are two different versions of CCP: Lenient and Strict. Lenient CCP is used in all U.S. versions of Microsoft Access, while Strict CCP is used in countries other than the United States.

If Lenient CCP does not find one of the acceptable products in the product list, it is a restriction violation. However, only a warning is given and installation does occur.

When Strict CCP perceives a violation, it halts installation with a violation message stating that installation cannot continue.

Product List

In the United States, an upgrade to Microsoft Access version 2.0 requires only a previous installation of Microsoft Access version 1.x.

A competitive upgrade in the United States requires a previous installation of any of the following products:

- Microsoft Access 1.x
- Microsoft Works for Windows 3.0, 2.0a, 2.0 (US, Z, F, G)
- Microsoft Works for MS-DOS 3.0, 2A, 2, 1.5, 1.05, 1.0 (US, Z, F, G)
- Microsoft FoxPro for MS-DOS 2.0, 2.5, 2.5a, 2.5b, 2.6
- Microsoft FoxPro for Windows 2.5, 2.5a, 2.5b, 2.6
- Microsoft Office 4.0-4.2
- Lotus Approach 1.0, 2.0, 2.1
- DataEase for MS-DOS 4.2, 4.5
- DataEase for Windows 1.0, 1.1
- FileMaker Pro for Windows 2.0, 2.1
- Borland dBASE III+ 1.1, IV 1.5, 2.0
- Borland Paradox for MS-DOS 3.5, 4.0, 4.5
- Borland Paradox for Windows 1.0, 4.5
- Symantec Q&A for MS-DOS 2.0, 3.0
- Symantec Q&A for Windows 4.0 (Checking for Q&A or QAWrite)
- Microrim R:Base 2.11, 3.0, 3.1, 4.0, 4.5, 4.5+
- Software Publishing Superbase4 for Windows 1.2,1.3
- Computer Associates Clipper 5.01
- PowerSoft Powerbuilder 3.0

Additional reference words: 2.00 Complying Magic Answer

KBCategory: kbsetup

KBSubcategory: StpOthr

INF: Explanation of the KeepTogether Property for Sections
Article ID: Q95648

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Form and report sections have a KeepTogether property that you can set to Yes or No. The KeepTogether property setting for sections determines whether Microsoft Access prints the controls located in that section on a single page or on more than one page.

The section's KeepTogether property affects the controls located in that section, for a single record.

MORE INFORMATION

=====

If the KeepTogether property is set to yes, Microsoft Access begins printing the controls located in that section at the top of the next page if it cannot print all of the controls on the current page. If it is set to no, Microsoft Access prints as many controls as it can on the current page and continues printing them on the next page.

If all the controls in a section require more than one page for a single record, Microsoft Access continues printing on a second page regardless of the KeepTogether setting.

For example, say the detail section of a report has its KeepTogether property set to Yes and all the controls in that section will not fit on the current printed page. In this case, Microsoft Access prints none of the controls on the current page. Instead, it begins printing the controls at the top of the next page. If it turns out the section requires more than a single page, Microsoft Access ignores the KeepTogether property value and continues printing the controls at the top of the next page.

The KeepTogether property affects the controls in a form or report section for a single record -- not groups of records. The KeepTogether property will not keep all the grouped records together, and it will not avoid the problem of abandoned group headers left on the bottom of a printed page.

Several other articles can help you learn how to create a label that will print at the top of each report page when a group of records span more than one page. To find these articles, query on the following words here in the Microsoft Knowledge Base:

access and continued and new and page

REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 313-314

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptProp

INF: Export Btrieve Table Without Existing Xtrieve FILE.DDF
Article ID: Q93688

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

To export a Btrieve table by using the Microsoft Access user interface, you must ensure that an Xtrieve dictionary file (FILE.DDF) exists. If the FILE.DDF file does not exist, use the macro action Transfer Database to export the table.

MORE INFORMATION

=====

Multiple Btrieve tables can exist in Xtrieve databases, just as Microsoft Access databases can have multiple tables. The FILE.DDF file describes the structure of the Xtrieve database.

Example

The following sample macro (TransBtrieve) transfers the Customers table from the sample database NWIND.MDB to a Btrieve Table named Customers in an Xtrieve database named C:\ACCESS\BTRV00000.DAT. If C:\ACCESS is a valid directory, the export occurs even if BTRV00000.DAT does not exist.

Use the following as a guide to create the sample macro. Note that the database name can be any complete file specification.

```
Macro Name      Action
-----
TransBtrieve    TransferDatabase

TransferDatabase Action
-----
Transfer Type:  Export
Database Type:  Btrieve
Database Name:  C:\ACCESS\BTRV00000.DAT
Object Type:    Table
Source:         Customers
Destination:    Customers
Structure Only: No
```

In addition, for Microsoft Access version 2.0, you must add IndexDDF=Maintain to the [Btrieve ISAM] section of the MSACC20.INI file in order for Microsoft Access to create the INDEX.DDF file.

NOTE: You must have the file WBTRVCALL.DLL in the \WINDOWS\SYSTEM subdirectory. If you have Microsoft Access installed locally, this file should be about 51K. If you're running Microsoft Access off a network, then this file will be about 13K and this macro won't work.

Microsoft Access creates the following four files if they do not exist:

FILE.DDF
INDEX.DDF
FIELD.DDF
BTRV00000.DAT

Transferring more than one table adds data to the existing files, but does not create more files as long as the database name is not changed.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 4, "Importing, Exporting, and Attaching," page 86

For more information in version 2.0 on [Btrieve ISAM] settings, see "Using Microsoft Access with Btrieve" then "Setting Btrieve Options" in the Contents of the ACREADME.HLP file, version 2.0, in the ACCESS directory.

Additional reference words: 1.00 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmBtrv

INF: Exporting a Parameter Query to a Spreadsheet or Text File
Article ID: Q124514

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to use the TransferText or TransferSpreadsheet action in a macro to export a parameter query. (A parameter query is a query that requires input in order to run.) You can export queries without parameters using the TransferText or TransferSpreadsheet macro action using the query's name for the action's Table Name argument. If you try to do this with a parameter query, however, you receive the following error message:

1 parameters were expected, but only 0 were supplied.

To export a parameter query, you can eliminate the parameter from the query and instead pass it using an Access Basic function called in the query's Criteria row.

MORE INFORMATION

=====

The following example demonstrates how to export a parameter query:

1. Open the sample database NWIND.MDB.
2. Create a new query based on the Customers table.
3. Drag the Company Name and the City fields from the field list to the query grid.
4. In the Criteria row for the City column, type "Find_City()" (without quotations marks).
5. Save the query as Query1 and then close it.
6. Create a new module and then enter the following sample code in the module:

```
Function Find_City()  
  
Find_City = Inputbox("Enter city name")  
' You could also use "= [Forms]![Customers]![City]" with the  
' assumption that the Customers form is open and that the City  
' field contains a value.  
End Function
```

7. Save the module as Module1 and then close it.
8. Create the following new macro:

Macro Name	Action
Macrol	TransferSpreadsheet

Macrol Action

TransferSpreadsheet	Transfer Type: Export
	Table Name: Query1
	File Name: C:\access\test.xls

9. Save the macro as Macrol and then close it.
10. Select the Macrol macro in the Database window and then choose the Run button.
11. When you are prompted "Enter city name," enter "London" (without quotation marks) and then choose the OK button. When you choose OK, a spreadsheet file called TEST.XLS is created in the ACCESS directory on drive C. It contains only records with a city of London.

REFERENCES

=====

For more information about the TransferText macro action, search for "TransferText" then "TransferText Action" using the Microsoft Access Help menu.

For more information about the TransferSpreadsheet macro action, search for "TransferSpreadsheet" then "TransferSpreadsheet Action" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00
 KBCategory: kbusage kberrmsg
 KBSubcategory: QryParm

INF: Exporting Tables from MS Word with AccessExporter Macro
Article ID: Q122247

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Microsoft Word version 6.0 for Windows contains a macro called AccessExporter that you can use to export tables from Word to Microsoft Access. The AccessExporter macro is in the TABLES.DOT template shipped with Word 6.0.

MORE INFORMATION

=====

The AccessExporter macro exports the selected portion of a Word table to a Microsoft Access table. When you run the macro, you are prompted for the name of the Microsoft Access database to which you want to export the table. You can export the Word table to an existing Microsoft Access table, or you can have the macro create a new table in Microsoft Access. You can also specify whether the first row in the Word table contains the table's field names.

The following steps demonstrate how to use the AccessExporter macro to export a table from Word 6.0 to Microsoft Access:

1. Start Word 6.0 and open the document that contains the table you want to export to Microsoft Access.
2. From the File menu, choose Templates, and then choose the Add button.
3. In the Directories box, select the WINWORD\MACROS subdirectory. In the File Name box, select the TABLES.DOT file. Choose OK, verify that the TABLES.DOT file is listed in the Global Templates And Add-ins box, and then choose OK again.
4. Click anywhere in the table you want to export, and then choose Select Table from the Table menu. (Or, use the mouse to select the portion of the table you want to export.)
5. From the Tools menu, choose Macro. In the Macro Name box, select AccessExporter, and then choose the Run button.
6. In the File Name box, select the Microsoft Access database to which you want to export the Word table, and then choose OK.
7. In the Export Table Options dialog box, select the appropriate options. Choose the Help button for a description of each option. Choose OK when you are ready to export the table.

The macro functions by exporting the selected portion of the Word table to a delimited text file, and then importing the text file into the Microsoft

Access database. If you choose to create a new table in Microsoft Access, the table will be called Winword.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr

INF: Exporting WordPerfect Secondary Files to Microsoft Access
Article ID: Q99317

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes how to export data from a WordPerfect (WP) print merge secondary file (or data file) to Microsoft Access.

MORE INFORMATION

=====

A WordPerfect secondary file (or data file) contains records, typically names and addresses, used in a print merge with a primary file (or form file), such as a form letter or mailing label document. To use data from a secondary file, the information must be saved in a comma or tab delimited (separated), unformatted text file. WordPerfect secondary files are neither comma nor tab delimited.

A typical comma delimited text file resembles the following:

```
"Joe","Smith","123 Main Street","Seattle, WA 98033"  
"Martha","Jones","4523 Williams Ave","","Redmond, WA 98052"
```

The equivalent WordPerfect secondary file resembles the following:

```
Joe{END FIELD}  
Smith{END FIELD}  
123 Main Street{END FIELD}  
Seattle, WA 98033{END FIELD}  
{END RECORD}  
<CRLF>  
Martha{END FIELD}  
Jones{END FIELD}  
4523 Williams Ave{END FIELD}  
{END FIELD}  
Redmond, WA 98052{END FIELD}  
{END RECORD}  
<CRLF>
```

where <CRLF> represents a hard page break. Note also that in Wordperfect versions later than 5.1, {END FIELD} and {END RECORD} will be shown as boldface without braces.

Problems: Secondary Files With Multiple-Line Field Entries

WordPerfect allows you to merge multiple lines of data in a single field. The following record has only two fields:

```
Joe Smith{END FIELD}
```

```
123 Main Street
Suite 101
Seattle, WA 98033{END FIELD}
{END RECORD}
<CRLF>
```

Delimited text files use carriage returns to mark the end of a record; these cannot be included in the field information. Each line in the secondary file must be made a unique field before it can be prepared as a delimited file. When you add the {END FIELD} code to a record, the new field is represented in every record in your file. Each record must contain the same number of fields and each field should either contain the same type of data or remain empty.

Using the {FIELD NAMES} Section in a Secondary File

Fields in a secondary file can be referenced from a primary file by number. These fields are numbered from top to bottom, or by a name that is assigned in the optional {FIELD NAMES} section, which appears at the top of the secondary file. The {FIELD NAMES} section in WP 5.1 may look like this

```
{FIELD NAMES}
Name~
Address~
~{END RECORD}
```

or like this:

```
{FIELD NAMES}Name~Address~~{END RECORD}
```

The {FIELD NAMES} section in WP versions later than 5.1 may look like this:

```
FIELDNAMES (Name;Address)ENDRECORD
```

If referenced by number, the Name field is 1 and the Address field is 2. A hard page break falls between this section and the first record.

Preparing the Comma Delimited Secondary File

WordPerfect for Windows:

WordPerfect for Windows can save secondary files directly in a comma delimited format. However, if you have a {FIELD NAMES} section in the file and want to preserve these names in Microsoft Access, you must reformat the file so that each name is separated by a comma and not by a tilde (~) character. Also, the fields should not appear on separate lines, nor be followed by carriage returns. The following {FIELD NAMES} section

```
{FIELD NAMES}
Name~
Address~
Phone~
~{END RECORD}
```

should be editing as follows:

```
{FIELD NAMES}Name,Address,Phone{END RECORD}
```

Creating the Comma Delimited File:

1. From the File menu, choose Save As.
2. In the Save As box, type a unique file name, preferably with a .TXT extension.
3. In the Format box, select ASCII Delimited Text (DOS), then choose Save.

WordPerfect for MS-DOS:

WordPerfect for MS-DOS cannot save secondary files directly in a delimited format. To create the comma delimited file, you need to create a primary file (or form file) and merge the secondary information into it. The primary document will contain fields already in comma delimited format. Once the data is merged, remove the hard page breaks separating each record. The resulting document can be saved as a text file that Microsoft Access can import.

Creating the Primary File (or form file) in WP 5.1:

1. Press F7, choose No, then choose No again to create a new WordPerfect document.
2. Type an opening quotation mark. Press SHIFT+F9, then 1 (Field).
3. In the Enter Field box, type the name of the first field if a {FIELD NAMES} section appears in the secondary file, or type "1" if there is no {FIELD NAMES} section.
4. Press ENTER. Type a closing quotation mark, followed by a comma.
5. Repeat steps 2-4 for each field in the secondary file, changing only the field name or field position number as appropriate. Do not type a comma after the last field.
6. Press ENTER to place a hard carriage return at the end of the record.
7. Press F10, type a unique filename in the Document To Be Saved box, and press ENTER. The primary document in WP 5.1 will resemble the following:

```
"{FIELD}name~","{FIELD}address~","{FIELD}phone~"
```

Creating the Primary File (or form file) in WP 6.0:

1. Press F7, choose No, then choose No again to create a new WordPerfect document.
2. Type an opening quotation mark. Press SHIFT+F9, then 1 (Form).

3. Choose 1 (Field). Type the name of the first field and choose OK; or choose List Field Names to select fields from an existing data file.
4. Type a closing quotation mark, followed by a comma.
5. Repeat steps 2-4 for each field in the secondary file, changing only the field name or field position number as appropriate. Do not type a comma after the last field.
6. Press ENTER to place a hard carriage return at the end of the record.
7. Press F10, type a unique filename in the Document To Be Saved box, and press ENTER. The form document in WP 6.0 will resemble the following:

"FIELD(name)", "FIELD(address)"

NOTE: It is acceptable for the lines to wrap on the screen.

Merging the Primary and Secondary Files:

1. Press F7, choose No, then choose No again to create a new WordPerfect document.
2. Press CTRL+F9, then 1 (Merge).
3. In the Primary File or Form File box, type the name of the primary file and press ENTER.
4. In the Secondary File or Data File box, type name of the secondary file and press ENTER.
5. For WP 6.0 only, choose Merge.

Removing the Residual Hard Page Breaks:

1. In WP 5.1, press ALT+F2, then N to disallow replacements. In WP 6.0, press ALT+F2.
2. At the -> Srch prompt, press CTRL+ENTER to enter a [HPg] code.
3. Press ALT+F2 twice to pass the Replace With message.

Saving As an Unformatted Text File in WP 5.1:

1. Press CTRL+F5, then 1 (DOS Text), then 1 (Save).
2. Type a valid document name, preferably with a .TXT extension, and press ENTER.

Saving As an Unformatted Text File in WP 6.0:

1. Press F10, type a valid document name, preferably with a .TXT extension.
2. Press TAB, choose DOS Delimited Text, then press ENTER.

WordPerfect is manufactured by WordPerfect Corporation, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

REFERENCES

=====

"WordPerfect for DOS Reference," version 5.1, pages 420-431

"WordPerfect for DOS Workbook," version 5.1, pages 237-246

"WordPerfect for Windows Reference," version 5.1, pages 300-308

"WordPerfect for Windows Workbook," version 5.1, pages 203-210

"WordPerfect for Windows Reference," version 5.2, pages 321-328

"WordPerfect for Windows Workbook," version 5.2, pages 227-236

"WordPerfect for DOS Reference," version 6.0, pages 412-422

"WordPerfect for DOS Learning WordPerfect," version 6.0, pages 195-204

"WordPerfect for Windows Reference," version 6.0, pages 453-473

"WordPerfect for Windows Learning WordPerfect," version 6.0, pages 201-210

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpPrtm

INF: Expressions to Count Yes, No, and Other Responses
Article ID: Q101076

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY
=====

This article lists sample expressions that you can use to count the occurrences of Yes, No, or Null in a field with a Yes/No data type.

MORE INFORMATION
=====

The following expressions can be used in a report's group footers or report footer to count the occurrences of Yes, No, or Null in a field named YesNoField the with data type Yes/No:

Expression	Sums What
=Sum(IIF([YesNoField],1,0))	Yes
=Sum(IIF([YesNoField],0,1))	No
=Sum(IIF(Not[YesNoField],1,0))	No
=Sum(IIF(IsNull[YesNoField],1,0))	Null

You can also create a related expression to count a specific value in a field. For example, the following sample expression counts all occurrences of the value 3 in a field called MyField.

=Sum(IIF([MyField]=3,1,0))

Example Using Sample Database NWIND.MDB

1. Open the sample database NWIND.MDB.
2. Use the Report Wizard to create a Groups/Totals report based on the Products table.
3. Select Category ID as the only field on your report.
4. Group on Category ID.
5. In the report's Design view, add an unbound text box in the Category ID Footer section with the ControlSource property for the box set to the following expression:

=Sum(IIF([Discontinued],1,0))

6. Add a second unbound text box with the ControlSource property for the box set to the following expression:

=Sum(IIF([Discontinued],0,1))

7. From the File menu, choose Print Preview. The first expression will count the number of products within each category that have the Discontinued field set to Yes. The second expression will count the number of products within each category that have the Discontinued field set to No.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptTotal

INF: Field and Record Validation Rule Enforced at Engine Level
Article ID: Q113874

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

Microsoft Access version 2.0 enforces field and record validation rules defined at the table level, regardless of the data input or editing method.

MORE INFORMATION

=====

In Microsoft Access version 1.x, field validation defined at the table level is not enforced by the database engine. The validation rule and validation text defined in table design are used as the defaults for the ValidationRule and ValidationText properties in forms. There is no record validation in Microsoft Access 1.x.

In Microsoft Access 2.0, the validation rule for both the field and table record are enforced at the engine level. The validation rules are enforced whenever you add or edit data, whether through a table's datasheet, a form bound to the table, an append query, an update query, Access Basic code, or by importing data from another table.

If you have converted a database from Microsoft Access 1.x to version 2.0, you can test to see if the data in the table meets the validation rules with the Test Validation Rules command, available by clicking the right mouse button on the title bar of the Table Design window.

The expression for a field validation rule cannot refer to fields or controls in a field validation rule, user-defined functions, Microsoft Access domain functions, aggregate functions, or the CurrentUser() or Eval() functions. The expression for a record validation rule cannot contain user-defined functions, Microsoft Access domain functions, aggregate functions, or references to forms, queries, or tables. Expressions can include only references to fields in the table. If you try to do field level validation on a table including any of the above options you will encounter the following error message:

Unknown or invalid reference '<Field Name>' in validation rule or default value in table '<Table Name>'

However, if you use the CurrentUser() or Eval() function in your validation rule you will receive this message instead:

Unknown function '<Function Name>' in validation expression or default value on '<[Table Name]>.[<Field Name>'

The table below identifies the interactions between the field and form validation rules:

Field Validation Rule Defined (Table Level)	Form Validation Rule Defined	Behavior
No	No	No validation is performed at any level.
Yes	No	Validation will be enforced regardless of method used to insert data.
No	Yes	Validation enforced at form level only. User can bypass the form validation rule when editing data outside the form. This is the same as Microsoft Access 1.x behavior.
Yes	Yes	When editing data in a form, form validation rules will be enforced first. If the value passes form validation, then the field validation rule is enforced. In order for data to be accepted, it must pass both rules.

REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 8 "Changing and Customizing Tables"

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: TblFldp

INF: Fill New Record with Data from Prev. Record Automatically
Article ID: Q88670

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

When you are creating new records, you may want to speed up the data entry process by having fields in the new record fill with values from fields in the previous record automatically.

This article demonstrates a sample Access Basic function called AutoFillNewRecord() that you can use to fill fields in a new record with values from fields in the previous record.

MORE INFORMATION

=====

One technique used to automate repetitive data entry is to press CTRL+apostrophe (') with your insertion point in a field. Microsoft Access then inserts the value contained in the previous record in the new field.

However, you may want certain fields in a new record to fill with data from the previous record automatically.

The following Access Basic function, AutoFillNewRecord(), can be used to fill either selected fields or all fields in a new record with values from the previous record:

1. Open the sample database NWIND.MDB.
2. Create the following Access Basic module:

NOTE: In the following sample code, an underscore (_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
' *****  
' DECLARATION SECTION  
' *****  
Option Explicit  
  
' *****  
' FUNCTION: AutoFillNewRecord  
'  
' PURPOSE:  
'   Speeds up repetitive data entry by automatically filling  
'   selected fields in a new record with data contained in fields in  
'   the previous record.  
'  
' USAGE:
```

```

' Set the OnCurrent property of the form to read:
'
'     =AutoFillNewRecord()
'
' If you want only certain fields filled in automatically, create
' an unbound text box on the form with the following properties:
'
'     ControlName: AutoFillNewRecordCriteria
'     Visible: No
'     DefaultValue: ;Company Name;Contact Name;Contact Title;Address;
'
' NOTE: The DefaultValue property should be a semicolon-delimited
' list of control names on the form. You must have a semicolon
' to the left and right of each entry to avoid confusion with
' other fields with similar names.
'
' If you want all fields automatically filled in, do not create the
' AutoFillNewRecordCriteria text box described above.
'
' *****
Function AutoFillNewRecord ()
    Dim ds As Dynaset, F As Form, RetVal As Variant, i As Integer
    Dim AutoFillCriteria As String, AutoFillField As Variant

    On Error Resume Next

    ' Get the active form.. and exit if it is not available
    Set F = Screen.ActiveForm
    If Err Then Exit Function

    ' See if we are on the new record by trying to reference
    ' .. the bookmark. An error will occur if it is a new record.
    RetVal = F.Bookmark

    ' If there was no error, we are not on new record, so exit.
    If Err = 0 Then Exit Function

    Err = 0    ' Reset error flag

    ' Get the form dynaset and go to the last record. This last
    ' .. record is used to fill the new record with information.
    Set ds = F.Dynaset
    ds.MoveLast

    ' If we cannot move to, or find, last record, exit the function.
    If Err <> 0 Then Exit Function

    DoCmd Echo False

    ' Step through all the fields in the last record of the dynaset
    For i = 0 To F.Count - 1

        ' Get the list of fields to auto fill. An error will occur
        ' .. if there is no criteria field specified, which is
        ' .. trapped and is examined later.

        AutoFillCriteria = F![AutoFillNewRecordCriteria]

```

```

' Do not fill the field in the new record by default.
AutoFillField = False

' Fill the field in the new record if:
'
' 1. There are criteria fields (Err=0) and the current
'     field name in the dynaset is found in the field criteria.
'
If (Err = 0 And InStr(1, AutoFillCriteria, ";" & _
    (F(i).ControlName) & ";")) Then
    AutoFillField = True
End If

' 2. No criteria fields were specified (Err<>0).
'     (Fill ALL fields if no criteria is specified on the form)
'
If Err <> 0 Then AutoFillField = True

Err = 0 ' Reset error flag

' If we are going to fill the field, copy the value from the
' .. dynaset field (in last record) to the new record field
' .. on the form.
If AutoFillField = True Then
    F(i) = ds((F(i).ControlName))

    ' Ignore error if one occurs during the fill.
    If Err <> 0 Then Err = 0
End If
Next i

DoCmd Echo True

End Function

```

3. Open the Customers form in Design view. Change the form's OnCurrent property to read:

```
=AutoFillNewRecord()
```

4. Create an unbound text box on the form with the following properties:

```

ControlName: AutoFillNewRecordCriteria
Visible: No
DefaultValue: ;Company Name;Contact Name;Contact Title;Address;

```

When you go to a new record, the Company Name, Contact Name, Contact Title, and Address fields are filled in automatically. If you want all fields to automatically fill in, you can either skip step 4 above or leave the DefaultValue property blank.

Additional reference words: 1.00 1.10 2.00 default value defaultvalue
KBCategory: kbusage
KBSubcategory: GnlDe

INF: Filling a Multiple-Column List Box Using Access Basic
Article ID: Q95903

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

You can use Access Basic to fill in the information for a list box or a combo box. On pages 82-85 of the Microsoft Access "Introduction to Programming" manual, an example is given on how to fill in a single-column list box. You can also use Access Basic to fill in a multiple-column list box or combo box.

MORE INFORMATION

=====

To create a multiple-column list box, you must set Code 4 in the Select Case statement to the number of columns that you want in your list box. In Code 6 of the Select Case statement, define the data that will display in each column and row of the list box. The following code is based on the example on page 84 of the Microsoft Access "Introduction to Programming" manual:

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

```
'The following function uses a select case statement to fill a  
'two-column and four-row list box. The function fills the first column  
'of the list box with the dates of the next four Mondays. The second  
'column is filled with the dates of the next four Tuesdays.  
'=====
```

```
Function ListMonTuesdays(fld As Control, id, row, col, code)  
    Dim offset  
    Select Case Code  
        Case 0  
            ListMonTuesdays=True           'Initialize.  
        Case 1  
            ListMonTuesdays=id           'Open.  
        Case 3  
            ListMonTuesdays=4           'Unique ID number for control  
        Case 4  
            ListMonTuesdays=2           'Number of rows.  
        Case 5  
            ListMonTuesdays=-1          'Number of columns.  
        Case 6  
            ListMonTuesdays=-1          'Column width.  
        Case 7  
            ListMonTuesdays=-1          'Use default width.  
    End Select  
  
'=====
```

```
'In the next Case statement:  
,
```

```
' Offset is the formula for finding the next four Mondays.  
' If column=0, then fill in with the dates for the next four  
' Mondays in column 1. If column=1, then fill in with the dates  
' for the next four Tuesdays in column 2.
```

```
'=====
```

```
Case 6                                'Get Date  
  Offset=abs((9-Weekday(Now))Mod 7)  
  If col=0 then  
    ListMonTuesdays=Format(Now()+offset+7*row,"mmm d")  
  Else  
    Offset=abs((10-Weekday(Now))Mod 7)  
    ListMonTuesdays=Format(Now()+offset+7*row,"mmm d")  
  End if  
End Select
```

End Function

REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.0, pages 82-85

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsCmbo

INF: FindEOM() Sample Function to Find Last Day of Month
Article ID: Q103184

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

This article describes a sample Access Basic function called FindEOM() that accepts a date as an argument and returns the last day of the month.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

The sample code below demonstrates how to set up the FindEOM() function. Enter the following in a new or existing module:

```
*****  
'Declarations Section of Module  
*****  
Option Explicit  
  
*****  
'FindEOM Function  
*****  
'This function takes a date as an argument and returns the last  
'day of the month.  
Function FindEOM (MyDate)  
    Dim NextMonth, EndOfMonth  
    NextMonth = DateAdd("m", 1, MyDate)  
    EndOfMonth = NextMonth - DatePart("d", NextMonth)  
    FindEOM = EndOfMonth  
End Function
```

Use the following sample expression to call the FindEOM() function:

```
=FindEOM(#5/5/93#)
```

To view the results of the FindEOM() function, type the following in the module's Immediate window:

```
?FindEOM(#5/5/93#)
```

Additional reference words: 1.00 1.10 2.00
KBCategory: kbprg

KBSubcategory: PgmOthr

INF: Fixes to Btrieve ISAM in MS Access Service Pack

Article ID: Q123594

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

The Microsoft Access version 2.0 Service Pack includes the following new Btrieve ISAM:

File Name	Size	Date
BTRV200.DLL	111,616	11/20/94 12:00a

MORE INFORMATION

=====

The new Btrieve ISAM fixes issues using Btrieve data with Microsoft Access. The new ISAM:

- Improves performance in queries based on multisegmented indexes and indexes based on type 8 (COBOL Numeric data type) fields.
- Fixes problems where incorrect results are produced in queries using indexes based on type 8 (COBOL Numeric data type) fields.
- Fixes the error that causes the error message "Unexpected error from external database driver [7]" to be returned to Microsoft Access when you are using a Btrieve version 6.x NLM.

The new Btrieve ISAM is included in the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927
TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

Additional reference words: 2.00 jet25 jet 2.5
KBCategory: kb3rdparty kberrmsg
KBSubcategory: IsmBtrv

INF: Fixes to Paradox ISAM in MS Access Service Pack

Article ID: Q123592

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

The Microsoft Access version 2.0 Service Pack includes the following new Paradox ISAM:

File Name	Size	Date
PDX200.DLL	233,328	11/20/94 12:00a

MORE INFORMATION

=====

The new Paradox ISAM corrects the following issues:

- The new ISAM corrects problems trying to create indexes for Paradox tables with a long directory path.
- Tables marked read-only cannot be renamed using the new Paradox ISAM.

The new Paradox ISAM is included in the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kb3rdparty

KBSubcategory: IsmPdcox

INF: Fixes to xBASE ISAM in MS Access Service Pack

Article ID: Q123590

The information in this article applies to:

- Microsoft Access version 2.0
-

SUMMARY

=====

The Microsoft Access version 2.0 Service Pack includes the following new xBASE ISAM:

File Name	Size	Date
XBS200.DLL	296,832	11/20/94 12:00a

MORE INFORMATION

=====

The new xBASE ISAM corrects the following issues using xBASE data with Microsoft Access:

- The new ISAM correctly reads the <filename>.INF file that gets created when you attach an xBASE table and provide a unique index. Using the prior ISAM can cause performance problems because indexes are not used correctly.
- The new ISAM translates extended characters in international code pages into Windows ANSI codes correctly. Using the prior ISAM, for example, extended character 130 could be translated as an underscore character instead of an accented "e."
- The new ISAM no longer generates "Invalid disk access" or "Disk I/O" error messages using some multiuser xBASE tables.
- The new ISAM expands the range of valid data in xBASE tables from 1/1/1753 through 12/31/2078 to 1/1/100 through 12/31/9999.

The new xBASE ISAM is included in the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kb3rdparty kberrmsg

KBSubcategory: IsmXbase

INF: Form and Report Modules Have Local Scope

Article ID: Q112726

The information in this article applies to:

- Microsoft Access version 2.0

SUMMARY

=====

This article describes the effect of "scope" of procedures and variables in form and report modules. Scope refers to the extent to which an identifier (that is, a constant, data type, variable, or routine) can be referenced in a program.

MORE INFORMATION

=====

The procedures and variables defined in a form or report module have "local" scope; that is, they and their values do not exist outside of that object. Attempting to call a function in a form or report module outside that form or report results in a "reference to undefined function or array" error message or other "unknown function name..." type of error. A call made to a subroutine from outside the form or report results in a "syntax error" message.

A reference to one of the form's variables from a different module produces unexpected results. Due to the nature of modules and procedures, a new variable of the same name is implicitly created with an Empty value (Variant type 0). If Option Explicit has been specified in the module that refers to the form's variable, then a "Variable not Defined" compile error occurs.

When you plan to create a procedure or variable in a form or report module, first determine that nothing outside of that form or report will require that the procedure or variable exist. Any variable or procedure you write in a form or report module should be useful to that form or report only.

If you need to call a procedure from multiple forms or reports, place the procedure in a global module. Do the same for any variables your code needs to access from more than one form or report.

The following example demonstrates the scope of a function in a form module and the errors that can occur:

1. Create a new database.
2. Create the following new form and name it Form1:

```
Form: Form1  
-----  
no RecordSource (unbound)  
  
Form1 Controls
```



```
-----  
command button: Button0  
  Caption: Form1_SayHi  
  OnClick: =Form1_SayHi()
```

3. In Form1's Design view, choose Code from the View menu. Form1's module will be displayed. Enter the following function in the module:

```
Function Form1_SayHi()  
  MsgBox "Hi!"  
End Function
```

4. Create the following new form, and call it Form2:

```
Form: Form2  
-----  
no RecordSource (unbound)  
  
Form2 Controls  
-----  
command button: Button0  
  Caption: Form1_SayHi  
  OnClick: =Form1_SayHi()
```

5. From the Database window, create a new module. Enter the following function in the module:

```
Function Run_Form1_SayHi()  
  Run_Form1_SayHi = Form1_SayHi()  
End Function
```

6. Open both forms. Click the command button on Form1. A message box with the text "Hi!" will be displayed. This works because the function is in scope in Form1.
7. Click the command button on Form2. An error message will be displayed. This is because the function is out of scope for Form2.
8. In the module's Immediate window, type the following and then press ENTER:

```
? Run_Form1_SayHi()
```

The error message "Reference to undefined Function or array" will be displayed. The error occurs because the function is still out of scope. Calling the function from anywhere except Form1 will result in an error.

REFERENCES =====

Microsoft Access "Building Applications," version 2.0, pages 59-60

Additional reference words: 2.00 cbf code behind forms
KBCategory: kbusage
KBSubcategory: FmsOthr

INF: Formatting a Label for Each Axis of a Graph

Article ID: Q99398

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
 - Microsoft Graph versions 3.0 and 5.0
-

SUMMARY

=====

The Microsoft Access GraphWizard does not give users the option to format a label for each axis of the graph. You can modify the labels in Microsoft Graph, but this change is not permanent. The next time the graph is updated, the formatting is lost.

This article describes the steps to change the format permanently by modifying the information that Microsoft Access sends to Microsoft Graph.

MORE INFORMATION

=====

When a graph is created using GraphWizard, a SQL statement is built. This statement is placed in the RowSource property of the graph object. Each time the form or report in which the graph resides is refreshed, the SQL statement is re-executed and the results are sent to Microsoft Graph.

This SQL statement can be modified by the user. We recommend that you copy it to a new query, where you can test your changes.

The example below describes the steps to change the format of the labels in a graph:

Procedure A - Create the Graph

For Microsoft Access 2.0:

1. Open the sample database NWIND.MDB. In the Database window, select the Orders table.
2. Choose the New Form button on the toolbar to create a new form.
3. Choose FormWizards. Select Graph and then choose OK.
4. Select the Order Date field from the Available Fields list and choose the right arrow to add it to the Fields for Graph list.
5. Repeat step 4 for Freight, then choose Next four times to accept the default (Group [Order Date] by: Year).
6. Choose Finish.

Note that the labels for the axes are: 1991, 1992, 1993, 1994

For Microsoft Access versions 1.0 and 1.1:

1. Open the sample database NWIND.MDB. In the Database window, select the Orders table.
2. On the Toolbar, choose the Forms button to create a new form.
3. Choose FormWizards. Select Graph and then choose OK.
4. To accept the defaults, press Next when the following message appears: "Which type of graph do you want?"
5. Select the Order Date field from the Available Fields list and choose the right arrow to add it to the Fields for Graph list.
6. Repeat step 5 for Freight, then choose Next twice to accept the default (Group [Order Date] by: Year). Choose Open.

Note that the labels for the axes are: 1989, 1990, 1991, 1992

Procedure B - Changing the Label Format

For Microsoft Access 2.0:

1. Open your form in Design view.
2. Select the graph. If the property sheet is not displayed, choose Properties from the View menu.
3. With your secondary mouse button, click the RowSource property and choose Build.
4. Change the field Year([Order Date]) to Format([Order Date],"YY"). Execute the query to verify that the years in the Order Date column are formatted as follows: 89, 90, 91, 92.
5. Close the query and save the changes.
6. Switch to Form view. The labels of the axes should now read as follows: 91, 92, 93, 94.

For Microsoft Access versions 1.0 and 1.1:

1. Switch to Design view. Select the chart object and display the property sheet.
2. Place your insertion point in the RowSource property and press SHIFT+F2 to open the Zoom window.
3. Highlight the entire SQL Statement to select it. Press CTRL+C to copy it to the Clipboard.
4. Create a new query by choosing the Query button on the Toolbar.

5. From the View menu, choose SQL.
6. Highlight all the text in the SQL dialog box to select it. Press CTRL+V to replace the existing text with the Clipboard contents.
7. Choose OK. The fields [Order Date] and [Order Amount] are placed in the query grid.
8. Change the field Year([Order Date]) to Format([Order Date],"YY"). Execute the query to verify that the years in the Order Date column are formatted as follows: 89, 90, 91, 92.
9. Switch to Design view. Copy the contents of the SQL dialog box to the Clipboard.
10. Return to the form. Select the chart object and paste the contents of the Clipboard into the RowSource property.
11. Switch to Form view. The labels of the axes should now read as follows: 89, 90, 91, 92.

Additional reference words: 1.00 1.10 2.00 chart ole embed requery label
KBCategory: kbusage
KBSubcategory: RptLabel

INF: Formatting Immediate IF (IIF) Statements in Reports

Article ID: Q106184

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

If a report that is based on a query contains a field that includes a null value from an Immediate If (IIf) statement, you will not be able to format the field properly on the report.

MORE INFORMATION

=====

There are two ways to work around this behavior. The first method is to format the IIf expression in the query. You can add a format statement to the beginning of the IIf statement. For example, the following statement would add formatting to the field:

```
Format(IIF([freight] = 32.07,[freight],null),"$0.00")
```

The second method is to use the IIf statement on the report directly, and not in the query. For example, you could add the following statement to the ControlSource property of the expression field's text box:

```
=Format(IIF([freight]=32.07,[freight],null),"$0.00")
```

Steps to Reproduce Behavior

1. Open the sample database NWIND.MDB.
2. Create the following new query based on the Customers and the Orders tables. Join the two tables on the Customer ID field. Save the query as TestIIF:

```
Query: TestIIF
```

Field: Company Name

Field: Order ID

Field: Freight

Field: Shipped Date

Field: MyField: IIf([freight]=32.07,[freight],null)

3. Use the ReportWizards to create a new single-column report based on the TestIIF query. Include all the fields on the report.
4. Choose Print Preview to verify that the MyField field is unformatted.
5. Choose Cancel to return to Design view. Try to format the field.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage
KBSubcategory: GnlFrmt

INF: Four Ways to Move to a Record from a Combo Box Selection
Article ID: Q100132

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
-

SUMMARY

=====

Listed below are four different methods of moving to a specific record based on a combo box selection:

1. In the AfterUpdate property of the combo box, call a macro using the FindRecord action.
2. In the AfterUpdate property of the combo box, call a macro using the ApplyFilter action.
3. Use a Form/Subform, with the combo box in the main form and the data in the subform, bound by the LinkMasterFields and LinkChildFields properties.
4. Base the form on a query joining two tables and bind the combo box to the field that controls the join, using the row fix-up or dynamic lookup techniques.

MORE INFORMATION

=====

These four methods are outlined in the sections below and are based on the sample database NWIND.MDB.

The following table compares the features (benefits and drawbacks) of the four methods:

Method Number:	1	2	3	4
Requires no code/macros			x	x
Saves on subforms	x	x		x
Can scroll to other records	x		x	x
Does not require a query	x	x	x	
Can edit records	x	x	x	

NOTE: These methods can also apply to text boxes.

Method 1

1. Create a form called ComboTest based on the Products table. You must include at least the Product ID field on the new form. (Additional fields can help to illustrate that you have changed records based on the value selected in the combo box.) Set the DefaultView

property to Single Form.

```
Object: Text Box
-----
ControlName: Product ID
ControlSource: Product ID
Visible: Yes
```

2. Create an unbound combo box called Comb01 with the properties described below. The combo box can be located anywhere on the form, but preferably in the form header or footer.

```
Object: Combo Box
-----
ControlName: Comb01
ControlSource: <leave blank>
RowSourceType: Table/Query
RowSource: Products
ColumnCount: 4
ColumnWidths: 0 in;0 in;0 in;2 in
BoundColumn: 1
After Update: Locate Product
```

3. Create a new macro called Locate Product with two actions:

```
Macro: Locate Product
-----
GoToControl
    ControlName: [Product ID]
FindRecord
    Find What: =Comb01
    Find First: Yes
```

4. Switch to Form view. Note that when you choose a product name in the combo box, you are moved to the record selected and the appropriate Product ID is displayed.

Another example of this method is illustrated in the Suppliers form in NWIND.MDB in Microsoft Access version 1.x, and in the Products And Suppliers form in Microsoft Access version 2.0.

Method 2

- ```

```
1. Repeat steps 1 and 2 from Method 1 described above.
  2. Create a macro called Locate Product with one action:

```
Macro: Locate Product

ApplyFilter
 Where:[Product ID]=Forms![ComboTest]![Comb01]
```

The Product ID field is the bound field in the combo box. While the combo box displays the Product Name information, it is bound to (or, holds internally) the value of the Product ID.



This method is similar to the Suppliers form in NWIND.MDB, which uses the buttons to filter records from A to Z. Also, see the Alpha Apply Filter Buttons macro.

#### Method 3

1. Create a new, unbound form. Add a combo box with the following properties:

Object: Combo Box

-----  
ControlName: Comb01  
ControlSource: <leave blank>  
RowSourceType: Table/Query  
RowSource: Products  
ColumnCount: 4  
ColumnWidths: 0 in;0 in;0 in;2 in  
BoundColumn: 1

2. Create a second form based on the Products table to use as a subform. Include at least the Product ID field. (Additional fields can help to illustrate that you have changed records based on the value selected in the combo box.) Set the DefaultView property to Single Form.

Object: Text Box

-----  
ControlName: Product ID  
ControlSource: Product ID  
Visible: Yes

3. Save the form, then use it to create a subform control on the first form by dragging it from the Database window to the Detail section of the first form.
4. Set the subform control properties as follows:

Object: Subform

-----  
LinkChildFields: [Product ID]  
LinkMasterFields: Comb01

By changing the value in Comb01, Microsoft Access ensures that the records in the subform match the combo box.

The Orders form in NWIND.MDB illustrates this method. The Order Details subform is related by the LinkMasterFields and LinkChildFields properties.

#### Method 4

1. Create a table called Prd containing a single field Product ID. Set the data type to Number and the field size to Long Integer. No primary key is necessary.

2. Create a single blank record.
3. Create a query called Prd Query based on a join between the Product ID fields of the Prd and Products tables. Include the following attributes in the query:

Query: Prd Query

-----  
Field: Product ID  
    TableName: Prd  
    Show: X  
Field: <any other fields you are interested in>  
    TableName: Products

4. Create a form based on Prd Query and add all fields. You must add at least the Product ID field.
5. Delete the text box control for Product ID and re-create it as a combo box, as follows:

Object: Combo box  
-----  
ControlName: Product ID  
ControlSource: Product ID  
RowSourceType: Table/Query  
RowSource: Products  
ColumnCount: 4  
ColumnWidths: 0 in;0 in ;0 in;2 in  
BoundColumn: 1

The Orders form in NWIND.MDB illustrates this method. It allows the customer name and address to be selected based on the Customer ID combo box. By changing the Customer ID in the Orders table, the related Customers record is changed and the corresponding fields are updated on the form. For another example, see the Products And Suppliers form.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, pages 270-284 and 545-548

Microsoft Access "User's Guide," version 1.1, pages 274-288 and 549-552

Microsoft Access "User's Guide," version 2.0, Chapter 17, "Creating Forms Based on More Than One Table"

For more information on the dynamic lookup or row fix-up techniques, query on the following words here in the Microsoft Knowledge Base:

dynamic and lookup

-or-

row and fix-up

For more information on how to reference controls on a form, search

for "Identifiers in Expressions" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 Forms Controls Text box

KBCategory: kbusage

KBSubcategory: FmsCmbo

## **INF: Function to Determine Retail or Run-Time Version**

**Article ID: Q103182**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

### SUMMARY

=====

This article demonstrates how to use a function to determine if your Microsoft Access application is running in the run-time or the retail version of Microsoft Access. This function can be used to prevent your application from being used in the retail version of Microsoft Access.

The SysCmd() function is new in Microsoft Access 1.1 and is documented in the Access Distribution Kit (ADK). This function enables you to display a meter in the Microsoft Access status bar. Although SysCmd() was designed to display a meter, the number 6 was added to return whether the application is running in the run-time or the retail version of Microsoft Access.

### MORE INFORMATION

=====

NOTE: This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

The following example demonstrates how to create a function called IsRunTime() that can be used to prevent an application from being run in the retail version of Microsoft Access.

1. Add the following function called IsRuntime() to a new or existing module in your application:

```
Function IsRuntime ()
 On Error GoTo ErrIsRuntime
 IsRuntime = SysCmd(6)
```

```
ByeIsRuntime:
 Exit Function
```

```
ErrIsRuntime:
 If (Err = 5) Then
 IsRuntime = False
 Else
 Error Err
 End If
 Resume ByeIsRuntime
```

End Function

Note that it is necessary to trap for Err = 5, Illegal Function Call, in case your application is being run in Microsoft Access version 1.0, which does not support the SysCmd(6) call.

2. Create the following macro and save it with the name AutoExec:

| Condition       | Action | Defined Below |
|-----------------|--------|---------------|
| Not IsRuntime() | MsgBox | 1             |
| ...             | Quit   | 2             |

AutoExec Actions

1. MsgBox  
    Message: Invalid setup, run YourAppName setup and try again.  
    Beep: Yes  
    Type: Critical  
    Title: YourAppName
2. Quit

Additional reference words: 1.10 2.00  
KBCategory: kbusage  
KBSubcategory: GnlApp

**INF: Function to Show Date/Time as Hours, Minutes, and Seconds**  
**Article ID: Q115918**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes a sample user-defined Access Basic function called GetTimeStr() that you can use to display a cumulative time amount as total hours, minutes, and seconds.

The GetTimeStr() function may be useful when you need to display the sum of the difference between the start time and the end time for a number of records.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

Given a table with each record having a start time and an end time, you can find the difference between the two times with the an expression similar to:

=[end time] - [start time]

This expression will result in a decimal value because Microsoft Access stores time as a number representing a fraction of 1 day. You can multiply the resulting value by 24 to see the value in hours. For example, 12:00 P.M. (noon) is stored as 0.5 (half the day has expired). To see the value in hours, multiply 0.5 by 24, to get 12 hours. 9:00 A.M. is stored as 0.375. Multiplying 0.375 by 24 results in 9 hours.

You can use the CSng() function to see the numeric equivalent of a time. For example, the function

? CSng(#12:00pm#)

results in 0.5.

The following function demonstrates how to see the difference in two times as a value in hours:

? CSng(#10:00am#-#9:00am#)\*24

This function returns a value of 1.

Given a table with 100 records, all of which have a start time of 9:00 A.M.

and an end time of 10:00 A.M., you could sum the difference in times for the records to get a numeric value of 4.16666666666667. Multiplying the numeric value of 4.16666666666667 by 24 would return a value of 100 hours.

This works well for whole-hour values. However, numeric values that do not represent whole hours, when multiplied by 24, will return decimal hour values, which may not be what you want. For example, the function

```
? CSng(#10:23am#)*24
```

will return an hour value of 10.38333.

The GetTimeStr() function returns values broken out by hours, minutes, and seconds.

#### The GetTimeStr() Function

---

Syntax: GetTimeStr(DateTime, Format)

| Argument | Description                                                                                                                                                                                                          |        |         |   |               |   |                 |   |                 |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|---------|---|---------------|---|-----------------|---|-----------------|
| DateTime | Date/Time data-type value that contains the data you want to use.                                                                                                                                                    |        |         |   |               |   |                 |   |                 |
| Format   | String of display format characters.<br>For consistency with Microsoft Access the following symbols are allowed:                                                                                                     |        |         |   |               |   |                 |   |                 |
|          | <table><thead><tr><th>Symbol</th><th>Meaning</th></tr></thead><tbody><tr><td>h</td><td>Display hours</td></tr><tr><td>n</td><td>Display minutes</td></tr><tr><td>s</td><td>Display seconds</td></tr></tbody></table> | Symbol | Meaning | h | Display hours | n | Display minutes | s | Display seconds |
| Symbol   | Meaning                                                                                                                                                                                                              |        |         |   |               |   |                 |   |                 |
| h        | Display hours                                                                                                                                                                                                        |        |         |   |               |   |                 |   |                 |
| n        | Display minutes                                                                                                                                                                                                      |        |         |   |               |   |                 |   |                 |
| s        | Display seconds                                                                                                                                                                                                      |        |         |   |               |   |                 |   |                 |

Remarks: To display a unit with a fixed number of leading zeros, repeat the symbol as many times as needed. For example, to display hours to the hundreds unit, use the format "hhh."

The first symbol will determine the units to be displayed. Each additional symbol must be delimited with a colon (:). Additional symbols will show additional units as remainders of the first unit. The following table demonstrates what will be displayed for various formats given a DateTime of 1.11 (1.11 days):

| Format  | Result                 |
|---------|------------------------|
| "h"     | 26 Hrs                 |
| "h:n"   | 26 Hrs 38 Mins         |
| "h:n:s" | 26 Hrs 38 Mins 24 Secs |
| "n"     | 3996 Mins              |
| "n:s"   | 3996 Mins 24 Secs      |
| "s"     | 95904 Secs             |

If the format does not begin with a valid symbol the result

```
is "#Error?"
```

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Option Compare Database 'Use database order for string comparisons.
Option Explicit
```

```
Function GetTimeStr (dt As Variant, frmt As String) As String
```

```
 Dim totalsec As Double
 Dim hrs As Variant
 Dim mins As Variant
 Dim secs As Variant
 Dim timestr As String
```

```
 Dim p As Integer
 Dim timefrmt As String
```

```
 If IsNull(dt) Then
 GetTimeStr = ""
 Exit Function
 Else
 totalsec = ((dt) * 86400)
 End If
```

```
 Select Case Left$(frmt, 1)
 Case "H"
 hrs = (totalsec \ 3600)
 mins = ((totalsec Mod 3600) \ 60)
 secs = (totalsec Mod 60)
 Case "N"
 hrs = 0
 mins = ((totalsec \ 24))
 secs = (totalsec Mod 60)
 Case "S"
 hrs = 0
 mins = 0
 secs = (totalsec)
 Case Else
 GetTimeStr = "#Error?"
 Exit Function
 End Select
```

```
 timestr = ""
```

```
 Do Until Len(frmt) = 0
```

```
 p = InStr(frmt, ":")
```

```
 If p = 0 Then
 timefrmt = frmt
 frmt = ""
```

```
 Else
 timefrmt = Left$(frmt, (p - 1))
 frmt = Mid$(frmt, (p + 1))
```



End If

Select Case Left\$(timefmt, 1)

Case "H"

    timestr = timestr & (Format\$(hrs, String\$(Len(timefmt), \_  
    "0")) & " Hrs ")

Case "N"

    timestr = timestr & (Format\$(mins, String\$(Len(timefmt), \_  
    "0")) & " Mins ")

Case "S"

    timestr = timestr & (Format\$(secs, String\$(Len(timefmt), \_  
    "0")) & " Secs")

End Select

Loop

GetTimeStr = timestr

End Function

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

## INF: Functions to Left and Right Pad Character Strings

Article ID: Q96458

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

An imported file may contain field values that have a combination of numeric and alphabetic characters. These fields do not sort in proper order if they contain variable numbers of characters. This article describes functions that you can use to pad these values with a selected character to produce values of consistent length.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic," in version 2.0.

### MORE INFORMATION

=====

Suppose you have a table that contains Customer ID numbers with values entered as follows:

```
Customer ID

123B
1231
1231B2
B123
```

In a query, these numbers would sort in ascending order as follows:

| Unpadded | Right Padded | Left Padded |
|----------|--------------|-------------|
| 1231     | 123100       | 001231      |
| 1231B2   | 1231B2       | 00123B      |
| 123B     | 123B00       | 00B123      |
| B123     | B12300       | 1231B2      |

Right padding does not change the sort order; however, it is useful if you need to make all values a consistent number of characters. Left padding, however, will allow proper sorting.

To create a left-padding function and a right-padding function, enter the following code in a new or existing module in your database.

Note: In the following sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

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

```
Option Explicit
Dim x As Integer
Dim PadLength As Integer
```

```
'=====
```

```
'The following function will left pad a string with a specified
'character. It accepts a base string which is to be left padded with
'characters, a character to be used as the pad character, and a
'length which specifies the total length of the padded result.
'=====
```

```
Function Lpad (MyValue$, MyPadCharacter$, MyPaddedLength%)
```

```
 Padlength = MyPaddedLength - Len(MyValue)
 Dim PadString As String
 For x = 1 To Padlength
 PadString = PadString & MyPadCharacter
 Next
 Lpad = PadString + MyValue
```

```
End Function
```

```
'=====
```

```
'The following function will right pad a string with a specified
'character. It accepts a base string which is to be right padded with
'characters, a character to be used as the pad character, and a
'length which specifies the total length of the padded result.
'=====
```

```
Function Rpad (MyValue$, MyPadCharacter$, MyPaddedLength%)
```

```
 Padlength = MyPaddedLength - Len(MyValue)
 Dim PadString As String
 For x = 1 To Padlength
 PadString = PadString & MyPadCharacter
 Next
 Rpad = MyValue + PadString
```

```
End Function
```

The following example shows an update query that would modify the Customer ID field by left padding the field with the 0 (zero) character. You will use the Lpad function that was created in step #2.

```
Update Query: Leftpad Customer Number
```

```

Field name: Customer ID
Update to: Lpad([Customer ID],"0",6)
```

Additional reference words: 1.00 1.10 2.00 constant  
KBCategory: kbusage  
KBSubcategory: ExrOthr

## **INF: Hiding the Database Window Menu Bar**

Article ID: Q109337

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

### SUMMARY

=====

In Microsoft Access version 1.x, in order to hide the menu bar associated with the Database window, you must hide the Database window itself. In Microsoft Access version 2.0, you can use the MenuBar property to hide the menu bar.

### MORE INFORMATION

=====

#### Microsoft Access version 1.x

-----

In version 1.x, the menu bar that appears when the Database window is selected is always the Database window menu bar. The run-time version of Microsoft Access hides the Database window by keeping it minimized, and not displaying the minimized window's icon. However, the Database window's menus and commands are still available on the menu bar.

To keep the Database window's menu bar from being displayed, you must hide the Database window. The following sample macro action demonstrates how to hide the Database window. Add this action to the executable application's Autoexec macro.

NOTE: Make sure to add this action before any OpenForm actions in the Autoexec macro in order to avoid hiding the forms after they are opened.

#### Action

-----

DoMenuItem  
    MenuBar: Database  
    MenuName: Window  
    Command: Hide

#### Microsoft Access version 2.0

-----

In version 2.0, the MenuBar property applies to the application object, forms, and reports, and can be set in the Autoexec macro for the application as a whole. You can use the MenuBar property for custom menu bars at all points in your application, even while in the Database window and reports.

### REFERENCES

=====

Microsoft Access "User's Guide," version 1.x, Chapter 24, "Using Macros for Application Design"

Microsoft Access "Building Applications," version 2.0, Chapter 2, "Structuring a User Interface"

Additional reference words: 1.00 1.10 2.00 ADK

KBCategory: kbui

KBSubcategory: UifOthr

## **INF: Hints for Printing Mailing Labels**

**Article ID: Q92807**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### 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 cannot 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.

Additional reference words: 1.00 1.10 2.00 w\_print wizards

KBCategory: kbusage

KBSubcategory: RptLabel

**INF: How First Line of Data is Used to Import Delimited Text**  
**Article ID: Q99401**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The first line of data is extremely important when you are importing delimited text files. Microsoft Access uses the first line of data to determine the following:

- The number of fields
- The data type for each field

For this reason, it is important to verify that the first line of data correctly reflects the rest of the data in the table. This article discusses some of the common problems you may encounter when the first line of data is incorrect and how to work around these problems.

MORE INFORMATION

=====

To specify the structure that Microsoft Access should use, you can use the following steps instead of relying on the first line of data:

1. Create a table in Microsoft Access that has the correct number of fields and the correct data type for each field.
2. From the File menu, choose Import.
3. Select Text (Delimited) from the Data Source list and choose OK.
4. Select the text file and choose OK.
5. In the Import Text Options dialog box, choose Append To Existing Table and specify the table you created in step 1.
6. Choose OK to import the file.

Determining the Number of Fields

-----

Microsoft Access scans the first line of data to determine the number of fields in the text file. If successive records contain more fields, these are ignored and no error message is given. For example, suppose you have a text file that looks like the following:

```
5959,"John Doe","123 Main Street"
5960, "George Brown","55 Orange Avenue","MainTown","WA","USA"
```

Because the first line only contains three fields, Microsoft Access creates the following table:



| Column 1 | Column 2     | Column 3         |
|----------|--------------|------------------|
| 5959     | John Doe     | 123 Main Street  |
| 5960     | George Brown | 55 Orange Avenue |

Note that the additional fields are ignored and no error message is generated.

#### Determining the Data Types

Microsoft Access scans the first line to determine the data type for each field. If the data type in the field is invalid, a "Type Conversion Failure" error message is generated in the Import Errors table. For example, if the following text file is imported

```
5959,"John Doe","123 Main Street",8/12/90
5960,"George Brown","55 Orange Avenue",81290
ABCD,"Jane Bell","78 West Street",8/12/90
```

Microsoft Access detects four fields with the following data types:

| Field | Data Type |
|-------|-----------|
| 1     | Number    |
| 2     | Text      |
| 3     | Text      |
| 4     | Date/Time |

The second record fails to convert because the number 81290 in the fourth field is not a Date/Time data type, as determined by the first record value (8/12/90). The third record fails because "ABCD" in the first field is not a number. As a result, Microsoft Access reports type conversion errors in each case.

#### Other Workarounds

In addition to import appending the text file to an existing table, it is also possible to use a text editor to edit the first line of data to correctly reflect the number of fields and data types in the file.

If a field in the first row contains a number that is supposed to belong in a text field, type double quotation marks around the number in the first row. Microsoft Access will then correctly interpret the field as a text field.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, pages 72-79

Additional reference words: 1.00 1.10 2.00 import

KBCategory: kb3rdparty

KBSubcategory: IsmTxd



**INF: How Microsoft Access Deletes Objects from a Database**  
**Article ID: Q90129**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The following is a description of what happens internally when an object (table, form, query, and so on) is deleted from a database in Microsoft Access.

MORE INFORMATION

=====

When you delete an object from the Database window, it is not physically deleted from the database at that moment. Instead, it is renamed to a temporary name (for example, ~TMPCLP1) and hidden, so that it disappears from the Database window. The renaming operation occurs very quickly, regardless of the object's size. The hidden object is physically deleted from the database once you take any action other than the Undo Deletion action.

Objects are deleted this way so that you can undo an accidental deletion, unless some other action has been taken that then makes "undoing" an object deletion impossible. Microsoft Access can undo commands only one level back; it cannot queue up multiple commands to undo actions.

Since the physical object deletion occurs on the next action taken, instead of when Delete is chosen, your computer may appear to be hung while it is performing the delete. If the object is large enough, the deletion can take some time.

NOTE: If you reboot your computer before the deletion is completed, the hidden temporary object remains in the database, wasting space. To remove hidden objects like this, 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 does 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.

Additional reference words: 1.00 1.10 2.00 Tables Deleting Erasing

KBCategory: kbusage

KBSubcategory: Gn1Othr

## **INF: How Microsoft Access Uses SHARE.EXE**

**Article ID: Q113355**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article describes the SHARE.EXE file-sharing utility, and how it is used by Microsoft Access to handle file sharing for multiuser databases.

### MORE INFORMATION

=====

#### Description of SHARE.EXE

-----

SHARE.EXE (Share) is a file-sharing utility included with MS-DOS versions 3.0 and higher. It is used by application programs that work with several files at a time. It is essential for sharing files on a network with multiple users.

In MS-DOS, Share is a terminate-and-stay-resident (TSR) utility. If you are running Windows for Workgroups version 3.11 in enhanced mode, Share is replaced by an enhanced mode virtual device driver (VxD) called VSHARE.386 (VShare). VShare was designed for Windows for Workgroups and the peer-to-peer networking it supports. Since Windows for Workgroups 3.11 uses only VShare, you can conserve memory by not loading Share. Consult your Windows for Workgroups documentation for more information about using VShare.

Since Share and VShare are functionally identical for Microsoft Access, only Share will be described further in this article.

Share gives applications an easy, well-defined way to prevent users from accessing the same file (or region of a file) at the same instant. Once Share is run, an application can use it to lock a file (or region of a file) so that only one person at a time can make changes to it, preventing file read and write conflicts. Most multiuser and network software packages use Share for their file and record locking tasks.

Microsoft Access uses Share to control file sharing between two or more instances of Microsoft Access running on the same system and using the same database, or to allow a workstation to run as a non-dedicated file server in a Windows for Workgroups environment. Share is also used by certain object linking and embedding (OLE) applications. OLE applications use Share to coordinate file usage, and OLE itself opens files using Share. Also, all of the installable ISAM drivers rely on Share.

#### How Share Works

-----

Share maintains two tables in memory: the FILES table and the LOCK table. The FILES table contains the complete pathname of each file that has been

opened, plus an internal file handle number and other housekeeping information. The LOCK table contains a list of internal file handle numbers and corresponding information on the various areas of each file that are locked. Share checks these tables whenever an application attempts to open or use a file or a region of a file, and tells the application whether or not the file or region is available.

Share uses at least one entry in each table for each file that is opened. The more files your computer opens and locks, the more space Share needs for its internal tables. You can control the size of Share's internal tables with two command line options: /F and /L. The /F option controls the amount of space allocated for the FILES table and the /L option controls the number of simultaneous locks that Share will allow.

#### The /F Parameter

-----

The /F parameter controls the size (in bytes) of the table that Share reserves for filenames and handles. The syntax for the /F parameter is

SHARE /F:<n>

where <n> is any number from 0 to approximately 62,000 (depending on the other applications loaded). The default is /F:2048 (or, 2K). Share stores the complete pathname of each file, plus 11 bytes of file handle and housekeeping information. You can figure the largest possible space requirement by multiplying the number of files in your CONFIG.SYS file by 71 (60 bytes for the longest possible pathname + 11 bytes for other information). For example, if your system has FILES = 255 in your CONFIG.SYS file, in the worst case (all 255 files are open) Share would require over 18,100 bytes for the FILES table.

On a network server, you must allocate enough space for all the files that will be opened by all the users on the network. The default value is 2048 bytes--enough space to hold the information for 66 files with paths averaging 20 characters, or about 28 files in a worst-case scenario. If you know that paths on your system average more than 20 characters or that you will be opening many files, you should use the /F parameter to give Share more space for the FILES table.

#### The /L Parameter

-----

The /L parameter controls the number of simultaneous locks that Share can handle. A low number of Share locks is a common problem for Microsoft Access users on networks. The syntax for the /L parameter is

SHARE /L:<n>

where <n> is an integer between 1 and approximately 3800 (depending on the size of other applications loaded). The default is 20 locks.

Opening a file on a server requires at least one lock. In addition, most network programs use several more locks per file. They lock individual records, and even individual fields within records. Most multiuser databases use many locks--sometimes 10 or more per file. On a network, with several users opening each file, Share's 20-lock default can be used up

almost instantly. The number of Share locks used by Microsoft Access varies, depending on the amount of data that is accessed. A rough estimate of the number of locks needed is 5 times the number of records that will be accessed at any given time. During the Microsoft Access installation process, the SETUP.EXE program automatically sets the number of locks to 500.

Use the /L parameter to increase the number of locks allowed to prevent Share lock errors. The /L setting should be at least the number of files you have specified in your CONFIG.SYS file. If you are running a multiuser program that opens many files, consider setting /L to at least twice the number of open files allowed, or in the case of Microsoft Access, 5 times the number.

#### Questions and Answers About Share and Microsoft Access

---

1. Q: Does Microsoft Access use Share if you open a database exclusively?

A: Yes. Share takes over all File Open calls.

2. Q: Do any of the ODBC drivers (such as those in the Drivers Pack, or third-party drivers) require Share?

A: Yes, all of the ODBC drivers require Share, whether in a multiuser or a stand-alone configuration.

3. Q: What is the highest number of locks Microsoft Access might need?

A: This is impossible to determine exactly, since it depends on the amount of data that is accessed at any given time. However, a rough estimate is 5 times the number of records that will be accessed at any given time.

4. Q: What memory does Share use?

A: Share uses conventional memory, while VShare uses memory from the Windows memory pool (extended, virtual, and so on). You do not need to run both Share and VShare, because VShare disables Share.

5. Q: Do Share and VShare compete with other applications for memory?

A: Share is an MS-DOS application that is loaded into the conventional memory area, so it does compete with other programs that use conventional memory. Also, Share requires a small amount of memory to store file and lock information. VShare is a Windows-managed VxD, so it can be allocated in any of the available Windows memory areas and does not impact other applications.

6. Q: Do transactions use Share?

A: Yes.

7. Q: Why do I sometimes get a "Share locks exceeded" message, while at other times I get an "Out of memory" error message?

A: If you have a Share locking problem, you get an error message identifying Share as the problem. An "Out of memory" message can be caused by a number of other things.

Additional reference words: 2.00 MU Multiuser Sharing

KBCategory: kbusage

KBSubcategory: GnlMu



## INF: How to Activate an Application with Class Name

Article ID: Q104710

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

The function AppActivate() activates an application using the title bar text of the application. You must use the exact title bar text or the function fails. For example, the title bar text of Microsoft Access is "Microsoft Access" unless an object (a form or report, for example) is maximized. If an object is maximized, the title bar text is "Microsoft Access - <Object Name>." Since knowing the exact title bar text ahead of time can be challenging, the following function demonstrates how to activate an application using the application's Class Name and Microsoft Windows API calls.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access.

### 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 Microsoft Access window has the class name "OMain."

You can determine the class name of an application by using SPY.EXE, supplied with the Microsoft Windows Software Development Kit (SDK) version 3.1.

To use the application's Class Name to activate the application, create a new module with the following Declaration section and function:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character to improve readability. Remove the underscore when re-creating this code in Access Basic.

NOTE: You may have some Microsoft Windows API functions defined in an existing Microsoft Access library or module; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.

```
'*****
' MODULE DECLARATION SECTION
'*****
Option Compare Database 'Use database order for string comparisons
Option Explicit
```

```
'Windows API Declarations
Declare Function ShowWindow% Lib "User" (ByVal hWnd%, ByVal nCmdShow%)
Declare Function FindWindow% Lib "user" (ByVal lpclassname As Any, _
 ByVal lpCaption As Any)
Declare Function SetActiveWindow% Lib "User" (ByVal hWnd%)
Const SW_SHOW = 9
```

```

' FUNCTION: AppActivateClass()
'
' PURPOSE: Used to activate another application by using its
' class name instead of the title bar text as AppActivate does.
'
' ARGUMENTS:
' lpClassName$ - The Class Name of the application.
'
' RETURNS:
' 0 - The application was not found - it either was not running
' or the class name is incorrect
' -1 - The application was activated

```

```
Function AppActivateClass (lpclassname$)
 Dim hWnd% 'the application's window handle
 Dim dummy 'Dummy variable

 'Get the Window Handle
 hWnd% = FindWindow%(lpclassname$, 0&)

 'Activate the Application
 dummy = SetActiveWindow%(hWnd%)
 dummy = ShowWindow%(hWnd%, SW_SHOW)

 'Return True if Application Running, or False if not.
 AppActivateClass = hWnd%
```

End Function

To use this function, call the function and supply the class name of the application to activate. The following example shows how to activate Microsoft Word for Windows:

```
x=AppActivateClass("opusapp")
```

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      | NOTEPAD.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    |                           |

Additional reference words: 1.00 1.10 2.00 WinIss ab caption  
KBCategory: kbprg  
KBSubcategory: PgmApi

**INF: How to Add a User to a Group with CreateUser Method**  
**Article ID: Q112063**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

To add a user to an existing group with Access Basic code, you must use the CreateUser method twice: once to create the new user for that Microsoft Access session, and again to create the user for the group.

MORE INFORMATION

=====

The following sample Access Basic function called CUser() demonstrates how to create a user named MyName for the Users group:

1. Open or create a module and enter the following:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function CUser (uname As String, upin As String, upwd As String, _
 gname As String)
 Dim u As User, w As Workspace, g As Group
 Set w = dbengine.workspaces(0) 'The default workspace.
 Set u = w.CreateUser(uname, upin, upwd) 'Create a user for the
 'Access session.
 w.users.Append u 'Add user to workspace.
 w.users.Refresh
 Set g = w.groups(gname)
 Set u = g.CreateUser(uname) 'Create a user for a
 group.
 g.users.Append u 'Add new user to group.
 g.users.Refresh
End Function
```

2. From the View menu, choose Immediate Window.
3. Type the following in the Immediate window, and then press ENTER:

```
? CUser ("MyName","1234","MyName","Users")
```

4. From the Security menu, choose Permissions, and then select the List Users option button.

Note that the new user appears in the Users/Group Name box.

REFERENCES

=====

For more information about the CreateUser method, search for "CreateUser" then "CreateUser Method" using the Microsoft Access Help menu.

Additional reference words: 2.00 Programming AB

KBCategory: kbprg

KBSubcategory: PgmObj

## INF: How to Add an Icon to Program Manager Using DDE

Article ID: Q109394

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a sample Access Basic function called CreatePROGMANIcon() that you can use to create an icon in the Microsoft Windows Program Manager using dynamic data exchange (DDE).

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

### MORE INFORMATION

=====

To set up the CreatePROGMANIcon() function, create a new module and enter the code below.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

#### Option Explicit

```
'*****
' FUNCTION: CreatePROGMANIcon
'
' PURPOSE:
' To create an icon in the Windows Program Manager.
'
' ARGUMENTS:
' CommandLine - The command line argument to execute
' when the icon is double-clicked.
' IconText - The text to appear under the icon.
' GroupName - The name of the group to place the
' icon in.
' RESULT:
' An icon is placed in the specified group. If the
' group does not exist, a new group is created.
'
' *****
Function CreatePROGMANIcon (CommandLine$, IconText$, GroupName$)
 Dim ChanNum As Integer
 Dim Groups As String
 Dim Exe As String
```

```

' Begin a DDE conversation with Program Manager.
ChanNum = DDEInitiate("PROGMAN", "PROGMAN")

' Request a tab delimited list of Program Manager groups.
Groups = DDERequest(ChanNum, "Progman")

' See if the requested group exists in the list.
' If not, create the group.
If Not InStr(1, GroupName, Groups) Then
 DDEExecute ChanNum, "[CreateGroup(" & GroupName & ")]"
End If

' Add an icon to the group with the specified text underneath.
Exe = "[AddItem(" & CommandLine & ", " & IconText & ",,)]"
DDEExecute ChanNum, Exe
DDETerminate ChanNum

End Function

```

#### How to Use the CreatePROGMANIcon() Function

---

Type the command below in an Immediate window and then press ENTER.

NOTE: In the following example, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this example.

```

? CreatePROGMANIcon("C:\WINDOWS\notepad.exe C:\AUTOEXEC.BAT", _
 "AUTOEXEC", "MAIN")

```

The sample command above uses the CreatePROGMANIcon() function to add an icon to the Main group in Program Manager. The icon will open the Windows Notepad text editor and load the C:\AUTOEXEC.BAT file into Notepad.

Note that when the icon is created, Program Manager uses the default icon supplied with the application specified in the command line. The following function is a modified version of the CreatePROGMANIcon() function with two additional arguments. These arguments are IconFile, which specifies the name of a file containing custom icons, and IconNum, which specifies which icon in the IconFile to use. Note that if you specify "", or null, for both of these arguments, the default icon will be used.

```

Function CreatePROGMANIcon2 (CommandLine$, IconText$, GroupName$, _
 IconFile$, IconNum$)

 Dim ChanNum As Integer
 Dim Groups As String
 Dim Exe As String

 ' Begin a DDE conversation with Program Manager.
 ChanNum = DDEInitiate("PROGMAN", "PROGMAN")

 ' Request a tab delimited list of Program Manager groups.
 Groups = DDERequest(ChanNum, "Progman")

 ' See if the requested group exists in the list.

```

```

' If not, create the group.
If Not InStr(1, GroupName, Groups) Then
 DDEExecute ChanNum, "[CreateGroup(" & GroupName & ")]"
End If

' Add an icon to the group with the specified text underneath.
Exe$ = "[AddItem(" & CommandLine & ", " & IconText & ", _
 " & IconFile & ", " & IconNum & ")]"
DDEExecute ChanNum, Exe
DDETerminate ChanNum

End Function

```

The following sample command uses the fourth icon in the MORICONS.DLL file that is supplied with Windows. This file contains multiple custom icons:

```

CreatePROGMANIcon2("C:\TEST\TEST.EXE ", "My App", "My Group", _
 "C:\WINDOWS\MOREICONS.DLL", "4")

```

The next sample command uses an icon in an .ICO file. Since there is only one icon in each .ICO file, leave the IconNum argument empty:

```

CreatePROGMANIcon2("C:\TEST\TEST.EXE ", "My App", "My Group", _
 "C:\ICONS\MY.ICO", "")

```

#### REFERENCES

=====

Windows Software Development Kit "Guide to Programming," pages 19-22

For more information on DDE and Program Manager, query on the following words here in the Microsoft Knowledge Base:

dde and progman

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr



## **INF: How to Add ToolTips to Form Controls**

**Article ID: Q119991**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article describes how to add ToolTips to form controls. ToolTips refers to the "balloon help" feature that displays the name of a control when the user passes the mouse pointer over a particular control.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

The following steps demonstrate how to create a pop-up form to display ToolTips for a form control, and how to create an Access Basic module with procedures for displaying and hiding the ToolTips form.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Create a new, unbound form with the following properties:

```
RecordSource: <blank>
ScrollBars: Neither
RecordSelectors: No
NavigationButtons: No
PopUp: Yes
BorderStyle: None
Width: 4 in
OnTimer: [Event Procedure]
```

```
OnTimer Event Procedure
```

```

```

```
Sub Form_Timer ()
 ShowToolTips Me
End Sub
```

NOTE: To create the event procedure, click in the OnTimer property field, then choose the Build button to the right of the field. Select Code Builder, then choose OK. After you enter the code in the form module, close the module.

2. Add a text box with the following properties to the form:

```
Name: TipText
```

```
Left: 0
Top: 0
Width: 4 in
Height: 0.166 in
BackColor: 8454143
BorderStyle: Clear
FontName: MS Sans Serif
FontSize: 8
TextAlign: Center
OnMouseMove: [Event Procedure]
```

```
OnMouseMove Event Procedure
```

```

Sub TipText_MouseMove (Button As Integer, Shift As Integer, _
X As Single, Y As Single)
 HideToolTips
End Sub
```

3. Delete the text box's label.
4. Set the detail section's Height property to:  
0.166 in
5. Save the form as ToolTips and then close it.
6. Create a new module and enter the following lines in the Declarations section:

```
Option Explicit

' Tip Delay (time to wait to display tip) constants (in
' milliseconds)
Const TipDelayIfHidden = 1000
Const TipDelayIfVisible = 100

' Tip size and placement adjustment constants
Const AdjustTipWidth = .75
Const AdjustTipWidthPixel = 5
Const AdjustTipLeft = .3
Const AdjustTipTop = .6

' Windows API constants and Declarations
Type POINTAPI
 X As Integer
 Y As Integer
End Type

Type RECT
 Left As Integer
 Top As Integer
 Right As Integer
 Bottom As Integer
End Type

Global Const SM_CXCURSOR = 13
Global Const SM_CYCURSOR = 14
```

```

Global Const SW_SHOWNOACTIVATE = 4
Global Const TwipsPerPixel = 15

Declare Sub GetCursorPos Lib "User" (lpPoint As POINTAPI)
Declare Sub GetWindowRect Lib "User" (ByVal hWnd%, lpRect As RECT)
Declare Sub MoveWindow Lib "User" (ByVal hWnd%, ByVal X%, _
 ByVal Y%, ByVal nWidth%, ByVal nHeight%, ByVal bRepaint%)
Declare Function GetDC Lib "User" (ByVal hWnd%) As Integer
Declare Function GetSystemMetrics Lib "User" (ByVal nIndex%) _
 As Integer
Declare Function GetTextExtent Lib "GDI" (ByVal hDC%, _
 ByVal lpString As String, ByVal nCount%) As Long
Declare Function ReleaseDC Lib "User" (ByVal hWnd%, ByVal hDC%) _
 As Integer
Declare Function ShowWindow Lib "User" (ByVal hWnd%, _
 ByVal CmdShow%) As Integer

' ToolTips variables to store tip information
Global TipPoint As POINTAPI
Global TipText As String
Global TipTextLast As String

```

7. Create the following four Access Basic procedures in the module:

```

Sub ToolTips (MyTipText)
' *****
' PURPOSE: Sets up ToolTips for view after a delay of
' TipDelayIfHidden or TipDelayIfVisible milliseconds
' using the ToolTips form's Timer event.
' USAGE: Call this procedure from the MouseMove event of any
' control for which you want to display ToolTips.
' PARAMETERS:
' MyTipText: The ToolTips caption text.
' *****
 Dim Tip As Form

 ' Get tip form and open if not yet opened.
 On Error Resume Next
 Set Tip = Forms!ToolTips
 If Err Then StartToolTips

 ' If the tip is already visible with the desired
 ' text, then exit.
 If Tip.Visible And TipText = MyTipText Then Exit Sub

 ' Record new tip information.
 TipTextLast = TipText ' Save last tip.
 TipText = MyTipText ' Save new tip text.
 GetCursorPos TipPoint ' Get and save the current mouse
 ' pointer position.

 ' Set new tip delay (time to wait before displaying the tip).

 ' If the tip is NOT visible...
 If Not Tip.Visible Then
 ' ...then set the standard delay to display the tip
 ' from a hidden state.

```

```

 Tip.TimerInterval = TipDelayIfHidden
Else
 ' ...otherwise the pointer is moving from tip to tip, so
 ' set a shorter delay. Set a shorter delay only if the
 ' shorter delay has not yet been set.
 If Tip.TimerInterval <> TipDelayIfVisible Then
 Tip.TimerInterval = TipDelayIfVisible
 End If
End If
End Sub

```

```

Sub HideToolTips ()
' *****
' PURPOSE: Hides ToolTips from view.
' USAGE: Call from the MouseMove event of all form sections
' that contain ToolTips-enabled controls.
' OPTIONAL (but recommended): Call from the MouseMove event
' of all form sections on all forms and from MouseMove
' events of controls that are not ToolTips-enabled.
' CALLED FROM: MouseMove event of the TipText text box, the
' ShowToolTips procedure, the MouseMove event of form
' sections, and non-ToolTips-enabled controls.
' *****
 Dim F As Form

 ' Get tip form and open if not yet opened.
 On Error Resume Next
 Set F = Forms!ToolTips
 If Err Then StartToolTips

 ' Hide tip and turn delay off.
 F.Visible = False
 F.TimerInterval = 0
End Sub

```

```

Sub ShowToolTips (Tip As Form)
' *****
' PURPOSE: Displays ToolTips caption.
' CALLED FROM: ToolTips form Timer event only.
' *****
 Dim P As POINTAPI
 Dim R As RECT
 Dim hDC As Integer
 Dim RetVal As Integer
 Dim TipLeft As Integer, TipTop As Integer
 Dim TipWidth As Integer, TipHeight As Integer

 ' Get the current mouse pointer position.
 GetCursorPos P

 ' If displaying the tip from a hidden state was delayed...
 If Tip.TimerInterval = TipDelayIfHidden Then
 ' ...and the mouse pointer position does not match the
 ' pre-delay position...
 If Not (P.X = TipPoint.X And P.Y = TipPoint.Y) Then
 ' ...then hide the tip and exit.
 HideToolTips
 End If
 End If
End Sub

```

```

 Exit Sub
 End If
End If

' Turn the tip delay off.
Tip.TimerInterval = 0

' Compute approximate TipWidth.
hDC = GetDC(0)
TipWidth = GetTextExtent(hDC, TipText, Len(TipText)) And &HFFFF&
RetVal = ReleaseDC(0, hDC)
TipWidth = (TipWidth * AdjustTipWidth) + AdjustTipWidthPixel

' Compute TipHeight.
GetWindowRect Tip.hWnd, R
TipHeight = R.Bottom - R.Top

' Compute TipLeft.
TipLeft = P.X + (GetSystemMetrics(SM_CXCURSOR) * AdjustTipLeft)
TipLeft = TipLeft - (TipWidth / 2)

' Compute TipTop.
TipTop = P.Y + (GetSystemMetrics(SM_CYCURSOR) * AdjustTipTop)

' Hide tip form.
Tip.Visible = False

' Set tip text.
Tip!TipText = TipText

' Set tip text box width.
Tip!TipText.Width = TipWidth * TwipsPerPixel

' Move and size tip form.
MoveWindow Tip.hWnd, TipLeft, TipTop, TipWidth, TipHeight, False

' Show tip form.
RetVal = ShowWindow(Tip.hWnd, SW_SHOWNOACTIVATE)
End Sub

Private Sub StartToolTips ()
' *****
' PURPOSE: Opens the ToolTips form.
' CALLED FROM: HideToolTips, ShowToolTips procedures only.
' *****
 DoCmd OpenForm "ToolTips", , , , , A_HIDDEN
End Sub

```

8. Save the module as ToolTips.

How to Use the ToolTips Form and ToolTips Procedures

---

To enable ToolTips for a control on a form, follow these steps:

1. Call the ToolTips procedure from the control's MouseMove event, supplying the text for the tip as an argument. For example, the

MouseMove event procedure for a command button called SaveRecord would look like:

```
Sub SaveRecord_MouseMove (Button As Integer, Shift As Integer, _
 X As Single, Y As Single)
 ToolTips "Save Record"
End Sub
```

2. Call the HideToolTips procedure from the MouseMove event of form sections holding any ToolTips-enabled controls. For example, the MouseMove event of a form detail section containing the SaveRecord button from the above example would look like:

```
Sub Detail0_MouseMove (Button As Integer, Shift As Integer, _
 X As Single, Y As Single)
 HideToolTips
End Sub
```

NOTE: You must call the HideToolTips procedure to dismiss ToolTips. To be sure that tips are dismissed when unnecessary, you should call the HideToolTips procedure from the MouseMove event of all controls that are not ToolTips-enabled, as well as from all form section MouseMove events. You should also call this procedure from forms that do not contain ToolTips, but that may be running in conjunction with ToolTips-enabled forms.

For example, if you have two buttons next to each other on a form, and one of the buttons is ToolTips-enabled and the other is not, ToolTips will not be dismissed if the mouse pointer passes from the ToolTips-enabled button to the other button unless the other button calls the HideToolTips procedure from its MouseMove event.

If you plan to use ToolTips for buttons on a toolbar, make sure to leave some part of the section surrounding the buttons visible, so that the section's MouseMove event can call the HideToolTips procedure.

#### Fine-Tuning the Size and Placement of ToolTips

---

The Declarations section of the ToolTips module contains some constants that you can use to adjust the size, placement, and behavior of the ToolTips form.

##### ToolTips Delay Constants:

The TipDelayIfHidden and TipDelayIfVisible constants specify the amount of time (in milliseconds) to wait before the ToolTips form is displayed. The TipDelayIfHidden constant specifies how long to wait if the mouse pointer is positioned over a ToolTips-enabled control and the tip is not yet visible. The TipDelayIfVisible constant specifies how long to wait if the mouse pointer is being moved from one visible tip to another.

##### ToolTips Size Constants:

The width of the ToolTips form is based on the width of the tip text, and is approximated using the GetTextExtent Windows API call and the AdjustTipWidth and AdjustTipWidthPixel constants. To increase the amount of

empty space to the left and right of the centered tip text, increase the value of the constants. To decrease the amount of empty space, decrease the constants.

NOTE: Tips with a number of wide letters (like "M" and "W") using a proportionally-spaced font may be truncated if the values of the AdjustTipWidth and AdjustTipWidthPixel constants are too small. Tips with a number of narrow characters (like "i" and "j") may contain too much empty space if these constants are too small.

#### ToolTips Position Constants:

The AdjustTipLeft and AdjustTipTop constants specify the position of the ToolTips in relation to the mouse pointer. Increase these values to position the ToolTips lower and further to the right. Decrease these values to position the ToolTips higher and further to the left. Note that you can use negative numbers for these values.

#### Closing the ToolTips Form

-----

When you are working with and debugging the ToolTips form and ToolTips code, the ToolTips form may get "stuck" on. If this happens, you will have to close the running, hidden ToolTips form manually. To close the form, run the following statement in the Immediate window:

```
DoCmd Close A_FORM, "ToolTips"
```

#### REFERENCES

=====

For more information about using ToolTips in Microsoft Access, search for "ToolTips: using" then "Displaying Information about a Toolbar or its Buttons" using the Microsoft Access Help menu.

Additional reference words: 2.00 tool

KBCategory: kbusage

KBSubcategory: FmsHowto

## **INF: How to Attach All the User Tables on a SQL Server**

**Article ID: Q123714**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article describes how to use a SQL pass-through query to obtain a list of all the user tables located on a SQL Server and then attach those tables.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

To create a SQL pass-through query to list and attach all the tables on a SQL Server, follow these steps:

1. Create a new query that is not based on any table.
2. From the Query menu, choose SQL Specific and then choose Pass-Through.
3. In the SQL Pass-Through Query window, type the following line:

```
SELECT Name FROM SysObjects Where Type ='u';
```

NOTE: If the SQL Server you will be using was installed with case sensitivity, you may have to type "Name" and "u" in uppercase letters.

4. From the View menu, choose Properties. Enter a complete ODBC connect string in the ODBCConnectStr property, or choose the Build button to select an ODBC data source. The following is an example of a valid ODBC connect string:

```
ODBCConnectStr: ODBC;DSN=opus;UID=john;PWD=steelers;DATABASE=pubs
```

5. Run the query to make sure that it returns one column called Name that lists the tables on the SQL Server.
6. Save the query as SQL\_Tables and then close it.
7. Create a new module and then enter the following line in the Declarations section:

```
Option Explicit
```

8. Enter the following code in the module.



NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Function attach_all ()
 Dim DB As Database
 Dim RS As Recordset
 Dim QD As QueryDef

 Set DB = DBEngine.Workspaces(0).Databases(0)
 ' Set the QueryDef variable to the pass-through query.
 Set QD = DB.OpenQueryDef("sql_tables")
 Set RS = DB.OpenRecordset(QD.name, DB_OPEN_SNAPSHOT)

 Do Until RS.EOF
 DoCmd TransferDatabase A_ATTACH, "<Sql database>", _
 QD.connect, A_TABLE, RS!Name, RS!Name
 RS.MoveNext
 Loop

 RS.Close
 QD.Close

End Function
```

9. To test the function, choose View from the Immediate window, type the following line in the Immediate window, and then press ENTER:

```
? attach_all()
```

All the tables on the SQL Server will be attached to the current database. The amount of time this process requires depends on the number of tables on the SQL Server.

#### REFERENCES

=====

For more information about SQL pass-through queries, search for "pass-through query" then "Creating a SQL Pass-Through Query" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: ObcSqlms

**INF: How to Automatically Detect If a Form Is Being Edited**  
**Article ID: Q122294**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes a sample function that you can use to determine whether a record on a form is being edited. You can use this method when you want different controls available on a form when a record is being edited from those available when the record is not being edited.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

When a pencil is displayed in the record selector, the record is being edited but has not yet been saved. When a triangle is displayed in the selector, the record has been saved and is not being edited. (Note: If the form's RecordSelector property is set to No, the symbols described here will not be displayed.)

The following steps demonstrate how to detect when a record is being edited on a form and enable an Undo Record button as appropriate:

1. Open the sample database NWIND.MDB.
2. Open the Employees form in Design view.
3. Create a text box with the following properties on the form:

```
Name: txtEditModeChange
ControlSource: =[Form].[Dirty] & EditModeChange([Form])
Visible: No
```

4. From the View menu, choose Code.
5. Type the following line in the form module's Declarations section:

```
Option Explicit
```

6. Create the following function in the module and then close the module:

```
Function EditModeChange (F As Form) As Variant
 If Me.Dirty Then
 Me!btnUndo.Enabled = True
 Else
 Me!btnUndo.Enabled = False
 End If
End Function
```

End If  
End Function

7. Use the Command Button Wizard to create an Undo Record button on the form.
8. Change the Undo Record button's properties to match the following properties:  
  
    Name: btnUndo  
    Enabled: No
9. View the form in Form view. Note that the Undo Record button is unavailable.
10. Modify any field in the current record and then press the TAB key. Note that the Undo Record button becomes available.
11. Press the ESC key twice or choose Undo from the Edit menu to undo the change you made to the record. Note that the Undo Record button becomes unavailable.

#### REFERENCES

=====

For more information about the Dirty property, search for "Dirty" then "Dirty Property" using the Microsoft Access Help menu.

Additional reference words: 2.00 onundo ab ondirty edit mode

KBCategory: kbusage

KBSubcategory: FmsEvt

## **INF: How to Avoid Abandoned Group Headers in Reports**

Article ID: Q100920

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a routine to keep a given number of records together on a page with their group header. The article also explains how to avoid an abandoned group header at the bottom of a page and how not to print the group header beyond a certain position on the page in a single-column or multiple-column report.

NOTE: The new KeepTogether property for groups in Microsoft Access version 2.0 gives you the ability to keep groups of like information together. This property is available in the Sorting And Grouping dialog box for reports. However, this property only prevents page breaks from occurring within a group, it does not prevent column breaks. To prevent column breaks from occurring within a group use one of the methods described in this article.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

### MORE INFORMATION

=====

The KeepTogether property is used to keep all of the controls for one record together in a particular section of a report. This property does not keep multiple records together, nor does it keep a group header and its detail records together.

Method 1 below helps you to find the location on the page beyond which you should not print a group header if you want to keep a given number of records together with that header.

Method 2 below helps you to determine the twip location by trial and error and describes two sample functions, one for a single-column report and the other for a multi-column report.

#### Method 1

-----

To keep a set number of records together on a single page with the group header, you need to know approximately how many records will be printed per group, or, more importantly, how many records you want to print with a particular group header. You can preview your report to find the group that prints the most records to determine how many records to keep with the group header.

For example, say you have determined that the largest group prints 24 records. The detail section of your report is 1/4 inch in height; thus, 24 detail records take up 6 inches on a page. You must also consider the top and bottom margins, which default to 1 inch each, and the height of your group header section.

Using the Report.Top property, you can determine where on the page the next section will print. If enough space remains to print your group header, all of its detail records, and the page footer (if any), plus the bottom margin, your group header will be printed on the same page as its detail records. If there is not enough blank space, delay printing the report (use a combination of the MoveLayout, NextRecord, and PrintSection properties to leave the rest of the page blank) and continue the group header information on the next page.

Below are sample measurements on which this article's calculations are based:

Page Length: 11 inches  
Detail.Height: 0.25 inch

Add the following values together:

Top Margin: 1 inch  
Page Header.Height: 1 inch  
Group Header.Height: 0.5 inch  
Group Footer.Height: 0.5 inch  
Page Footer.Height: 1 inch  
Bottom Margin: 1 inch  
-----  
Total = 5 inches

To determine the blank space left on the page for detail records, subtract the total above from your Page Length property value (11 inches - 5 inches = 6 inches).

Divide the maximum space for detail records by the Detail.Height property value (6 inches / 0.25 inch). A total of 24 detail records will fit on a page. Force a new page if fewer than 8 records will print with the header (for example, if the formula returns either less than the sum of 8 \* 0.25 inches or less than 2 inches of blank space).

To determine where the top of the 2-inch limit is, or the point beyond which you should not print the group header, add the following values together:

Top margin 1 inch  
+ Page Header.Height 1 inch  
+ Remainder of detail section (6 inches - 2 inches)  
  
= 1 inch + 1 inch + (6 inches - 2 inches)

Final calculation for the function: = 6 inches or (6 \* 1440 twips)

To implement this calculation in a module, use the following steps:

1. Enter the =PrintOK() function in the OnPrint property of your

group header. (This function assumes a report name of MyReport.)

2. Type the following lines in the Declarations section of a module:

```
Option Explicit

Function PrintOK()
 Dim R As Report
 Set R = Reports!MyReport
 If R.Top > (6 * 1440) Then
 R.MoveLayout = TRUE
 R.PrintSection = FALSE
 R.NextRecord = FALSE
 End If
End Function
```

Method 2

-----

Another way to avoid an abandoned group header at the bottom of a page, or not to print the group header beyond a certain position on the page, is to replace (6\*1440) in the function above with the twip location beyond which you do not want to print.

The report tests the Top property; if the condition is True, the report does not print the section. Instead, it moves down the report layout and attempts to print the section until the condition is False, which occurs at the top of the next page.

For a multi-column report, the function must be modified as follows:

```
Option Compare Database
Option Explicit
'=====
'
' Function to force to new column if at bottom of column.
' R is the name of the Report being formatted, as in
' Reports![ReportName], and
' Bottom is the value (in inches) at which a new column is forced.
' This function should be called from the OnPrint property of the Group
' Header.
'=====
Function IfBottom (R As Report, Bottom As Integer)
 Dim YPos
 Static LastPos

 YPos = R.Top
 If YPos > Bottom * 1440 Then 'Beyond this position, go to a new column.
 R.MoveLayout = True
 R.NextRecord = False
 R.PrintSection = (YPos = LastPos) 'Set to True when at Max position.
 LastPos = YPos
 End If
End Function
```

NOTE: This function will not work properly if Item Layout is set to Horizontal (in Print Setup) for the columns.

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbusage  
KSubcategory: RptSort

## **INF: How to Base Subforms on SQL Pass-Through Queries**

**Article ID: Q112746**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article describes how to use an SQL pass-through query for a form's record source. It assumes that you know how to build and use SQL pass-through queries.

The following considerations should be kept in mind when you are using SQL pass-through queries for forms' record sources:

- SQL pass-through query record sources are read-only.
- The Query Builder does not save the connect string unless you save the query.
- Subforms can use SQL pass-through queries as record sources only if you set the record source in the main form's OnLoad property. Subreports cannot use SQL pass-through queries as record sources.

### MORE INFORMATION

=====

#### SQL Pass-Through Queries Are Read-Only

-----

Forms based on SQL pass-through queries are read-only because SQL pass-through queries are read-only. The recordset returned by an SQL pass-through query is a snapshot, or read-only recordset. This behavior is by design. In order for the form to be updateable, base your form on an attached table with a unique index.

#### The Query Builder Does Not Save the Connect String

-----

The RecordSource property's Query Builder displays a window that looks similar to a query's Design window. You use this window to build the SQL string or query for the RecordSource property. The Query Builder will set the RecordSource property to an SQL string if the SQL string is not saved as a query. If you save the string as a query, the name of the query will be used as the RecordSource property.

When you are using an SQL pass-through query created using the Query Builder, the ODBC connect string defined in that query will not be returned as part of the RecordSource property SQL string. This can produce the error message "Couldn't find input table or query '[Table from ODBC Server]'" because without the ODBC connect string, the form will look for a local table, instead of a table on the server. If you save the SQL string as a query, the RecordSource property will contain the name of the query instead



of the SQL string, and the form will be able to retrieve the remote data correctly.

#### Subforms Cannot Directly Use SQL Pass-Through Queries as Record Sources

---

The best way to use server-based data in a subform is to attach the remote table and then base the subform on the attachment.

If you base a subform directly on an SQL pass-through query, you will receive the following error message when you open the form:

The crosstab query underlying a subform or subreport must have fixed column headings.

If you have to use an SQL pass-through query as a subform's record source, you can dynamically set the subform's RecordSource property with the main form's Load event. The following is an example of an OnLoad property event procedure in the main form MainForm that dynamically sets the RecordSource property in the subform SPTSubForm:

```
Sub MainForm_OnLoad
 Me!SPTSubForm.Form.RecordSource = "[<Name of SQL pass-through query>]"
End Sub
```

The subform itself should be saved without a RecordSource property. Note that the subform will not return data if the LinkMasterFields and LinkChildFields properties are set. If you need to filter the records in the subform based on a value in the main form, either base your query on attached tables, or modify the QueryDef so the query includes the value in the main form.

#### REFERENCES

=====

For more information about SQL pass-through queries, search for "pass-through query" then "Creating a Pass-Through Query" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsProp

**INF: How to Bring a Subtotal from a Subform to a Main Form**  
**Article ID: Q119134**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article demonstrates how to correctly reference a subtotal control on a subform from a main form.

MORE INFORMATION

=====

If you try to use an expression similar to

```
=Sum([MySubForm].Form![Extended Price])
```

on a main form to sum a reference to a subform control that contains a subtotal, you will receive a "#Name?" error message.

Instead, place the sum expression in the subform's form footer, and then reference that expression from the main form.

The following example demonstrates how to reference a control containing a subtotal on a subform from a main form:

1. Open the sample database NWIND.MDB.
2. Create a new form based on the Order Details Extended query. This form will be the subform.
3. Place three text box controls with the following properties in the form's detail section:

```
Text Box Control
 Name: Extended Price
 ControlSource: Extended Price
Text Box Control
 Name: Order ID
 ControlSource: Order ID
Text Box Control
 Name: Product Name
 ControlSource: Product Name
```

NOTE: In Microsoft Access version 1.x, the Name property is called the ControlName property.

4. Set the form's DefaultView property to Datasheet.
5. From the Format menu, choose Form Header/Footer.
6. Place a text box with the following properties in the form footer:

Text Box  
Name: SubTotal  
ControlSource: =Sum([Extended Price])

7. Save the form as MySubForm and then close it.
8. Create a form based on the Order Details table. Set the form's DefaultView property to Single Form. This form will be the main form.
9. Place two text boxes with the following properties in the form's detail section:

Text Box  
Name: Customer ID  
ControlSource: Customer ID  
Text Box  
Name: Order ID  
ControlSource: Order ID

10. Embed the MySubForm form in the main form's detail section by dragging it from the Database window to the main form. Set the subform control's properties as follows:

Subform/Subreport  
Name: MySubForm  
SourceObject: MySubForm  
LinkChildFields: Order ID  
LinkMasterFields: Order ID

11. Add a text box control with the following properties to the main form's detail section. This control will reference the subtotal on the subform:

Text Box  
Name: Order Subtotal  
ControlSource: =[MySubForm].Form![SubTotal]

12. Save the form as MyMainForm.

When you view the MyMainForm form in Form view, the subtotal from the subform will be displayed on the main form.

#### REFERENCES

=====

For more information about referencing subform controls, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q113352  
TITLE : INF: How to Refer to a Control on a Subform or Subreport

For more information about referring to controls on a subform or subreport, search for "subforms: referring to controls" then "Referring to Controls on a Subform or Subreport" using the Microsoft Access Help menu.

Microsoft Access "User's Guide," version 1.0, Chapter 11, "Using Expressions in Forms," pages 296-298

Microsoft Access "User's Guide," version 1.1, Chapter 11, "Using Expressions in Forms," pages 300-302

Microsoft Access "User's Guide," version 2.0, Chapter 18, "Using Expressions in Forms," pages 457-459

Additional reference words: 1.00 1.10 2.00 totals

KBCategory: kbusage kberrmsg

KBSubcategory: FmsSubf

## **INF: How to Calculate a Credit Card Expiration Date**

**Article ID: Q99941**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

If you enter a credit card expiration date (month/year) in a date/time field on a form, Microsoft Access assumes that the card expires on the first day of the month. This article includes a function that correctly calculates the expiration date as the last day of the month.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

### MORE INFORMATION

=====

Use the following function, `ExpirationDay()`, in the `AfterUpdate` property of an expiration text box on your form. The function will replace the date entered in the text box with the date computed for the last day of the month/year.

Given a date/time field bound to a text box with the `ControlName` property set to "Expiration", the `AfterUpdate` property should read:

```
=ExpirationDay([expiration])
```

Create a module with the following function:

```
Option Explicit
Function ExpirationDay (MyDate)
 Dim NextMonth
 If IsNull(MyDate) Then Exit Function
 NextMonth = DateAdd("m", 1, MyDate)
 MyDate = NextMonth - DatePart("d", NextMonth)
End Function
```

You can use the following derivative of the `ExpirationDay()` function in calculated fields in a query:

NOTE: In the following sample code, an underscore (`_`) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function ExpirationDay (MyDate)
 Dim NextMonth
 Is IsNull(MyDate) Then Exit Function
```

```
NextMonth = DateAdd("m", 1, MyDate)
ExpirationDay = NextMonth - DatePart("d", NextMonth)
End Function
```

Additional reference words: 1.00 1.10 2.00 date

KBCategory: kbprg

KBSubcategory: PgmOthr

## **INF: How to Calculate Averages Excluding Zero Values**

**Article ID: Q109364**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

When you use the Avg() function in a report to average a set of values, the function uses records containing zero values in the calculation. In some cases you may not want to include records with zero values in this calculation.

This article includes two examples of how to calculate an average for all the non-zero values in a set by counting the number of non-zero values in the set of values and then using that total with a running sum calculation.

### MORE INFORMATION

=====

#### Example One

-----

In this example, one text box displays the number of non-zero values, and the other text box displays the average for the set:

1. Open the sample database NWIND.MDB.
2. Use a Report Wizard to create a new Groups/Totals report based on the Order Details table. This report will calculate the average discount for each Product ID.
3. Include the Product ID and Discount fields on the report. Group the report on the Product ID field, grouped as Normal.
4. Open the report in Design view.
5. Create an unbound text box in the report's detail section. Place this text box to the left of the Discount text box. The new text box will display the count of non-zero values. Set the box's ControlSource property to:

```
=IIF([Discount]=0 or [Discount] is null,0,1)
```

This expression returns 0 if the value of the Discount field is equal to zero or null, otherwise it returns 1. The Avg() function automatically excludes nulls as well.

6. Set the text box's RunningSum property to Over Group, and its Name property to "Count of Data" (without quotation marks).

NOTE: In Microsoft Access version 1.x, the Name property is called the ControlName property.

7. Create another text box in the report's footer section. This text box will display the result of the average calculation.

8. Set the ControlSource property of this text box to:

```
=Sum([Discount])/[Count of Data]
```

9. Preview the report.

The left column displays a running count of non-zero (and non-null) Discounts and the group footer displays an average Discount based on the running count.

#### Example Two

-----  
This example uses a user-defined Access Basic function that is the functional inverse of the NullToZero() function in the NWIND database:

1. Open the NWIND database and create a new module. Enter the following line in the Declarations section of the module:

```
Option Explicit
```

2. Enter the following code in the module:

```
Function ZeroToNull(MyValue As Variant) As Variant
 If MyValue = 0 Or MyValue = Null Then
 ZeroToNull = Null
 Else
 ZeroToNull = MyValue
 End If
End Function
```

3. Repeat the example above, but use the following expression in place of the expression in step 5:

```
=ZeroToNull([Discount])
```

You can also use the ZeroToNull() function in a query.

#### REFERENCES

=====

Microsoft Access "Language Reference," version 1.1, pages 44, 248-249, 414-415, and 461-462

For more information about the Avg() function, search for "Avg" then "Avg Function" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 nonzero pure

KBCategory: kbusage

KBSubcategory: RptTotal



## INF: How to Call Functions Using a String Variable

Article ID: Q100164

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how Access Basic or user-defined functions can be called when the function name is stored in a string variable. This method provides a functionality similar to that of pointers to functions in other programming languages.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

### MORE INFORMATION

=====

Calling a function in Access Basic when the function name is stored in a variable allows significant programming flexibility. This method is described in the following procedure:

1. Store the function name and parameters in a string variable.
2. Use the Eval() function in Access Basic to evaluate the string. The process of evaluating the variable calls the function.
3. Open a new module or a previously created module and enter the following code:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----
'GLOBAL DECLARATIONS SECTION
'-----
Option Explicit

'-----
' The CallMyArray() function creates an array of strings, then
' loops, using the Eval() function, to call each element of the array.
'-----

Function CallMyArray ()
 Dim MyArray$()

 For i = 0 To 2
 ReDim Preserve MyArray$(i)
```

```
 MyArray$(i) = "MyFunc" & i & "(" & i & ")"
 Next i

 For i = 0 To 2
 x = Eval(MyArray(i))
 Next i
End Function
```

```
'-----
' The first function called by CallMyArray().
'-----
```

```
Function MyFunc0 (nParam)
 MsgBox "This is function: " & nParam
End Function
```

```
'-----
' The second function called by CallMyArray().
'-----
```

```
Function MyFunc1 (nParam)
 MsgBox "This is function: " & nParam
End Function
```

```
'-----
' The third function called by CallMyArray().
'-----
```

```
Function MyFunc2 (nParam)
 MsgBox "This is function: " & nParam
End Function
```

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.0, pages 7-14

Additional reference words: 1.00 1.10 2.00 string Eval

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: How to Change Default Template for Forms and Reports**  
**Article ID: Q95922**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Microsoft Access uses templates to define the characteristics of new forms and reports when Access Wizards are not used. These templates contain information about the default properties for the sections and controls the new form (or report) contains.

There are two templates: a Normal Form template and a Normal Report template. You can change the defaults used in these templates by defining new templates or by modifying the Normal templates.

MORE INFORMATION

=====

The information in the Normal templates used by Microsoft Access includes the following:

- Default section dimensions
- Default control properties
- Options to include page headers/footers

To change the default settings, either modify a report or form and save it as Normal, or rename the templates.

Note that the names you enter for the form and report templates must actually exist in your database. The SYSTEM.MDA file retains the template names that you define.

To change form or report template names:

1. Open a Microsoft Access database.
2. From the View menu, choose Options.
3. In the Category box, select Form & Report Design.
4. In the Items box, type the name of the form or report you want to use in the Form Template or Report Template field.

To use your templates with other databases, you must copy or export the templates.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, pages 267-268 (forms) and  
446-447 (reports)

Additional reference words: 1.00 1.10 2.00 templates

KBCategory: kbusage

KBSubcategory: FmrProp

**INF: How to Change the "Microsoft Access" Window Caption**  
**Article ID: Q95932**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

There is no Microsoft Access property to set the main caption of the Microsoft Access window. To change the default "Microsoft Access" caption to your own text, you must call the Windows application programming interface (API) SetWindowText() function.

MORE INFORMATION

=====

To change the default "Microsoft Access" caption to your own text string, create a new module in Microsoft Access and add the following function below with the appropriate declaration section.

Notes:

- You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.
- In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----
' GLOBAL DECLARATION
'-----
Option Compare Database
Option Explicit
Declare Function FindWindow% Lib "User" (ByVal lpClassName As Any, _
 ByVal lpWindowName As Any)
Declare Sub SetWindowText Lib "User" (ByVal hWnd%, ByVal lpString$)

'-----
' Function: SetCaption ()
' This function will set the caption of
' Microsoft Access's main window.
'-----
Function SetCaption ()
 Dim hWnd%
 hWnd% = FindWindow%("OMain", 0&)
 Call SetWindowText(hWnd%, "This is cool!")
End Function
```

To change the caption when Microsoft Access is started, you can execute the RunCode macro action with SetCaption() as an argument and

save the macro as Autoexec. When you open this database, this macro will execute and change the "Microsoft Access" caption to "This is cool!"

#### REFERENCES

=====

"Microsoft Windows Software Development Kit Programmer's Reference Volume 2: Functions," version 3.1

Additional reference words: 1.00 1.10 2.00 windows API

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: How to Change the Caption on the Microsoft Access Window**  
**Article ID: Q92684**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to use a function called from a macro to change the Microsoft Access window default title bar. If you name the macro Autoexec, the title bar will be changed automatically every time the database containing the macro is opened.

MORE INFORMATION

=====

The following steps demonstrate how to change the title bar caption:

1. Open a new database and name it TEST.MDB.
2. Open a new blank form. Place at least one control on the form. Save the form as MainForm and then close it.
3. Create the following new macro, save it as Macro1, and then close it:

| 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. Enter the following code for the ChangeAccessCaption() function in the module:

```
Function ChangeAccessCaption (Caption$)
 Dim X%
 Dim hParent%

 ' Example of how to set the caption bar
 DoCmd OpenForm "MainForm" 'Open the main startup form
 X% = Forms!MainForm.hwnd 'Get its handle
 hParent% = GetParent(GetParent(X%)) 'Get the form' parent
 Call SetWindowText(hParent%, Caption$)

End Function
```

6. Save the module as Module1 and then close it.
7. Run Macro1. The title for the Microsoft Access window should change

to the text you specified as the function parameter in step 3.

Additional reference words: 1.00 1.10 2.00 header caption

KBCategory: kbprg

KBSubcategory: PgmApi



## INF: How to Change the Font for the Zoom Box

Article ID: Q96996

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to modify the system UTILITY.MDA database to change the default font used in the Zoom box of Microsoft Access.

### MORE INFORMATION

=====

To change the font for the Zoom box, do the following:

1. Before you open Microsoft Access, copy the UTILITY.MDA file to MYUTIL.MDA. The UTILITY.MDA file is generally stored in your Microsoft Access directory.
2. Start Microsoft Access. From the File menu, choose Open Database, select MYUTIL.MDA, and choose the OK button.

NOTE: When you open this database, you are prompted to choose the OK button in response to several error messages. These errors are caused by duplicate modules loaded from the UTILITY.MDA database. The messages will be similar to the following:

Tried to load module with duplicate procedure definition: BuilderZoom

3. In Design view, open the ZoomForm form. Select the unbound text box control Zoom, and change the following properties to the ones noted below or to a font of your choice.

FontName: Microsoft Sans Serif  
FontSize: 12  
FontWeight: Semi-bold

NOTE: Select a font and a size that suit your needs.

4. Close the database and save your changes to the ZoomForm form. Exit Microsoft Access.
5. Use any ASCII text editor, such as NOTEPAD.EXE, to modify the MSACCESS.INI file in your Windows directory. You should modify the MSACC20.INI file if you are using Microsoft Access version 2.0.

Change the following line in your [Options] section from

```
[Options]
UtilityDB=C:\ACCESS\UTILITY.MDA
```

-to-

```
[Options]
;UtilityDB=C:\ACCESS\UTILITY.MDA
```

Add a new line so the section reads:

```
[Options]
;UtilityDB=C:\ACCESS\UTILITY.MDA
UtilityDB=C:\ACCESS\MYUTIL.MDA
```

6. Save the MSACCESS.INI file or MSACC20.INI file.

When you next start Microsoft Access and use the Zoom box for editing, the font, size, and weight reflect the changes you made.

Additional reference words: 1.00 1.10 2.00 zoombox font expressions editing  
KBCategory: kbusage  
KBSubcategory: GnlDe

## **INF: How to Change the Order of Columns in a Graph**

Article ID: Q109315

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

By default, a graph sorts the X-axis fields in alphabetical or numeric order. This article describes two ways to sort the X-axis fields in a different order.

### MORE INFORMATION

=====

There are two ways to change the ordering of fields in a graph. The first way is to add an Order By clause to the SQL statement in the graph's RowSource property. The second way is to create a query that orders the fields the way you want them, and then use the query for the graph's RowSource property.

#### How to Create a Sample Graph

---

For Microsoft Access version 2.0:

1. Open the sample database NWIND.MDB.
2. Create a new, blank form, and then add a graph object to the form.
3. In the GraphWizard dialog box, select the Sales By Category query as the data source for the graph.
4. Select the Category Name field in the Available Fields box and then choose the ">" button. Then select the Product Sales field and choose the ">" button. Click Next three times, and then click Finish.
5. View the form in Form view. Note that the Category Name records are listed in alphabetical order.

For Microsoft Access version 1.x:

1. Open the sample database NWIND.MDB.
2. Create a new, blank form, and then add a graph object to the form.
3. In the GraphWizard dialog box, select the Sales By Category query as the data source for the graph. Then, click the Bar Chart button, and then choose Next.
4. Select the Category Name field in the Available Fields box and then choose the ">" button. Then select the Product Sales field and choose the ">" button. Choose Next.

5. Choose Design.
6. View the form in Form view. Note that the Category Name records are listed in alphabetical order.

#### How to Change the Graph's Sorting Order

---

##### Method 1:

The following example demonstrates how to add an Order By clause to the SQL statement in a graph's RowSource property:

1. View the form in Design view.
2. Select the Graph.
3. From the View menu, choose Properties.
4. Select the RowSource property, and then press SHIFT+F2 to zoom the window. The RowSource property looks like:

```
SELECT DISTINCTROW [Category Name] AS [Sales by Category],
SUM([Sales by Category].[Product Sales]) AS [Product Sales]
FROM [Sales by Category]
GROUP BY [Category Name];
```

5. Type "Order By SUM([Sales by Category].[Product Sales]) desc" (without the quotation marks) before the semicolon at the end of the SQL statement. The SQL statement should look like:

```
SELECT DISTINCTROW [Category Name] AS [Sales by Category],
SUM([Sales by Category].[Product Sales]) AS [Product Sales]
FROM [Sales by Category]
GROUP BY [Category Name]
Order By SUM([Sales by Category].[Product Sales]) desc;
```

6. Choose OK.
7. View the form in Form view. Note that the Category Name records are now listed in descending order of sales.

##### Method 2:

The following example describes how to create and use a query for the graph's RowSource property:

For Microsoft Access version 2.0:

1. Create a form with a graph using the above steps.
2. View the form in Design view.
3. With your secondary mouse button, click the graph object, and then choose Properties.

4. With your secondary mouse button, click the Row Source property and then choose Build.
5. For the Product Sales field, change the sort order to Descending.
6. Close the Query window and save the changes.
7. Switch to Form view.

For Microsoft Access version 1.x:

1. Create a form with a graph using the above steps.
2. Make a copy of the Sales By Category query. Call the copy Sales By Category 2.
3. Open the Sales By Category 2 query in Design view. Create the following fields in the query:

```
Field: Sales by Category:Category Name
 Total: Group By
 Sort: <leave blank>
Field: Product Sales
 Total: Sum
 Sort: Desc
```

Note that the first field is named "Sales by Category:Category Name" because the name of the first field becomes the graph's title.

4. Remove the Product Name field.
5. Save the query.
6. View the form in Design view. Replace the SQL text in the graph's RowSource property with the Sales By Category 2 query.
7. View the form in Form view.

The columns will be in descending order.

#### REFERENCES

=====

For more information on ordering fields, search for "order by" then "ORDER BY Clause (SQL)" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 queries msgraph

KBCategory: kbtool

KBSubcategory: IntpGrph

## **INF: How to Change the Status Bar Text Using SysCmd()**

**Article ID: Q105511**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article demonstrates how to use the SysCmd() function to display custom messages in the Microsoft Access status bar.

NOTE: This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

### MORE INFORMATION

=====

The SysCmd() function is documented in the manual "The Secrets of AccessWizards," which is included with the Microsoft Access Distribution Kit (ADK) version 1.1. You can use this function to change the text that appears in the status bar. For example, you might want to change the status bar to read "Sorting..." while a sort operation is taking place, and then remove it when the operation is complete.

The SysCmd() function syntax is as follows:

```
SysCmd(<action> [, <text>])
```

Replace <action> with a numeric expression identifying the action to take. Valid entries are as follows:

- 4 - Set the status bar text to the text argument
- 5 - Reset the status bar text

Replace <text> with a string expression that will appear left aligned in the status bar.

The SysCmd() function returns null, which is not used.

### Setting Status Bar Text

-----

When the action argument is 4, the string expression in the text argument will appear in the status bar. For example, type the following expression in the Immediate window and the message "Sorting..." will appear in the status bar:

```
? SysCmd(4, "Sorting...")
```

## Removing or Resetting the Status Bar Text

---

When the action argument is 5, the status bar text set by the previous SysCmd() function is removed. For example, type the following in the Immediate window and the "Sorting..." message will be removed:

```
? SysCmd(5)
```

The text argument can contain about 80 characters. Since the text in the status bar uses a proportional font, the actual number of characters you can display is determined by the total width of all the characters in the text argument.

You cannot set the status bar text to an empty string. If you want to remove the existing text in the status bar, set the text argument to a single space.

## How to Use SysCmd() in an Access Basic Function

---

The following sample function will place a message in the status bar, display a message box, and then remove the status bar text:

1. Create a new module called Status Bar Test.
2. Enter the following statement in the Declarations section:

```
Option Explicit
```

3. Enter the following function:

```
Function StatusBar ()
 DimRetVal As Variant
 RetVal = SysCmd(4, "The rain in Spain falls mainly ...")
 MsgBox "Press OK when you are ready to finish!"
 RetVal = SysCmd(5)
End Function
```

4. To run the function, type the following in the Immediate window, and then press ENTER:

```
? StatusBar()
```

## REFERENCES

---

Microsoft Access Distribution Kit "The Secrets of AccessWizards," version 1.1, page 30

For information on how to use the status bar progress meter, query on the following words here in the Microsoft Knowledge Base:

progress and meter and updating

Additional reference words: 1.00 1.10 2.00 StatusBar progressmeter  
KBCategory: kbprg

KBSubcategory: PgmOthr



**INF: How to Check a Menu Item Using Access Basic**  
**Article ID: Q90811**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Microsoft Access does not have any built-in macro or Access Basic commands that you can use to place a check mark beside or remove a check mark from a menu command. To do this in Microsoft Access, you must use Access Basic code to call Microsoft Windows application programming interface (API) functions.

MORE INFORMATION

=====

The following Windows API function is used to check and uncheck a menu item:

CheckMenuItem% (hMenu%, wIDCheckItem%, wCheck%)

-----

This function places check marks next to or removes check marks from menu items in the pop-up menu specified by the hMenu% parameter.

hMenu%            Identifies the menu.

wIDCheckItem%    Specifies the menu item to be checked.

wCheck%           Specifies how to check the menu item. The wCheck% parameter can be a combination of the MF\_CHECKED or MF\_UNCHECKED with MF\_BYPOSITION or MF\_BYCOMMAND flags. These flags can be combined by using the bitwise OR operator. The values are described as follows:

| Value         | Meaning                                                                                                            |
|---------------|--------------------------------------------------------------------------------------------------------------------|
| -----         |                                                                                                                    |
| MF_BYCOMMAND  | Specifies that the idCheckItem parameter gives the menu-item identifier (MF_BYCOMMAND is the default).             |
| MF_BYPOSITION | Specifies that the idCheckItem parameter gives the position of the menu item (the first item is at position zero). |
| MF_CHECKED    | Selects the item (adds check mark).                                                                                |
| MF_UNCHECKED  | Clears the item (removes check mark).                                                                              |

Return Value The return value specifies the previous state of the item. It is either MF\_CHECKED or MF\_UNCHECKED. The return value is -1 if the menu item does not exist.

NOTE: Top-level menu items (items on the Microsoft Access menu bar) cannot have a check mark.

When determining the position of a menu command, separator bars count as commands. For example, to get to the Import menu command on the File menu in the Database window, use the arguments 0 (for the File menu) and 7 (for the Import command). Positions 3 and 6 correspond to the separator bars on the File menu.

#### Example

-----  
The example below creates a menu, associated with a form, that can be used to check and uncheck a menu command.

1. Create a new macro. After adding the following actions and their associated properties, save the macro and name it "Menu Checking Macro." To display the "Macro Name" column, choose Macro Name from the View menu.

| Macro Name | Action  | Function Name          |
|------------|---------|------------------------|
| &Check     | RunCode | Check_Menu_Item(0,0)   |
| &Uncheck   | RunCode | Uncheck_Menu_Item(0,0) |

NOTE: The first parameter in the function indicates which menu on the menu bar to use. All menus are zero based. For example, if your menu bar contains the File, Edit, Window, and Help menus, and you specify 2 as the first parameter for one of these functions, the function will work on the Window menu. A 2 indicates the Window menu instead of the Edit menu because the parameter uses base 0 instead of base 1; therefore, 0 indicates the File menu, 1 indicates the Edit menu, and so on.

The second parameter in the function indicates which menu command (on the menu indicated in the first parameter) to place the check mark beside. This parameter also uses base 0. To refer to the first command on the menu, use 0; to refer to the second, use 1; and so on.

2. Create a new macro. After adding the action below and its associated property, save the macro as "Custom Menu."

Action  
-----  
AddMenu  
Menu Name:           &Check  
Menu Macro Name:    Menu Checking Macro

3. Create a new blank form and display the Properties window of the form by choosing Properties from the View menu.

4. In Microsoft Access 1.X, set the OnMenu property to "Custom Menu." In Microsoft Access 2.0, set the MenuBar 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 sample Access Basic code listed at the end of this document. Save the module as "Menu Checking Code."
7. From the Database window, select the "Menu Checking Form" form and then choose the Open button to display the form in Form view. The normal Microsoft Access menu will disappear and be replaced by the custom menu that was designed above.
8. There are two options on the Check menu. If you choose the Check command, the command will be checked. If you choose the UnCheck command, the Check command will be unchecked.

#### Sample Access Basic Code

---

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```

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

Declare Function GetMenu% Lib "user" (ByVal hWnd%)
Declare Function GetSubMenu% Lib "user" (ByVal hSubMenu%, ByVal nPos%)
Declare Function CheckMenuItem% Lib "user" (ByVal hSubMenu%, _
 ByVal nPos%, ByVal Flag%)
Declare Function FindWindow% Lib "user" (ByVal lpClassName As Any, _
 ByVal lpCaption As Any)
Declare Function IsZoomed% Lib "user" (ByVal hWnd%)

Const MF_BYPOSITION = &H400
Const MF_BYCOMMAND = &H0
Const MF_CHECK = &H8
Const MF_UNCHECKED = &H0
Const MyNull = 0&
Const ClassName = "OMain"

Dim ChWnd% ' Handle to the Microsoft Access window.
Dim hMenuTop% ' Handle to the Microsoft Access menu.
Dim hSubMenu% ' Handle to the pop-up menu associated with hMenuTop%.
Dim ItemID% ' Command ID associated with a specified menu item.
Dim ReturnVal% ' Function return value may be used depending on
 ' the function.

'=====
' This function displays a check mark next to the menu item.
'=====
Function Check_Menu_Item% (TopLevel%, SubLevel%)

```

```

'If the form is maximized, the system menu is added to the forms
'menu bar, so increment the actual TopLevel%
If (IsZoomed(Screen.ActiveForm.hWnd)) Then
 TopLevel% = TopLevel% + 1
End If

Call Get_Menu_Handles(TopLevel%)
' The following statement must appear on one line.
Check_Menu_Item% = CheckMenuItem(hSubMenu%, SubLevel%,_
 MF_BYPOSITION Or MF_CHECK)
End Function

'=====
' This function initializes:
' - The window handles associated with the Microsoft Access form.
' - The handle to the menu of the specified window.
' - The menu handle of the specified pop-up menu of the Window menu.
' The variables here are global to the database.
'=====
Sub Get_Menu_Handles (TopLevel%)
 ChWnd% = FindWindow(Classname, MyNull)
 hMenuTop% = GetMenu(ChWnd%)
 hSubMenu% = GetSubMenu(hMenuTop%, TopLevel%)
End Sub

'=====
' This function will uncheck a menu item.
'=====
Function UnCheck_Menu_Item% (TopLevel%, SubLevel%)
 Call Get_Menu_Handles(TopLevel%)
 ' The following statement must appear on one line.
 UnCheck_Menu_Item% = CheckMenuItem(hSubMenu%, SubLevel%,_
 MF_BYPOSITION Or MF_UNCHECKED)
End Function

```

#### REFERENCES

=====

For more information on custom menus, search for "customizing menus" using the Microsoft Access Help menu.

For more information on similar programming features, query on the following words here in the Microsoft Knowledge Base:

GetSubMenu

Additional reference words: 1.00 1.10 2.00 menus checkmark

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: How to Check for Duplicate Values in Primary Key Fields**  
**Article ID: Q102527**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

When you are adding records, Microsoft Access does not normally check the values in primary key fields for duplicates until you move to the next record. If you enter an invalid or duplicate value in a control and move to the next record, you may invalidate all previous entries. However, there is a method to force an immediate check for duplicate values. This article describes how to check for duplicate values immediately.

MORE INFORMATION

=====

The following example uses the Customers form in the sample database NWIND.MDB. The Customers form is based on the Customers table; Customer ID is the primary key in the table:

1. Create the following new macro. Make sure to choose Conditions from the View menu before you begin entering this macro:

NOTE: In the following sample macro, an underscore ( \_ ) is used as a line-continuation character. Remove the underscore when re-creating this macro.

| MacroName | Condition                                                                             | Action                    |
|-----------|---------------------------------------------------------------------------------------|---------------------------|
| IsItDup   | DCount("[customer id]","customers",_<br>"[customer id]=Form.[customer id]")<>0<br>... | MsgBox<br><br>CancelEvent |

IsItDup Actions

-----  
MsgBox  
    Message: Duplicate ID  
CancelEvent

2. Open the Customers form in Design view. Change the BeforeUpdate property of the Customer ID field as follows:

Text Box: Customer ID  
    BeforeUpdate: IsItDup

After you make this change, adding a duplicate value in the Customer ID field will result in an error message. The insertion point will remain in the field so that you can change the value.

Additional reference words: 1.00 1.10 2.00 count before update fasttips  
KBCategory: kbusage

KBSubcategory: FmsOthr

## **INF: How to Communicate with NetDDE**

**Article ID: Q99939**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article demonstrates how to use NetDDE to communicate with Microsoft Access on a remote computer. This article assumes that you are familiar with DDE communications on a local computer, and with the capabilities of Microsoft Access as a DDE server.

This article also assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

### MORE INFORMATION

=====

In order to establish a link to a remote application, a DDE share must be created on the remote computer. A DDE share is a name that represents an application name, a topic, and any permissions for that topic. A DDE conversation can only occur with applications and topics assigned to share names on the remote computer. Note that a utility called Network DDE Share Manager (provided with the Windows for Workgroups Resource Kit) can be used to create and edit DDE share names.

#### How to Create the DDE Share

-----

The following steps demonstrate how to create a DDE share for the Employees table in the sample database NWIND.MDB:

1. Start the Network DDE Share Manager. From the Share menu, choose New.
2. Type "Northwind Employees" (without the quotation marks) in the Share Name box.
3. Type "MSACCESS.EXE" (without the quotation marks) in the Application Name box and "NWIND.MDB;TABLE Employees" (without the quotation marks) in the Topic Name box. The password information is optional and can be left empty.
4. From the File menu, choose Save.

#### How to Communicate with the Remote Microsoft Access DDE Share

-----

NOTE: Before running the following function, make sure that Microsoft

Access is running on the remote computer and that the NWIND.MDB database is open.

Create the following module function in Microsoft Access on your local computer:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Option Explicit
Function GetRemoteEmployeeInfo()
 Dim iChan As Integer
 Dim EmpRecord As String, FieldNames As String

 iChan = DDEInitiate("\\RemoteMachineName\NDDE$", _
 "Northwind Employees")

 FieldNames = DDERequest(iChan, "FieldNames")
 MsgBox FieldNames, 0, "Employee Table Field Names"

 On Error Resume Next
 EmpRecord = DDERequest(iChan, "FirstRow")
 Do Until EmpRecord = ""
 MsgBox EmpRecord, 0, "Employee Record"
 EmpRecord = ""
 EmpRecord = DDERequest(iChan, "NextRow")
 Loop
End Function
```

Change the first argument of the DDEInitiate command (RemoteMachineName) to reference the name of your remote computer.

When you run this function from the Immediate window, a message box will display a list of field names in the Employees table. Next, a series of message boxes will appear displaying each record from the table. The field data will be delimited by a tab character.

#### The DDEInitiate Statement

-----

Note the arguments used in the DDEInitiate command to establish the link with the remote share. The application name is the net DDE server application NDDE\$ located on the remote computer "\\RemoteMachineName". NDDE\$ is a system service application that is run automatically by Windows for Workgroups in the background. This application is always running and present when Windows for Workgroups is running and cannot be terminated.

When NDDE\$ receives the topic "Northwind Employees" it checks the list of available DDE shares on the local computer. If it finds the share, it will validate any permissions on the share and prompt for a password if one is required. After the validation is complete, NDDE\$ establishes a local DDE link with the application name and topic assigned to the share name.

Once the link has been established, communication with the client application proceeds as if it were another application on the local computer. NDDE\$ becomes the liaison between the two applications across the



network.

#### REFERENCES

=====

For more information on Microsoft Access as a DDE server, query on the following words here in the Microsoft Knowledge Base:

dde and server

Microsoft Windows for Workgroups "Resource Kit," Chapter 11

Additional reference words: 1.00 1.10 2.00 dynamic data exchange

KBCategory: kbinterop

KBSubcategory: IntpDde

## INF: How to Compare Data Among Records in a Query

Article ID: Q122243

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

You usually use Access Basic code to compare data among records in a table. This article describes how to use a query instead of code to compare data among records.

### MORE INFORMATION

=====

The following example demonstrates how to use a query to compare data among the records in the Orders table in the sample database NWIND.MDB. This query finds orders that were followed by another order within 60 days:

1. Open the sample database NWIND.MDB.
2. Create a new query. In the Add Table dialog box, add the Orders table twice. The table will be added as Orders\_1 the second time.
3. Join the tables on the Customer ID field.
4. Create the following columns in the query grid.

NOTE: In the following query, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this query.

```
Field: Customer ID
 Table: Orders
 Sort: Ascending
Field: Order Date
 Table: Orders
 Sort: Ascending
Field: DaysBetweenOrders: DateDiff("d",[Orders].[Order Date],_
 [Orders_1].[Order Date])
 Criteria: Between 1 And 60
Field: Order ID
 Table: Orders
Field: NextOrderDate: Order Date
 Table: Orders_1
```

5. Run the query. Every order that was followed by another order within 60 days will be returned.

### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 11, "Designing

Select Queries," page 227

Additional reference words: 2.00 previous next

KBCategory: kbusage

KBSubcategory: QryHowto

**INF: How to Control the Number of Records Printed Per Page**  
**Article ID: Q119075**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes a method that you can use to control the number of records printed per page on a report.

MORE INFORMATION

=====

You can control the number of records printed per page on a report by setting the page break's Visible property. The following example demonstrates how to print three records per page on a report:

1. Open the sample database NWIND.MDB.
2. Create a new macro with the following action, and then save the macro as Detail Page Break:

Macro Action

-----  
SetValue

Item: [PageBreak].Visible

Expression: iif([TheCounter] Mod 3 = 0, "Yes", "No")

3. Follow these steps to create a report based on the Products And Suppliers query:
  - a. In the Database window, choose the Report button, then choose New.
  - b. In the Select A Table/Query box, select the Products And Suppliers query, then choose the Report Wizards button.
  - c. In the Which Wizard Do You Want box, select Groups/Totals, then choose OK.
  - d. In the "Which fields do you want on your report?" screen, select Product Name in the Available Fields box, then choose the ">" button. Repeat using the Company Name, Contact Name, City, and Country fields. Choose the Next button.
  - e. In the "Which fields do you want to group by?" screen, choose the Next button.
  - f. In the "Which fields do you want to sort by?" screen, select Product Name in the Available Fields box, then choose the ">" button. Choose the Next button.

- g. In the "What style do you want for your report? screen, select the Executive option button, then choose the Next button.
  - h. In the "What title do you want for your report?" screen, select the Modify The Report's Design option button, then choose the Finish button.
4. Add a text box with the following properties to the detail section. Place the text box directly on top of the Product Name control. This control will act as a counter for the number of records in the report:

```
Text Box
Name: TheCounter
ControlSource: =1
Visible: No
RunningSum: Over All
```

NOTE: In Microsoft Access version 1.x, the Name property is called the ControlName property.

- 5. Set the detail section's OnFormat property to the Detail Page Break macro.
- 6. Add a page break control to the bottom left corner of the detail section, beneath the Product Name control.
- 7. Set the page break control's Name property to PageBreak.
- 8. Preview the report. The first page of the report should contain the following three records:

| Product Name | Company Name        | Contact Name     | City      | Country   |
|--------------|---------------------|------------------|-----------|-----------|
| Alice        | Pavlova, Ltd.       | Ian Devling      | Melbourne | Australia |
| Aniseed      | Exotic Liquids      | Charlotte Cooper | London    | UK        |
| Boston Crab  | New England Seafood | Robb Merchant    | Boston    | USA       |

If the last record is missing, or is only partially displayed, the PageBreak control is not down far enough in the report's detail section.

You can modify this example to print any number of records per page by changing the expression in the Detail Page Break macro from "mod 3" to "mod x" where x is the number of records you want to print per page. For example, if you want to print 11 records per page, change the expression to read:

```
iif([TheCounter] mod 11 = 0, "Yes", "No")
```

#### REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, "SetValue Action," pages 437-438, and "Visible Property," pages 495-496

Microsoft Access "User's Guide," version 2.0, pages 689-693

Microsoft Access "User's Guide," version 1.1, pages 577-579

Microsoft Access "User's Guide," version 1.0, pages 573-576

Additional reference words: 1.00 1.10 2.00 page break

KBCategory: kbusage

KBSubcategory: RptsLyt

**INF: How to Convert Currency or Numbers into English Words**  
**Article ID: Q95640**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

By using four functions you can convert a currency or other numeric value into its equivalent in English words. For example, you can change 24.50 into the following text: Twenty Four DOLLARS and Fifty CENTS.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

Note that none of the words in the result are hyphenated. That is, the result shows Twenty Four, not Twenty-Four. Revise the code if hyphenation is important to you.

MORE INFORMATION

=====

The following code creates four functions: NumToWord(), GetWord(), GetTens(), and GetDigit():

```
' Note: Enter each command on a single line. The underscore (_) on
' the end of some lines is a line continuation character. It's there
' to improve readability. When you enter the line, omit the underscore
' and combine all continued lines into a single line.
```

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

```
'=====
```

```
'*** This is the main function call
```

```
'=====
```

```
Function NumToWord(NumVal)
 Dim NTW, NText, Dollars, Cents, NWord, TotalCents As String
 DIM DecPlace, TotalSets, Cnt, LDollHold As Integer
 ReDim NumParts(9) As String 'Array for Amount (sets of three)
 ReDim Place(9) As String 'Array containing place holders
 Dim LDoll As Integer 'Length of the Dollars Text Amount

 Place(2) = " Thousand " '
 Place(3) = " Million " 'Place holder names for money
 Place(4) = " Billion " 'amounts
 Place(5) = " Trillion " '
End Function
```

```

NTW = "" 'Temp value for the function
NText = Trim(Str(NumVal)) 'String representation of amount
DecPlace = InStr(Trim(NText), ".") 'Position of decimal 0 if none
Dollars = Trim(Left(NText, IIf(DecPlace = 0, Len(NumVal), _
 DecPlace - 1)))

LDoll = Len(Dollars)
Cents = Trim(Right(NText, IIf(DecPlace = 0, 0, Abs(DecPlace - _
 Len(NText))))))

If Len(Cents) = 1 Then
 Cents = Cents & "0"
End If
If (LDoll Mod 3) = 0 Then
 TotalSets = (LDoll \ 3)
Else
 TotalSets = (LDoll \ 3) + 1
End If
cnt = 1
LDollHold = LDoll
Do While LDoll > 0
 NumParts(cnt) = IIf(LDoll > 3, Right(Dollars, 3), _
 Trim(Dollars))
 Dollars = IIf(LDoll > 3, Left(Dollars, (IIf(LDoll < 3, 3, _
 LDoll)) - 3), "")
 LDoll = Len(Dollars)
 cnt = cnt + 1
Loop
For cnt = TotalSets To 1 Step -1 'step through NumParts array
 NWord = GetWord(NumParts(cnt)) 'convert 1 element of NumParts
 NTW = NTW & NWord 'concatenate it to temp variable
 If NWord <> "" Then NTW = NTW & Place(cnt)
Next cnt 'loop through
If LDollHold > 0 Then
 NTW = NTW & " DOLLARS and " 'concatenate text
Else
 NTW = NTW & " NO DOLLARS and " 'concatenate text
End If
TotalCents = gettens(Cents) 'Convert cents part to word
If TotalCents = "" Then TotalCents = "NO" 'Concat NO if cents=0
NTW = NTW & TotalCents & " CENTS" 'Concat Dollars and Cents
NumToWord = NTW 'Assign word value to function
End Function 'End NumToWord function, Return to calling program

```

```

'=====
' The following function converts a number from 0 to 999 to text
'=====

```

```

Function GetWord (NumText)
 Dim GW As String, x As Integer
 GW = "" 'null out temporary function value
 If Val(NumText) > 0 Then
 For x = 1 To Len(NumText) 'loop the length of NumText times
 Select Case Len(NumText)
 Case 3:
 If Val(NumText) > 99 Then
 GW = GetDigit(Left(NumText, 1)) & " Hundred "
 End If
 NumText = Right(NumText, 2)
 Case 2:

```



```

 GW = GW & GetTens(NumText)
 NumText = ""
 Case 1:
 GW = GetDigit(NumText)
 Case Else
 End Select
 Next x
 End If
 GetWord = GW 'assign function return value
End Function 'End function GetWord - Return to calling program

```

```

=====
' The following function converts a number from 10 to 99 to text
=====

```

```

Function GetTens (TensText)
 Dim GT As String
 GT = "" 'null out the temporary function value
 If Val(Left(TensText, 1)) = 1 Then ' If value between 10-19
 Select Case Val(TensText)
 Case 10: GT = "Ten" '
 Case 11: GT = "Eleven" '
 Case 12: GT = "Twelve" '
 Case 13: GT = "Thirteen" ' Retrieve numeric word
 Case 14: GT = "Fourteen" ' value if between ten and
 Case 15: GT = "Fifteen" ' nineteen inclusive.
 Case 16: GT = "Sixteen" '
 Case 17: GT = "Seventeen" '
 Case 18: GT = "Eighteen" '
 Case 19: GT = "Nineteen" '
 Case Else
 End Select
 Else ' If value between 20-99
 Select Case Val(Left(TensText, 1))
 Case 2: GT = "Twenty " '
 Case 3: GT = "Thirty " '
 Case 4: GT = "Forty " '
 Case 5: GT = "Fifty " ' Retrieve value if it is
 Case 6: GT = "Sixty " ' divisible by ten
 Case 7: GT = "Seventy " ' excluding the value ten.
 Case 8: GT = "Eighty " '
 Case 9: GT = "Ninety " '
 Case Else
 End Select
 GT = GT & GetDigit(Right(TensText, 1)) 'Retrieve ones place
 End If
 GetTens = GT ' Assign function return value.
End Function 'End function GetTens - return to calling program

```

```

=====
' The following function converts a number from 1 to 9 to text
=====

```

```

Function GetDigit (Digit)
 Select Case Val(Digit)
 Case 1: GetDigit = "One" '
 Case 2: GetDigit = "Two" '
 Case 3: GetDigit = "Three" '
 Case 4: GetDigit = "Four" ' Assign a numeric word value
 End Select
End Function

```

```

 Case 5: GetDigit = "Five" ' based on a single digit.
 Case 6: GetDigit = "Six" '
 Case 7: GetDigit = "Seven" '
 Case 8: GetDigit = "Eight" '
 Case 9: GetDigit = "Nine" '
 Case Else: GetDigit = "" '
 End Select
End Function 'End function GetDigit - return to calling program

```

#### How to Use the Sample Code

-----

To change a currency value into English words, call the NumToWord() function giving the number as the argument. NumToWord() calls the other functions as needed and returns the result in English words.

There are two ways to call the NumToWord function. You can call it in Access Basic code as follows:

```

Amount = 24.50
WordValue = NumToWord(Amount)

```

Or you can call NumToWord() as a parameter in a SetValue action in a macro. For example, set Amount to 24.50 and use the following in a SetValue action in a macro:

```

NumToWord(Amount)

```

Additional reference words: 1.00 1.10 2.00 AB functions macros programs  
 KBCategory: kbprg  
 KSubcategory: PgmOthr

## INF: How to Convert Twips to Pixels

Article ID: Q94927

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Because Microsoft Access stores dimension/location properties as twips, it may be necessary to convert to pixels in certain cases, such as when calling a Windows API function. This article describes how to do this.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

### MORE INFORMATION

=====

The ConvertTwipsToPixels() function below can be used to convert twips to pixels. Note that pixels are not always square (the height and width are not the same), so it is necessary to pass in the desired "direction" to use (horizontal or vertical).

To add this function to your application, do the following:

1. Create a new module and enter the following Windows API declarations and constants into the Declarations section:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Option Explicit
Declare Function GetDC% Lib "User" (ByVal hw%)
Declare Function ReleaseDC% Lib "User" (ByVal hw%, ByVal hDC%)
Declare Function GetDeviceCaps% Lib "GDI" (ByVal hDC%, _
 ByVal iCapability%)

Const WU_LOGPIXELSX = 88
Const WU_LOGPIXELSY = 90
```

2. Add the following function:

```
Function ConvertTwipsToPixels (nTwips%, nDirection%) As Integer
 Dim hDC% 'Handle to device
 Dim nPixelsPerInch%
 Const nTwipsPerInch = 1440
 hDC% = GetDC(0)
 If (nDirection% = 0) Then 'Horizontal
```

```
 nPixelsPerInch% = GetDeviceCaps(hDC%, WU_LOGPIXELSX)
Else
 'Vertical
 nPixelsPerInch% = GetDeviceCaps(hDC%, WU_LOGPIXELSY)
End If
hDC% = ReleaseDC(0, hDC%)
ConvertTwipsToPixels = (nTwips% / nTwipsPerInch%) * nPixelsPerInch
End Function
```

3. To call this function, pass the number of twips you want to convert, and another parameter indicating whether the horizontal or vertical measurement (0 for horizontal, non-zero for vertical). The following is a sample call:

```
OldTwips = 2377
NewPixels = ConvertTwipsToPixels(OldTwips, 0)
```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

## **INF: How to Correctly Back Up Multiuser Databases**

**Article ID: Q109816**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

In order to back up a multiuser database correctly, you must make sure that you have exclusive access to the database. If you back up a database while others are using it, you risk damaging the backup database.

### MORE INFORMATION

=====

You must have exclusive access to a database in order to back it up correctly. If you back up a database that is being modified by other users, your backup copy of the database may be damaged or corrupted. When you open such a backup database in Microsoft Access, you may receive the error message "database is corrupted." If you repair and compact the database, you may find that data in the database is truncated.

The only way to ensure the integrity of your backup database is to achieve a static copy of the database. You must have exclusive access to all the tables in the database at the same time in order to do so. Once you have exclusive access to the database, you can copy or export the database, and then archive the copy or export.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: GnlMu

# INF: How to Create a Custom Startup "Splash" Screen

Article ID: Q101374

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

## SUMMARY

=====

This article explains how to create a custom startup "splash" screen for your Microsoft Access application.

## MORE INFORMATION

=====

A startup "splash" screen (title screen) is used by programmers to display important information about a program during startup and to give the user something colorful to look at and read while waiting for the program to complete its initial processing, such as reading and setting up user-defined preferences.

A typical splash screen contains the following information:

- The program name and version number
- The registered user's name and company
- Copyright information
- The developer's company name and logo

This article contains two Access Basic functions to implement splash functionality.

## Creating a New Splash Module

-----

1. Create a new Access Basic module with the following functions:

```
!*****
' MODULE NAME: Splash
' DECLARATION SECTION
'*****
Option Explicit

Dim gSplashStart ' The time when the splash screen opened.
Dim gSplashInterval ' The minimum time to leave the splash screen up.
Dim gSplashForm ' The name of the splash screen form.

'*****
' FUNCTION: SplashStart()
'
' PURPOSE: Used to invoke the splash screen form specified by the
' SplashForm argument.
'
' ARGUMENTS:
```

```

' SplashForm - The name of the form to use as the splash
' screen.
' SplashInterval - The minimum time, in seconds, that the splash
' screen must remain active on the screen.
'
'*****
Function SplashStart (ByVal SplashForm As String, ByVal SplashInterval_
As Integer)
' Open the splash form.
DoCmd OpenForm SplashForm

' Set the starting time.
gSplashStart = Timer

' Record the global information.
gSplashInterval = SplashInterval
gSplashForm = SplashForm
End Function

'*****
' FUNCTION: SplashEnd()
'
' PURPOSE: Used to close the splash screen form opened by the
' SplashStart() function. This function checks to ensure that
' the splash screen remains active until the user-specified
' interval has expired.
'
'*****
Function SplashEnd ()
Dim RetVal

' Loop until the splash screen has been active for
' .. the desired interval.
Do Until Timer - gSplashStart > gSplashInterval
' Yield control so other applications can process.
RetVal = DoEvents()
Loop

' Close the splash screen.
DoCmd Close A_FORM, gSplashForm

End Function

```

2. Save the new module as Splash.

#### How to Use the SplashStart() and SplashEnd() Functions

---

1. Create an unbound form with the text and graphics you want to display on the splash screen. For more information, see the "Setting Form Properties" section below.
2. Create a macro with the following actions and save it as AutoExec:

| Action  | Arguments |
|---------|-----------|
| RunCode | 1.        |

<any startup macro actions required by your program> 2.  
RunCode 3.

#### AutoExec Actions

---

1. RunCode  
    Function Name: SplashStart("YourSplashFormNameHere", 5)
2. <any startup macro actions required by your program .. optional>
3. RunCode  
    Function Name: SplashEnd()

#### Setting Form Properties

---

For best results, the splash form should have its form properties set as follows:

ScrollBars: Neither  
PopUp: Yes  
Modal: Yes  
RecordSelectors: No

#### Suppressing the Default Splash Screen

---

To prevent the default Microsoft Access splash screen from appearing, you can use the run-time version of Microsoft Access, provided with the Microsoft Access Distribution Kit (ADK) for version 1.1, or the Microsoft Access Developer's Toolkit (ADT) for version 2.0.

Additional reference words: 1.10 2.00 Title Intro  
KBCategory: kbprg  
KBSubcategory: PgmOthr



## **INF: How to Create a Custom Toolbar Using a Form**

**Article ID: Q113304**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article describes how to create a custom, floating toolbar using a Microsoft Access form. Using a form for a custom toolbar provides the following advantages:

- Labels, combo boxes and list boxes can be placed on the toolbar.
- Buttons with custom bitmaps can be placed on the toolbar.
- You can prevent users from closing or moving the toolbar.
- You can take advantage of form modules to store the code being used by the controls on the toolbar with the toolbar itself.
- You can disable the shortcut menus from being used on the toolbar.

### MORE INFORMATION

=====

The example below demonstrates how to create a custom, floating toolbar that has buttons to find, save, delete, and undo a record, as well as buttons to navigate among the records, including buttons to go to the first record, go to the previous record, go to the next record, go to the last record, and add a new record.

1. Open the sample database NWIND.MDB and create a new, blank form.
2. Set the following form-level properties:

```
Caption: Toolbar
ShortcutMenu: No
ScrollBars: Neither
RecordSelectors: No
NavigationButtons: No
PopUp: Yes
BorderStyle: Dialog
MinButton: No
MaxButton: No
```

3. Activate the toolbox by choosing Toolbox from the View menu. Make sure the Wizard button is chosen (the button should appear sunken).
4. Choose the command button tool, and then click anywhere in the form.
5. In the Categories box, select Record Navigation. In the When Button Is Pressed box, select Find Record. Choose Finish.
6. Drag the new button to the upper left corner of the detail section.
7. Repeat steps 4-6 to create the other buttons for the toolbar. Use the

following selection combinations from the Categories and the When Button Is Pressed boxes. Move each of the new buttons directly to the right of the button before it.

| Categories        | When button Is Pressed |
|-------------------|------------------------|
| Record Operations | Save Record            |
| Record Operations | Delete Record          |
| Record Operations | Undo Record            |
| Record Navigation | Go to First Record     |
| Record Navigation | Go to Previous Record  |
| Record Navigation | Go to Next Record      |
| Record Navigation | Go to Last Record      |
| Record Operations | Add New Record         |

8. Drag the bottom of the detail section up so that it is flush with the bottom of the buttons. Drag the right side of the detail section so that it is flush with the right side of the last button.
9. From the View menu, choose Code.
10. Create the following Access Basic function in the toolbar form module:

Option Explicit

```
Function ActivateToolbarForm ()
 On Error Resume Next
 Forms(Me.Tag).SetFocus
 If Err Then
 ActivateToolbarForm = False
 Else
 ActivateToolbarForm = True
 End If
End Function
```

This function will be used to reactivate the form the toolbar is floating on, so that the chosen operation is performed on that form rather than against the toolbar form itself.

11. For each button on the toolbar form, insert the following line of code at the very top of the button's OnClick event procedure. To insert the code, select a button, click the secondary mouse button in the OnClick property field, and then choose Build.

```
If ActivateToolbarForm() = False Then Exit Sub
```

For example, the code for the Search button might look like the following:

```
Sub Button0_Click ()
 If ActivateToolbarForm() = False Then Exit Sub
 On Error GoTo Err_Button0_Click

 DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, 10, , A_MENU_VER20

Exit_Button0_Click:
 Exit Sub
```

```

Err_Button0_Click:
 MsgBox Error$
 Resume Exit_Button0_Click
End Sub

```

This code ensures that the form the toolbar is floating on is selected for the chosen operation.

12. Save the form with the name `Toolbar`, and then close the form.
13. Create a new Access Basic module. Add the following code:

```

Option Explicit

Sub SetToolbarForm (F As Form)
 If IsLoaded("Toolbar") Then Forms![Toolbar].Tag = F.Name
End Sub

```

The `SetToolbarForm` subroutine uses the `IsLoaded()` function that is in the Utility Functions module in `NWIND.MDB`. You should copy this function for use in other databases.

14. For every form that you intend to use the custom toolbar with, add the following line of code to the form's `OnActivate` property:

```
SetToolbarForm Me
```

For this example, add the line of code above to the `OnActivate` property of the `Customers` and `Employees` forms:

- a. Open the `Employees` form in Design view.
- b. From the View menu, choose Code.
- c. Select Form in the first combo box on the code toolbar.
- d. Select Activate in the second combo box on the code toolbar.
- e. Add the above line of code so the subroutine appears as follows:

```

Sub Form_Activate ()
 SetToolbarForm Me
End Sub

```

- f. Repeat steps a-e for the `Customers` form.

This line of code instructs the custom toolbar to store the name of the form to be used when a toolbar button is chosen. This ensures that the toolbar actions are performed against the active form.

#### Using the Custom Toolbar

-----

Open the `Customers` and `Employees` forms in Form view, and then open the `Toolbar` form. Switch back and forth between the `Employees` form and the `Customers` form, using the navigation buttons on the custom toolbar to

navigate among the form records.

#### Suggested Enhancements to the Custom Toolbar

---

- You may want to automate the loading and closing of a custom toolbar form from another form's load and unload events. For example, to have the custom toolbar open only with the Customers form, add the following code to the Customers form's OnLoad and OnUnload properties:

```
Sub Form_Load ()
 DoCmd OpenForm "Toolbar"
End Sub
```

```
Sub Form_Unload (Cancel As Integer)
 DoCmd Close A_FORM, "Toolbar"
End Sub
```

- You can set the BorderStyle property of the toolbar form to None, so that no border and no caption bar appears. This will prevent users from being able to move the toolbar around.
- You may want to automatically position the toolbar form to a specific location on the screen. To do this, use a MoveSize macro action in the toolbar form's OnLoad property. The following sample code will position the toolbar form to the upper left corner of the screen:

```
DoCmd MoveSize 0, 0
```

Additional reference words: 1.00 1.10 2.00 tool bar

KBCategory: kbusage

KBSubcategory: FmsOthr

**INF: How to Create a Cyclic Tab Order in a Form**  
**Article ID: Q112064**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY  
=====

This article describes a technique that you can use to create a cyclic tab order on a form, so that pressing TAB in the last control on a form moves you back to the first control on the form (instead of moving you to the first control on the next record), and pressing SHIFT+TAB in the first control on a form moves you to the last control on the form.

MORE INFORMATION  
=====

The following example demonstrates how to create a cyclic tab order on the Customers form in the sample database NWIND.MDB. This technique uses two transparent command buttons, one at the top and one at the bottom of the form's tab order. When you press TAB in the last control, or SHIFT+TAB in the first control, you run a macro attached to the OnEnter property of the transparent command button at the top or bottom of the form's tab order.

1. Create a macro called CyclicCustomers with the following actions:

| MacroName | Action                                  |
|-----------|-----------------------------------------|
| GotoFirst | GotoControl<br>ControlName: Customer ID |
| GotoLast  | GotoControl<br>ControlName: Fax         |

2. Open the Customers form in Design view and add the following two command buttons with no labels:

Command Button  
-----  
ControlName: btnGotoLast  
OnEnter: CyclicCustomers.GotoLast  
Transparent: Yes

Command Button  
-----  
ControlName: btnGotoFirst  
OnEnter: CyclicCustomers.GotoFirst  
Transparent: Yes

NOTE: Make these buttons very small, and position them so that users of the form are unlikely to choose them accidentally.

3. Set the form's OnOpen property to:

OnOpen: CyclicCustomers.GotoFirst

This is necessary since the btnGotoLast button will be the first control in the form's tab order. If you do not set the form's OnOpen property to CyclicCustomers.GotoFirst, the first command button will be selected when the form is opened, moving the focus to the last control on the form.

4. From the Edit menu, choose Tab Order.
5. Scroll to the bottom of the list of control names to find the new btnGotoLast and btnGotoFirst controls. Select the btnGotoLast control and drag it to the top of the list. btnGotoFirst should remain at the bottom of the list.
6. View the form in Form view. The first Customer ID in the Customers table is displayed.
7. Press SHIFT+TAB. The Fax field (the last field in the form) is selected.
8. Press TAB. The Customer ID field is selected.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsEvt

**INF: How to Create a Function to Capture Screens of Your Forms**  
**Article ID: Q100973**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can use Microsoft Windows API functions to capture screens of any object on the Microsoft Windows desktop. You can perform screen captures of Microsoft Access, Microsoft Access objects, other Microsoft Windows applications, or the entire desktop.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

The following sample Access Basic code will capture an object on the screen to the Clipboard. Note that you can change the handle AccessHwnd in the sample code below to create screen captures of other applications, including specific objects within Microsoft Access.

After capturing a screen, you can open Paintbrush (PBRUSH.EXE) and choose the Paste command from the Edit menu. The screen capture from Microsoft Access will be loaded into the Paintbrush window.

To run this code, call the ScreenDump() function from an event, a macro, or the module's Immediate window. To run ScreenDump() from the Immediate window, type the following, and then press ENTER:

```
?ScreenDump()
```

To set up the ScreenDump() function, open a new or existing module and insert the code listed below.

Notes:

- In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.
- You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.

Code for the ScreenDump() Function:

```
Option Compare Database
Option Explicit
```

```
Type RECT_Type
 left As Integer
 top As Integer
 right As Integer
 bottom As Integer
End Type
```

```
Declare Function GetActiveWindow% Lib "User" ()
Declare Function GetDesktopWindow% Lib "User" ()
Declare Sub GetWindowRect Lib "User" (ByVal Hwnd%, _
 lpRect As RECT_Type)
Declare Function GetDC% Lib "User" (ByVal Hwnd%)
Declare Function CreateCompatibleDC% Lib "GDI" (ByVal hdc%)
Declare Function CreateCompatibleBitmap% Lib "GDI" (ByVal hdc%, _
 ByVal nWidth%, ByVal nHeight%)
Declare Function SelectObject% Lib "GDI" (ByVal hdc%, ByVal hObject%)
Declare Function BitBlt% Lib "GDI" (ByVal hDestDC%, ByVal X%, _
 ByVal Y%, ByVal nWidth%, _
 ByVal nHeight%, ByVal hSrcDC%, _
 ByVal XSrc%, ByVal YSrc%, _
 ByVal dwRop&)
Declare Function OpenClipboard% Lib "User" (ByVal Hwnd%)
Declare Function EmptyClipboard% Lib "User" ()
Declare Function SetClipboardData% Lib "User" (ByVal wFormat%, _
 ByVal hMem%)
Declare Function CloseClipboard% Lib "User" ()
Declare Function ReleaseDC% Lib "User" (ByVal Hwnd%, ByVal hdc%)
Declare Function DeleteDC% Lib "GDI" (ByVal hdc%)
```

```
Global Const SRCCOPY = &HCC0020
Global Const CF_BITMAP = 2
```

```
Function ScreenDump ()
 Dim AccessHwnd%, DeskHwnd%
 Dim hdc%
 Dim hdcMem%
 Dim rect As RECT_Type
 Dim junk%
 Dim fwidth%, fheight%
 Dim hBitmap%

 DoCmd Hourglass True

 '-----
 ' Get window handle to Windows and Microsoft Access
 '-----
 DeskHwnd = GetDesktopWindow()
 AccessHwnd = GetActiveWindow()

 '-----
 ' Get screen coordinates of Microsoft Access
 '-----
 Call GetWindowRect(AccessHwnd, rect)
 fwidth = rect.right - rect.left
```



```

fheight = rect.bottom - rect.top

'-----
' Get the device context of Desktop and allocate memory
'-----
hdc = GetDC(DeskHwnd)
hdcMem = CreateCompatibleDC(hdc)
hBitmap = CreateCompatibleBitmap(hdc, fwidth, fheight)

If hBitmap <> 0 Then
 junk = SelectObject(hdcMem, hBitmap)

 '-----
 ' Copy the Desktop bitmap to memory location
 ' based on Access coordinates.
 '-----
 junk = BitBlt(hdcMem, 0, 0, fwidth, fheight, hdc, rect.left, _
 rect.top, SRCCOPY)

 '-----
 ' Set up the Clipboard and copy bitmap
 '-----
 junk = OpenClipboard(DeskHwnd)
 junk = EmptyClipboard()
 junk = SetClipboardData(CF_BITMAP, hBitmap)
 junk = CloseClipboard()
End If

'-----
' Clean up handles
'-----
junk = DeleteDC(hdcMem)
junk = ReleaseDC(DeskHwnd, hdc)

DoCmd Hourglass False

End Function

```

Additional reference words: 1.00 1.10 2.00 dump  
KBCategory: kbprg  
KBSubcategory: PgmApi

## INF: How to Create a Multiple-Selection List Box

Article ID: Q121356

-----  
The information in this article applies to:

- Microsoft Access versions 2.0  
-----

### SUMMARY

=====

In a normal list box, you can only select one item at a time. This article describes how to create a multiple-selection list box, in which you can select more than one item at a time.

The example below demonstrates how to create a two-column list box on a form. The first column will display an "X" if the row is selected, or will be blank if the row is not selected. The second column will contain the values that you can select. This article also demonstrates how to create a semicolon-delimited list of the items selected in the list box.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The following example demonstrates how to create and use a multiple-selection list box. This example uses user-defined Access Basic functions to fill the list box. For additional information about how to fill a list box with an Access Basic function, please see the "References" section later in this article.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new module and enter the following lines in the module's Declarations section:

```
Option Explicit
```

```
Type MultiSelectArray_TYPE
```

```
 Selected As String ' Holds "X" or "" indicating selection
 Display As Variant ' The value to display in the list box
 ' row. Example: John Smith
 Value As Variant ' The value to store for the row
 ' selection. Example: 535-86-9328 (John's
 ' SSN)
```

```
End Type
```

```

Dim MultiSelectArray() As MultiSelectArray_TYPE
Dim MultiSelectRows

' Flag indicating if the list is being updated (new selection)
' or being filled.
Global UpdateMultiSelect

```

3. Add the following two functions to the module:

```

Function MultiSelect (fld As Control, id As Long, Row As _
Long, Col As Long, Code As Integer)
'*****
' CALLED FROM: The RowSourceType property of a list box.
' EXAMPLE:
' RowSourceType: MultiSelect
'*****
 Dim RetVal: RetVal = Null

 Select Case Code
 Case LB_INITIALIZE
 ' Is the list being updated by a new selection?
 If UpdateMultiSelect Then
 ' If so, ignore refilling the list.
 UpdateMultiSelect = False
 Else
 ' Otherwise, fill the MultiSelect array.
 MultiSelectRows = MultiSelectFillArray()
 End If
 RetVal = MultiSelectRows

 Case LB_OPEN
 RetVal = Timer ' Unique ID number for control.

 Case LB_GETROWCOUNT
 ' Return the number of rows in the MultiSelect array.
 RetVal = UBound(MultiSelectArray) + 1

 Case LB_GETCOLUMNCOUNT
 ' Return the number of columns to display.
 RetVal = 2

 Case LB_GETCOLUMNWIDTH
 RetVal = -1 ' Use the default width.

 Case LB_GETVALUE
 Select Case Col
 Case 0 ' Selected
 RetVal = MultiSelectArray(Row).Selected
 Case 1 ' Display
 RetVal = MultiSelectArray(Row).Display
 End Select

 Case LB_END ' End

 End Select

 MultiSelect = RetVal

```

End Function

```
Function MultiSelectUpdate (C As Control)
'*****
' CALLED FROM: The AfterUpdate property of the list box.
' EXAMPLE:
' AfterUpdate: =MultiSelectUpdate([<YourListBoxName>])
'*****
' Update the MultiSelect array selection by toggling
' the "X" in the selected row.
Select Case MultiSelectArray(C).Selected
 Case ""
 MultiSelectArray(C).Selected = "X"
 Case "X"
 MultiSelectArray(C).Selected = ""
End Select

' Set the flag indicating an update.
UpdateMultiSelect = True

' Requery the list.
C.Requery
```

End Function

4. Create a user-defined function called MultiSelectFillArray() that will fill the MultiSelect array with the values to be displayed in the list box. Following are two examples of the MultiSelectFillArray() function. The first example demonstrates how to fill the array with a list of names from the Employees table. The second example demonstrates how to fill the array with a list of field names from the Employees table. Enter only one of these functions in the module:

```
Function MultiSelectFillArray ()
'*****
' PURPOSE: Fills the MultiSelect array with a list of
' names from the Employees table.
' CALLED FROM: The MultiSelect() function's initialization
' code to fill the list box array with values.
'*****
 Dim DB As Database
 Dim RS As Recordset
 Dim i As Integer
 Dim RecordCount As Integer

 Set DB = DBEngine.Workspaces(0).Databases(0)
 Set RS = DB.OpenRecordset("Employees", DB_OPEN_SNAPSHOT)

 ' Get record count.
 RS.MoveLast
 RecordCount = RS.RecordCount
 RS.MoveFirst

 ' Resize the MultiSelect array to the number of Employee
 ' records.
 ReDim MultiSelectArray(0 To RecordCount - 1)
```

```

' Fill the MultiSelect array by setting:
' Selected to "" (clearing "X").
' Display to [First Name] space [Last Name].
' Value to [Employee ID].
For i = 0 To RecordCount - 1
 MultiSelectArray(i).Selected = ""
 MultiSelectArray(i).Display = RS![First Name] & " " _
 & RS![Last Name]
 MultiSelectArray(i).Value = RS![Employee ID]
 RS.MoveNext
Next i

' Return the number of rows in the array (RecordCount).
MultiSelectFillArray = RecordCount

End Function

Function MultiSelectFillArray ()
'*****
' PURPOSE: Fills the MultiSelect array with a list of
' field names from the Employees table.
' CALLED FROM: The MultiSelect() function's initialization
' code to fill the list box array with values.
'*****
 Dim DB As Database
 Dim RS As Recordset
 Dim i As Integer

 Set DB = DBEngine.Workspaces(0).Databases(0)
 Set RS = DB.OpenRecordset("Employees", DB_OPEN_SNAPSHOT)

 ' Resize the MultiSelect array to the number of Employee
 ' fields.
 ReDim MultiSelectArray(0 To RS.Fields.Count - 1)

 ' Fill the MultiSelect array by setting:
 ' Selected to "" (clearing "X").
 ' Display to the name of the field.
 ' Value to the name of the field.
 For i = 0 To RS.Fields.Count - 1
 MultiSelectArray(i).Selected = ""
 MultiSelectArray(i).Display = RS(i).Name
 MultiSelectArray(i).Value = RS(i).Name
 Next i

 ' Return the number of rows in the array (the number of
 ' fields).
 MultiSelectFillArray = RS.Fields.Count

End Function

```

5. Create a blank, new form.

6. Add a list box with the following properties to the form:

Name: EmployeeFields

```

ControlSource: <blank>
RowSourceType: MultiSelect
RowSource: <blank>
ColumnCount: 2
ColumnWidths: 0.15 in.
BoundColumn: 0
Width: 1.6 in
Height: 1.5 in
AfterUpdate: =MultiSelectUpdate([EmployeeFields])

```

7. View the form in Form view. Depending on which example you chose in step 4, you will see a list box containing either a list of names from the Employees table, or a list of field names from the Employees table. You can select or unselect an item in the list box by clicking the item with the mouse. An item is selected if an "X" appears in the column to the left of the item. Note that if you use the UP ARROW and DOWN ARROW keys to move the selection cursor up or down in the list box, each item that you move past is toggled.

NOTE: If you manually requery the list box by using the Requery macro action or the Requery method, or by pressing the F9 key in the list box, all the selections in the list box will be cleared.

#### How to Create a Semicolon-Delimited List of the Items Selected in the List Box

-----

The following example demonstrates how to use the MultiSelect array to create a semicolon-delimited list of the items selected in the list box:

1. Add the following function to the module that you created in step 2 above:

```

Function MultiSelectSemicolonList ()
 Dim i
 Dim Result
 Result = ""
 For i = 0 To UBound(MultiSelectArray)
 If MultiSelectArray(i).Selected = "X" Then
 Result = Result & MultiSelectArray(i).Value & ";"
 End If
 Next i

 ' Remove the last semicolon.
 If Result <> "" Then Result = Left(Result, Len(Result) - 1)

 MultiSelectSemicolonList = Result
End Function

```

2. Add a text box with the following property to the form that you created in step 5 above:

```
ControlSource: =MultiSelectSemicolonList()
```

3. View the form in Form view. When you make selections in the list box, the text box will be updated to display a list of the items selected.

## REFERENCES

=====

For more information about using an Access Basic function to fill a list box, search for "filling list boxes/combo boxes" then "Filling a List Box or Combo Box Using an Access Basic Function" using the Microsoft Access version 2.0 Help menu.

Additional reference words: 2.00 multiselect

KBCategory: kbusage

KBSubcategory: FmrCodeb

## **INF: How to Create a Multiuser Custom Counter**

**Article ID: Q88159**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

You can define a field as a counter in a Microsoft Access table. When a field is defined as a counter, Microsoft Access manages the numbering in the field. When you add a record, Microsoft Access assigns the next available value to the field. You have no control over this value.

You may want to use a unique counter that is not immediately sequential in nature. For example, you may want to use a counter that decrements, or a counter that that steps according to some value.

This article demonstrates a sample user-defined Access Basic function that you can use to create a custom counter field.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The following steps outline in general how to create a custom counter. Further below is a more detailed description on how to create a custom counter.

1. Create a separate table that will maintain the next available custom counter. This table will have one field and one record, with the value of the next available counter in this one record.
2. From within Access Basic, open this counter table and retrieve the value stored there.
3. Increment the value retrieved and store the number back into the database.
4. Close the table and use the value in the appropriate table as the next available counter.

In a single-user environment, the steps described above can be done with macro actions. In a multiuser environment, Access Basic is needed to handle the event where this counter table is locked. Macro actions do not provide the locking control needed in the multiuser environment.

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 ()

 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

'=====
'The following error routine should be replaced with a custom error
'routine. This example only resumes execution if an error occurs. If
'a record locking error occurs this is fine. But any non-record
'locking error will result in an infinite loop.
'=====

Next_Custom_Counter_Err:
 MsgBox "Error " & err & ": " & error$
 If ERR <> 0 then Resume
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. Choose 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().

NOTE: In Microsoft Access version 2.0, the OnPush property is called the OnClick property.

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 switch to Form view. Click the command button; a message box will display the value of the next counter.

Additional reference words: 1.00 1.10 2.00 network duplicate  
KBCategory: kbusage

KBSubcategory: TblDatyp

## INF: How to Create a Parameter In() Statement

Article ID: Q100131

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a way to create a query by using the In() operator with multiple values.

The method shown below uses a query that calls a function, which then is passed two parameters. The first parameter is the name of a field that exists in the table on which the query is based. The second parameter prompts the user to enter a list of values. The function processes the user's entries as the list of multiple parameters for the In() operator.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The following example is based on the sample database NWIND.MDB:

1. Create a new module with the following two functions:

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

Option Explicit

'=====
```

```
' The GetToken() function defines the delimiter character.
'=====
```

```
Function GetToken (stLn, stDelim)
 Dim iDelim as Integer, stToken as String
 iDelim = InStr(1, stLn, stDelim)
 If (iDelim <> 0) Then
 stToken = LTrim$(RTrim$(Mid$(stLn, 1, iDelim - 1)))
 stLn = Mid$(stLn, iDelim + 1)
 Else
 stToken = LTrim$(RTrim$(Mid$(stLn, 1)))
 stLn = ""
 End If
 GetToken = stToken
End Function
```

```

'=====
' The InParam() function is the heart of this article. When
' the query runs, this function causes a query parameter
' dialog box to display so that you can enter a list of values.
' The values you enter are interpreted in the same way as if you
' had entered them within the parentheses of the In() operator.
'=====
Function InParam (Fld, Param)
 Dim stToken as String
 'The following two lines are optional, making queries
 'case-insensitive
 Fld = UCase(Fld)
 Param = UCase(Param)
 If IsNull(Fld) Then Fld = ""
 Do While (Len(Param) > 0)
 stToken = GetToken(Param, ",")
 If stToken = LTrim$(RTrim$(Fld)) Then
 InParam = -1
 Exit Function
 Else
 InParam = 0
 End If
 Loop
End Function

```

2. Close and save the module.
3. Create a new query based on the Customers table. Drag any fields you want to the query grid.
4. Add the following field to the query grid:

```

 FieldName: InParam([Customer ID],[Enter a list of IDs separated
 by commas and no spaces:])
 Show: False
 Criteria: True

```

NOTE: The value InParam(...) shown for the FieldName should be entered as one statement on a single line. The InParam() function works with Integer fields as well as with Text fields.

5. From the Query menu, choose Parameters. Enter the following parameter with a Text data type:

```

 Enter a list of ID's separated by commas and no spaces:

```

6. Choose OK and run the query.

You are prompted to enter a list of parameters. The following message is displayed in the dialog box:

```

 Enter a list of ID's separated by commas and no spaces:

```

7. Type the following values, without spaces:

```

 BLUEL,CACTP,DOLLC

```

NOTE: In version 2.0, type

CHOPS,EASTC,FAMIA

All records meeting the above criteria are displayed.

Additional reference words: 1.00 1.10 2.00 Queries AB parameters IN

KBCategory: kbusage

KBSubcategory: QryParm

## **INF: How to Create a Running Sum on a Form**

**Article ID: Q103183**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

While reports have a RunningSum property, forms do not. This article describes a method to create a running sum on a form.

### MORE INFORMATION

=====

1. You need a unique, sequential ID field, such as a Counter or a Date/Time field.

NOTE: If such a field does not already exist, you can create one by opening the table in Design view and adding a new field.

2. Start Microsoft Access and open the sample database NWIND.MDB.
3. Create a new, blank form based on the Orders table.
4. Set up the form as follows:

NOTE: The underscore (\_) is used in the code below as a line-continuation character to improve readability. You must enter the entire text as one line.

Form: Form1

-----

RecordSource: Orders

Form Controls

-----

Text Box: Order ID

ControlSource: Order ID

Text Box: Order Amount

ControlSource: Order Amount

Text Box: Total

ControlSource: =DSum("[Order Amount]", "Orders", \_  
"[Order ID] <= Forms![Form1]![Order ID]")

The DSum() function calculates the sum total of all orders with an Order number less than or equal to the currently displayed order number. For this reason, it is important to have a unique and sequential ID field.

Additional reference words: 1.00 1.10 2.00 subtotal query

KBCategory: kbusage

KBSubcategory: FmsOthr





## INF: How to Create a Top 10 Report

Article ID: Q109319

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a method for creating a Top 10 report, which is a report that lists the first 10 records, based on the report's sort order.

Note that in Microsoft Access version 2.0, you can base a report on a top values query. For more information about creating a top values query in Microsoft Access 2.0, search for "TopValues" then "TopValues Property" using the Microsoft Access 2.0 Help menu.

### MORE INFORMATION

=====

The following example demonstrates how to create a Top 10 report using a calculated control's RunningSum property to provide a dynamic counter for each record. A macro called from the report's OnFormat property then cancels the formatting of records whose counter falls outside the range you want to print:

1. Open the sample database NWIND.MDB.
2. Create the following new report, and then save it as Top 10 Report:

Report: Top 10 Report

-----  
RecordSource: Sales By Sale Amount

Detail Section

-----  
OnFormat: Top 10 Macro

Text Box

-----  
ControlName: Sale Amount  
ControlSource: Sale Amount

NOTE: In Microsoft Access 2.0, the ControlName property is called the Name property.

Text Box

-----  
ControlName: Order ID  
ControlSource: Order ID

Text Box

-----  
ControlName: Company Name

ControlSource: Company Name

Text Box

-----  
ControlName: Counter  
ControlSource: =1  
Visible: No  
RunningSum: Over All

3. From the View menu, choose Sorting And Grouping. Enter the following in the Sorting And Grouping window, and then close the window:

Field/Expression: Sale Amount  
Sort Order: Descending

4. Create the following new macro, and then save it as Top 10 Macro:

Macro: Top 10 Macro  
-----  
Condition: [Counter]>10  
Action: CancelEvent

5. Preview the report. The counter control will number each record, and the macro will prevent any record whose counter is greater than 10 from being printed. Since the report is sorted on Sale Amount, the 10 largest sales will be printed.

Notes

-----

- If you use this method with a Grouping/Totals report, the totals will include the records that are not printed, so the totals may not appear to be correct. To avoid this discrepancy, you can create a dynamic counter in the query. For additional information about this method, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q94397  
TITLE : Adding a Dynamic Counter to a Query to Count Records

- You can use this method with a Grouping/Totals report to show the top 10 records per group. To do this, change the counter control's RunningSum property from Over All to Over Group.
- You can use this method with a Grouping/Totals report to show the top 10 groups. To do this, add the counter control to the report's group header instead of to the detail section, and add the macro to the OnFormat property of the group header, the detail section, and the group footer.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 19, "Sorting and Grouping Data," pages 450-464

Additional reference words: 1.00 1.10 2.00 runningsum ten

KBCategory: kbusage

KBSubcategory: RptOthr



## **INF: How to Create an AfterUndo Form Event**

**Article ID: Q123595**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article demonstrates how to create and use a form module procedure called AfterUndo. The AfterUndo procedure runs when edits made to the current record are undone.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

When you choose Undo Record from the Edit menu or press the ESC key twice to undo changes to the current record in a form, there is no built-in form event that is triggered. The AfterUndo procedure simulates an AfterUndo event so that you can restore calculations that may have been set while the record was being edited.

Follow these steps to create the AfterUndo procedure:

1. Open the form to which you want to add the AfterUpdate procedure in Design view and then add a text box with the following properties to the form:

```
Name: txtEditModeChange
ControlSource: =[Form].[Dirty] & CheckUndo([Form])
Visible: No
```

2. Add the following event procedure to the form's AfterUpdate property:

```
Sub Form_AfterUpdate ()
 ' Since the record is being saved, which changes the edit mode, the
 ' bookmark should be reset so that it will appear to the CheckUndo()
 ' function that the user moved to another record.
 PrevBookmark = Null
End Sub
```

3. Add the following lines to the form module's Declarations section:

```
Option Explicit
Dim PrevBookmark
```

4. Add the following procedures to the form module:

```
Function CheckUndo (F As Form) As Variant
```

```

Dim CurrBookmark

' Is the record clean (not dirty)?
If Not F.Dirty Then
 ' If so, get the current bookmark.
 On Error Resume Next
 CurrBookmark = F.bookmark

 ' If an error occurred, this is the new record.
 If Err Then CurrBookmark = "NewRecord"

 ' Determine if the edit change occurred on the same record (the
 ' record was undone, as opposed to moving to another record).
 If CurrBookmark = PrevBookmark Then
 ' The record was undone.
 AfterUndo
 Else
 ' The record was not undone (moved to another
 ' record). Record the bookmark of the current
 ' record for the next iteration.
 PrevBookmark = CurrBookmark
 End If
End If

End Function

Sub AfterUndo ()
 ' Add the code you want to run when the record is undone here.
End Sub

```

#### How to Use the AfterUndo Procedure

-----

The following example demonstrates how to use the AfterUndo procedure:

1. Open the sample database NWIND.MDB.
2. Open the Employees form in Design view and follow steps 1-4 above.
3. Add the following code to the AfterUpdate procedure:

```

Sub AfterUndo ()
 MsgBox "Undone"
End Sub

```

4. View the form in Form view.
5. Modify any field in any record in the form.
6. From the Edit menu, choose Undo Record. The "Undone" message box appears.
7. From the Records menu, choose Go To then choose New.
8. Type any text in any field in the new record.
9. Press the ESC key. The "Undone" message box appears.

## REFERENCES

=====

For more information about determining whether a record on a form is being edited, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122294

TITLE : INF: How to Detect If a Form Is Being Edited

For more information about undoing changes, search for "undo" then "Undoing Changes" using the Microsoft Access Help menu.

For more information about event procedures, search for "event procedures" then "Creating an Event Procedure" using the Microsoft Access Help menu.

Additional reference words: 2.00 onundo

KBCategory: kbusage

KBSubcategory: FmsEvt

**INF: How to Create an SQL Pass-Through Query in Access Basic**  
**Article ID: Q112108**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

This article demonstrates how to create an SQL pass-through (SPT) query in Access Basic. You can use SPT queries to pass SQL statements directly to an ODBC data source, avoiding the need to attach tables.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual, Chapter 3, "Introducing Access Basic."

MORE INFORMATION

=====

You can use SPT queries to send commands directly to an ODBC database server (such as Microsoft SQL Server). Using Access Basic, you can write a function that will create an SPT query. When you run the query, it will send commands directly to the ODBC database server for processing.

The following example demonstrates how to create an SPT query in Access Basic:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

1. Create the following sample Access Basic code:

```
'-----
' Declarations Section
'-----
Option Explicit

'-----
' FUNCTION: CreateSPT()
' PURPOSE:
' Creates an SQL pass-through query using Access Basic using the
' supplied arguments:
' SPTQueryName: the name of the query to create
' SQLString: the query's SQL string
' ConnectString: the ODBC connect string, this must be at
' least "ODBC;"
'-----
Function CreateSPT (SPTQueryName As String, SQLString As String, _
 ConnectString As String)
 Dim mydatabase As Database, myquerydef As QueryDef
```

```
Set mydatabase = DBENGINE.Workspaces(0).Databases(0)
Set myquerydef = mydatabase.CreateQueryDef(SPTQueryName)

myquerydef.connect = ConnectString
myquerydef.sql = SQLString
myquerydef.Close
End Function
```

2. To call this function, type the following in the Immediate window:

```
? CreateSPT("MySptQuery", "sp_help", "ODBC;")
```

This will create the query MySptQuery and add it to the list of queries in the Database window. When you run MySptQuery it will prompt you for the ODBC connect string and run the stored procedure sp\_help, which returns a list of tables from the SQL Server.

To create an SPT query called Test that would return a list of all the records from the Authors table located on the Red server in the Pubs database, you would type

```
? CreateSPT("Test", "Select * from authors", _
"ODBC;DSN=Red;Database=Pubs;USID=JOE;PWD=JOE")
```

in the Immediate window. This example also includes the UserId and password (both "Joe") arguments in the ODBC connect string. Note that if you do not supply at least "ODBC;" as the connect string, you will receive the following error message:

```
Syntax error in SELECT statement.
```

#### REFERENCES =====

For more information about ODBC connect strings, query on the following words here in the Microsoft Access Knowledge Base:

ODBC and Connect and Usid

For more information about the syntax of SQL pass-through queries, please consult the documentation for your server.

Additional reference words: 2.00 spt  
KBCategory: kbusage  
KBSubcategory: ObcOthr



**INF: How to Create Custom Domain Function Similar to DCount()  
Article ID: Q103401**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Custom domain functions can be used to provide functionality other than that provided by the standard domain functions DLookup(), DMin(), DLast(), and so on.

This article demonstrates how to write a custom domain function that is similar to the DCount() function. In addition, it demonstrates a sample function, DFix(), to overcome limitations in concatenating variables in criteria strings. Information about the limitations of custom domain functions is at the end of this article.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

The DRecCount() sample custom domain function is useful when the domain is a totals or aggregate query based on an attached SQL table. The DRecCount() custom domain function will accurately return the count of such a dynaset.

To create these examples, open a new module within Microsoft Access and add the following functions with the appropriate declaration section:

```
'-----
' GLOBAL DECLARATION
'-----
Option Compare Database
Option Explicit

Function DRecCount (FieldName, DomainName, Criteria)
'-----
' Use DRecCount to return a count
' of records when the domain is a query based on a
' totals/aggregate query on an attached SQL table.
'-----
Dim MyDB As Database, Myset As Dynaset

If VarType(FieldName) <> 8 Or Len(FieldName) = 0 Then
 MsgBox "You Must Specify a Field name", , "DRecCount"
```

```

Exit Function
End If
If VarType(DomainName) <> 8 Or Len(DomainName) = 0 Then
 MsgBox "You Must Specify a Domain name", , "DRecCount"
 Exit Function
End If
If VarType(Criteria) <> 8 And Not IsNull(Criteria) Then
 MsgBox "Invalid Criteria", , "DRecCount"
 Exit Function
End If

Set MyDB = CurrentDB()
Set Myset = MyDB.CreateDynaset(DomainName)

If FieldName <> "*" Then
 If Len(Criteria) > 0 Then
 Criteria = Criteria & " AND "
 End If
 Criteria = Criteria & "[" & FieldName & "] Is Not Null"
 Myset.Filter = Criteria
 Set Myset = Myset.CreateDynaset()
End If
If Myset.EOF Then
 DRecCount = 0
Else
 Myset.MoveLast
 DRecCount = Myset.recordcount
End If
Myset.Close
MyDB.Close
End Function

Function DFix (ByVal T, DQuote As Integer)
'-----
' Fixes string arguments that are passed
' to Criteria in domain functions
'
' DQuote should be TRUE or -1 if Double Quotes (") delimit Criteria
' DQuote should be FALSE or 0 if Single Quotes (') delimit Criteria
'
' e.g. this gives an error
' (note the quote (') in the data)
' X="Mike's Diner"
' A=DRecCount("*","Clients","Name='" & X & "'")
'
' Use either:
' X=DFix("Mike's Diner",False)
' Or:
' A=DRecCount("*","Clients","Name='" & DFix(X,False) & "'")
'-----
Dim P As Integer, OldP As Integer, Q As String * 1

If VarType(T) = 8 Then
 If DQuote = 0 Then
 Q = "'"
 Else
 Q = """"
 End If

```

```

End If
P = InStr(T, Q)
Do While P > 0
 OldP = P + 2
 T = Left$(T, P) & Q & Mid$(T, P + 1)
 P = InStr(OldP, T, Q)
Loop
End If
DFix = T
End Function

```

#### General Limitations to Custom Domain Functions

---

- You cannot use Forms!FormName!ControlName or Form.ID in quotation marks. The following example is not allowed:

```
DRecCount("ID", "LOG", "Name=Form.Name")
```

Instead, use:

```
DRecCount("ID", "LOG", "Name = '" & DFix(Name, False) & "'")
```

Note that the DFix() function is another custom function that is demonstrated in this article.

- You must always specify a criteria, even if it is "" or Null.
- Do not place field names or domain names in brackets. For example, use "First Name" instead of "[First Name]".

NOTE: DFix() can be used with other criteria strings, such as the FindRecord method, or when building custom SQL criteria.

#### REFERENCES

=====

For more information on custom domain functions, query on the following words here in the Microsoft Knowledge Base:

custom and domain and function

Additional reference words: 1.00 1.10 2.00 Modules ab  
 KBCategory: kbprg  
 KBSubcategory: PgmOthr

**INF: How to Create Custom Navigation (VCR) Buttons on a Form**  
**Article ID: Q104683**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The navigation (VCR) buttons that appear on a form's horizontal scroll bar provide a convenient way of navigating among records.

The following information describes how to create custom navigation buttons on your own forms.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

The following sample module demonstrates Access Basic functions that can be used to create custom buttons for navigating among first, last, previous, and next records on a form. Functions are provided to navigate through forms as well as subforms.

To create custom navigation buttons on a form, create a new module with the following Access Basic code:

```
'*****
' MODULE DECLARATION SECTION
'*****
Option Explicit
DimRetVal As Variant

'*****
' MODULE FUNCTIONS
'*****

Function GotoFirstRecord ()
 RetVal = GotoRecord(A_FIRST)
End Function

Function GotoLastRecord ()
 RetVal = GotoRecord(A_LAST)
End Function

Function GotoNextRecord ()
 RetVal = GotoRecord(A_NEXT)
End Function
```

```

Function GotoPrevRecord ()
 RetVal = GotoRecord(A_PREVIOUS)
End Function

Function GotoFirstSubRecord (SubControlName As String)
 DoCmd GoToControl SubControlName
 RetVal = GotoFirstRecord()
End Function

Function GotoLastSubRecord (SubControlName As String)
 DoCmd GoToControl SubControlName
 RetVal = GotoLastRecord()
End Function

Function GotoNextSubRecord (SubControlName As String)
 DoCmd GoToControl SubControlName
 RetVal = GotoNextRecord()
End Function

Function GotoPrevSubRecord (SubControlName As String)
 DoCmd GoToControl SubControlName
 RetVal = GotoPrevRecord()
End Function

Function GotoRecord (Direction)
 On Error Resume Next
 DoCmd GoToRecord , , Direction
End Function

```

#### How to Use the Custom Navigation Functions on a Form

---

The steps listed below demonstrate how to use the Access Basic functions detailed above to add custom navigation buttons to the Orders and the Orders Subform forms in the sample database NWIND.MDB included with Microsoft Access:

1. Open the Orders form in Design view.
2. Create four buttons on the form to facilitate navigation among the Orders records. Place these buttons side by side, with the following properties:

```

Command Button : btnGotoFirstRecord
Caption: <<
OnClick: =GotoFirstRecord()
(OnPush in version 1.x)

```

```

Command Button : btnGotoPrevRecord
Caption: <
OnClick: =GotoPrevRecord()
(OnPush in version 1.x)

```

```

Command Button : btnGotoNextRecord
Caption: >
OnClick: =GotoNextRecord()
(OnPush in version 1.x)

```

```
Command Button : btnGotoLastRecord
Caption: >>
OnClick: =GotoLastRecord()
(OnPush in version 1.x)
```

3. Create four buttons on the Orders Subform form to facilitate navigation among the records. Place these buttons side by side, with the following properties:

```
Command Button : btnGotoFirstSubRecord
Caption: <<
OnClick: =GotoFirstSubRecord("Orders Subform")
(OnPush in version 1.x)
```

```
Command Button : btnGotoPrevSubRecord
Caption: <
OnClick: =GotoPrevSubRecord("Orders Subform")
(OnPush in version 1.x)
```

```
Command Button : btnGotoNextSubRecord
Caption: >
OnClick: =GotoNextSubRecord("Orders Subform")
(OnPush in version 1.x)
```

```
Command Button : btnGotoLastSubRecord
Caption: >>
OnClick: =GotoLastSubRecord("Orders Subform")
(OnPush in version 1.x)
```

NOTE: These buttons can be placed on the main form or on the subform. The argument being passed to the function ("Orders Subform") is the name of the subform control.

```
record selector vcr
Additional reference words: 1.00 1.10 2.00 scrollbar navigate goto
KBCategory: kbusage
KBSubcategory: FmsOthr
```

## INF: How to Create For Next Loops In a Macro

Article ID: Q90815

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

To perform a For Next loop in a Microsoft Access macro, you need to use a combination of correct macro structure and the RunMacro action with appropriate information supplied in either the Repeat Count or Repeat Expression parameters.

To enhance your macro, you can use a different macro structure and include a call to an Access Basic function that maintains a counter, or use the SetValue macro action to maintain and update a counter value in a text box on a hidden form. The benefit of maintaining a counter is that you can use the value of the counter in your macro processing just as in a Access Basic For Next loop.

### MORE INFORMATION

=====

To perform a For Next loop in a macro, you would use a macro similar to the following examples. To create the following sample macros, open a new macro and choose Macro Names from the View menu. Add the following macro names and actions, then save this macro group as "For\_Next\_Loops".

#### For Next Loop Using Repeat Count

-----

| Macro Name     | Action   |
|----------------|----------|
| For_Next_Loop1 | 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<br>RunMacro |
| Loop3          | MsgBox<br>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<br>RunMacro |
| Loop3          | MsgBox<br>RunCode   |



### For\_Next\_Loop3 Actions

```

RunCode
 Function Name: =LoopCount(1)
RunMacro
 MacroName: For_Next_Loops.Loop3
 Repeat Expression: LoopCount(3)<=10
```

### Loop3 Actions

```

MsgBox
 Message: ="Loop Count: " & LoopCount(3)
RunCode
 Function Name: =LoopCount(2)
```

In this example, the For\_Next\_Loop3 macro runs the LoopCount(1) function by using the RunCode macro action. This causes the function to initialize the counter to a value of 1. It then runs the Loop2 macro using the RunMacro action until the Repeat Expression parameter is false. LoopCount(3), which is used in the conditional expression, returns the current value of the counter. Therefore, when the counter reaches 11, the loop has executed 10 times, and when the RunMacro action checks the condition, it evaluates to false and will not call the Loop3 macro again. The Loop3 macro displays a message box that indicates which instance of the loop is being executed by referring to the counter value returned by the LoopCount(3) function call. The Loop3 macro then increments the counter using the RunCode macro action, which calls the function LoopCount(2).

The benefits of this example are that a counter is available that indicates which instance of the loop is being executed and you do not need to create and open a form with a control on it to contain the counter value. The drawback to this example is that you need to create the LoopCount() Access Basic function prior to using this example; however, after you create the function it is available anywhere in Microsoft Access.

The Access Basic function procedure LoopCount() returns the current value of a counter that the function declares and maintains. Depending on which parameter is passed (1, 2, or any other number), LoopCount() will initialize, increment, or return the counter's current value. Below is the Access Basic code for the LoopCount() function procedure.

```
Function LoopCount (Action)
 Static LoopCounter ' Static variable to hold counter
 If Action = 1 Then ' Initialize counter
 LoopCounter = 0
 ElseIf Action = 2 Then ' Increment counter
 LoopCounter = LoopCounter + 1
 End If
 LoopCount = LoopCounter ' Return value
End Function
```

### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.0, chapter 3,  
pages 6-7 and 10-11

Microsoft Access "Language Reference," version 1.0, pages 246-248,  
316-317, 412-414, and 437-438

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: McrOthr

## INF: How to Create Synchronized Combo Boxes

Article ID: Q98660

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to create a combo box that is filtered to list only items that are related to an item selected in a prior combo box.

### MORE INFORMATION

=====

In the following example, the first combo box lists the category names from the Categories table, and the second combo box lists the product names from the Products table. When you select a category name in the first combo box, the second combo box is filtered to list only the product names for that category:

1. Open the sample database NWIND.MDB.
2. Create the following new query based on the Categories table, and then save it as Categories Lister:

Query: Categories Lister

-----  
Field name: Category ID  
Show: Yes  
Field name: Category Name  
Show: Yes  
Sort: Ascending

3. Create the following new query based on the Products table, and then save it as Products Lister:

NOTE: In the following sample query, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this query.

Query: Products Lister

-----  
Field name: Product ID  
Show: Yes  
Field name: Product Name  
Show: Yes  
Sort: Ascending  
Field name: Category ID  
Show: Yes  
Sort: Ascending  
Criteria: IIF(IsNull([Forms]![Selector Form]![Category \_  
Selection]),[Category ID],[Forms]![Selector \_  
Form]![Category Selection])

Note that using the IIf() and IsNull() functions in this query lets you see a list of all the products if no category is selected.

4. Create the following new macro, and then save it as Reset Product Selection:

| Macro Name              | Action      | Argument          |
|-------------------------|-------------|-------------------|
| Reset Product Selection | Requery     | Product Selection |
|                         | GoToControl | Product Selection |

5. Create the following new unbound form, and then save it as Selector Form:

Object: Combo Box

-----  
ControlName: Category Selection  
RowSourceType: Table/Query  
RowSource: Categories Lister  
ColumnCount: 2  
ColumnWidths: 0 in;1 in  
BoundColumn: 1  
After Update: Reset Product Selection

Object: Combo Box

-----  
ControlName: Product Selection  
RowSourceType: Table/Query  
RowSource: Products Lister  
ColumnCount: 3  
ColumnWidths: 0 in;1 in; 0 in  
BoundColumn: 1

6. View the form in Form view. In the Category Selection box, select a category and note that the Product Selection box lists only the products for that category.
7. In the Category Selection box, select a different category. Note that the Product Selection box is reset and lists the appropriate products for the new category.

#### REFERENCES

=====

For more information on requerying, search for "requery" then "Requery Action" using the Microsoft Access Help menu.

Microsoft Access "User's Guide," version 1.0, pages 237 and 541

You can find information, instructions, and examples in the Solutions sample application (SOLUTION.MDB) included with Microsoft Access version 2.0. For more information about how to list only items that are related to an item selected in a prior combo box, open the SOLUTION.MDB database usually located in the ACCESS\SAMPAPPS directory. Select Get More Mileage From Combo Boxes, List Boxes, Subforms, And Subreports in the Select A Category Of Examples box, then select Limit The Contents Of One List Based

On The Value Selected In Another in the Select An Example box.

Additional reference words: 1.00 1.10 2.00 combobox parameter listbox

KBCategory: kbusage

KBSubcategory: FmsCmbo

**INF: How to Create Tasks in MS Project for Windows Using DDE**  
**Article ID: Q103273**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Before data can be sent to Microsoft Project version 3.0 for Windows, a task must exist in the active project. This article contains sample code to start Project, create 10 tasks, and fill the task fields with data.

MORE INFORMATION

=====

This article assumes that the user has some knowledge of Access Basic and is intended as a demonstration of the technique required to create a task in Project. The following function is not intended to be replicated exactly.

1. Create a new module in a Microsoft Access database. Add the following statement to the Declarations section of the module:

```
'-----
'Declarations Section
'-----
Option Explicit
```

2. Add the following function to the module:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----
'Start of Function
'-----
```

Function dde\_createtasks (ProjectPath)

```
 Const DDE_ERROR = 282 'Define DDE return error.
 Dim Temp, Mynum, Chan1, Chan2 'Define variables.
```

```
 On Error GoTo StartUp 'If Project is not open, start it.
```

```
 Chan1 = DDEInitiate("winproj", "system")
 'Initiate a channel to Project.
```

```
 On Error GoTo QuitNow:
```

```
 Chan2 = DDEInitiate("winproj", DDERequest(Chan1, "ActiveProject"))
 'Initiate a channel to the active Project.
```

```

For Mynum = 1 To 10
 DDEExecute Chan2, "EditGoto .ID=[" & Mynum & "]"
 'Go to the task.
 DDEExecute Chan2, "SetField .Field=[Name] .Value=MyNewTask" & _
 Mynum & " .Create=[Yes]"
 'Create a new task.
 DDEPoke Chan2, "T((" & Mynum & "), (Name,Duration))", "Task_
 Number " & Mynum & Chr(9) & Mynum & "d"
 'Send information to Project.
Next

```

```

DDETerminateAll 'Terminate the DDE channel.

```

```

Exit Function

```

```

QuitNow:

```

```

 DDETerminateAll 'Terminate the DDE channel.
 Exit Function

```

```

StartUp:

```

```

 If Err = DDE_ERROR Then
 Temp = Shell(ProjectPath, 6) 'Start Project.
 Resume ' Re-execute DDEInitiate() function.
 End If

```

```

End Function

```

```

'-----
'End of Function
'-----

```

3. To test this function, type the following in the Immediate window

```
?dde_createtasks ("")
```

where "" is the path to Project.

Project installs two files, MACRO.WRI and DDEINFO.WRI, in the program directory that contain more information about macro commands and dynamic data exchange (DDE) functionality. These files can be opened in Microsoft Windows Write.

REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00 dde

KBCategory: kbinterop

KBSubcategory: IntpOthr

**INF: How to Create, Debug, and Use a Microsoft Access Library**  
**Article ID: Q88175**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes a Microsoft Access library and discusses 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 Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

Access Basic Library Definition

-----

When you write a Microsoft Access application, such as the sample database NWIND.MDB, 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 were not the case, you could not use a Wizard outside of the database that the Wizard program and system objects reside in. In order to make 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 the .INI file (MSACCESS.INI in Microsoft Access version 1.x, MSACC20.INI in version 2.0) and add an line to the Libraries section. When you open the .INI file 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 .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 its own tables that are to be written to at



any point in your program, you would specify "rw" rather than "ro." For example, to add an application called STOCKAPP.MDB that uses its own tables that can be modified, add the following line to the .INI file:

```
stockapp.mdb=rw
```

The complete [Libraries] section might look like:

```
[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. After restarting Microsoft Access you would be able to 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 database objects from STOCKAPP.MDB, you cannot see them in the Database window.

Note that 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

---

##### Microsoft Access version 1.x:

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 message boxes at milestone areas of the code so that you always have an idea of which code is being executed.

##### Microsoft Access version 2.0:

Add the following line to the [Options] section of the MSACC20.INI file. If the [Options] section does not exist, add it to the bottom of the file.

```
DebugLibraries=True
```

The completed section might look like:

```
[Options]
DebugLibraries=True
```

This enables the code debugger to appear when a run-time error occurs in a library module. You can also get to the library code using the View Procedures command when you are in a module's Design view. Note that this will tend to reduce performance because all the library code will have to be recompiled. Remove this option when the code debugger is not necessary.

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 CurrentDB() returns the database object from the current open user database. Because of this, it is easy to confuse one for the other.

Domain Functions  
-----

Domain functions should be used with caution in library modules as they can only process user tables and not library tables.

Macros in a Library  
-----

Microsoft Access version 1.x:

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.

Microsoft Access version 2.0:

In Microsoft Access 2.0, you can call a macro from a Library database. However, due to lack of error handling, they should be avoided except for implementing custom menus.

Keys May Be Redefined in User Database  
-----

Since the user database may contain an AutoKeys macro to redefine the meaning of keys, you should use caution when using SendKeys in your Library application. Use other macro actions where appropriate and use hot-keys and {tab} rather than arrow keys when manipulating a dialog box.

## USysAddIns Table

-----

The Microsoft Access 2.0 Add-In Manager will process entries in the USysAddIns table of a library database to update the MSACC20.INI [Libraries] and [Menu Add-Ins] sections. This makes installing and uninstalling libraries more manageable for users. See Chapter 15 of the Microsoft Access "Building Applications" manual for more information.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: MdlLib

**INF: How to Declare C++ Functions Called from Access Basic**  
**Article ID: Q109338**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

The following steps describe how to call C++ dynamic-link library (DLL) functions in Access Basic:

1. Make sure to define the functions within your C++ code with the "extern" declaration. The following example demonstrates the syntax for the extern declaration:

```
extern "C" {return type} FAR PASCAL __export {function name}
 ({parameters})
```

An example of an actual declaration in your C++ code might look like:

```
extern "C" int FAR PASCAL __export my_func ()
```

To call the sample function above you would define the Declare statement as follows:

```
Declare Function my_func Lib "MYLIB.DLL" () As Integer
```

2. Make sure that the declaration is not set for case-sensitive exports.

REFERENCES

=====

For more information on calling DLLs, search on "declare external procedures" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 Programming

KBCategory: kbprg

KBSubcategory: PgmApi

## **INF: How to Delete Duplicate Records from a Table**

**Article ID: Q109329**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to remove duplicate records from a table using primary keys. This is particularly useful for tables with more than 10 fields, where it is not possible to use Group By in queries.

### MORE INFORMATION

=====

The following example demonstrates how to remove duplicate records from a table:

1. Make a copy of the table from which you want to remove the duplicate records. To copy a table, select the table in the Database window and then choose Copy from the Edit menu. Then choose Paste from the Edit menu. Enter a new name for the table and select the Structure Only option. Choose OK.
2. Open the new table in Design view and select the field(s) which you want to check for duplicates. To make your selection the primary key for the table press the Primary Key button or choose Set Primary Key from the Edit menu.
3. Save and then close the table.
4. Create an append query and append the data from the original table into the new table. Because the primary key fields in the new table will not permit duplicates, the error message

Errors were encountered...Proceed anyway?

will occur when you run the append query. Choose OK. The new table will contain the data from the original table, but without duplicates.

NOTE: A fast way to remove duplicate records from a small table is to select the table in the Database window, choose Copy from the Edit menu, then choose Paste from the Edit menu. In the Paste Table As dialog box, type a new name for the table in the Table Name box, and select the Append Data To Existing Table option button. When you choose OK, a new table will be created with the data from the original table, without the duplicate records.

### REFERENCES

=====

For more information on deleting duplicate records, query on the following words here in the Microsoft Knowledge Base:

query and unique and duplicate and filter

Microsoft Access "User's Guide," version 1.0, pages 28-29, 32-33, and 174-178

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryMktbl

**INF: How to Detect When MS Access Is Activated or Deactivated**  
**Article ID: Q124393**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

This article describes how to detect and run code when Microsoft Access is activated or deactivated. Switching to Microsoft Access from another application activates Microsoft Access, and switching from Microsoft Access to another application deactivates Microsoft Access.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The following steps demonstrate how to create a form called Access\_Activation that will detect when Microsoft Access is activated or deactivated. The form contains two empty procedures called Access\_Activate and Access\_Deactivate into which you can insert the code you want to run when Microsoft Access is activated or deactivated. When the Access\_Activation form is open in Form view, switching to Microsoft Access from another application will cause the Access\_Activate procedure to run. Switching from Microsoft Access to another application will cause the Access\_Deactivate procedure to run.

The Access\_Activation form must be open in order to detect when Microsoft Access is activated and deactivated. To make sure the form is always open, you could open the form in a hidden state in your database's Autoexec macro. To open the form in a hidden state, set the Window Mode argument of the OpenForm macro action to Hidden.

1. Open any database and create a blank new form. Set the following properties for the form:

```
OnTimer: [Event Procedure]
TimerInterval: 500
```

2. From the View menu, choose Code. Type the following lines in the form module's Declarations section.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Option Explicit
Declare Function GetActiveWindow Lib "User" () As Integer
Declare Function GetParent Lib "User" (ByVal hWnd As Integer) _
```

```
As Integer
Declare Function GetFocus Lib "User" () As Integer
```

3. Enter the following code for the OnTimer property event procedure:

```
Sub Form_Timer ()
 Dim RetVal As Integer
 Dim CurrhWnd As Integer
 Dim AccesshWnd As Integer
 Static ActiveApphWnd As Integer

 ' Get the Microsoft Access window handle (hWnd).
 AccesshWnd = GetAccesshWnd()

 ' Get the active application window handle (hWnd).
 CurrhWnd = GetActiveApphWnd()

 ' The first time through, just record the current
 ' window handle.
 If ActiveApphWnd = 0 Then
 ActiveApphWnd = CurrhWnd
 Exit Sub
 End If

 ' Determine if the current window handle differs from the
 ' previous window handle (focus change).
 If CurrhWnd <> ActiveApphWnd Then
 ' Record the current window handle.
 ActiveApphWnd = CurrhWnd

 ' Determine if the current handle is the Microsoft Access
 ' handle (activate Microsoft Access?).
 If ActiveApphWnd = AccesshWnd Then
 Access_Activate
 Else
 Access_Deactivate
 End If
 End If
End Sub
```

4. Enter the following procedures in the form module:

```
Sub Access_Activate ()
 ' Insert the code that you want to run when Microsoft Access
 ' is activated here.
End Sub
```

```
Sub Access_Deactivate ()
 ' Insert the code that you want to run when Microsoft Access
 ' is deactivated here.
End Sub
```

```
Function GetAccesshWnd ()
 GetAccesshWnd = GetTopMosthWnd(Me.hWnd)
End Function
```

```
Function GetActiveApphWnd ()
```



```
 GetActiveAppWnd = GetTopMosthWnd(GetActiveWindow())
End Function
```

```
Function GetTopMosthWnd (ByVal hWnd)
 Dim hWndTopMost As Integer
 hWndTopMost = hWnd

 ' Find the top window without a parent window.
 While hWnd <> 0
 hWndTopMost = hWnd
 hWnd = GetParent(hWnd)
 Wend

 GetTopMosthWnd = hWndTopMost
End Function
```

5. Close the module and save the form as Access\_Activation.

#### How to Use the Access\_Activation Form

---

The following steps demonstrate how to use the Access\_Activation form:

1. Open any database and create the Access\_Activation form as described above.
2. Substitute the following Access\_Activate procedure for the Access\_Activate procedure you created in step 4 above:

```
Sub Access_Activate ()
 MsgBox "Microsoft Access was activated"
End Sub
```

3. View the Access\_Activation form in Form view.
4. Press ALT+TAB to switch from Microsoft Access to another application.
5. Press ALT+TAB to switch back to Microsoft Access. A message box with the text "Microsoft Access was activated" is displayed.

Additional reference words: 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

## INF: How to Determine If a Form Exists in a Database

Article ID: Q92825

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0  
-----

### SUMMARY

=====

This article discusses how to obtain a dynaset containing the names of all the forms in a database whether or not they are active.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

If you want to obtain a complete list of forms whether or not they are active, you can create a dynaset that queries MSysObjects for all the forms stored in the current database. MSysObjects stores the names of every object in a database, and each record includes the type of object represented by a code and stored in a field called Type. The code that represents forms in the Type field is -32768. Given this information, you can create a dynaset that obtains the names of all the objects in MSysObjects that are forms:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Dim DB As Database, S As Dynaset
Set DB = 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 of all the forms in the current user database. You can then process the information as shown in the example below. The procedure below accepts a form name and opens it if it exists in the database, or presents a message indicating that it does not exist in the database:

```
Option Explicit
Sub FormOpener (FormToOpen$)
 Dim DB As Database
 Dim S As Dynaset
 Set DB = CurrentDB()
 Set S = DB.CreateDynaset("SELECT Name FROM MSysObjects _
 WHERE Type = -32768;")
```

```

S.FindFirst "Name = '" & FormToOpen$ & "'"
If S.NoMatch Then
 MsgBox FormToOpen$ & " does not exist in this database"
Else
 DoCmd OpenForm FormToOpen$
End If
End Sub

```

In Microsoft Access version 2.0, you can use the following method that uses data access objects (DAO):

```

Option Explicit
Sub DAOFormOpener (FormToOpen$)
 Dim mydb As Database, x As Integer
 Set mydb = dbengine(0)(0)

 For x = 0 To dbengine(0)(0).containers("forms").documents.count - 1
 If FormToOpen$ = _
 dbengine(0)(0).containers("forms").documents(x).name Then
 DoCmd OpenForm FormToOpen$
 Exit Sub
 End If
 Next x
 MsgBox FormToOpen$ & " does not exist in this database."
End Sub

```

Additional reference words: 1.00 1.10 2.00 VT

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: How to Determine if a Specific Windows Program Is Running**  
**Article ID: Q88167**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

There may be times when you only want one instance of an application to run under Microsoft Windows. For example, 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 use of memory and system resources. If your program determines if the application is already running, it can either start the application for the first time, or bring the existing application to the front.

To determine if a specific program is running, call the Windows application programming interface (API) function FindWindow(). FindWindow() returns the handle of the window whose class is given by the lpClassName parameter and whose window name, or caption, is given by the lpCaption parameter. If the returned value is zero, the application is not running.

MORE INFORMATION

=====

When an application is started from the Program Manager it registers the class name of the form. The window class provides information about the name, attributes, and resources required by your form. The Microsoft Access window has a class name of "OMain." For additional command class names, see further below.

By calling FindWindow() with a combination of a specific program's class name and/or the title bar caption, your program can determine whether that specific program is running.

You can determine the class name of an application by using SPY.EXE, which is supplied with the Microsoft Windows Software Development Kit (SDK) version 3.1.

If the window has a caption bar title, you can also use the title to locate the instance of the running application. This caption text is valid even when the application is minimized to an icon.

The following example shows three ways to determine if the Windows 3.1 Calculator is running. To create the program, do the following:

1. From the Program Manager File menu, choose Run and enter CALC.EXE to start Calculator.
2. From the File menu, choose New to create a new database.
3. From the File menu, choose New, then select Form to create a new blank form.

4. When the form is displayed, from the View menu, choose Toolbox.
5. Select the command button tool on the toolbar, then double-click 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() |

NOTE: In Microsoft Access version 2.0, the OnPush property has become the OnClick property.

7. Close and save the form as "FindWindow Form."
8. From within the Database window, open a new module and enter the sample Access Basic code below:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```

'=====
'Declarations section
'=====
Option Explicit
Declare Function FindWindow% Lib "user" (_
 ByVal lpClassName As Any, _
 ByVal lpCaption As Any)
'=====
'End of Declarations section
'=====

'=====
'Code section
'=====

Function CalculatorUp ()

 Const lpClassName = "SciCalc"
 Const lpCaption = "Calculator"

 'This demonstrates three different ways to call FindWindow:
 '1. The ClassName only.
 '2. The Caption only.
 '3. Both the ClassName and the Caption

 MsgBox "Calculator Handle = " & FindWindow(lpClassName, 0&)
 MsgBox "Calculator Handle = " & FindWindow(0&, lpCaption)
 MsgBox "Calculator Handle = " & FindWindow(lpClassName, lpCaption)

 'This function could return the existence of a window.
 CalculatorUp = FindWindow(lpClassName, 0&)
End Function

```

```
'=====
'End of code section
'=====
```

- 9. Close and save the module as "FindWindow Code."
- 10. From within the Database window, open the 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                             |

REFERENCES  
=====

"Programming Windows: the Microsoft Guide to Writing Applications for Windows 3," by Charles Petzold, Microsoft Press, 1990

"Peter Norton's Windows 3.0 Power Programming Techniques," by Peter Norton and Paul Yao, Bantam Computer Books, 1990

Microsoft Windows 3.0 Software Development Kit "Reference Volume 1"

Additional reference words: 1.00 1.10 2.00 WinIss windows api  
KBCategory: kbprg

KBSubcategory: PgmApi

## INF: How to Determine If a Table or Query Exists

Article ID: Q113549

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article describes a sample user-defined function, `IsTableQuery()`, that you can use to determine whether a table or query exists in a database. The sample function uses the `TableDefs` and `QueryDefs` data access objects (DAO) collections.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

The following example demonstrates how to use the sample user-defined function `IsTableQuery()` to determine whether a table or query exists in a database:

1. Open the sample database `NWIND.MDB`.
2. Create a new module and add the following line to the module's Declarations section:

```
Option Explicit
```

3. Enter the following function in the module:

```

' FUNCTION: IsTableQuery()
'
' PURPOSE: Determine if a table or query exists.
'
' ARGUMENTS:
' dbName: The name of the database. If the database name
' is "" the current database is used.
' TName: The name of a table or query.
'
' RETURNS: True (it exists) or False (it does not exist).
'

Function IsTableQuery (dbName As String, TName As String) As Integer
 Dim Db As Database, Found As Integer, Test As String
 Const NAME_NOT_IN_COLLECTION = 3265

 ' Assume the table or query does exist.
 Found = False
```



```

' Trap for any errors.
On Error Resume Next

' If the database name is empty...
If Trim$(DbName) = "" Then
 ' ...then set Db to the current Db.
 Set Db = DBEngine.Workspaces(0).Databases(0)
Else
 ' Otherwise, set Db to the specified open database.
 Set Db = DBEngine.Workspaces(0).OpenDatabase(DbName)

 ' See if an error occurred.
 If Err Then
 MsgBox "Could not find opened database " & DbName
 IsTableQuery = False
 Exit Function
 End If
End If

' See if the name is in the Tables collection.
Test = Db.TableDefs(TName).Name
If Err <> NAME_NOT_IN_COLLECTION Then Found = True

' Reset the error variable.
Err = 0

' See if the name is in the Queries collection.
Test = Db.QueryDefs(TName$.name)
If Err <> NAME_NOT_IN_COLLECTION Then Found = True

DB.Close

IsTableQuery = Found

End Function

```

4. From the View menu, choose Immediate Window.

5. Type the following line in the Immediate window, and then press ENTER:

```
If IsTableQuery("", "Orders") = True Then Print "Found it!"
```

6. Try using other valid database names, and other table and query names, in the line above.

#### REFERENCES

=====

For more information about the TableDefs and QueryDefs collections, search for "TableDefs" and "QueryDefs" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: How to Determine If Current Database Is Open Exclusively**  
**Article ID: Q117539**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to use Access Basic to determine if the current database is open exclusively in Microsoft Access version 2.0.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

When a database is open exclusively, no other process or program can access the database's .MDB file. When a database is open nonexclusively, other processes can get shared access to the .MDB file. This article describes how to use the Open statement in Access Basic to determine if the database is open exclusively.

The Open statement's Shared argument lets you attempt to access a file in shared mode. If that file is the current database, and it is open exclusively, a "Permission denied" error message will occur. No error message will occur if the database is open nonexclusively, or shared. You can trap for this error to determine in what mode the database is open.

The following sample function demonstrates how to use the Open statement to determine in what mode the current database is open. This function is most useful to Access Basic programmers who want to warn users that the database being opened is in the wrong mode:

1. Open the sample database NWIND.MDB exclusively.
2. Create a new module and enter the following line in the declarations section:

Option Explicit

3. Enter the following function in the module:

```
Function IsCurDBExclusive20 () As Integer
' Purpose: Determine if the current database is exclusive.
' Returns: 0 if database is not open exclusively.
' -1 if database is open exclusively.
' Err if any error condition is detected.
```

```
Dim db As Database
Dim hFile As Integer
```

```

hFile = FreeFile

Set db = dbengine.workspaces(0).databases(0)
If Dir$(db.name) <> "" Then
 On Error Resume Next
 Open db.name For Binary Access Read Write Shared As hFile
 Select Case Err
 Case 0
 IsCurDBExclusive20 = False
 Case 70
 IsCurDBExclusive20 = True
 Case Else
 IsCurDBExclusive20 = Err
 End Select
 Close hFile
 On Error GoTo 0
Else
 MsgBox "Couldn't find " & db.name & "."
End If
End Function

```

4. From the View menu, choose Immediate Window.
5. Type the following line in the Immediate window, then press ENTER:

```
If IsCurDBExclusive20()=True Then MsgBox "It's Exclusive!"
```

The message "It's Exclusive!" should appear.

6. Try opening the database nonexclusively and then repeating step 5. The message box should not appear.

#### REFERENCES

=====

For more information about the Open statement, search for "Open" then "Open Statement" using the Microsoft Access Help menu.

Additional reference words: 2.00 multi-user  
 KBCategory: kbusage kberrmsg  
 KBSubcategory: GnlMu

**INF: How to Determine If Current User Is in a Particular Group**  
**Article ID: Q119707**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

This article describes a sample function that you can use to determine if the current user of the database is a member of a particular group, such as the Admins group. This function can be useful when you need to determine whether the user should be allowed to access certain commands or carry out certain tasks.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The following steps describe how to create and use the sample function `CurrentUserInGroup()`:

1. Create a new module and enter the following line in the Declarations section:

```
Option Explicit
```

2. Enter the following code in the module:

```
Function CurrentUserInGroup (groupname As String)

'Purpose: Determines if the current user is in a specified group.
'Accepts: The name of a group
'Returns: True if the current user is a member of the specified group,
' false if the current user is not a member of the group.
'Assumes: The existence of a user called Developer in the Admins group,
' with no password.

On Error GoTo err_CurrentUserInGroup

Dim Myworkspace As Workspace, i As Integer
Dim mygroup As Group, myuser As User

'Create a new workspace as a member of the Admins group.
Set Myworkspace = DBEngine.CreateWorkspace("Special", "developer", "")

Set mygroup = Myworkspace.groups(groupname)
Set myuser = Myworkspace.users(CurrentUser())

For i = 0 To mygroup.users.count - 1
```

```
 If mygroup.users(i).name = myuser.name Then
 CurrentUserInGroup = True
 Exit Function
 End If
Next i
```

```
CurrentUserInGroup = False
Myworkspace.Close
```

```
Exit Function
```

```
err_CurrentUserInGroup:
 If Err = 3265 Then
 MsgBox "Invalid group name"
 Else MsgBox Error(Err)
 End If
 Myworkspace.Close
 Exit Function
```

```
End Function
```

3. From the View menu, choose Immediate Window.
4. Type the following line in the Immediate window, and then press ENTER:

```
? CurrentUserInGroup("admins")
```

If the current user is a member of the Admins group, -1 (true) is returned.  
If the current user is not a member of the Admins group, 0 (false) is returned.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 14,  
"Securing Your Application"

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: PgmOthr

**INF: How to Determine Number of Working Days Between Two Dates**  
**Article ID: Q97757**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article explains how to calculate the number of working days between two dates.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

There is no function in Microsoft Access to determine the number of working days between two given dates. To determine this, you need to call a user-defined Access Basic function. Note that the function below includes the start date and the end date, so the number of days between 02/02/93 and 02/03/93 equals one. The code for the sample function Work\_Days() is below.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

Option Explicit

```
Function Work_Days (BegDate As Variant, EndDate As Variant) As Integer
' Note that this function does not account for holidays.
 Dim WholeWeeks As Variant
 Dim DateCnt As Variant
 Dim EndDays As Integer

 BegDate = DateValue(BegDate)
 EndDate = DateValue(EndDate)
 WholeWeeks = DateDiff("w", BegDate, EndDate)
 DateCnt = DateAdd("ww", WholeWeeks, BegDate)
 EndDays = 0
 Do While DateCnt < EndDate
 If Format(DateCnt, "ddd") <> "Sun" And
 Format(DateCnt, "ddd") <> "Sat" Then
 EndDays = EndDays + 1
 End If
 DateCnt = DateAdd("d", 1, DateCnt)
 Loop
 Work_Days = WholeWeeks * 5 + EndDays
End Function
```

To call the function, you can pass either a valid string or an actual date value. The following are two ways to call this function from the Immediate window:

```
?Work_Days("01/01/93", "12/31/96") (returns 1043)
```

```
?Work_Days(#03/05/93#, #04/06/93#) (returns 23)
```

Additional reference words: 1.00 1.10 2.00 modules Day/Time

KBCategory: kbprg

KBSubcategory: PgmOthr

## INF: How to Determine the Current Screen Resolution (Size)

Article ID: Q113458

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a sample user-defined Access Basic function that you can use to determine the current screen resolution. This function is useful in determining if you are running in standard VGA mode (640 x 480) or Super VGA mode (800 x 600 or 1024 x 768).

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The example below demonstrates how to create and use the sample GetScreenResolution() function.

### Notes:

- You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.
- In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscores when re-creating this code in Access Basic.
- In Microsoft Access version 2.0, you can create code in a form module. Code in a form module is restricted to that form. The Type statement used in this example must be global. You can use this code in a form's module if the Type statement is in the Declarations section of a global Access Basic module. For more information about the Type statement, search for "type" then "Type Statement" using the Microsoft Access Help menu.

1. Create a new Access Basic module, and enter the following Declarations and function:

```
' *****
' DECLARATIONS SECTION

Option Explicit

Type RECT
```



```

 x1 As Integer
 y1 As Integer
 x2 As Integer
 y2 As Integer
End Type

Declare Function GetDesktopWindow Lib "User" () As Integer
Declare Function GetWindowRect Lib "User" _
 (ByVal hWnd As Integer, rectangle As RECT) As Integer

' *****
' FUNCTION: GetScreenResolution()
'
' PURPOSE:
' To determine the current screen size or resolution.
'
' RETURN:
' The current screen resolution. Typically one of the following:
' 640 x 480
' 800 x 600
' 1024 x 768
' *****
Function GetScreenResolution () as String
 Dim R As RECT
 Dim hWnd As Integer
 Dim RetVal As Integer
 hWnd = GetDesktopWindow()
 RetVal = GetWindowRect(hWnd, R)
 GetScreenResolution = (R.x2 - R.x1) & "x" & (R.y2 - R.y1)
End Function

```

2. From the View menu, choose Immediate Window.

3. Type the following in the Immediate window, and then press ENTER:

```
? GetScreenResolution()
```

The current screen resolution will be displayed in the Immediate window. If the current resolution is standard VGA, the following will be displayed in the Immediate window:

```
640x480
```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: How to Determine Version of INSTCAT.SQL File on Server**  
**Article ID: Q100467**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Using the SQL Server ODBC driver with Microsoft Access requires that you run the INSTCAT.SQL file on the ODBC installation disks. This article describes the SQL stored procedure that you can use to query for the version number of the INSTCAT.SQL file that was run against the SQL Server computer.

MORE INFORMATION

=====

The following syntax can be executed in a pass-through query in Microsoft Access version 2.0, as well as in the Query window of SQL Server Administration Facility (SAF) for MS-DOS, OS/2, Microsoft Windows or any other operating system in which you can execute a stored procedure:

```
sp_server_info 500
```

The attribute\_value field of the returned row contains the version number. If an empty recordset is returned, it is likely that the INSTCAT.SQL file was not run on the server. The result will resemble the following:

| attribute_id | attribute_name    | attribute_value |
|--------------|-------------------|-----------------|
| 500          | SYS_SPROC_VERSION | 01.00.2101      |

OS/2 is manufactured by International Business Machines Corporation, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr

## INF: How to Determine Whether a Menu Item Is Checked

Article ID: Q95935

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article discusses how you can make application programming interface (API) calls to determine whether or not a menu item is checked (selected). This method can be used, for example, when you want to determine whether a form is in Form view or Datasheet view. If a form is in Datasheet view, the Datasheet menu item on the View menu is checked. Thus, determining whether this menu item is checked allows you to determine whether the form is in Datasheet view.

The program code in this article comes from the ORDENTRY.MDB sample database and is presented in this article in a condensed version.

### MORE INFORMATION

=====

To determine whether a menu item is selected, you can use the IsMenuChecked() function listed in the code at the end of this article. IsMenuChecked() accepts two parameters indicating the position of the desired menu item; IsMenuChecked() returns true if the item is selected and false if it is not.

To find the position of the menu item you want to select, begin at 0 (zero) and count from the left-most menu item until you reach the item you want. This determines the first number. Then, from the selected menu, begin with zero and count down the submenu until you reach the submenu item you want. This determines the second number. (NOTE: separator bars are considered submenu items, and always return a True from the IsMenuChecked() function.)

For example, suppose you have the following menus:

```
File Edit View

Open Cut Form Design
Close Copy Form
Exit Paste Datasheet
```

To determine whether Datasheet is checked, you would call IsMenuChecked() with the values 2 and 2, because the View menu is in position 2 and the Datasheet item on the View submenu is in position 2. Note the same menu with numbered rows and columns to illustrate this further:

```
 0 1 2
File Edit View

```

```

0 Open Cut Form Design
1 Close Copy Form
2 Exit Paste Datasheet

```

When determining the position of a menu command, separator bars count as commands. For example, to get to the Import menu command on the File menu in the Database window, use the arguments 0 (for the File menu) and 7 (for the Import command). Positions 3 and 6 correspond to the separator bars on the File menu.

For an example of how to use this function, follow these steps in the sample database NWIND.MDB:

1. Copy the Access Basic code at the bottom of this article to a new module.
2. Open the Customers form in Design view.
3. Select the Customer ID text box.
4. Specify the following for the OnDblClick property in the property sheet:

```
OnDblClick: =MsgBox(Str(IsMenuChecked(2,2)))
```

5. Switch to Form view.

Note that Datasheet on the View menu is not checked because the form is not in Datasheet view.

6. Double-click the Customer ID text box.

A zero appears indicating that the Datasheet item on the View menu is not checked.

7. Switch to Datasheet view.

Notice that the Datasheet item on the View menu is now checked because the form is in Datasheet view.

8. Double-click the Customer ID text box.

This time, a -1 appears indicating that the Datasheet item on the View menu is checked.

The following is the code for the sample function IsMenuChecked():

```

Option Compare Database 'Use database order for string comparisons.
Option Explicit

Declare Function IsZoomed% Lib "User" (ByVal hWnd%)
Declare Function GetMenu% Lib "User" (ByVal hWnd%)
Declare Function GetSubMenu% Lib "User" (ByVal hMenu%, ByVal nPos%)
Declare Function GetMenuState% Lib "User" (ByVal hMenu%, ByVal idItem%,
 ByVal fuFlags%)
Declare Function GetActiveWindow% Lib "User" ()
Declare Function GetParent% Lib "User" (ByVal hwin%)

```

```
Declare Function GetClassName% Lib "User" (ByVal hwin%, ByVal stBuf$,
 ByVal cch%)
```

```
Const WU_MF_BYPOSITION = &H400
Const WU_MF_CHECKED = &H8
' OMain is the predefined classname of the
Global Const WU_WC_ACCESS = "OMain"
```

```
Function IsMenuChecked (iMenu%, iItem%) As Integer
 Dim hMainMenu%
 Dim hMenu%
 Dim Flags%

 '-----
 'If the window is maximized, there is an additional
 'Control menu added to the MDI window.
 '-----
 If (IsZoomed(Screen.ActiveForm.hWnd)) Then
 iMenu% = iMenu% + 1
 End If

 '-----
 ' Get the appropriate handle to the menu item.
 '-----
 hMainMenu% = GetMenu(GetAccessHwnd())
 hMenu% = GetSubMenu(hMainMenu%, iMenu%)
 Flags% = WU_MF_BYPOSITION Or WU_MF_CHECKED

 '-----
 ' Call the API that returns the state of the menu.
 '-----
 IsMenuChecked = (GetMenuState(hMenu%, iItem%, Flags%) <> 0)
End Function
```

```
Function StWindowClass (hWnd As Integer) As String
 Const cchMax = 255
 Dim cch%
 Dim stBuff As String * cchMax

 '-----
 ' Get the class name of the window relating to hWnd
 ' and return this as the function.
 '-----
 cch% = GetClassName(hWnd, stBuff, cchMax)
 If (hWnd% = 0) Then
 StWindowClass = ""
 Else
 StWindowClass = (Left$(stBuff, cch%))
 End If
End Function
```

```
Function GetAccessHwnd () As Integer
 Dim hWnd%

 '-----
 ' Keep getting the handle of the parent, until the
 ' ClassName = "OMain", which is the class name of
```

```
' the Microsoft Access window.
'-----
hWnd% = GetActiveWindow()
While ((StWindowClass(hWnd%) <> WU_WC_ACCESS) And (hWnd% <> 0))
 hWnd% = GetParent(hWnd%)
Wend
GetAccessHwnd = hWnd%
End Function
```

#### REFERENCES

=====

For more information, query on the following word here in the Microsoft Knowledge Base:

GetSubMenu

Additional reference words: 1.00 1.10 2.00 basic

KBCategory: kbprg

KBSubcategory: PgmApi

## **INF: How to Determine Which Snaking Column Is Current**

Article ID: Q123712

---

The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article describes how to determine which snaking column in a report is current. This determination is useful for:

- Setting columns so that each has an equal number of items.
- Drawing separation lines between the columns.

### MORE INFORMATION

=====

You can determine which column is current by checking the report's Left property. In a report with snaking columns, the Left property value is always greater in the second column than in the first column. For example, assume a report with the following dimensions:

Left margin: 1 inch  
Right margin: 1 inch  
Report width: 3.5 Inch  
Column spacing: 0  
Items across: 2

In the first column, the report's Left property is 1 inch. In the second column, the report's Left property is 4.5 inches (left margin + report width = second column Left property). Note that values in the Left property are measured in twips. 1 inch contains 1440 twips.

You can determine if the second column is current without knowing the report's dimensions by comparing the value of the Left property with the width of the report. If the value of the Left property is greater than the width of the report, the second column is current. Note that this method works only if you select the Same As Detail check box in the Print Setup dialog box. (To select this check box, choose Print Setup from the File menu, and then choose the More button.)

You can use this information in conjunction with the report's OnFormat property to perform a variety of tasks, such as ensuring that each column has an equal number of items and drawing lines between the columns.

### REFERENCES

=====

For more information about snaking columns, search for "snaking columns" then "Creating Snaking Columns in a Report" using the Microsoft Access Help menu.

Additional reference words: 2.00  
KBCategory: kbusage  
KBSubcategory: RptWizsn



## **INF: How to Dial a Phone Number in Microsoft Access**

**Article ID: Q93696**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a sample user-defined Access Basic function, `DialNumber()`, that you can use to dial a telephone number from Microsoft Access using your computer's modem. This method uses Microsoft Windows application programming interface (API) function calls.

Although this function will work with Microsoft Access version 2.0, version 2.0 has a new `AutoDialer` feature that can be used for dialing telephone numbers. For more information about the `AutoDialer` feature, search for "AutoDialer" using the Microsoft Access 2.0 Help menu.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

To dial a phone number in Microsoft Access, you can use Microsoft Windows API calls in the dynamic-link library (DLL) `USER.EXE`. There are 17 communication API calls available in `USER.EXE`, but only the following 3 are used to dial a phone number: `OpenComm()`, `CloseComm()`, and `WriteComm()`.

The following steps demonstrate how to create and use the `DialNumber()` function.

NOTE: In the following sample code, an underscore (`_`) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

1. Open the sample database `NWIND.MDB`.
2. Create a new Access Basic module with the following in the Declarations section:

```
Option Explicit
```

```
' The number of seconds to wait for the modem to dial before
' .. resetting the modem. If the phone hangs up prematurely
' .. try increasing this value by small increments.
Const WAITSECONDS = 4
```

```
Declare Function OpenComm Lib "User" (ByVal lpComName$, _
 ByVal wInQueue%, ByVal wOutQueue%) As Integer
```

```

Declare Function CloseComm Lib "User" (ByVal nCid%) As Integer
Declare Function WriteComm Lib "User" (ByVal nCid%, _
 ByVal lpBuf$, ByVal nSize%) As Integer

```

```

Const ID_CANCEL = 2
Const MB_OKCANCEL = 1
Const MB_ICONSTOP = 16, MB_ICONINFORMATION = 64

```

3. Add the following function to the module:

```

' *****
' FUNCTION: DialNumber()
'
' PURPOSE: To dial a telephone number using the computer's modem
'
' ARGUMENTS:
' PhoneNumber: The telephone number to dial
'
' CommPort: The communications port the modem is connected
' to. Typically, modems are found on COM2, however,
' they can be configured for any COM port.
'
' EXAMPLE:
' Type the following in the Immediate window using a modem
' connected to the COM2 port:
'
' ? DialNumber("555-1212", "COM2")
' *****
Function DialNumber (PhoneNumber, CommPort As String)
 Dim Msg As String, MsgBoxType As Integer, MsgBoxTitle As String
 Dim ModemCommand As String
 Dim OpenPort As Integer
 Dim RetVal As Integer
 Dim StartTime
 Dim CR As String: CR = Chr$(13)
 Dim LF As String: LF = Chr$(10)

 ' Ask the user to pick up the phone
 Msg = "Please pickup the phone and choose OK to dial " & _
 & PhoneNumber
 MsgBoxType = MB_ICONINFORMATION + MB_OKCANCEL
 MsgBoxTitle = "Dial Number"
 If MsgBox(Msg, MsgBoxType, MsgBoxTitle) = ID_CANCEL Then
 Exit Function
 End If

 ' Open the communications port
 OpenPort = OpenComm(CommPort, 1024, 128)
 If OpenPort < 0 Then
 Msg = "Unable to open communication port " & CommPort
 GoTo Err_DialNumber
 End If

 ' Send the telephone number to the modem
 ModemCommand = "ATDT" & PhoneNumber & CR & LF
 If WriteComm(OpenPort, ModemCommand, Len(ModemCommand)) < 0 Then

```

```

 Msg = "Unable to dial number " & PhoneNumber
 GoTo Err_DialNumber
 End If

 ' Wait WAITSECONDS seconds for the phone to dial
 StartTime = Timer
 While Timer < StartTime + WAITSECONDS
 DoEvents
 Wend

 ' Reset the modem and take it off line
 ModemCommand = "ATH0" & CR & LF
 RetVal = WriteComm(OpenPort, ModemCommand, Len(ModemCommand))

 ' Close the communications port
 RetVal = CloseComm(OpenPort)

 Exit Function

Err_DialNumber: 'This is not an On Error routine
 Msg = Msg & CR & CR & "Make sure no other devices are using _
 communication port " & CommPort
 MsgBoxType = MB_ICONSTOP
 MsgBoxTitle = "Dial Number Error"
 MsgBox Msg, MsgBoxType, MsgBoxTitle

End Function

```

4. Open the Employees form in Design view.
5. Place a command button with the following properties on the form, next to the Home Phone field.

Microsoft Access 1.x:

```

ControlName: btnDialPhone
Caption: Dial
OnPush: =DialNumber([Home Phone], "COM2")

```

Microsoft Access 2.0:

```

Name: btnDialPhone
Caption: Dial
OnClick: =DialNumber([Home Phone], "COM2")

```

6. View the form in Form view. To dial an employee's home phone number, choose the Dial button.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: How to Dim (Gray) Menu Items with Access Basic**  
**Article ID: Q88940**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Access Basic does not have a command that allows you to dim (make unavailable) a menu item. There are also no properties associated with a form that enable you to set this menu characteristic.

To change such characteristics of a menu item, you can use Windows application programming interface (API) functions.

NOTE: This will only work for custom menus created with macros. It will not work for the default Microsoft Access menus. This technique will also not work for custom menu items created with the DoMenuItem action. For example, if you add a command to the menu with the DoMenuItem action and specify the PrintPreview arguments for the action, Microsoft Access overrides any API function that you write to check or gray this menu command.

MORE INFORMATION

=====

The following Windows API function is used to dim a menu item:

EnableMenuItem% (hMenu%, wIDEnableItem%, wEnable%)

-----

This function enables, disables, or grays a menu item.

|                |                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| hMenu%         | Specifies the menu.                                                                                                                                               |
| wIDEnableItem% | Specifies the menu item to be checked. The wIDEnableItem% parameter can specify pop-up menu items as well as menu items.                                          |
| wEnable%       | Specifies the action to take. It can be a combination of MF_DISABLED, MF_ENABLED, and MF_GRAYED. These values can be combined by using the bitwise 'OR' operator. |
| Return Value   | The return value specifies the previous state of the menu item. The return value is -1 if the menu item does not exist.                                           |

Example

-----

The following example designs a menu that dims a menu item:

1. Create a new macro. After adding the following actions and their associated properties, save the macro and name it "Menu Manipulation Macro." To display the "Macro Name" column, choose the Macro Names command from the View menu.

| Macro Name | Action  | Function Name         |
|------------|---------|-----------------------|
| GrayItem   | RunCode | Gray_Menu_Item(0,0)   |
| UnGray     | RunCode | UnGray_Menu_Item(0,0) |

When determining the position of a menu command, separator bars count as commands. For example, to get to the Import menu command on the File menu in the Database window, use the arguments 0 (for the File menu) and 7 (for the Import command). Positions 3 and 6 correspond to the separator bars on the File menu.

2. Create a new macro. After adding the action below and its associated property, save the macro as "Custom Demo Menu."

| Macro Name                                | Action                  | Function Name |
|-------------------------------------------|-------------------------|---------------|
| Top Level Menu                            | AddMenu                 |               |
| [Top Level Menu].AddMenu Action Arguments |                         |               |
| Menu Name                                 | &Gray                   |               |
| Menu Macro Name                           | Menu Manipulation Macro |               |

3. Create a new blank form and display the Properties window of the form by choosing the Properties command from the View menu.
4. Set the OnMenu property to "Custom Demo Menu."
5. From the File menu, choose Close to close the form. Save the form as "Menu Manipulation Form."
6. Create a new module from the Database window. Within the new module, enter the Access Basic code listed further below. Save the module as "Menu Manipulation Code."
7. From the Database window, select the Menu Manipulation form and then choose the Open button to display the form in Form view. The normal Microsoft Access menu disappears and is replaced by the custom menu you created in the previous steps.
8. There are two options on the menu. Choose the GrayItem command and the command is dimmed (grayed). Choosing the UnGray command makes the GrayItem menu command available.

NOTE: In the following sample code, an underscore \_ is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```

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

Declare Function FindWindow% Lib "user" (ByVal lpClassName_
 As Any, ByVal lpCaption As Any)
Declare Function GetMenu% Lib "user" (ByVal hWnd%)

```

```

Declare Function GetSubMenu% Lib "user" (ByVal hSubMenu%, _
 ByVal nPos%)
Declare Function EnableMenuItem% Lib "user" (ByVal hMenu%, _
 ByVal wItem%, ByVal wEnable%)
Declare Function IsZoomed% Lib "User" (ByVal hWnd%)
Declare Sub DrawMenuBar Lib "User" (ByVal hWnd%)

Const MF_BYPOSITION = &H400
Const MF_GRAYED = &H1

Const MyNull = 0&
Const ClassName = "OMain"

Dim ChWnd% 'handle to the Microsoft Access window.
Dim hMenuTop% 'handle to the Microsoft Access menu.
Dim hSubMenu% 'handle to the pop-up menu
Dim ItemID% 'command ID associated with menu item.
Dim ReturnVal%

'=====
'This function initializes:
'
' - The window handles associated with the Microsoft Access form.
' - The handle to the menu of the specified window.
' - The menu handle of the specified pop-up menu of the window menu.
'
'The variables here are global to the database.
'=====
Sub Get_Menu_Handles (TopLevel%)

 ChWnd% = FindWindow(ClassName, MyNull)
 hMenuTop% = GetMenu(ChWnd%)
 hSubMenu% = GetSubMenu(hMenuTop%, TopLevel%)

End Sub

'=====
'This function dims a menu item. The text of a dimmed
'menu item appears in light gray text on the menu,
'but does not allow the user to select the item either by
'mouse or keypad. The macro action associated with the
'menu item does not execute when the user tries to select
'the menu item.
'=====
Function Gray_Menu_Item (TopLevel%, SubLevel%)

 'If the form is maximized, the system menu is added to the forms
 'menu bar, so increment the actual TopLevel%
 If (IsZoomed(Screen.ActiveForm.hWnd)) Then
 TopLevel% = TopLevel% + 1
 End If

 Call Get_Menu_Handles(TopLevel%)
 Gray_Menu_Item = EnableMenuItem(hSubMenu, SubLevel%, _
 MF_GRAYED Or MF_BYPOSITION)

 DrawMenuBar ChWnd%

```

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%)
```

```
 'If the form is maximized, the system menu is added to the forms
 'menu bar, so increment the actual TopLevel%
```

```
 If (IsZoomed(Screen.ActiveForm.hWnd)) Then
```

```
 TopLevel% = TopLevel% + 1
```

```
 End If
```

```
 Call Get_Menu_Handles(TopLevel%)
```

```
 UnGray_Menu_Item = EnableMenuItem(hSubMenu, SubLevel%, _
 Not MF_GRAYED And MF_BYPOSITION)
```

```
 DrawMenuBar ChWnd%
```

End Function

#### REFERENCES

```
=====
```

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

menu and system and findwindow and getmenu and getsubmenu

Additional reference words: 1.00 1.10 2.00 grey out

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: How to Direct DDE to a Specific Instance of a Server**  
**Article ID: Q105659**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

SUMMARY

=====

You can run multiple instances of the same application under Microsoft Windows. When you communicate with an application using DDE, it may be important to be able to identify which instance of an application should respond to your DDE commands.

You can initiate a DDE conversation with a specific instance of an application by appending the application's Task ID number to the application name argument in the DDE initiate command.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

Each instance of each application launched in Windows is assigned a unique number called a Task ID. This number can be used to distinguish among multiple instances of an application.

To initiate a DDE conversation using the System topic with an instance of Microsoft Excel that has a Task ID of 34567, you could use the following sample command:

```
Chan = DDEInitiate("excel34567", "System")
```

If you do not specify a Task ID number, you will initiate a conversation with the first instance of the application that Windows finds. Note that the order in which Windows finds instances of an application does not always correspond to the order in which the applications were started.

How to Get the Task ID of an Application

-----

You can get the Task ID of an application by doing one of the following:

- Use the Access Basic shell command to launch the application. The shell command will return the application's Task ID if the application is launched successfully.



-or-

- Use an Access Basic function to find the Task ID. The following sample function can be used to find an application's Task ID:

#### Notes

-----

- In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.
- You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.

#### Option Explicit

```
Declare Function GetActiveWindow Lib "User" () As Integer
Declare Function GetWindowWORD Lib "User" (ByVal hWnd As Integer, _
 ByVal nIndex As Integer) As Integer
Const GWW_HINSTANCE = (-6)

Function GetActiveTaskID ()
 GetActiveTaskID = GetWindowWORD(GetActiveWindow(), _
 GWW_HINSTANCE)
End Function
```

The following Access Basic code demonstrates the use of both of these techniques in getting the Task ID.

```
Function TestTaskID()
 Dim TaskID
 TaskID = Shell("excel", 1)
 MsgBox TaskID & " and " & GetActiveTaskID()
End Function
```

#### How to Use an Application's Task ID in a DDE Conversation

-----

The following example demonstrates how to use Task IDs to indicate which instance of an application should be used in a DDE conversation. This sample code will launch two instances of Microsoft Excel, start a DDE conversation with each instance, and poke data to Sheet1 of each instance:

```
Function DDEWithTwoInstances ()
 Dim TaskID1, TaskID2
 Dim Chan1, Chan2

 ' Start 2 instances of Excel and record Task ID numbers
 TaskID1 = Shell("excel.exe", 1)
 TaskID2 = Shell("excel.exe", 1)

 ' Initiate DDE conversations with both instances using
 ' the Task ID numbers. Trap an error that may happen
 ' if we try to initiate a conversation and the copy of
```

```
' Excel is still trying to start from the shell command.
```

```
On Error GoTo ExcelNotYetReady
Chan1 = DDEInitiate("excel" & TaskID1, "Sheet1")
On Error GoTo 0
```

```
On Error GoTo ExcelNotYetReady
Chan2 = DDEInitiate("excel" & TaskID2, "Sheet1")
On Error GoTo 0
```

```
' Poke information into each instance of Excel
```

```
DDEPoke Chan1, "R1C1", "This is Task1!"
DDEPoke Chan2, "R1C1", "This is Task2!"
```

```
DDETerminateAll
```

```
Exit Function
```

```
ExcelNotYetReady:
DoEvents
Resume
```

```
End Function
```

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.1, pages 139-147

Additional reference words: 1.00 1.10 2.00 Dynamic Data Exchange ADK

KBCategory: kbinterop

KBSubcategory: IntpDde

**INF: How to Disable PAGE UP and PAGE DOWN Keys in a Form (2.0)**  
**Article ID: Q114506**

---

The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes the following two methods that you can use to disable the PAGE UP and PAGE DOWN keys in a form:

- Open the form as a Dialog form using the OpenForm macro action.
- or-
- Trap the PAGE UP and PAGE DOWN keys in the KeyDown event.

MORE INFORMATION

=====

Method 1: Opening the Form as Dialog Using the OpenForm Macro Action

---

The PAGE UP and PAGE DOWN keys will be inoperative in a form if the form is opened using an OpenForm macro action with the Window Mode argument set to Dialog. To demonstrate this technique, create and run a new macro with the following actions in the sample database NWIND.MDB:

```
OpenForm
 Form Name: Customers
 Filter Name: <leave empty>
 Where Condition: <leave empty>
 View: Form
 Data Mode: Edit
 Window Mode: Dialog
```

The drawback to this technique is that a Dialog form cannot use a custom menu, nor will you be able to switch to another form while the Dialog form is open.

NOTE: The Microsoft Access Wizards use this technique to limit navigation in multiple-page forms.

Method 2: Trapping PAGE UP and PAGE DOWN in the KeyDown Event

---

The PAGE UP and PAGE DOWN keys can be detected and trapped in a control by adding the following lines of Access Basic code to the KeyDown event for all the controls on a form where the PAGE UP and PAGE DOWN keys should be disabled:

```
If KeyCode = KEY_PRIOR Or KeyCode = KEY_NEXT Then
 KeyCode = 0
```

End If

NOTE: The constants for KeyCode are defined in the CONSTANT.TXT file, which is included with Microsoft Access. To use these constants in your code, you must either declare them yourself or copy them from the CONSTANT.TXT file and paste them into the Declarations section of your module.

The following example demonstrates how to detect and trap the PAGE UP and PAGE DOWN keys in a control's KeyDown event.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the Customers form in Design view.
2. Create an event procedure with the following code for the KeyDown event for each control on the form:

```
If KeyCode = KEY_PRIOR Or KeyCode = KEY_NEXT Then
 KeyCode = 0
End If
```

NOTE: To create each event procedure, select a control, click the Build button to the right of the OnKeyDown property, and then select Code Builder.

For example, the event procedure code for the Customer ID field will look like the following:

```
Sub Customer_ID_KeyDown (KeyCode As Integer, Shift As Integer)
 If KeyCode = KEY_PRIOR Or KeyCode = KEY_NEXT Then
 KeyCode = 0
 End If
End Sub
```

3. Close the module and then switch the form to Form view.

#### REFERENCES

=====

For more information about Dialog forms, search for "OpenForm" then "OpenForm Action" using the Microsoft Access Help menu.

For more information about the KeyDown event, search for "KeyDown" then "KeyDown, KeyUp Events" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 pgup pgdn

KBCategory: kbusage

KBSubcategory: FmsOthr

## **INF: How to Display Intervals Greater Than 24 Hours**

**Article ID: Q109320**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

When you are displaying an interval of greater than 24 hours in a calculated control, the interval will not be displayed correctly. Depending on the format of the control, you will see just the hours and minutes, or you will see an incorrect date, such as 12/31/1889.

This article demonstrates three sample functions you can add to a calculated control to correctly display intervals of greater than 24 hours.

### MORE INFORMATION

=====

NOTE: This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access.

To find the number of hours elapsed between two dates, you can subtract the values in Access Basic. For example, the sample code

```
StartDate=#6/1/93 8:00AM#
EndDate=#6/1/93 1:00PM#
?Format(EndDate-StartDate,"hh:mm")
```

will return a value of 05:00, which is the correct result.

However, the sample code

```
StartDate=#6/1/93 8:00AM#
EndDate=#6/3/93 3:00PM#
? Format(EndDate-StartDate,"hh:mm")
```

will return a value of 07:00, which is incorrect. The correct result is 55:00.

The following three sample functions will display the elapsed time correctly:

NOTE: In the following sample code, an underscore ( \_ ) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

1. Open a module and enter the following in the Declarations section:

```
Option Explicit
Global StartDate
```

Global EndDate

2. Enter the following functions:

```
'-----
' This function accepts a time interval and returns
' a value, such as "55:23" for 55 hours and 23 minutes.
'-----
Function HoursMinutes (Interval)
 Dim Hours As Long, Minutes As Integer
 If IsNull(Interval) Then
 HoursMinutes = Null
 Else
 Hours = Int(Interval * 24)
 Minutes = CInt((Interval * 24 - Hours) * 60)
 If Minutes = 60 Then
 Minutes = 0
 Hours = hours + 1
 End If
 HoursMinutes = Hours & ":" & Format(Minutes, "00")
 End If
End Function

'-----
' This function accepts a time interval and returns
' a value, such as "1 Day 23 Hours."
'-----
Function DaysHours (Interval)
 Dim Days As Long, Hours As Long
 If IsNull(Interval) Then
 DaysHours = Null
 Else
 Days = Int(Interval)
 Hours = CInt((Interval - Days) * 24)
 If Hours = 24 Then
 Hours = 0
 Days = Days + 1
 End If
 DaysHours = Days & IIf(Days = 1, " Day ", " Days ") & _
 Hours & IIf(Hours = 1, " Hour", " Hours")
 End If
End Function

'-----
' This function accepts a time interval and returns
' a value, such as "3 Days 17:43".
'-----
Function DaysHoursMinutes (Interval)
 Dim Days As Long, Remainder As Double
 If IsNull(Interval) Then
 DaysHoursMinutes = Null
 Else
 Days = Int(Interval)
 Remainder = Interval - Days
 If Remainder > #23:59:59# Then
 Remainder = 0
 Days = Days + 1
```

```
End If
DaysHoursMinutes = Days & IIf(Days = 1, " Day ", " Days ") & _
Format(Remainder, "hh:mm") & " Hours"
End If
End Function
```

3. Open a new module, and type the following in the module's Immediate window:

```
StartDate=#6/1/93 8:00AM#
EndDate=#6/3/93 3:05PM#
?HoursMinutes(EndDate-StartDate)
```

The HoursMinutes() function will return a value of 55:05.

4. Type:

```
?DaysHours(EndDate-StartDate)
```

The DaysHours() function will return the value "2 Days 7 Hours."

5. Type:

```
?DaysHoursMinutes(EndDate-StartDate)
```

The DaysHoursMinutes() function will return the value "2 Days 07:05."

#### REFERENCES

=====

For more information on calculating time differences, query on the following words here in the Microsoft Knowledge Base:

calculate and timevalue

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

## INF: How to Display Line Numbers on Subform Records

Article ID: Q120913

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article describes how to create and use a sample user-defined Access Basic function to display the current line or row number in a subform.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

The following examples demonstrate how to create and use the sample function GetLineNumber():

#### How to Create the GetLineNumber() Function

-----

1. Open the sample database NWIND.MDB.
2. Create a new module and enter the following line in the Declarations section:

Option Explicit

3. Enter the following code in the module.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Function GetLineNumber (F As Form, KeyName As String, KeyValue)
 Dim RS As Recordset
 Dim CountLines

 On Error GoTo Err_GetLineNumber

 Set RS = F.RecordsetClone

 ' Find the current record.
 Select Case RS.Fields(KeyName).Type
 ' Find using numeric data type key value?
 Case DB_INTEGER, DB_LONG, DB_CURRENCY, DB_SINGLE, _
 DB_DOUBLE, DB_BYTE
 RS.FindFirst "[" & KeyName & "]" = " & KeyValue
 ' Find using date data type key value?
```



```

 Case DB_DATE
 RS.FindFirst "[" & KeyName & "]" = "#" & KeyValue & "#"
 ' Find using text data type key value?
 Case DB_TEXT
 RS.FindFirst "[" & KeyName & "]" = "'" & KeyValue & "'"
 Case Else
 MsgBox "ERROR: Invalid key field data type!"
 Exit Function
 End Select

 ' Loop backward, counting the lines.
 Do Until RS.EOF
 CountLines = CountLines + 1
 RS.MovePrevious
 Loop

 Bye_GetLineNumber:
 ' Return the result.
 GetLineNumber = CountLines

 Exit Function

 Err_GetLineNumber:
 CountLines = 0
 Resume Bye_GetLineNumber

End Function

```

The GetLineNumber() function requires the following three arguments:

- The form object on which to place line numbers.
- The name of the unique key field in the subform's underlying table. If the record source does not have a single unique key field, add a field with a Counter data type to the underlying table for this purpose.
- The current key field value.

You could use the following sample expression as the ControlSource property setting of a text box on a subform, given that the subform's underlying table has a field called ID as its unique key field:

```
=GetLineNumber(Form,"ID",[ID])
```

How to Use the GetLineNumber() Function

---

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the Order Details table in Design view, add the following field to the table, and then save the table:

```

Field Name: ID
Data Type: Counter

```

This field will serve as the required single unique key field for the table.

2. Open the Order Details Extended query in Design view, add the ID field from the Order Details table to the query grid, and then save the query.
3. Open the Orders Subform form in Design view and add the following text box to the form:

Name: #

ControlSource: =GetLineNumber(Form, "ID", [ID])

4. From the Edit menu, choose Tab Order. Drag the "#" field from the bottom of the Custom Order list to the top, and then choose OK.
5. Save and then close the form.
6. Open the Orders form in Form view and move among the orders to find orders with multiple line items.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsSubf

## INF: How to Dynamically Create a Filter

Article ID: Q109334

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a method of dynamically creating a filter by applying expressions on a form using a macro. The same technique can also be adapted to a Query by Form (QBF) method.

The article assumes that you are familiar with form and macro design.

### MORE INFORMATION

=====

The "+" and "&" operators are used to concatenate (join) two string values. However, each operator behaves differently when used on null-valued strings. For example, if you use a null variable "Name" in the expression "Dear "+[Name], then the expression evaluates to null, whereas if you have the expression "Dear "&[Name], then the expression evaluates to "Dear ".

You can use these different behaviors to build a string for use in the Where condition of an ApplyFilter or OpenForm macro action. You combine the strings using the + operator to concatenate values to dynamically build each separate part of the filter (which becomes null if no value is entered), then use the & operator to concatenate the separate strings.

The example below, using the sample database NWIND.MDB, uses a form and a macro to create a 3-field filter:

1. Open the sample database NWIND.MDB.
2. Open the Employees form in Design view.
3. Choose Save As from the File menu and save the form as Employees 2.
4. Widen the Employees 2 form and add the following controls to the new blank area in the detail section to the right of the photo:

NOTE: The labels should be to the left of their corresponding text boxes.

Label

-----  
Caption: Search ID

Text Box

-----  
Name: SID  
ControlSource: <leave blank>

AfterUpdate: Filter.Make null 2

NOTE: In Microsoft Access version 1.x, The Name property is called ControlName.

Label

-----  
Caption: Search First Name

Text Box

-----  
Name: SName  
ControlSource: <leave blank>  
Validation Rule: Not Like "\*" Or Is Null  
Validation Text: Can't use apostrophe in filter text.  
AfterUpdate: Filter.Make null

Label

-----  
Caption: Search Hire Date

Text Box

-----  
Name: SDate  
ControlSource: <leave blank>  
Validation Rule: Not Like "\*" Or Is Null  
Validation Text: Can't use apostrophe in filter text.  
AfterUpdate: Filter.Make null

NOTE: In the following example, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this line.

Text Box

-----  
Name: Where Text  
ControlSource: =Mid((" And [Employee ID]="+[SID]) & \_  
(" And [First Name] Like '"+[SName]+'') & \_  
(" And [Hire Date] Like '"+[SDate]+''),6) \_

Command Button

-----  
Caption: Apply Filter  
OnClick: Filter.Apply Filter

NOTE: In Microsoft Access version 1.x, the OnClick property is named OnPush.

## 5. Create the following macro group:

Macro Group: Filter

-----  
Macro Name: Make null  
Condition: IsNull(Screen.ActiveControl)  
Action: StopMacro  
Description: This macro is for text fields.

Condition: Len(Screen.ActiveControl)=0  
Action: SetValue  
Item: Screen.ActiveControl  
Expression: Null  
Description: Make an empty text field null.

Macro Name: Make null 2  
Condition: IsNull(Screen.ActiveControl)  
Action: StopMacro  
Description: This macro is for numeric fields.  
  
Condition: Not IsNumeric(Screen.ActiveControl)  
Action: SetValue  
Item: Screen.ActiveControl  
Expression: Null  
Description: Turn non-numeric values into nulls.

Macro Name: Apply Filter  
Condition: IsNull([Where Text])  
Action: ApplyFilter  
Where Condition: True  
Description: No Where Text, so select all records and stop.  
  
Condition: ...  
Action: StopMacro  
  
Condition: Len([Where Text]) > 255  
Action: MsgBox  
Message: You have selected too many criteria in the filter.  
Title: Apply Filter  
Description: If filter too long, display warning and stop.  
  
Condition: ...  
Action: StopMacro  
  
Action: ApplyFilter  
Where Condition: =[Where Text]  
Description: Apply the filter.

6. Save the macro and the form.

7. View the form in Form view and enter values in the search fields.

Note that the data for the Apply Filter's Where condition is constructed in the Where text control. If you do not want users to see this, set the Visible property of the Where text field to False.

#### Limitations

-----

- The length of the ControlSource property for the Where text box cannot exceed 255 characters. Microsoft Access will not let you type an expression longer than this. One workaround to this limitation is to use several text boxes to build each part of the criteria, then use a final text box to combine the various parts into a single Where clause.

- The string that the expression produces cannot exceed 255 characters. The second test in the Apply Filter macro above tests for this condition. If the filter string is longer than 255 characters, you will receive a warning that the filter is too long and the macro will halt.

To modify the above example for a QBF method, add the controls to a blank unbound form and change the ApplyFilter actions to OpenForm actions. To select all the records using OpenForm actions, leave the Where condition blank instead of using True.

#### REFERENCES

=====

For more information on the ApplyFilter action, search for "ApplyFilter" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 macros forms

KBCategory: kbusage

KBSubcategory: McrActn

## INF: How to Edit Library Databases

Article ID: Q114682

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes two methods for editing Microsoft Access library databases.

A library database is a database that contains a collection of database objects that can be loaded automatically when you start Microsoft Access. You specify that a database is to be loaded by adding its name to the [Libraries] section of the MSACCESS.INI (version 1.x) or MSACC20.INI (version 2.0) file, or by installing it using the Add-in Manager (version 2.0). Library database files usually have .MDA extensions.

Since library databases are loaded when Microsoft Access starts, you cannot open them directly for editing. If you try to do so, you will receive an error message similar to:

The database 'c:\access\filename.mda' is already open as a library database

### MORE INFORMATION

=====

NOTE: You should back up library database files before you modify them.

#### Method 1

-----

NOTE: This method applies only to Microsoft Access version 2.0.

1. While Microsoft Access is not running, open the MSACC20.INI file located in your Windows directory with any text editor (such as Windows Notepad).
2. Add the line "DebugLibraries=True" (without quotation marks) to the [Options] section of the file. The section will look like:

```
[Options]
Systemdb=c:\Access\SYSTEM.MDA
UtilityDB= c:\Access\Utility.MDA
DebugLibraries=True
```

3. Save and then close the file.
4. Start Microsoft Access.
5. Open any database.

6. Open any module.
7. From the View menu, choose Procedures.
8. In the Databases box, select the library that you want to edit.

#### Method 2

- 
1. While Microsoft Access is not running, open the MSACCESS.INI (version 1.x) or MSACC20.INI (version 2.0) file in your Windows directory in any text editor (such as Windows Notepad).
  2. Locate the entry for the library file that you want to edit. The entry will be similar to:

```
Wzlib.mda=rw
```

3. Comment the line out by adding a semicolon (;) at the beginning of the line. The entry should be similar to:

```
;Wzlib.mda=rw
```

4. Save and then close the .INI file.
5. Start Microsoft Access.
6. From the File menu, choose Open Database.
7. Select the library you want to open. Note that by default, the Open Database dialog box only displays files with .MDB extensions. Either type the name of the library you want to open, or type "\*.MDA" (without quotation marks) in the File Name box to display files with .MDA extensions.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 15, "Using Library Databases and Dynamic-Link Libraries"

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: MdlLib



**INF: How to Fill a List Box with Database Object Names**  
**Article ID: Q124344**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article demonstrates how to use data access objects (DAO) to fill a list box or combo box with the names of database objects. The following two sample user-defined functions are used to fill the list box or combo box:

- The GetNames() function. This function fills an array with object names from the active database.
- The FillNameList() function. This function calls the GetNames() function and fills the list box or combo box.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The following steps demonstrate how to fill a list box or combo box with database object names:

1. Open the sample database NWIND.MDB.
2. Create a blank new form called Form1.
3. Place the following two list box controls on the form:

Name: ListBox1

-----  
RowSourceType: Value List  
RowSource: 1;Table;2;Query;3;Form;4;Reports;5;Macros;6;Modules  
ColumnCount: 2  
ColumnWidths: 0.25 in;0.75 in  
BoundColumn: 2

Name: ListBox2

-----  
RowSourceType: FillNameList

4. Create a macro called Macro1 with the following action:

Macro1 Action

-----

Requery

## Macrol Arguments

-----  
Control Name: ListBox2

5. Set the ListBox1 list box's AfterUpdate property to Macrol.
6. Create a new module called Module1 and enter the following sample code in the module. Note that the FillNameList() function requires that two variables be declared in the module's Declarations section so that data can be shared with the GetNames() function. The GetNames() function requires an object type and an array name argument. The function fills the array with the database object names and returns the number of rows in the array. Allowable object names are: Tables, Query, Forms, Reports, Macros, and Modules.

```
'*****
' Declarations section of the module
'*****

Option Compare Database ' Use database order for string
 ' comparisons.

Option Explicit

Dim list() As String
Dim entries

'*****
' Function FillNameList()
'*****

Function FillNameList (fld As Control, id, row, col, code)
' Accepts a control, an identifier, a row, a column, and a code.

On Error GoTo ErrorHandler

Dim ReturnVal
Dim x

If IsNull(Forms![Form1]![ListBox1]) Then
 x = "Tables"
Else
 x = Forms![Form1]![ListBox1]
End If

ReturnVal = Null
Select Case code
 Case 0 ' Initialize.
 entries = 0
 entries = GetNames(x, list())
 ReturnVal = entries
 Case 1 ' Open.
 ReturnVal = Timer ' Unique ID number for the control.
 Case 3 ' Get the number of rows.
 ReturnVal = entries
 Case 4 ' Get the number of columns.
 ReturnVal = 1
 Case 5 ' Get the column width.
```

```

 ReturnVal = -1 ' Use the default width.
 Case 6 ' Get the data.
 ReturnVal = list(row)
 Case 9 ' End.
 ReDim list(0)
 entries = 0
End Select
FillNameList = ReturnVal

ErrorHandler:
Resume Next

End Function

'*****
' Function GetNames()
'*****

Function GetNames (objtype As String, names() As String)
Dim Conta As Container, Db As Database, I, Arlen

Set Db = DbEngine.Workspaces(0).Databases(0)
Arlen = 0

If objtype = "Macros" Then
 objtype = "Scripts" ' Macros are called scripts internally.
End If

Select Case objtype
 Case "Tables"
 If Db.Tabledefs.Count <> 0 Then
 Arlen = Db.Tabledefs.Count
 ReDim list(0 To Arlen - 1)
 I = 0
 For I = 0 To (Arlen) - 1
 names(i) = Db.Tabledefs(i).Name
 Next I
 End If
 Case "Query"
 If Db.Querydefs.Count <> 0 Then
 Arlen = Db.Querydefs.Count
 ReDim list(0 To Arlen - 1)
 I = 0
 For I = 0 To (Arlen) - 1
 names(i) = Db.Querydefs(i).Name
 Next i
 End If
 Case Else
 Set Conta = Db.Containers(objtype)
 If Conta.Documents.Count <> 0 Then
 Arlen = Conta.Documents.Count
 ReDim list(0 To Conta.Documents.Count - 1)
 I = 0
 For I = 0 To (Arlen) - 1 ' Fill the Names array
 ' with object names.
 names(i) = Conta.Documents(i).Name
 Next I
 End If
 End Select
End Function

```

```
 End If
 End Select
 Getnames = Arlen ' Return the length of the array to the
 ' FillNameList() function.

 End Function
```

7. View Form1 in Form view and select an object type in ListBox1. ListBox2 displays the names of objects of the type selected in ListBox1.

#### REFERENCES

=====

For more information about Container objects, search for "Container" then "Container Object, Containers Collection" using the Microsoft Access Help menu.

For more information about filling list boxes, search for "filling list boxes/combo boxes" then "Filling a List Box or Combo Box Using an Access Basic Function" using the Microsoft Access Help menu.

Additional reference words: 2.00 ab listfunction

KBCategory: kbusage

KBSubcategory: FmsCmbo

**INF: How to Fill an Array Using Access Basic**  
**Article ID: Q109727**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes two methods of filling an array using Access Basic. The first method fills an array when the number of elements in the array is known. The second method fills an array when the number of elements in the array is not known.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

This example demonstrates how to fill a one-dimensional array when the number of elements is known:

1. Open the sample database NWIND.MDB.
2. Create a new module with the following sample code:

Option Explicit

Function FillOneDimArray ()

Dim i As Long

Dim DB As Database, SS As Snapshot

Dim RecordCount As Long

On Error GoTo ErrorHandler

Set DB = CurrentDB()

Set SS = DB.CreateSnapshot("Employees")

' Get number of records.

SS.MoveLast

RecordCount = SS.RecordCount

' Create the (zero-based) array.

' Address elements starting from row 0 rather than 1.

ReDim AnArray(RecordCount - 1)

' Fill the array.

SS.MoveFirst

For i = 0 To RecordCount - 1

```

 AnArray(i) = SS.[Last Name]
 SS.MoveNext
 Next i

 ' Print the array contents to the Immediate window.
 For i = 0 To RecordCount - 1
 Debug.Print AnArray(i)
 Next i

 SS.Close
 DB.Close

 Exit Function

ErrorHandler:
 MsgBox Error
 Exit Function

End Function

```

3. Choose Immediate Window from the View menu. Type the following in the Immediate window and press ENTER:

```
? FillOneDimArray()
```

The following will be printed to the Immediate window:

```

Davolio
Fuller
Leverling
Peacock
Buchanan
Suyama
King
Callahan
Dodsworth

```

The next example demonstrates how to fill an array when the number of elements is not known. A Dynaset will be used to fill the array. Records can be added to or removed from the Dynaset while the array is being filled:

1. Open the sample database NWIND.MDB.
2. Create a new module with the following sample code:

```

Function FillIndefArray ()
 Dim DB As Database, DS As Dynaset, Count As Integer
 Dim AnArray()
 Dim i As Long

 Set DB = CurrentDB()
 Set DS = DB.CreateDynaset("Employees")

 ' Create the (zero-based) array with only one row.
 ' NOTE: The Preserve keyword allows the size of the array
 ' to be adjusted without the contents being destroyed.

```

```

' Adjusting the size of an array using Preserve can
' cause your code to go significantly slower. It is better
' to create an array with a known number of elements prior
' to filling it.
Count = 0
ReDim Preserve AnArray(0)

' Fill the array.
DS.MoveFirst
Do Until DS.eof
 ' Fill the array row with the last name.
 AnArray(Count) = DS![Last Name]

 ' Increase the number of elements in the array
 ' by one to accommodate the next record.
 ReDim Preserve AnArray(UBound(AnArray) + 1)
 Count = Count + 1

 DS.MoveNext
Loop

' Remove the remaining empty array row.
ReDim Preserve AnArray(UBound(AnArray) - 1)

DS.Close

' Print the array contents to the Immediate window.
For i = 0 To Count - 1
 Debug.Print AnArray(i)
Next i

End Function

```

3. From the View menu, choose Immediate Window. Type the following in the Immediate window, and then press ENTER:

```
? FillIndefArray()
```

The following will be printed to the Immediate window:

```

Davolio
Fuller
Leverling
Peacock
Buchanan
Suyama
King
Callahan
Dodsworth

```

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.0, Chapter 3, "Access Basic Essentials," and Chapter 4, "Debugging Your Access Basic Code"

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbprg  
KBSubcategory: PgmOthr



**INF: How to Filter a Subform by Changing the Record Source**  
**Article ID: Q112796**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

You cannot use the ApplyFilter action to apply a filter to a subform. If you try to do so, the filter will be applied to the main form instead. In Microsoft Access version 1.x, you can achieve some of this functionality by basing the subform on a parameter query. In Microsoft Access version 2.0, however, you can get the same results by changing the subform's RecordSource property.

MORE INFORMATION

=====

The ApplyFilter action dynamically restricts or sorts the records in a table, or the records from a form or report's underlying table or query. Although you cannot use this action with a subform, you can change the subform's RecordSource property to get the same results.

The following example demonstrates how to change the sort order of the Categories Subform subform in the sample database NWIND.MDB by changing the subform's RecordSource property.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the NWIND database and create a new query based on the Products table. Include all the fields from the table in the query.
2. Set the Unit Price column's Sort row to Ascending.
3. Save the query as Unit Price Sort.
4. Open the Categories form in Design view and then add a command button to the form.
5. Set the command button's OnClick property to the following event procedure:

```
If Me![Categories Subform].Form.RecordSource = "Products" Then
 Me![Categories Subform].Form.RecordSource = "Unit Price Sort"
Else
 Me![Categories Subform].Form.RecordSource = "Products"
End If
```

6. View the Categories form in Form view.
7. Choose the command button you added in step 4.

As you continue to choose the command button, the form's sort order will toggle between the Unit Price and the Product ID fields.

#### REFERENCES

=====

For more information about the RecordSource property, search for "RecordSource" then "Changing a Form or Report's Underlying Table or Query" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsProp

**INF: How to Find a Record in a Form Using Access Basic**  
**Article ID: Q114556**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes a method you can use to find a record in a form using the form's recordset and a bookmark.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

The Microsoft Access "User's Guide" demonstrates a method using the FindRecord macro action that can be used to find records in a form by selecting a key value from a list. (See the Microsoft Access "User's Guide," version 1.0, pages 547-548, or version 1.1, pages 551-552, or version 2.0, pages 651-652.) The method described in the User's Guide is limited in the following ways:

- It requires that the key search field be located on the form.
- You are limited to searching for a single key value.

The example below demonstrates a sample user-defined Access Basic function called FindRecord\_RS() that uses the FindFirst recordset method to find a record in a form:

NOTE: In the following example, an underscore ( \_ ) at the end of a line is used as a line-continuation character. Remove the underscores from the ends of the lines when re-creating this example.

1. Create a new module with the following in the Declarations section:

```
Option Explicit
```

2. Add the following function to the module:

```
Function FindRecord_RS (SQLWhere)
 Dim DS As Dynaset
 Set DS = Screen.ActiveForm.Dynaset
 DS.FindFirst SQLWhere
 If DS.NoMatch Then
 MsgBox "No record found!"
 Else
 Screen.ActiveForm.Bookmark = DS.Bookmark
 End If
End Function
```

```
End If
End Function
```

The FindRecord\_RS() function's SQLWhere argument must be a string expression that is any valid SQL WHERE clause (without the word WHERE).

The following are examples of valid SQL WHERE clauses:

```
"[Customer ID] = 3"
```

```
"[Customer ID] = " & [Find Customer]
```

If the field is a text field, the WHERE clause may look like the following:

```
"[Customer ID] = '3'"
```

```
"[Customer ID] = '" & [Find Customer] & "'"
```

NOTE: You can use single or double quotation marks to delimit text values. The following examples are functionally identical to the two examples above:

```
"[Customer ID] = ""3"""
```

```
"[Customer ID] = """ & [Find Customer] & """""
```

The SQL WHERE clause to find a record for a particular customer on a given date might look like the following:

```
"[Customer ID] = 3 AND [Order Date] = #1/1/94#"
```

```
"[Customer ID] = " & [Find Customer] & " AND [Order Date] = #" & _
[Find Order Date] & "#"
```

Using the FindRecord\_RS Function

---

Microsoft Access Version 1.x:

The Suppliers form in the sample database NWIND.MDB has a combo box called Select Company To Find that lists available suppliers. When you select a supplier, the Find Company macro is run to find the supplier matching the selection using the FindRecord action.

The following steps demonstrate how to use the FindRecord\_RS() method instead of the Find Company macro:

1. Open the Suppliers form in Design view.
2. Select the Select Company To Find combo box in the form footer, and alter the AfterUpdate property from

```
Find Company
```

```
to:
```

```
=FindRecord_RS("[Company Name] = "" & [Company Pick List] & """)
```

When you make a selection in the combo box, the FindRecord\_RS() function will find a record in the Suppliers form whose Company Name field matches your selection in the combo box.

Microsoft Access Version 2.0:

The Products And Suppliers form in the sample database NWIND.MDB has a combo box called Select Product To Find that lists the available products. When you select a product, the Product Pick List.Find Product macro is run to find the product record matching the selection using the FindRecord macro action.

The following example demonstrates how to use the FindRecord\_RS() method instead of the Product Pick List.Find Product macro:

1. Open the Products And Suppliers form in Design view.
2. Select the Select Product To Find combo box in the form header, and alter the AfterUpdate property from

```
Product Pick List.Find Product
```

to:

```
=FindRecord_RS("[Product Name] = "" & [Product Pick List] & """)
```

When you select an item in the combo box, the FindRecord\_RS() function will find a record in the Products And Suppliers form whose Product Name field matches your selection in the Select Product To Find combo box.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 22, "Using Macros with Forms," pages 547-548

Microsoft Access "User's Guide," version 1.1, Chapter 22, "Using Macros with Forms," pages 551-552

Microsoft Access "User's Guide," version 2.0, Chapter 25, "Using Macros with Forms," pages 651-652

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsEvt

## INF: How to Find a Record Using a Bound Control (2.0)

Article ID: Q115189

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article demonstrates how to use a single bound control both for finding records and entering data in a form.

You can use this technique to find whether the value entered in a field already exists in another record. If the value does exist in another record, the record containing that value will be displayed. If the value does not exist in another record, you can continue entering data for the current record.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

The Microsoft Access "User's Guide" demonstrates a method that you can use to find records in your database by selecting a key value from a list. (See "Finding a Record by Selecting a Value from a List.") This method is limited in that it requires two controls. One control is an unbound control that is used to select the value to look for, and the other control is a bound control in which data entry for that value occurs.

The following example demonstrates how to use a single, bound control to achieve the same functionality.

NOTE: The field to which the control is bound cannot be a required field and it cannot have a validation rule that allows null values. Make sure that the field's Required property is set to No and that its ValidationRule property does not allow null values.

1. Open the sample database NWIND.MDB. Create a new module with the following declarations and functions:

```
' *****
' DECLARATIONS SECTION
' *****
Option Explicit
Dim Found

Function Find_BeforeUpdate ()
 Dim RS As Recordset, C As Control
 Set C = Screen.ActiveControl
 Set RS = Screen.ActiveForm.RecordsetClone
```

```

On Error Resume Next

' Try to find a record with a matching entry in
' the form's dynaset as a numeric field.
RS.FindFirst "[" & C.ControlName & "]" = " & C

' If it was not a numeric field (error 3070),
' try finding the record as a text field.
If Err = 3070 Then
 Err = 0
 RS.FindFirst "[" & C.ControlName & "]" = " & C & ""
End If

On Error GoTo 0

' If any unexpected error happened, present to user and exit.
If Err <> 0 Then
 MsgBox "Err " & Err & ": " & Error$, 48
 DoCmd CancelEvent
 Exit Function
End If

' If a record was found, save the found record's bookmark.
If RS.NoMatch Then
 Found = Null
Else
 Found = RS.Bookmark
End If

' If the record was found...
' ...cancel the BeforeUpdate event
' ...undo changes made to the current record
' ...and TAB to the next control to trigger the OnExit routine.
If Not IsNull(Found) Then
 DoCmd CancelEvent
 SendKeys "{ESC 2}{TAB}", False
End If
End Function

Function Find_OnExit ()
' If the record was found, cancel the OnExit routine to stay
' in the control and go find the record.
If Not IsNull(Found) And Found <> "" Then
 DoCmd CancelEvent

 ' Synchronize the form record with the found record.
 Screen.ActiveForm.Bookmark = Found

 Found = Null
End If
End Function

```

2. Open the Customers form in Design view. Change the following properties for the Customer ID field:

```
BeforeUpdate: =Find_BeforeUpdate()
```

OnExit: =Find\_OnExit()

3. View the form in Form view. In the Customer ID field, type "AROUT" (without quotation marks) and then press ENTER. Microsoft Access will find and display the Around The Horn customer.
4. In the Customer ID field, type "POPSI" (without quotation marks) and then press ENTER. Since this key value does not exist, you can continue entering data for the record.

#### How the Sample Functions Work

-----

The Find\_BeforeUpdate() function uses the FindFirst method to search the dynaset the form is based on to see if the value entered in the control exists in the table.

If the value does not exist, the global variable Found is set to NULL and the function exits. If the value does exist, the global variable Found is set to the bookmark of the found record to be used by the Find\_OnExit() function.

Before the found record can be presented, the BeforeUpdate event must be canceled, and a SendKeys action must send two ESC keys to undo changes to the current record.

Next, the SendKeys action sends a TAB key to exit the field. This event triggers the Find\_OnExit() function, which checks to see if the find was successful. If it was, the CancelEvent action is run to prevent exiting the control, and then the form record is synchronized with the found record by setting its bookmark equal to the bookmark of the found record. Found is then reset back to NULL.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, pages 651-652

Additional reference words: 2.00 findrecord

KBCategory: kbusage

KBSubcategory: FmsOthr



## INF: How to Find Access Basic Error Codes

Article ID: Q105666

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

The Microsoft Access manuals do not contain a listing of all the error codes and messages you can receive when you are running custom Access Basic functions. To obtain the error codes and messages, you can run a custom Access Basic function and use the Microsoft Access Help system.

### MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

There are approximately 700 error messages that can result from running Access Basic code. To print a list of the error values and descriptions to the Immediate window, add the sample function below to a module:

```
Function PrintErrorStrings ()
 For i = 1 To 32767
 E$ = Error$(i)
 If E$ <> "User-defined error" And E$ <> "Reserved Error" Then
 Debug.Print i; Error$(i)
 End If
 Next i
End Function
```

The following example demonstrates how to save the error code list in a table:

1. Create a new table with the following fields. Save the table as Error Codes:

Table Name: Error Codes

-----  
FieldName: Err  
    DataType: Number  
    FieldSize: Integer  
FieldName: Error  
    DataType: Text  
    FieldSize: 255

2. Revise the PrintErrorStrings() function as follows:

```

Function PrintErrorStrings ()
 Dim MyDB As Database, MyTable As Table
 Set MyDB = CurrentDB()
 Set MyTable = MyDB.OpenTable("Error Codes")
 For i = 1 To 32767
 E$ = Error$(i)
 If E$ <> "User-defined Error" And E$ <> "Reserved Error" Then
 MyTable.AddNew
 MyTable![Err] = i
 MyTable![Error] = E$
 MyTable.Update
 End If
 Next i
 MyTable.Close
 MyDB.Close
End Function

```

3. Type the following in the Immediate window, and then press ENTER:

```
? PrintErrorStrings()
```

The error codes and descriptions will be written to the Error Codes table.

The Microsoft Access Help system contains descriptions of the errors. However, the error values are not in the Help system. To determine the error value, match the description to the values returned by the PrintErrorStrings() function.

#### REFERENCES

=====

For more information on error messages, search for "error messages: reference" then "Error Messages Reference" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00  
 KBCategory: kbprg  
 KSubcategory: PgmErr

## INF: How to Find If an Object Exists in the Database

Article ID: Q90989

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a sample user-defined function that you can use to check for the existence of an object in the current database before you create a new object (such as a table, form, or report) in the database.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

To create the function, open a new module and enter the appropriate code for your version of Microsoft Access.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

Microsoft Access Version 1.x

-----

```
'*****
'Declarations section of the module
'*****

Option Explicit

'*****
' FUNCTION: ObjectExists_lx()
'
' PURPOSE:
' Determines whether the specified object exists in the
' current database.
'
' ARGUMENTS:
' ObjectType - A string representing the object whose names are to
' be retrieved. ObjectType can be one of the
' following:
'
' Tables, Queries, Forms, Reports, Macros, Modules
'
' ObjectName - A string representing the name of the specified
' ObjectType.
```

```

'
' RETURNS:
' True (-1), if the object exists.
' False (0), if the object does not exist.
'
' NOTES:
' This function uses information stored in the MSysObjects
' system table of the current database. The system tables
' are undocumented and are subject to change in future versions of
' Microsoft Access.
'
'*****
Function ObjectExists_1x% (ObjectType As String, ObjectName As String)
 On Error Resume Next

 Dim db As Database
 Dim ss As Snapshot
 Dim SQL
 Dim Msg As String

 SQL = "Select Name, Type from MSysObjects Where Type="

 ObjectExists_1x% = 0

 Select Case ObjectType
 Case "Tables"
 SQL = SQL & "1 And [Name] = '" & ObjectName & "' Order By Name;"
 Case "Queries"
 SQL = SQL & "5 And [Name] = '" & ObjectName & "' Order By Name;"
 Case "Forms"
 SQL = SQL & "-32768 And [Name] = '" & ObjectName & "' Order By _
 Name;"
 Case "Reports"
 SQL = SQL & "-32764 And [Name] = '" & ObjectName & "' Order By _
 Name;"
 Case "Macros"
 SQL = SQL & "-32766 And [Name] = '" & ObjectName & "' Order By _
 Name;"
 Case "Modules"
 SQL = SQL & "-32761 And [Name] = '" & ObjectName & "' Order By _
 Name;"
 Case Else
 Msg = "Object Name '" & ObjectName & "' is an invalid"
 Msg = Msg & " argument to function ObjectExists_1x!"
 MsgBox Msg, 16, "ObjectExists_1x"
 Exit Function
 End Select

 Set db = CurrentDB()
 Set ss = db.CreateSnapshot(SQL)

 ss.MoveLast
 If ss.RecordCount > 0 Then
 ObjectExists_1x% = -1
 End If

End Function

```

Microsoft Access Version 2.0

```

'*****
'Declarations section of the module
'*****
```

Option Explicit

```
'*****
' FUNCTION: ObjectExists_20()
'
' PURPOSE:
' Determines whether the specified object exists in the
' current database.
'
' ARGUMENTS:
' ObjectType - A string representing the object whose names are to
' be retrieved. ObjectType can be one of the
' following:
'
' Tables, Queries, Forms, Reports, Macros, Modules
'
' ObjectName - A string representing the name of the specified
' ObjectType.
'
' RETURNS:
' True (-1), if the object exists.
' False (0), if the object does not exist.
'
' NOTES:
' This function uses data access objects (DAO) to determine if
' the specified object exists in the current database.
'
'*****
```

```
Function ObjectExists_20% (ObjectType$, ObjectName$)
On Error Resume Next
```

```
Dim Found_Object%, Find_Object As String, ObjectNum As Integer
Dim DB As Database, T As TableDef
Dim Q As QueryDef, C As Container
Dim Msg As String
Found_Object% = -1
Set DB = dbengine(0)(0)
```

```
Select Case ObjectType$
Case "Tables"
 Find_Object = DB.TableDefs(ObjectName$).Name

Case "Queries"
 Find_Object = DB.QueryDefs(ObjectName$).Name

Case Else
 If ObjectType$ = "Forms" Then
 ObjectNum = 1
```

```

ElseIf ObjectType$ = "Modules" Then
 ObjectNum = 2
ElseIf ObjectType$ = "Reports" Then
 ObjectNum = 4
ElseIf ObjectType$ = "Macros" Then
 ObjectNum = 5
Else
 Msg = "Object Name "" & ObjectType & "" is an invalid"
 Msg = Msg & " argument to function ObjectExists_20!"
 MsgBox Msg, 16, "ObjectExists_20"
 Exit Function

End If

Set C = DB.Containers(ObjectNum)
Find_Object = C.Documents(ObjectName$).Name

End Select

If Err = 3265 Or Find_Object = "" Then
 Found_Object% = 0
End If

ObjectExists_20% = Found_Object%

End Function

```

#### How to Use the Function

-----

1. With the module containing the function open in Design view, choose Immediate Window from the View menu.
2. If you are using Microsoft Access version 1.x, type the following line in the Immediate window, then press ENTER:

```
?ObjectExists_1x("Tables","Employees")
```

If you are using Microsoft Access version 2.0, type the following line in the Immediate window, then press ENTER:

```
?ObjectExists_20("Tables","Employees")
```

If a table named Employees exists in the current database, -1 will be returned. If no table named Employees exists, 0 will be returned.

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.1, Chapter 8, "Manipulating Data," pages 124-127

Microsoft Access "Building Applications," version 2.0, Chapter 7, "Objects and Collections," pages 170-181

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: MdlDao

## INF: How to Find the Number of Days In a Month

Article ID: Q109703

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article demonstrates two methods of returning the total number of days in the current month. The first method uses a query, and the second method uses a user-defined Access Basic function.

### MORE INFORMATION

=====

#### Query Method

-----

Create the following new query based on any table:

NOTE: In the following example, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this example.

Query: QueryTest

-----  
Field: DateDiff("d",Date()-(Day(Date()- 1)), DateSerial(Year \_  
                  (Date()),(Month(Date())+1),1))  
Total: First  
Show: True

This query will return an integer for the number of days in the current month. For example, if the current month is April, the query will return the integer 30.

#### User-Defined Access Basic Function

-----

NOTE: This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

Create the following sample function in a new or existing module:

```
'*****
'Declarations Section of Module
'*****
Option Compare Database
Option Explicit
```



```

'*****
'DaysInMonth Function
'*****
' This function takes a date as an argument and returns
' the total number of days in the month.
Function DaysInMonth (MyDate)
 Dim NextMonth, EndOfMonth
 NextMonth = DateAdd("m", 1, MyDate)
 EndOfMonth = NextMonth - DatePart("d", NextMonth)
 DaysInMonth = DatePart("d",EndOfMonth)
End Function

```

Type the following expression in the module's Immediate window:

```
?DaysInMonth(Date())
```

The number of days in the current month will be returned. To obtain the number of days in the month of a specific date, enter the specific date as follows:

```
?DaysInMonth(#11/4/93#)
```

The integer 30 will be returned in the Immediate window, since there are 30 days in the month of November.

To use this function in a query or form, use an equal sign (=) instead of the question mark (?) before the function name:

```
=DaysInMonth(<date value or variable>)
```

#### REFERENCES

```
=====
```

For more information on date functions, search for "date" and then check each of the date function topics using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

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

**Article ID: Q90810**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### 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), in the message anywhere you want to start a new line.

### MORE INFORMATION

=====

For example, if you had the message:

Notice: This is an Important Message!

And instead of having it all display on one line, or letting Microsoft Access determine where to break the message onto a new line, you want to force it to display as:

Notice:  
This is an Important Message!

To force the message to display like this you must enter the message as a string expression. Use the following syntax:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

In a Macro:

-----

```
Action

MsgBox
 Message: ="Notice:" & Chr(13) & Chr(10) & "This is an Important _
 Message!"
```

NOTE: You must use the equal sign with the MsgBox action to tell the action you are using a String Expression as the message argument.

In Access Basic:

-----

```
MsgBox "Notice:" & Chr(13) & Chr(10) & "This is an Important_
 Message!"
```

NOTE: When using the MsgBox statement or MsgBox function in Access Basic you do not need the equal sign before the message because this

statement/function expects a string expression as the message argument.

#### REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: McrArg

## **INF: How to Format Data in Combo Boxes and List Boxes**

**Article ID: Q120904**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

Microsoft Access displays data in list boxes and combo boxes whose RowSource property is set to a table or query using the format defined in the underlying fields' Format property.

If you want the data in a list box or combo box to be displayed with a different format, you can create a query and set the fields' formats using the query's Format property, or you can use the Format() function in an expression.

If a list box or combo box's RowSource property is set to an SQL statement, a field's Format property will have no affect on how the data is displayed, since SQL statements do not contain or specify any formatting information.

### MORE INFORMATION

=====

To change the format for a field in a query, follow these steps:

1. Select the field in the query grid.
2. From the View menu, choose Properties.
3. Select one of the predefined formats in the Format property box, or enter an expression to format the field.
4. Close the property sheet.

The following are examples of expressions for formatting fields in a query:

- The expression

```
format([MyDate],"dddd mmmm d, yyyy")
```

will display the date 1/1/94 as:

```
Saturday January 1, 1994
```

- The expression

```
format([MyNumber],"$#,##0;($#,##0)")
```

will display the number 1256 as

```
$1,256
```

and will display the number -1256 as:

(\$1,256)

#### REFERENCES

=====

For more information about formatting data, search for "Format" then "Formatting a Field" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsCmbo

**INF: How to Get and Use the Microsoft Access Window Handle**  
**Article ID: Q113881**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes a sample user-defined Access Basic function that you can use to get the handle to the Microsoft Access window. This article also describes how to use the sample function to:

- Minimize, maximize, and restore the Microsoft Access window.
- Determine if the Microsoft Access window is minimized, maximized or restored.
- Move and size the Microsoft Access window.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

Every window in the Microsoft Windows environment has a unique number, or window handle, assigned to it that is used to identify the window. The window handle is a required argument for many Microsoft Windows application programming interface (API) functions.

The following steps describe how to create the sample function GetAccessshWnd() that you can use to get the Microsoft Access window handle:

Notes:

- In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.
- You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.

1. Create a new module.

2. Add the following lines to the module's Declarations section:

Option Explicit

```

Declare Function GetActiveWindow Lib "User" () As Integer
Declare Function GetParent Lib "User" (ByVal hWnd As Integer) _
 As Integer

```

3. Enter the following function in the module:

```

Function GetAccessshWnd ()
 Dim hWnd As Integer
 Dim hWndAccess As Integer

 ' Get the handle to the currently active window.
 hWnd = GetActiveWindow()
 hWndAccess = hWnd

 ' Find the top window without a parent window.
 While hWnd <> 0
 hWndAccess = hWnd
 hWnd = GetParent(hWnd)
 Wend

 GetAccessshWnd = hWndAccess

End Function

```

How to Use the GetAccessshWnd() Function

---

General Use:

1. Choose Immediate Window from the View menu in the module's Design view.
2. Type the following and then press ENTER:

```
? GetAccessshWnd()
```

The window handle for the Immediate window will be returned.

How to Minimize, Maximize, or Restore the Microsoft Access Window:

1. Add the following to the module's Declarations section:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```

Declare Function ShowWindow% Lib "User" (ByVal hWnd%, _
 ByVal nCmdShow%)

Global Const SW_MAXIMIZE = 3
Global Const SW_SHOWNORMAL = 1
Global Const SW_SHOWMINIMIZED = 2

```

2. Enter the following functions in the module:

```

Function AccessMinimize()
 AccessMinimize = ShowWindow(GetAccessshWnd(), SW_SHOWMINIMIZED)
End Function

```

```

Function AccessMaximize()
 AccessMaximize = ShowWindow(GetAccessshWnd(), SW_MAXIMIZE)
End Function

```

```

Function AccessRestore()
 AccessRestore = ShowWindow(GetAccessshWnd(), SW_SHOWNORMAL)
End Function

```

3. To run these functions, type "? <function name>()" (without quotation marks) in the module's Immediate window.

How to Determine If the Microsoft Access Window Is Minimized, Maximized, or Restored:

1. Add the following to the module's Declarations section:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```

Declare Function IsIconic Lib "User" (ByVal hWnd As Integer) _
 As Integer
Declare Function IsZoomed Lib "User" (ByVal hWnd As Integer) _
 As Integer

```

2. Enter the following functions in the module:

```

Function IsAccessMaximized ()
 If IsZoomed(GetAccessshWnd()) = 0 Then
 IsAccessMaximized = False
 Else
 IsAccessMaximized = True
 End If
End Function

```

```

Function IsAccessMinimized ()
 If IsIconic(GetAccessshWnd()) = 0 Then
 IsAccessMinimized = False
 Else
 IsAccessMinimized = True
 End If
End Function

```

```

Function IsAccessRestored ()
 If IsAccessMaximized() = False And _
 IsAccessMinimized() = False Then
 IsAccessRestored = True
 Else
 IsAccessRestored = False
 End If
End Function

```

3. To run these functions, type "? <function name>()" (without quotation marks) in the module's Immediate window. For example, type

```
? IsAccessMaximized()
```



to return a -1 if the Microsoft Access window is maximized, or a 0 if it is not.

How to Move and Size the Microsoft Access Window:

1. Add the following to the module's Declarations section:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Declare Sub MoveWindow Lib "User" (ByVal hWnd As Integer, _
 ByVal X As Integer, ByVal Y As Integer, _
 ByVal nWidth As Integer, ByVal nHeight As Integer, _
 ByVal bRepaint As Integer)
```

2. Enter the following function in the module:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function AccessMoveSize (iX As Integer, iY As Integer, _
 iWidth As Integer, iHeight As Integer)
 MoveWindow GetAccesshWnd(), iX, iY, iWidth, iHeight, True
End Function
```

To move the Microsoft Access window to the upper left corner of the screen and size it to the standard VGA display size of 640 x 480 pixels, type the following in the module's Immediate window:

```
? AccessMoveSize(0, 0, 640, 480)
```

On a computer configured with the standard VGA video driver, this will give the Microsoft Access window the appearance of being maximized, although it is really restored and sized to fill the screen. Note that the dimensions you supply to this function are in pixels.

#### REFERENCES

=====

For information about how to determine the current screen resolution, query on the following words here in the Microsoft Knowledge Base:

get and screen and resolution

Additional reference words: 1.00 1.10 2.00 hWnd hWindow hWin

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: How to Get Form Window's Right and Down Measurements**  
**Article ID: Q121100**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to create and use an Access Basic Sub procedure called GetFormDimensions to obtain a form window's right and down measurements for use in the MoveSize macro action. This article also demonstrates how to position a form immediately below or to the right of a second form.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

In Microsoft Access version 2.0, you can use the WindowWidth and WindowHeight form properties to determine the current width and height of the window a form is contained in. However, there are no properties to determine the window's right and down measurements (these measurements are commonly used in the MoveSize macro action to position a form window).

The right and down measurements, along with the width and height, are useful if you want to position a form side-by-side with another form. The following examples demonstrate how to create and use the GetFormDimensions Sub procedure to obtain these measurements and use them to position a form to the right of, and then below, another form using the MoveSize macro action.

NOTE: In the following sample code, an underscore (   ) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

How to Create the Sub Procedure  
-----

1. Open the sample database NWIND.MDB.
2. Create a new module and enter the following lines in the module's Declarations section:

```
Option Explicit

Type Rect
 x1 As Integer
 y1 As Integer
 x2 As Integer
```

```

 y2 As Integer
End Type

Declare Sub GetWindowRect Lib "User" (ByVal hWnd As Integer, _
 lpRect As Rect)
Declare Function GetDC Lib "User" (ByVal hWnd As Integer) _
 As Integer
Declare Function ReleaseDC Lib "User" (ByVal hWnd As Integer, _
 ByVal hDC As Integer) As Integer
Declare Function GetDeviceCaps Lib "GDI" (ByVal hDC As Integer, _
 ByVal nIndex As Integer) As Integer
Declare Function GetActiveWindow Lib "User" () As Integer
Declare Function GetParent Lib "User" (ByVal hWnd As Integer) _
 As Integer
Declare Function GetClassName Lib "User" (ByVal hWnd As Integer, _
 ByVal lpClassName As String, ByVal nMaxCount As Integer) _
 As Integer

Global Const TWIPSPERINCH = 1440

```

3. Create the following procedures in the module:

```

Sub GetFormDimensions (F As Form, FRight, FDown, FWidth, FHeight)
'*****
' PURPOSE: Returns the right, down, width, and height
' measurements of a form window in twips.
' ARGUMENTS:
' F: The form object whose measurements are to be determined.
' FRight, FDown, FWidth, FHeight: Measurement variables
' that will return the dimensions of form F "by reference."
' NOTE: The FWidth and FHeight values will be equivalent to
' those provided by the form WindowWidth and WindowHeight
' properties.
'*****
 Dim FormRect As Rect
 Dim MDIClient As Rect
 Dim MDIClientRight, MDIClientDown

 ' Get the screen coordinates and window size of the form.
 ' The screen coordinates are returned in pixels measured
 ' from the upper left corner of the screen.
 GetWindowRect F.hWnd, FormRect
 FRight = FormRect.x1
 FDown = FormRect.y1
 FWidth = FormRect.x2 - FormRect.x1
 FHeight = FormRect.y2 - FormRect.y1

 ' Convert the measurements from pixels to twips.
 ConvertPIXELSToTWIPS FRight, FDown
 ConvertPIXELSToTWIPS FWidth, FHeight

 ' If the form is not a pop-up form, adjust the screen
 ' coordinates to measure from the top of the Microsoft
 ' Access MDIClient window. Position 0,0 for a pop-up form
 ' is the upper left corner of the screen, whereas position
 ' 0,0 for a normal window is the upper left corner of the
 ' Microsoft Access client window below the menu bar.

```

```

If GetWindowClass(F.hWnd) <> "OFormPopup" Then
 ' Get the screen coordinates and window size of the
 ' MDIClient window.
 GetWindowRect GetParent(F.hWnd), MDIClient
 MDIClientRight = MDIClient.x1
 MDIClientDown = MDIClient.y1
 ConvertPIXELSToTWIPS MDIClientRight, MDIClientDown

 ' Adjust the form dimensions from the MDIClient
 ' measurements.
 FRight = FRight - MDIClientRight
 FDown = FDown - MDIClientDown
End If
End Sub

Sub ConvertPIXELSToTWIPS (X, Y)
'*****
' PURPOSE: Converts the two pixel measurements passed as
' arguments to twips.
' ARGUMENTS:
' X, Y: Measurement variables in pixels. These will be
' converted to twips and returned through the same
' variables "by reference."
'*****
 Dim hDC As Integer, hWnd As Integer, RetVal As Integer
 Dim XPIXELSPERINCH, YPIXELSPERINCH
 Const LOGPIXELSX = 88
 Const LOGPIXELSY = 90

 ' Retrieve the current number of pixels per inch, which is
 ' resolution-dependent.
 hDC = GetDC(0)
 XPIXELSPERINCH = GetDeviceCaps(hDC, LOGPIXELSX)
 YPIXELSPERINCH = GetDeviceCaps(hDC, LOGPIXELSY)
 RetVal = ReleaseDC(0, hDC)

 ' Compute and return the measurements in twips.
 X = (X / XPIXELSPERINCH) * TWIPSPERINCH
 Y = (Y / YPIXELSPERINCH) * TWIPSPERINCH
End Sub

Function GetWindowClass (hWnd) As String
'*****
' PURPOSE: Retrieve the class of the passed window handle.
' ARGUMENTS:
' hWnd: The window handle whose class is to be retrieved.
' RETURN:
' The window class name.
'*****
 Dim Buff As String
 Dim BuffSize As Integer
 Buff = String$(255, " ")
 BuffSize = GetClassName(hWnd, Buff, 255)
 GetWindowClass = Left$(Buff, BuffSize)
End Function

```

How to Use the GetFormDimensions Sub Procedure

- 
1. Open the sample database NWIND.MDB.
  2. Add the following Sub procedure to the module that you created in step 2 above:

```
Sub MoveEmployeesAround ()
 Dim frmCust As Form, frmEmp As Form
 Dim CustRight, CustDown, CustWidth, CustHeight
 Dim EmpRight, EmpDown, EmpWidth, EmpHeight

 Set frmCust = Forms!Customers
 Set frmEmp = Forms!Employees

 GetFormDimensions frmCust, CustRight, CustDown, _
 CustWidth, CustHeight
 GetFormDimensions frmEmp, EmpRight, EmpDown, _
 EmpWidth, EmpHeight

 frmEmp.SetFocus

 MsgBox "Move Employees to the right of Customers!"
 DoCmd MoveSize CustRight + CustWidth, CustDown

 MsgBox "Move Employees below Customers!"
 DoCmd MoveSize CustRight, CustDown + CustHeight
End Sub
```

3. Open the Customers and Employees forms in Form view.
4. Select the module, and then choose Immediate Window from the View menu.
5. Type the following line in the Immediate window, and then press ENTER:

```
MoveEmployeesAround
```

6. In the message box that appears, choose OK. The Employees form will be positioned immediately to the right of the Customers form, and then another message box will appear.
7. In the second message box, choose OK. The Employees form will be positioned immediately below the Customers form.

NOTE: Depending on the position and size of the Customers form, the Employees form may be positioned so that you cannot see it.

#### REFERENCES

=====

For more information about the MoveSize macro action, search for "MoveSize" then "MoveSize Action" using the Microsoft Access Help menu.

For more information about the WindowHeight and WindowWidth properties, search for "WindowHeight" then "WindowHeight, WindowWidth Properties" using the Microsoft Access Help menu.

Additional reference words: 2.00  
KBCategory: kbusage  
KSubcategory: FmsProp

**INF: How To Get Microsoft Works Files into Microsoft Access**  
**Article ID: Q93698**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

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 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. Choose the Import button. The file is imported as a Microsoft Access table with the same name as the .DBF file.

#### Microsoft Works for MS-DOS Database files

---

Save your Microsoft Works 2.0 database files in the format "Text & Commas". Microsoft Access imports this file type as Text (delimited) files.

Save your Microsoft Works 3.0 database files in dBase III or dBase IV format following the method outlined above.

#### Microsoft Works spreadsheet files (either version)

---

Spreadsheet files, created in either the MS-DOS or Windows version of Works, can be imported by Microsoft Access with little effort. If the first row of the spreadsheet contains a name for each column this can be recognized as the name of each field in the new Microsoft Access table. If there is only data in the spreadsheet Microsoft Access creates default field names for the table. Most data types are retained, with these exceptions:

- Fields formatted as TRUE or FALSE are converted to Number fields. A FALSE value is converted to 0 and a TRUE value is converted to 1.
- Formula fields are converted to Number fields with values.

#### Conversion process:

1. Start Microsoft Access and open the .MDB file you want to import the Microsoft Works spreadsheet file to.
2. From the File menu, choose Import.
3. Select Lotus (WKS) from the format list.
4. Select the name of the .WKS file from the list. If it does not appear, you may need to change the drive and directory to the correct location.
5. If the first row of the spreadsheet contains the column names, select the option "First Row Contains Field Names".
6. Select the Import button. The file will be imported as a Microsoft Access table with the same name as the .WKS file.

Depending upon the type of data in your Microsoft Works files, you may choose either method to convert files to Microsoft Access tables.



## REFERENCES

=====

For more information about sharing Microsoft Works data with other programs, see the Microsoft Works "User's Guide."

Microsoft Access "User's Guide," version 1.0, pages 64-65 and 72-75

Additional reference words: 1.00 1.10 2.00 spreadsheet dbase

KBCategory: kbinterop

KBSubcategory: IntpOthr

**INF: How to Get Record Position Number from Record Selector**  
**Article ID: Q103260**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Microsoft Access does not provide a way to retrieve the position number found in the record selector at the bottom of the form or datasheet.

This article demonstrates a sample user-defined Access Basic function called GetRecordPosition() that you can use to retrieve the current record position. This article also explains how to use the GetRecordPosition() function to create your own custom record position box on a form.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

The number that appears in the record selector is actually not a record number. It is a number used to indicate the record's relative position in the current record set, not the record's natural position in the table.

The following Access Basic function, GetRecordPosition(), can be used to get the record's relative position in the record set. Add this function to a new or existing module.

```
'Global Declaration Section
Option Explicit
```

```
'*****
' FUNCTION: GetRecordPosition
'
' PURPOSE:
' Gets the number representing the relative position of the
' record in the currently active form. This is the number that
' appears in the record selector found at the bottom of a form
' or datasheet.
'
' RETURN:
' The relative record position. If there are no records in
' the active form's dynaset, or if there is no active form,
' Null is returned.
'
'*****
```

```

Function GetRecordPosition ()
 Const ERR_NO_CURRENT_RECORD = 3021
 Dim DS As Dynaset
 Dim RecordPos As Long

 On Error GoTo Err_GetRecordPosition

 Set DS = Screen.ActiveForm.Dynaset
 DS.Bookmark = Screen.ActiveForm.Bookmark
 RecordPos = 0

 While Not DS.BOF
 DS.MovePrevious
 RecordPos = RecordPos + 1
 Wend

 GetRecordPosition = RecordPos

Bye_GetRecordPosition:
 Exit Function

NoCurrentRec_GetRecordPosition:
 On Error Resume Next
 DS.MoveLast
 If Err <> 0 Then
 ' There must be no records in the set, so new record is 1
 GetRecordPosition = 1
 Else
 ' We are on last record of set, so count + 1
 GetRecordPosition = DS.RecordCount + 1
 End If
 GoTo Bye_GetRecordPosition

Err_GetRecordPosition:
 Select Case Err
 Case ERR_NO_CURRENT_RECORD
 Resume NoCurrentRec_GetRecordPosition
 Case Else
 Resume Bye_GetRecordPosition
 End Select
End Function

```

#### Creating a Record Position Box on a Form

---

Use the following steps to create your own custom record position box on a form using the GetRecordPosition() function:

1. Create a new macro called Refresh Record Position with the following macro action:

```

Requery
 ControlName: Record Position

```

2. Open your form in Design view. Make sure that the Form Hdr/Ftr option is selected on the Layout menu.

3. Create a text box in the footer with the following properties:

Name: Record Position (ControlName: Record Position in version 1.x)  
ControlSource: =GetRecordPosition()

4. Set the following properties on the form (not on the control):

OnCurrent: Refresh Record Position  
AfterUpdate: Refresh Record Position

The current record position will display in the text box in the footer of the form either when you navigate to a different record or when a new record is created.

NOTE: The AfterUpdate property of the form is set in the event a new record is created and written to disk. The current record position may need to account for this new record.

#### REFERENCES

=====

For more information on record numbers, query on the following words here in the Microsoft Knowledge Base:

access and record and numbers and position

Additional reference words: 1.00 1.10 2.00 VCR

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: How to Get the Name and Path of the Current Database**  
**Article ID: Q113919**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

This article describes a sample user-defined Access Basic function that uses data access objects (DAO) to return the name and path of the current database.

MORE INFORMATION

=====

A Database object in the Databases collection has a Name property that will return the name and path of the database.

The following Access Basic function demonstrates how to access the Name property of the current database:

1. Create a new module with the following in the Declarations section:

Option Explicit

2. Enter the following function in the module:

```
Function GetNamePath ()
 Dim MyDB As Database

 ' Set MyDB to the current database.
 Set MyDB = DBEngine.Workspaces(0).Databases(0)

 ' Return the value in the Name property.
 GetNamePath = MyDB.Name
End Function
```

3. From the View menu, choose Immediate Window.
4. In the Immediate window, type the following, and then press ENTER:

```
? GetNamePath()
```

The path and name of the current database will be displayed in the Immediate window.

NOTE: Do not confuse this function with the CurDir\$() function, which returns the path of the current directory. The current directory may not be the directory where the current database is located.

REFERENCES

=====

For more information about the Database object and Databases collection, search for "Database" then "Database Object, Databases Collection," using the Microsoft Access Help menu.

Additional reference words: 2.00 AB Programming

KBCategory: kbprg

KBSubcategory: PgmOthr

## INF: How to Give Users the Ability to Change Passwords

Article ID: Q100970

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

### SUMMARY

=====

This article describes how to distribute a secure application with the Microsoft Access Distribution Kit or Developer's Toolkit and still give your users the ability to change their passwords.

### MORE INFORMATION

=====

To achieve this functionality, do the following:

1. Make the Database window active so that you can access the Security menu. (The best method for this is to select an object in the Database window through a macro).
2. From the Security menu, choose Password. The user then changes the password.
3. Once the user has closed the Password dialog box, set the focus back to the form you called the macro or function from.

Here is a sample macro:

Macro:Change Password

-----

Actions

-----

SelectObject

DoMenuItem

SelectObject

Action arguments

-----

SelectObject

Object Type: Macro

Object Name: Autoexec

In Database window: Yes

DoMenuItem

Menu Bar: Database

Menu Name: Security

Command: Change Password

Subcommand:

SelectObject

Object Type: Form

Object Name: Form1  
In Database window: No

The Database window appears, as well as the dialog box to change the password. You cannot select anything in the Database window because the Change Password dialog box is modal.

In the Microsoft Access Distribution Kit or Developer's Toolkit, the functionality is identical except that the Database window is not displayed.

Additional reference words: 1.00 1.10 2.00 container adk adt  
KBCategory: kbusage  
KBSubcategory: ScrtOthr



## INF: How to Group Column Headings in a Crosstab Query

Article ID: Q109949

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to group values in a crosstab query using a fixed number of headings, rather than having a column for every value.

### MORE INFORMATION

=====

The following example demonstrates how to use the Switch() function to group non-numeric values under four fixed headings. The headings are A-F, G-O, P-Z, and Other.

1. Open the sample database NWIND.MDB.
2. Create the following query based on the Orders and Customers tables.

NOTE: In the following example, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this example.

Query: Switch Limits Headings

-----  
Type: Crosstab Query  
Join: Orders.[Customer ID] <-> Customers.[Customer ID]  
Field: Country  
    Table: Customers  
    Total: Group By  
    Crosstab: Row Heading  
Field: Expr1: Switch([Company Name] Like "[A-Fa-f]\*", "A-F", \_  
    [Company Name] Like "[G-Og-o]\*", "G-O", [Company Name]\_  
    Like "[P-Zp-z]\*", "P-Z", True, "Other")  
    Total: Group By  
    Crosstab: Column Heading  
Field: Order ID  
    Table: Orders  
    Total: Count  
    Crosstab: Value

Note that both "A-F" and "a-f" are required in the Switch() function's argument because the argument is case sensitive. The "True" in the argument acts like an Else condition, collecting data that does not match any of the previous conditions.

3. Run the query.

### REFERENCES

=====

For more information about crosstab queries, search for "crosstab queries" then "Summarizing Data Using a Crosstab Query" using the Microsoft Access Help menu.

For more information about the Switch() function, search for "Switch" then "Switch Function" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryCross

**INF: How to Hide the Combo Box Drop-Down Arrow on a Form**  
**Article ID: Q109724**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article demonstrates a method to hide the drop-down arrow on a combo box until the combo box is selected.

MORE INFORMATION

=====

The following steps demonstrate how to hide the drop-down arrow on a combo box when the combo box is not selected:

1. Open the sample database NWIND.MDB.
2. Open the Orders form in Design view.
3. Place a rectangle control on the form. Size and move the rectangle control so that it completely covers the drop-down arrow on the Salesperson combo box.
4. From the View menu, select Palette. Select a color in the Fill line to match the form's background color.
5. View the form in Form view.

The drop-down arrow on the Salesperson combo box will only be visible when the combo box is selected.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 9, "Designing Forms," pages 235-245, version 1.1, pages 239-249

Additional reference words: 1.00 1.10 2.00 hide combo box

KBCategory: kbusage

KBSubcategory: FmsCmbo

## INF: How to Hide the Database Window at Startup

Article ID: Q88160

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

During normal startup of Microsoft Access, the Database window is visible. If you do not want to give a user access to the Database window, you can create an Autoexec macro that will hide the Database window. This article describes how to create a macro to do this.

### MORE INFORMATION

=====

When a Microsoft Access database is first loaded, Microsoft Access checks for an Autoexec macro and runs it if it exists. The following macro will hide the Database window from the user:

1. From the File menu, choose New to create a new database. Give the database a unique name.
2. From the File menu, choose New and then choose Macro to create a new macro.
3. Add the following actions:

| Action       | Comment                                                                             |
|--------------|-------------------------------------------------------------------------------------|
| SelectObject | Select an object 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.
6. Close and reopen the database.

The macro will hide the Database window. However, menu items (Show window on the File menu) will still exist to show the Database window. To hide these items, you can create a custom menu for your application, or disable the Show command on the File menu.

#### REFERENCES

=====

For more information on custom menus, search for "customizing menus" using the Microsoft Access Help menu.

For more information on using Windows API function calls to disable menu commands, query on the following words here in the Microsoft Knowledge Base:

GetMenu and GetSubMenu

Additional reference words: 1.00 1.10 2.00 gray grey dimmed container

KBCategory: kbusage

KBSubcategory: McrActn

## **INF: How to Import a Microsoft Windows Cardfile Data File**

**Article ID: Q98797**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a sample user-defined Access Basic function that you can use to import a Microsoft Windows Cardfile (.CRD) file.

### MORE INFORMATION

=====

The function below can be used to import a Microsoft Windows Cardfile data file. To use this function, you need to add it to a new or existing Microsoft Access module. In addition, you need to create the following table in your database before running the function:

Table Name: CardFile

-----

Field: Title

Type: Text

Length: 50

Field: Comment

Type: Text

Length: 255

You can run the function from the Immediate window, or you can call it from a command button by setting a command button's OnPush property to:

```
=Import_CRD()
```

NOTE: In Microsoft Access version 2.0, the OnPush property is named the OnClick property.

Make sure to substitute the correct path and filename for the .CRD file you want to import in the function below. Or, you can pass the path and filename to the function as a string argument.

Code for the Import\_CRD() Function

-----

```
Option Compare Binary
```

```
Option Explicit
```

```
Type CRD_Record_Type
```

```
Reserved As String * 6
```

```
Position As Long
```

```
Flag As String * 1
```

```
Index As String * 40
```

```
EndOfRecord As String * 1
```

```
End Type
```

```

Function Import_CRD ()
 Const VER30 = &H474D&
 Const VER31 = &H5252&
 Const VER30_NUMRECORDS_OFFSET = &H4&
 Const VER31_NUMRECORDS_OFFSET = &H8&
 Const VER30_FIRSTENTRY = &H6&
 Const VER31_FIRSTENTRY = &HA&

 Dim MyDB As Database
 Dim MyTable As Table

 Dim FileNum#
 Dim NumberOfCards%, NextRecord&, I%, VersionInfo%, EntryLength%
 Dim CRD_Record As CRD_Record_Type
 Dim Message$
 Dim n As Integer

 Set MyDB = CurrentDB()
 Set MyTable = MyDB.OpenTable("CardFile")

 FileNum# = FreeFile
 'Be sure to supply the correct path and filename for the .CRD file.
 Open "c:\my_cards.crd" For Binary As FileNum#

 'The number of files in the .CRD file is at the 4 bytes offset.
 Get #FileNum, , VersionInfo%
 If VersionInfo% = VER31 Then
 Get #FileNum, VER31_NUMRECORDS_OFFSET, NumberOfCards%
 NextRecord& = VER31_FIRSTENTRY
 ElseIf VersionInfo% = VER30 Then
 Get #FileNum, VER30_NUMRECORDS_OFFSET, NumberOfCards%
 NextRecord& = VER30_FIRSTENTRY
 End If

 For I% = 1 To NumberOfCards%

 Get #FileNum, NextRecord&, CRD_Record
 Get #FileNum, CRD_Record.Position + 3, EntryLength%

 Message$ = Space$(EntryLength%)

 Get #FileNum, CRD_Record.Position + 5, Message$

 MyTable.AddNew
 n = InStr(CRD_Record.Index, Chr$(0))
 If n > 1 Then
 MyTable.Title = Left$(CRD_Record.Index, n - 1)
 ElseIf n = 1 Then
 MyTable.Title = ""
 Else
 MyTable.Title = CRD_Record.Index
 End If

 n = InStr(Message$, Chr$(0))
 If n > 1 Then
 MyTable.Comment = Left$(Message$, n - 1)
 End If
 Next I%
End Function

```

```
 ElseIf n = 1 Then
 MyTable.Comment = ""
 Else
 MyTable.Comment = Message$
 End If
 MyTable.Update

 NextRecord& = NextRecord& + Len(CRD_Record)

Next I%
Close
End Function
```

#### REFERENCES

=====

For information about the format of .CRD files, query on the following words here in the Microsoft Knowledge Base:

cardfile and format and graphic and absolute and hexadecimal

Additional reference words: 1.00 1.10 2.00 .crd

KBCategory: kbinterop

KBSubcategory: IntpOthr



## INF: How to Import FoxBase for Macintosh Files

Article ID: Q109331

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

FoxBase for Macintosh files can be imported into or attached to Microsoft Access by specifying a dBASE III data source file type.

### MORE INFORMATION

=====

Although you can import or attach FoxBase for Macintosh files in Microsoft Access by specifying a dBASE III data source file type, make sure to note the following restrictions:

- You cannot import FoxBase for Macintosh files that contain fields with a PICT data type. The PICT data type is similar to the Microsoft Access OLE data type. If you import a file that contains a field with a PICT data type, you will receive the following error message:

External table isn't in the expected format.

- You can import FoxBase for Macintosh files that contain memo fields only if the associated .DBT file is present. The .DBT file must be located in the same directory as the FoxBase (.DBF) file. The .DBT file will be incorporated into the Microsoft Access table and is not needed by Microsoft Access after the import process is complete.

### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 4, "Importing, Exporting, and Attaching," pages 60-64

Additional reference words: 1.00 1.10 2.00 foxbase import macintosh

KBCategory: kb3rdparty

KBSubcategory: IsmXbase

**INF: How to Import Several dBASE Databases Simultaneously**  
**Article ID: Q88764**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

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

Many new Microsoft Access users want to immediately import data from the systems they are currently using. The Import dialog box in Microsoft Access allows you to import one table at a time, which usually is sufficient. However, some users have numerous tables to import, or they may want to import multiple tables regularly.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

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

The batch table lists the tables that you want to import. The Access Basic procedure reads the table and imports each foreign table listed there. To do this, use the following procedure:

1. Create a table called Batch Import with the following structure:

| Field name       | Type | Length |
|------------------|------|--------|
| Source Directory | Text | 50     |
| Source Database  | Text | 50     |
| Imported Name    | Text | 50     |
| Type of Table    | Text | 50     |

2. Enter information in the new Batch Import table about the tables you want to import. The fields in the Batch Import table should be filled out as follows:

- Source Directory: This is the full path for the location of the foreign database file (for example, C:\DBASE).
- Source Database: This is the name and extension of the dBASE

database you want to import (for example, CUSTOMER.DBF).

- Imported Name: This is the name you want the table to have once it is imported into Microsoft Access (for example, Customers).
- Table Type: This can be either dBASE III or dBASE IV. Specify dBASE III for both dBASE III and dBASE III PLUS databases.

For example, to import a dBASE IV database called EMPLOYEE.DBF from the C:\DBASE4 directory and a dBASE III database called ORDERS.DBF from the D:\DBASE3\DATA directory, you would fill out the fields in the Batch Import table as follows:

| Source Directory | Source Database | Imported Name  | Table Type |
|------------------|-----------------|----------------|------------|
| C:\DBASE4        | EMPLOYEE.DBF    | Employee Table | dBASE IV   |
| D:\DBASE3\DATA   | ORDERS.DBF      | Orders Table   | dBASE III  |

3. Create a new module, and enter the following code:

NOTE: In the following example, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this example.

```
Sub BatchImport ()
 Dim B_DB As Database, B_TBL As Table
 Set B_DB = CurrentDB()
 Set B_TBL = B_DB.OpenTable("Batch Import")

 DoCmd Hourglass True
 B_TBL.MoveFirst
 Do Until B_TBL.EOF
 DoCmd TransferDatabase A_IMPORT, B_TBL![Type of Table],_
 B_TBL![Source Directory], A_TABLE, B_TBL![Source Database],_
 B_TBL![Imported Name], False
 B_TBL.MoveNext
 Loop
 DoCmd Hourglass False
End Sub
```

4. Import the foreign databases as follows:

- a. Open a module.
- b. From the View menu, choose Immediate Window.
- c. Type the following command, and then press ENTER:

```
BatchImport
```

The pointer will become an hourglass and remains so until all of your databases are imported. This process may take several minutes, depending on the sizes of the databases.

dBASE III, dBASE III PLUS, and dBASE IV are manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding the performance or reliability of

these products.

Additional reference words: 1.00 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmXbase

## INF: How to Index an Existing Field with DAO

Article ID: Q112107

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article demonstrates how to create a compound (multiple-field) index in an existing table using data access objects (DAO).

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual, Chapter 3, "Introducing Access Basic."

### MORE INFORMATION

=====

An Index object contains the fields that are being indexed, and usually contains only a single field. The Index object created in the example below will have two fields appended to it, creating a multiple-field index.

The example below demonstrates how to create a multiple-field index with DAO:

1. Open the sample database NWIND.MDB.
2. Create the following new table:

Table: Interviews

-----  
FieldName: CustomerID  
    DataType: Number  
    FieldSize: Long Integer  
FieldName: InterviewerID  
    DataType: Number  
    FieldSize: Long Integer  
FieldName: Appointment  
    DataType: Date/Time

3. Save the table as Interviews without creating a primary key, then close it.
4. Create a new module and enter the following Declaration:

Option Explicit

5. Create the following function in the module:

'-----  
'PURPOSE: Adds a multiple-field index to a table.

```

'ACCEPTS: Nothing.
'RETURNS: Nothing.
'-----
Function AddMultiIndex ()
 Dim DB As Database, TDef As TableDef
 Dim Idx As Index, Fld As Field

 Set DB = DBEngine.Workspaces(0).Databases(0)

 ' Open the table definition.
 Set TDef = DB.TableDefs("Interviews")

 ' Create an index called PrimaryKey for this TableDef
 ' .. and turn on the Primary and Required properties.
 Set Idx = TDef.CreateIndex("PrimaryKey")
 Idx.Primary = True
 Idx.Required = True
 Idx.IgnoreNulls = False

 ' Create an index field with the same name as a table field,
 ' .. then append it to the index.
 Set Fld = Idx.CreateField("CustomerID")
 Idx.fields.Append Fld

 ' Do the second field the same way.
 Set Fld = Idx.CreateField("InterviewerID")
 Fld.Attributes = DB_DESCENDING
 Idx.fields.Append Fld

 ' Append the index to the TableDef.
 TDef.indexes.Append Idx

End Function

```

6. Save the module as DAOaddMultiIndex.
7. From the View menu, choose Immediate Window.
8. Type the following in the Immediate window, then press ENTER:

```
? AddMultiIndex()
```

9. Close the Immediate window and the module.
10. Open the Interviews table in Design view. Note the compound primary key on the CustomerID and InterviewerID fields.

#### REFERENCES

=====

For more information about Index objects, search for "Index" then "Index Object, Indexes Collection" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr



**INF: How to Make a Combo Box Default to First Item in List**  
**Article ID: Q105519**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

When you move to a new record on a form that has a combo box or a list box, the combo box will be blank or the list box will not have any value selected. The combo box or list box may have a table or query defined in its RowSource property that provides the list of items to display in the box. Since the data in the underlying RowSource property will vary with the addition or deletion of records, it is difficult to know what item will appear at the top of the list when the form is used.

This article describes how to force a list box or combo box to default to the first row in the underlying list. The methods outlined in this article work only if the combo box is bound to a field.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

Microsoft Access Version 2.0

-----

In Microsoft Access version 2.0, you can use the ItemData method to cause a list box or combo box to default to any row. To have the first row selected automatically, set the box's DefaultValue property to the following:

```
=[<MyCombo>].ItemData(0)
```

Microsoft Access Version 1.x

-----

The first method below uses a user-defined Access Basic function, and the second method uses the built-in DLookup() function, to display the first value in the list automatically.

Method 1:

The following example demonstrates a sample Access Basic function called GetFirst() that can be used to find the first item in the underlying table or query. The function's result can be used by the DefaultValue property to automatically select the first item in the list.



To create the GetFirst() function, add the following to a new or existing module:

```
Option Explicit
```

```
Function GetFirst (BoundColName As String, RowSource As String)
 Dim DB As Database
 Dim DS As Dynaset

 Set DB = CurrentDB()
 Set DS = DB.CreateDynaset(RowSource)

 On Error Resume Next
 DS.MoveFirst
 If Err = 0 Then GetFirst = DS(BoundColName)

End Function
```

Note that the first argument of the GetFirst() function is the name of the field that is used as the BoundColumn property for the combo box. The second argument is the name of the table or query specified in the RowSource property of the combo box.

The following example demonstrates how to use GetFirst() to automatically select the first employee in the Salesperson combo box on the Orders form in the sample database NWIND.MDB:

1. Open the Orders form in Design view.
2. Select the Salesperson combo box. Display the property sheet by choosing Properties from the View menu.
3. Set the DefaultValue property to the following:

```
=GetFirst("Employee ID", "Employee List")
```

4. View the form in Form view.
5. From the Records menu, choose Data Entry.

Note that the combo box automatically displays "Buchanan, B.L."

Differences Between GetFirst() and DFirst():

The GetFirst() function is similar to the built-in DFirst() aggregate function. However, DFirst() may return unexpected results when used to find the first item in a list.

If the underlying table or query is indexed, the value returned by DFirst() will be the first indexed record. Otherwise, DFirst() will return items in the actual order in which they were entered in the database. Therefore, if the RowSource property of a combo box is a query that sorts the data by a non-indexed field, DFirst() may not return the expected value.

For example, if you change the DefaultValue property of the Salesperson combo box on the Orders form to

```
=DFirst("[Employee ID]", "Employee List")
```

the item returned will be "Davolio, Nancy," which is not the first item in the combo box, but the first indexed item in the Employees table.

Method 2:

This method uses the DLookup() function to look up the first record in the list. The expression will be the field referred to in the BoundColumn property (or the ControlSource property) of the combo box or list box. The domain will be the same table or query that the combo box or list box uses as its RowSource property. The optional criteria will not be used so that the DLookup() function will return the first record.

The following example demonstrates how to use DLookup() to automatically select the first employee in the Salesperson combo box on the Orders form in the sample database NWIND.MDB:

1. Open the Orders form in Design view.
2. Select the Sales Person combo box. View the property sheet by choosing Properties from the View menu.
3. Set the DefaultValue property to the following expression:

```
=DLookup("[Employee ID]","Employee List")
```

4. View the form in Form view.
5. From the Records menu, choose Data Entry.

Note that the combo box automatically displays "Buchanan, B.L."

#### REFERENCES

=====

For more information about the ItemData method, search for "ItemData" then "ItemData Method" using the Microsoft Access version 2.0 Help menu.

Microsoft Access "User's Guide," version 1.0, Chapter 9, "Designing Forms," pages 233-242

Microsoft Access "User's Guide," version 1.1, Chapter 9, "Designing Forms," pages 237-246

Microsoft Access "Language Reference," versions 1.0 and 1.1, pages 131-132, 139-140, and 144-146

Additional reference words: 1.00 1.10 2.00 listbox combobox

KBCategory: kbusage

KBSubcategory: FmsCmbo

**INF: How to Make Report Footer Print at Bottom of Last Page**  
**Article ID: Q119783**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

The report footer is printed at the end of a report, immediately following the last section in the report. Since it immediately follows the last section in the report, the report footer may not be printed at the bottom of the last page. This article describes a method of printing the report footer information at the bottom of the last page by including the information in a page footer and making the information visible only on the last page of the report.

MORE INFORMATION

=====

The following example demonstrates a method that compares the Page and the Pages properties to determine if the current page is the last page of the report, and if so, makes the information in the page footer visible. Note that in order to use the Pages property in Access Basic, the report must contain a text box whose ControlSource property uses the Pages property.

1. Create a new report and place the following label control in the report's page footer section:

Label

-----  
Name: ReportFooterText  
Caption: This is the last page  
Visible: No

2. Create the following text box control in the page footer section:

Text Box

-----  
Name: PageNumber  
ControlSource: =[Page] & " of " & [Pages]

3. Set the page footer's OnPrint property to the following event procedure:

```
Sub PageFooter2_Print (Cancel As Integer, PrintCount As Integer)

 If Me.page = Me.pages Then
 ReportFooterText.visible = True
 Else
 ReportFooterText.visible = False
 End If

End Sub
```

## REFERENCES

=====

For more information about the Page and the Pages properties, search for "Page" then "Page, Pages Properties" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: RptLayout

**INF: How to Make the Immediate Window Available at All Times**  
**Article ID: Q88925**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The Immediate window is helpful for debugging Access Basic code in that you can try out an expression to see if it returns the expected value, or any value at all. The Immediate window can also be helpful for debugging macros, learning how to use expressions, and more. However, the Immediate window is only available when a module window is selected. This article describes how you can make the Immediate window available at all times.

MORE INFORMATION

=====

You can type an expression in the Immediate window and see the results of that expression, without having to write and run an Access Basic procedure. To run an expression in the Immediate window, use the Access Basic "?" command, as demonstrated 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, set the value of a variable and then test for the value of the variable with the following:

```
MyVar = 2
? MyVar
```

The "?" statement returns the number 2 since that is the value you assigned to the variable MyVar.

The Immediate window is helpful for debugging Access Basic code, because you can suspend execution of the procedure and set the values of variables in the procedure to see what kind of result you would get with different values. You can also use the Immediate window to suspend a procedure and check for the value of a variable or expression to see if it is calculating as you expect it to.

In the same way that an Immediate window can be helpful for debugging code, it can also be helpful for other 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 with the DoCmd command to see if the macro action produces the desired results.

However, the Immediate window is only available when a module window is selected. You can make the Immediate window available at all times by following these steps:

1. In the database window, choose Macro.
2. If you already have a macro called AutoKeys, open it in design mode. Otherwise, choose New.
3. If the Macro Name column does not appear in the macro sheet, choose Macro Names from the View menu.
4. If the Condition column does not appear in the macro sheet, choose Conditions from the View menu.
5. Add the following action and its corresponding property:

| Macro name | Condition                              | Action     |
|------------|----------------------------------------|------------|
| -----      |                                        |            |
| ^I         |                                        | SendKeys   |
|            |                                        | DoMenuItem |
|            |                                        | SendKeys   |
|            | MsgBox ("IW already on the screen?")=7 | DoMenuItem |

SendKeys Action Parameters

-----

KeyStrokes: {F11}

Wait: Yes

DoMenuItem Action Parameters

-----

Menu Bar: Database

Menu Name: View

Command: Modules

SendKeys Action Parameters

-----

KeyStrokes: %N

Wait: Yes

DoMenuItem Action Parameters

-----

Menu Bar: Module

Menu Name: View

Command: Immediate Window

6. Save and then close 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.

Note that a small module window 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 the 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.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: How to Minimize, Maximize, and Restore Microsoft Access**  
**Article ID: Q89597**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Use Microsoft Windows API calls in Microsoft Access modules to maximize, minimize, and restore Microsoft Access. Microsoft Access does not provide a function to perform these actions in a module or macro.

MORE INFORMATION

=====

Once the following sample functions are defined in a module they can be used in macros as RunCode actions. Define the functions with these steps:

1. From the File menu, choose New, then Module. In the Declarations section add both of the following declarations:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

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

2. Create the function MaximizeAccess():

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

3. Create the function MinimizeAccess():

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

4. Create the function RestoreAccess():

```
Function RestoreAccess ()
 Dim ActiveWnd%, Restoreit%
 ActiveWnd% = GetActiveWindow()
 Restoreit% = ShowWindow(ActiveWnd%, 1)
End Function
```



5. The following sample macro action will minimize the Microsoft Access window:

| Action  | FunctionName     |
|---------|------------------|
| -----   | -----            |
| RunCode | MinimizeAccess() |

#### REFERENCES

=====

"Microsoft Windows Software Development Kit," Microsoft Press, 1992

"Programming Windows: the Microsoft Guide to Writing Applications for Windows 3," Charles Petzold. Microsoft Press, 1990

"Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference" Volumes 1-6, Microsoft Press, 1992

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: How to Move List Box Selections to a Second List Box**  
**Article ID: Q97518**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to create a pair of list boxes to imitate the multiple selection behavior used in Microsoft Access wizards.

MORE INFORMATION

=====

The example below demonstrates how to create a new form that contains two list boxes and three command buttons. You can add items to or delete items from the list using the command buttons or the properties of the list boxes:

1. Create the following new table:

Table: Table1

-----

Field Name: List (primary key)

    DataType: Text

    Field Size: 15

    Caption: Items that will be provided in list

Field Name: Selected

    DataType: Text

    Field Size: 5

    Caption: Indicates if the item has been selected

2. View the table in Datasheet view. Add five records to the table. For each record, type any text in the List field (for example a list of colors or cities), and type "YES" (without quotation marks) in the Selected field.

3. Create the following two new queries based on Table1:

Query: Select Yes

-----

Field Name: List

    Show: Yes

    Criteria: [selected] = "YES"

Query: Select No

-----

Field Name: List

    Show: Yes

    Criteria: [selected] = "NO"

4. Enter the following three sample functions in a new or existing module:

```

'=====
'The following function opens the table and changes the
'selected value from YES to NO, and then executes the
'query for the two list boxes so that they will display
'the updated values.
'=====
Function ADD ()
 Dim MyDB As Database
 Dim MyTable As Table
 Dim y As Control

 Set MyDB = CurrentDB()
 Set MyTable = MyDB.OpenTable("Table1")
 Set y = forms!SelectList.field0

 MyTable.index = "PrimaryKey"
 MyTable.Seek "=", y

 MyTable.Edit
 MyTable.[selected] = "no"
 MyTable.Update

 MyTable.Close

 DoCmd Requery "field0"
 DoCmd Requery "field1"
End Function

'=====
'The following function opens the table and changes the
'selected value from NO to YES, and then executes the
'query for the two list boxes so that they will display
'the updated values.
'=====
Function Del ()
 Dim MyDB As Database
 Dim MyTable As Table
 Dim y As Control

 Set MyDB = CurrentDB()
 Set MyTable = MyDB.OpenTable("Table1")
 Set y = forms!SelectList.field1

 MyTable.index = "primarykey"
 MyTable.Seek "=", y
 MyTable.Edit
 MyTable.[selected] = "yes"
 MyTable.Update
 MyTable.Close

 DoCmd Requery "field0"
 DoCmd Requery "field1"
End Function

'=====
'The following function sets all values in the Selected

```

```
'field to YES, and then executes the query for the two
'list boxes so that they will display the updated
'values.
```

```
'=====
```

```
Function Clear ()
 Dim MyDB As Database
 Dim MyTable As Table

 Set MyDB = CurrentDB()
 Set MyTable = MyDB.OpenTable("Table1")

 On Error GoTo erhandle
 MyTable.MoveFirst

 Do Until MyTable.EOF
 MyTable.Edit
 MyTable.[selected] = "yes"
 MyTable.Update
 MyTable.MoveNext
 Loop

erhandle:
 Resume Next

 DoCmd Requery "field0"
 DoCmd Requery "field1"
End Function
```

5. Create the following new, unbound form, and then save it as SelectList:

```
Object: list box

ControlName: field0
ControlSource:
RowSourceType: Table/Query
RowSource: Select Yes
OnDblClick: =ADD()

Object: list box

ControlName: field1
ControlSource:
RowSourceType: Table/Query
RowSource: Select No
OnDblClick: =Del()

Object: command button

ControlName: Button One
Caption: Clear
OnPush: =Clear()

Object: command button

ControlName: Button Two
Caption: Add item
```

OnPush: =Add()

Object: command button

-----

ControlName: Button Three

Caption: Delete item

OnPush: =Del()

The first list box, Field0, displays the items to be selected. The second list box, Field1, displays the items that are selected.

#### REFERENCES

=====

For more information on how to select more than one item from the same list box, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q121356

TITLE : INF: How to Create a Multiple-Selection List Box

Additional reference words: 1.00 1.10 2.00 listbox combo multiple fill  
function wizard

KBCategory: kbusage

KBSubcategory: FmsCmbo

## **INF: How to Open UTILITY.MDA**

**Article ID: Q93929**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

If you try to open the UTILITY.MDA file, you receive the following error message:

The database "UTILITY.MDA" is already open as a library database.

To open the UTILITY.MDA file, you must create a copy of UTILITY.MDA, rename the copy, and then open it in Microsoft Access.

### MORE INFORMATION

=====

Use the following steps to open the UTILITY.MDA file.

NOTE: UTILITY.MDA is a collection of tables, forms, and modules used by Microsoft Access. You should always have a backup copy of UTILITY.MDA; this is a vital database to Microsoft Access and any changes in it may result in unexpected behavior.

1. Copy UTILITY.MDA to UTILITY.SAV.
2. In Microsoft Access, open UTILITY.SAV by choosing Open from the File menu.
3. Choose OK when the following error message is displayed:

Tried to load module with duplicate procedure definition:  
<procedure name>

After you make any changes to the copy of UTILITY.MDA, you can rename the original UTILITY.MDA file and substitute the edited copy of the UTILITY.MDA file for the one that Microsoft Access uses when the application is opened.

Additional reference words: 1.00 1.10 2.00 utility

KBCategory: kbusage kberrmsg

KBSubcategory: GnlOthr

**INF: How to Optimize Microsoft Access Version 2.0 Performance**  
**Article ID: Q112117**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article contains tips for optimizing the performance of Microsoft Access version 2.0. Much of this information is taken from the Microsoft Access 2.0 Help file.

Note that the optimal setting for each item may vary with the type of computer on which you run Microsoft Access. It is usually best to change only one setting at a time and then monitor database performance for improvement.

MORE INFORMATION

=====

To optimize the general performance of Microsoft Access version 2.0:

- Use the Add-in Manager to uninstall library databases that contain Microsoft Access Wizards, builders, and other add-ins you do not want. This reduces Microsoft Access memory consumption and load time.
- Make more memory available by closing applications and terminate-and-stay-resident (TSR) programs that you are not using. Usually, these applications are loaded from the AUTOEXEC.BAT and CONFIG.SYS files.
- Make sure your Microsoft Windows virtual memory (swap file) setting is large enough, and of type "permanent" rather than "temporary." In general, the virtual memory setting plus available RAM should be no less than 25 MB. It should be more if you will be running several memory-intensive applications simultaneously.

To check or change the virtual memory setting, start Microsoft Windows Control Panel. Double-click the 386 Enhanced icon, then choose Virtual Memory. To change the setting, choose Change. Make sure to select Permanent in the Type box. Choose OK to save your changes. Please see the Microsoft Windows "User's Guide" for more detailed information on virtual memory settings.

- Periodically run a disk defragmentation utility such as MS-DOS version 6.0 Defrag to keep files 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 files may increase since it may have to go to several physical locations on the disk to retrieve the entire file.
- Use 32-bit disk access, and 32-bit file access in Windows 3.11 and later. In Windows Control Panel, double-click the 386 Enhanced icon,

choose the Virtual Memory button, then choose the Change button. Make sure the Use 32-Bit Disk Access check box is selected. Also select the Use 32-Bit File Access option, if it is available in your version of Windows.

- Increase the RAM on your computer. Microsoft Access requires a minimum of 6 MB, but additional RAM improves performance.
- Make the WinCacheSize parameter for SMARTDrive (or similar settings for other disk caches) in your CONFIG.SYS file no larger than necessary for effective caching. For computers with limited RAM, try completely disabling software caching such as SMARTDrive.
- Do not use any of your RAM for a RAM disk.
- Set the Buffers parameter in your CONFIG.SYS file to at least 40.
- When you are opening databases that are not in a multiuser environment, select the Exclusive check box in the Open Database dialog box.
- When you are using databases that other users do not need to share, install Microsoft Access and all your databases on your local hard disk rather than on a network server.
- Create only as many indexes as necessary. Although indexes can speed access to data, it is possible to "over index" a table so that it is slow adding, deleting, and updating records.
- Create indexes for joined fields.
- In a multiple-field index, use only as many fields in the index as necessary.
- Use Rushmore query optimization in your queries whenever possible. For detailed information on how to do this, search for "Rushmore technology" then "Optimizing Queries with Rushmore Technology" using the Microsoft Access Help menu.
- If you have a wallpaper (full-screen background) bitmap on your Windows desktop, replace it with a solid color or pattern bitmap, or no bitmap at all. For a standard VGA display, this can free about 256K of RAM. For a 1024 x 768 pixel display with 256 colors, this can free about 750K of RAM. (Your actual RAM savings depends on your video display.)

#### REFERENCES

=====

For more information about optimizing Microsoft Access performance, search for "optimizing performance" using the Microsoft Access Help menu.

Additional reference words: 2.00 General Speeding Improving

KBCategory: kbusage

KBSubcategory: GnlOthr



**INF: How to Optimize Microsoft Access Version 2.0 Queries**  
**Article ID: Q112112**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article contains tips you can use to help you optimize your Microsoft Access version 2.0 query performance.

MORE INFORMATION

=====

This article assumes that all the tables in your application are local. If your application uses attached tables the information in this article still applies, but there are additional performance issues that pertain to attached tables that can impact query performance. For more information about improving performance using remotely attached tables, query on the following words here in the Microsoft Knowledge Base:

ODBC and Optimizing and Tables

The Query Optimizer

-----

The Microsoft Jet database engine contains several components, but the most important to queries (and the most complex) is the Optimizer. The Optimizer is "cost-based," meaning that it assigns a time cost to each query task and then chooses the least expensive list of tasks to perform that generates the desired result set. The longer a task takes to perform, the more costly or expensive it is considered to be.

To decide which query strategy to use, the Optimizer uses statistics. These statistics are based on the number of records in a table, the number of data pages in a table, the location of the table, whether or not indexes are used, and so on. Based on these statistics, the Optimizer chooses the best internal query strategy for dealing with a particular query. Statistics calculations are done whenever a query is compiled. A query is compiled whenever you save any changes made to it or to any of the tables underlying the query. If you have added a significant number of rows to your table since a query was last compiled, you should open and save the query to cause it to recompile and thus update the statistics.

You can improve the statistics and thereby reduce overhead search time and enable the Optimizer to make your queries run faster by using the tips described below.

Tips to Improve Query Performance

-----

- Compact your database. To do this, choose Compact Database from the File menu when no database is open. Compaction can speed up queries because

it writes all the data in a table into contiguous pages on the hard disk. Scanning sequential pages is much faster than scanning fragmented pages.

- Avoid expressions in query output. Expressions in query output can cause query optimization problems if the query is used as the input to another query. In the example below, query Q1 is used as the input for query Q2:

```
Q1: SELECT IIF([MyColumn]="H","Hello","Goodbye") AS X
 FROM MyTable;
```

```
Q2: SELECT *
 FROM Q1 WHERE X="HELLO";
```

Since the IIF() expression in Q1 cannot be optimized, Q2 also cannot be optimized. If an expression gets buried deeply enough in a query tree, you can forget that it is there. As a result, your entire string of queries cannot be optimized.

A better way to write the example query above is:

```
Q1: SELECT *
 FROM MyTable WHERE MyColumn = "H";
```

If expressions are necessary in the output, try to place them in a control on a form or report.

- Place GROUP BY clauses in the same table as aggregates. This is an issue when you are joining two tables. For example, if you join two tables on the Customer Name field and then run a query that performs a GROUP BY operation on the Customer Name field, make sure that both the GROUP BY field (Customer Name) and the field that is in the aggregate (Sum, Count, and so on) come from the same table.
- When you are creating a "totals" query, use the GROUP BY clause on as few fields as possible. The more fields in the GROUP BY clause, the longer the query takes to execute.
- If possible, place a GROUP BY clause on a table and then join it to another table, rather than joining the two tables and doing the GROUP BY in the same query as the join. For example, instead of this query:

```
Q1: SELECT Orders.[Company ID], Count(Orders.[Order ID]) AS
 [CountOfOrder ID]
 FROM Customers INNER JOIN Orders ON Customers.[Customer ID] =
 Orders.[Customer ID]
 GROUP BY Orders.[Company Name];
```

Break the query into two separate queries, such as:

```
Q1: SELECT Customers.[Company ID]
 FROM Customers
 GROUP BY Customers.[Company ID];
```

```
Q2: SELECT Orders.[Customer ID], Count(Orders.[Order ID]) AS
 [CountOfOrder ID]
 FROM Q1 INNER JOIN Orders ON Q1.[Customer ID] =
```

```
Orders.[Customer ID]
GROUP BY Orders.[Customer ID];
```

- When you are joining tables, try to index the fields on both sides of a join. This can speed query execution by allowing the query optimizer to use more sophisticated internal join strategy.
- Index fields as much as possible. If a database is not updated frequently, then an index should be placed on all fields that are used in a join or in a restriction. With the use of Rushmore query optimization technology in Microsoft Access version 2.0, the Microsoft Jet database engine is able to take advantage of multiple indexes on a single table, which makes indexing multiple fields advantageous.
- Try to construct your queries so that Rushmore technology can be used to help optimize them. Rushmore is a data-access technology that permits sets of records to be queried very efficiently. With Rushmore, when you use certain types of expressions in query criteria, your query will run much faster.

Rushmore does not automatically speed up all of your queries. You must construct your queries in a certain way for Rushmore to be able to improve them. For more information about how to optimize queries with Rushmore technology, search for "Rushmore technology" then "Optimizing Queries with Rushmore Technology" using the Microsoft Access Help menu.

- Use COUNT(\*) rather than COUNT([Column Name]) to determine the number of records in a table. This is because there are special optimizations in the Microsoft Jet database engine that allow COUNT(\*) to be executed much faster than COUNT([Column Name]).

#### REFERENCES

=====

For more information about optimizing Microsoft Access performance in general, search for "optimizing performance" then "Optimizing Microsoft Access Performance" using the Microsoft Access Help menu.

Additional reference words: 2.00 general speeding improving  
KBCategory: kbusage  
KBSubcategory: Gn1Othr

## **INF: How to Optimize MS Access Using ODBC Data Sources**

Article ID: Q113551

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article describes several tips for improving performance when you are accessing data from an ODBC data source.

### MORE INFORMATION

=====

Try these tips to improve performance with ODBC data sources:

- Restrict the amount of data that you request from the server. Do not ask for more data than you need. Use queries to select only the fields and rows that you need.
- Use only the functionality you need. Snapshots are less powerful than dynasets and are not updateable, but may be faster, particularly for small recordsets without Memo or OLE Object fields.
- Use attached tables to access server data. Do not use "direct" server access (that is, opening remote databases and executing queries against them). Instead, attach tables and use the attachments, or use pass-through queries.
- Use list and combo boxes wisely. On a form, each list box, combo box, subform, and control containing a total requires a separate query. Against local data, performance may be adequate. Against remote data, however, long delays may occur when you open a form since each query must be sent to the server and a response returned before the form can open.
- Avoid large combo boxes. Including a combo box with hundreds, or even thousands, of choices based on a local table may yield acceptable response time, especially if you define an appropriate index on the local table. Against a remote table, however, such a combo box will yield sluggish performance as it drains server and network resources as it fetches data to fill the list. It is best to limit the number of rows returned to the combo box when you are working with remote data. You can also break up the data into smaller combo boxes (bearing the tip above in mind).
- Use the Find command only on smaller recordsets. The Microsoft Jet database engine optimizes the Find command to work well against local recordsets of almost any size, and against remote recordsets of reasonable size. However, when you are working with large remote recordsets (thousands of records or more), you should instead use a filter or query and be careful to use restrictions that your server can process.

- Make sure that queries are sent to the server for processing. The most important factor in query performance against remote data is ensuring that your server executes as much of the query as possible. The Microsoft Jet database engine attempts to send the entire query to your server, but evaluates locally any query clauses and expressions that are not generally supported by servers or by your particular server. Functionality not supported by servers in general includes:
  - Operations that cannot be expressed in a single SQL statement. This can occur when you use a query as an input to another query, or when your query's FROM clause contains a Totals query or DISTINCT query. Often, you can rearrange your queries to calculate totals after all other operations.
  - Operations that are Microsoft Jet database engine-specific extensions to SQL, such as crosstab queries, TOP queries, and reports with multiple levels of grouping and totals. Note that simple crosstab queries can be sent to servers.
  - Expressions that contain Microsoft Access-specific operators or functions. The Microsoft Access financial functions and statistical aggregates have no server equivalents.
  - User-defined Access Basic functions that take remote columns as arguments. These functions do not exist on the server, but must process remote column data. However, if a user-defined function returns a single value and does not reference a remote column, the function is evaluated locally and its value is sent to the server for processing.
  - Mixing text and numeric data types in operators or UNION query outputs. Most servers lack the data-type leniency in Microsoft Access. Because of this, use explicit conversion functions where appropriate.
  - Heterogeneous joins between local tables and remote tables, or between remote tables in different ODBC data sources. Joins between small local tables and large remote tables, where the join column is indexed, may result in a remote index join. In a remote index join, one query for each row in the local table is sent to the server and only the joining rows are returned.
  - Non-remotable expressions, or expressions that cannot be sent remotely, because they cannot be evaluated by your server. Non-remoteable output expressions (those in the SELECT clause) do not force local evaluation of your query unless they occur in a Totals query, a DISTINCT query, or a UNION query. Non-remoteable expressions in other clauses (WHERE, ORDER BY, GROUP BY, HAVING, and so on) force at least part of your query to be evaluated locally.
- Servers differ in some areas of supported functionality. When you attach a remote table, the Microsoft Jet database engine queries the ODBC driver for its capabilities. If the required functionality is supported by the driver and the server, the Microsoft Jet database engine sends the operation to the server for processing. If not, the Microsoft Jet database engine performs the operation locally. Areas of differing

support include (but are not limited to) the following:

- Outer joins. Note that the Microsoft Jet database engine does not send multiple outer joins to a server, although many inner joins may accompany a single outer join.
- Numeric, string, and date/time functions (such as Log(), Mid\$(), DatePart(), and so on).
- Conversion functions (such as CInt(), CStr(), CVDDate(), and so on).

Additional reference words: 2.00 Attaching ODBC

KBCategory: kbinterop

KBSubcategory: IntpOthr

**INF: How to Pass an Array As an Argument to a Procedure**  
**Article ID: Q116140**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY  
=====

In order to pass an array as an argument to an Access Basic procedure, you must include the array name followed by opening and closing parentheses. If you do not include the parentheses, the error message "Parameter type mismatch" will result.

MORE INFORMATION  
=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

The following example demonstrates the correct syntax for procedure calls with an array:

```
Result = YourFunctionName(YourArrayName())

YourSubName YourArrayName()
```

The procedure you are calling must fully declare the argument and include the opening and closing parentheses.

The following example demonstrates the correct syntax for procedure declarations with an array:

```
Function YourFunctionName (InArray() As String) As Integer

Sub YourSubName (InArray() As String)
```

The following example demonstrates a sample user-defined function that loads an array with string values:

1. Create a new module and enter the following code:

NOTE: In the following sample code, an underscore ( \_ ) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
'-----
'GLOBAL DECLARATIONS SECTION
'-----
Option Explicit
```

```
'-----
'The function LoadArray() loads an array called MyArray
'with string values. After loading the array, the function
'calls a procedure that displays all array elements to the
'Immediate window.
'-----
```

```
Function LoadArray()
 Dim i as Integer
 ReDim MyArray(10) As String
 For i = 1 to 10
 MyArray(i) = "Test Value: " & i
 Next i
 DisplayArray MyArray()
End Function
```

```
'-----
'LoadArray() Sub Procedure
'-----
```

```
Sub DisplayArray (InArray() As String)
 Dim i as Integer
 For i = 1 to UBound(InArray)
 Debug.Print InArray(i)
 Next i
End Sub
```

2. From the View menu, choose Immediate Window.

3. In the Immediate window, type the following line and then press ENTER:

```
? LoadArray()
```

#### REFERENCES =====

Microsoft Access "Introduction to Programming," version 1.0, Chapters 2 and 5

Microsoft Access "Building Applications," version 2.0, Chapters 5 and 6

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg kberrmsg

KBSubcategory: MdlOthr



**INF: How to Play .WAV Sounds on Events in Microsoft Access**  
**Article ID: Q95647**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Microsoft Access does not have a built-in function to play sound files on events, such as when a form is opened or closed. However, you can use the Microsoft Windows 3.1 application program interface (API) through Access Basic code to create a user-defined function to play sound files.

This article assumes you are familiar with Access Basic and Windows APIs. In this article, the use of error trapping has been omitted to keep the information as clear and concise as possible.

MORE INFORMATION

=====

Follow these steps to create a user-defined function to play sound files:

1. Create a new module with a single function named PlaySound().

Note: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

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

Option Explicit

Declare Function sndplaysound% Lib "mmsystem" (ByVal filename$, _
 ByVal snd_async%)

'=====
' The following function PlaySound calls the Windows API function
'=====

Function PlaySound (msound)
 Dim XX%
 XX% = sndplaysound(msound, 1) ' play mmwave sound
 If XX% = 0 Then MsgBox "The Sound Did Not Play!"
End Function
```

2. Save the module as Play It Sam.

The following steps use a command button to trigger the sound, but you can use other events to trigger the sound.

3. Add a command button to a form you want to play the sound from.

Object: Command Button

-----  
ControlName: Push\_Button\_1  
Caption: Play Chimes  
OnPush: =PlaySound("C:\WINDOWS\CHIMES.WAV")

(Where C:\WINDOWS\ is the location of the sound file and CHIMES.WAV is the sound file.)

When the form is open in Form view, the sound is ready to play by clicking the command button. You can assign this function to the form's OnOpen property if you want chimes to play when you open a form.

NOTE: This example does not have error trapping. Unexpected results may occur if the sound file is not in the location specified or does not exist.

If the sound (.WAV file) is in the table as an OLE field, the sound can be added to the form out of the field list. You can then use a macro that does a GoToControl [olefield], DoMenuItem form-edit-object-<verb>. You can run this from a button on the form. It will go to the OLE field and edit the object (the default edit is play for a sound).

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," Chapters 1-5

"Microsoft Windows Software Development Kit," Microsoft Press, 1992

"Programming Windows: the Microsoft Guide to Writing Applications for Windows 3", Charles Petzold. Microsoft Press, 1990

"Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference," Volumes 1 - 6, Microsoft Press, 1992

Additional reference words: 1.00 1.10 2.00 sound

KBCategory: kbprg

KBSubcategory: PgmApi

## INF: How to Position Microsoft Access on the Screen

Article ID: Q95934

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

To position Microsoft Access on the screen, you need to call the SetWindowPos() Windows API function. The article gives a brief demonstration of how to position Microsoft Access on the screen.

### MORE INFORMATION

=====

To position Microsoft Access on the screen, you will need to use the SetWindowPos() application programming interface (API) function included in the USER.EXE dynamic link library (DLL) included with Windows. Implement this function as follows:

1. Open or create a new module in Microsoft Access.
2. Within the Declarations section, add the following declares, constants, and comments.

NOTE: In the following sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'=====
' Global Declarations
'=====
Option Compare Database
Option Explicit

Declare Function FindWindow% Lib "user" (ByVal lpClassName As Any, _
 ByVal lpCaption As Any)
Declare Sub SetWindowPos Lib "User" (ByVal hWnd%, _
 ByVal hWndInsertAfter%, ByVal X%, ByVal Y%, _
 ByVal cx%, ByVal cy%, ByVal wFlags%)

'Inserts the window to precede the position window in the Z-Order.
'This parameter must be a window handle or one of the following
'constants.

Global Const HWND_TOP = 0

'values for wFlags%

Global Const SWP_NOZORDER = &H4 'Ignores the hWndInsertAfter
```

3. Include the following function:

```
Function SizeAccess ()
 Dim cX%, cY%, cHeight%, cWidth%, h%
 'get handle to Microsoft Access
 h% = FindWindow("Omain", 0&)

 cX% = 0
 cY% = 0
 cWidth% = 640
 cHeight% = 480

 'position Microsoft Access
 SetWindowPos h%, HWND_TOP, cX%, cY%, cWidth%, cHeight%,_
 SWP_NOZORDER

End Function
```

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbprg  
KBSubcategory: PgmApi

## **INF: How to Prevent Users from Creating New Databases**

**Article ID: Q123483**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article describes how to use the security constant DB\_SEC\_DBCREATE to prevent database users from creating new databases. This constant can be set and removed only with Access Basic code.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

When you negate the security constant DB\_SEC\_DBCREATE for a user, you remove that user's permission to create new databases. The following example demonstrates how to use the DB\_SEC\_DBCREATE constant to remove the permission to create new databases from all users in the Users group:

1. Quit Microsoft Access if it is running. Copy the SYSTEM.MDA file (usually in the ACCESS directory) to a file with another name. For example, copy the file to a file named SYSTEM.BAK.
2. Start Microsoft Access and open the sample database NWIND.MDB.
3. Create a new module and enter the following line in the Declarations section of the module:

Option Explicit

4. Enter the following code in the module:

```
Function Remove_DBCreate()
Dim d As Database, c As Container
SystemDB = "c:\access\system.mda" ' Use path for your system.
Set d = DBEngine(0).OpenDatabase(SystemDB)
Set c = d.Containers!Databases
c.Username = "Users"
c.Permissions = c.Permissions And Not DB_SEC_DBCREATE
End Function
```

5. From the View menu, choose Immediate Window.
6. In the Immediate window, type the following line and then press ENTER:

?Remove\_DBCreate()

7. Quit Microsoft Access and then restart it. Log in as a member of the Users group.
8. From the File menu, choose New Database. In the File Name box, type "c:\db1.mdb" (without quotations marks) and then choose OK. You receive the following message:

No permissions on 'C:\DB1.MDB'

Note that if a member of the Users group has permission to create new databases by also belonging to another group (such as the Admins group), that user will still be able to create new databases. Microsoft Access uses the most liberal permissions of the groups a user belongs to.

#### REFERENCES

=====

For more information about permissions, search for "Permissions" then "Permissions Property (Data Access)" using the Microsoft Access Help menu.

Additional reference words: 2.00 rights

KBCategory:

KBSubcategory: Gn1Othr

**INF: How to Print a Blank Line Every Nth Line in a Report**  
**Article ID: Q120842**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to add blank lines between the printed lines on a report. You can use this method to add a blank line after a set number of lines. For example, you could use this method to add a blank line after every five lines of data in your report.

MORE INFORMATION

=====

The following example demonstrates how to add a blank line after every five lines in a report:

1. Open the sample database NWIND.MDB and create a new report based on the Employees table. Choose the Report Wizards button and then follow these steps:
  - a. In the "Which Wizard do you want?" screen, select Tabular and then choose the OK button.
  - b. In the Available Fields box, select Employee ID and then choose the ">" button. Repeat for Last Name, First Name, and Birth Date, and then choose the Next button.
  - c. In the Available Fields box, select Birth Date and then choose the ">" button. Choose the Next button.
  - d. In the "What style do you want for your report?" screen, choose the Next button.
  - e. In the "What title do you want for your report?" screen, type "Employee Birthdays" (without quotation marks), and then choose the Finish button.
2. View the new report in Design view.
3. From the View menu, choose Code.
4. Enter the following code in the module:

```
Option Compare Database
Dim cLines As Integer
Const cMaxLine=5
```

This code declares the cLines variable as an integer, and the cMaxLine constant as five. You can set the cMaxLine constant to insert a blank line after as many lines as you want. For example,

to add a blank line after every eight lines in the report, set cMaxLine=8.

5. In the Object box on the toolbar, select Report. In the Procedure box on the toolbar, select Open. Enter the following code in the module:

```
Sub Report_Open (Cancel As Integer)
 cLines = 0
End Sub
```

This code initializes the cLines variable to zero.

6. In the Object box, select Detail1. In the Procedure box, select Format. Enter the following code in the module:

```
Sub Detail1_Format (Cancel As Integer, FormatCount As Integer)
 If cLines Mod (cMaxLine+1) = 0 Then
 Me.NextRecord = False
 Me.PrintSection = False
 End If
 cLines = cLines + 1
End Sub
```

This code adds a blank line by setting the NextRecord and PrintSection properties.

7. Close the module and then preview the report. Note that there is a blank line in the report after every five lines.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, pages 677-680

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: RptLayou



**INF: How to Print a Conditional Page Header in a Report**  
**Article ID: Q105516**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to print a conditional page header using an example that forces the page header, which contains the column header labels, to print on consecutive pages of the report if a data group spans more than one page.

The technique described in this article works for Microsoft Access versions 1.x and 2.0. However, an alternative method for version 2.0 is described in the Solutions sample application. See the "References" section of this article for a complete reference to this method.

MORE INFORMATION

=====

Reports print group header information once per group, even if the group of data spans more than one page. In reports, a common layout is to place the grouping value in the group header along with the column labels. However, since this data is not repeated when a group spans more than one page, the column header labels are not repeated on the subsequent page(s).

To create conditional headers, you must create a new page before each group of data and set a flag from the group header to determine when the page header should print. Then, from the group footer, set the flag back so that the page header does not print at the beginning of the next group.

The following steps illustrate this procedure in the sample database NWIND.MDB:

1. Create a new form called FlagForm. This form is used to store a global variable.
2. Create one unbound control and set the following property:

    Name: Flag

NOTE: In Microsoft Access version 1.x, the Name property is called the ControlName property.

3. Create the following new macro, and name it Test:

| Macro Name  | Condition                 | Action      |
|-------------|---------------------------|-------------|
| SetFlag     |                           | SetValue    |
| ShrinkPageH | Forms![FlagForm]![Flag]=1 | CancelEvent |

FlagOn  
FlagOff

SetValue  
SetValue

Test actions

-----  
SetFlag  
  Item: Forms![FlagForm]![Flag]  
  Expression: 1  
CancelEvent: no arguments  
FlagOn  
  Item: Forms![FlagForm]![Flag]  
  Expression: 0  
FlagOff  
  Item: Forms![FlagForm]![Flag]  
  Expression: 1

4. Create a new blank report called PrintHeaders based on the Products table:
  - a. Set the following report header properties:

  OnFormat: Test.SetFlag  
  Height: 0
  - b. Set the page header property:

  OnFormat: Test.ShrinkPageH
  - c. Create a label in the page header section with the following property:

  Caption: Continued from Previous Page...
  - d. Create a grouping for the Category ID with following properties:

  Group Header: Yes  
  Group Footer: Yes
  - e. Set the following group header properties:

  OnFormat: Test.FlagOn  
  ForceNewPage: Before Section
  - f. Create a control in the group header section bound to the Category Id field.
  - g. Create a text box control bound to the Product Name field in the detail section.
  - h. Set the height property for the detail section to 2 inches.  
(This is to create groups of data that span more than one page.)
  - i. Set the group footer property to:

  OnFormat: Test.FlagOff
5. View the FlagForm form in Datasheet view.

6. Preview the PrintHeaders report.

7. Scroll through the report. Note the page headers for the groups.

#### REFERENCES

=====

You can find information, instructions, and examples in the Solutions sample application (SOLUTION.MDB) included with Microsoft Access version 2.0. For more information about printing a conditional page header, open the SOLUTION.MDB database usually located in the ACCESS\SAMPAPPS directory. Select "Control what you print on reports" in the Select A Category Of Examples box, then "Repeat a group name at the top of a column or page" in the Select An Example box.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptProp

## **INF: How to Print a Constant Number of Lines Per Group**

**Article ID: Q119073**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to create a constant number of lines per data group on a report. You can use this method to consistently print 15 lines per data group, regardless of the actual number of items within the group. Note that if the group contains more than 15 items, the function used by this method will not apply.

### MORE INFORMATION

=====

The steps below demonstrate how to create a report in the sample database NWIND.MDB with a constant number of lines per datagroup. You can use this method in a main report or subreport:

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new report based on the Products table.
3. Group the data by the Category ID field. Set the GroupHeader property to Yes.
4. Place a text box control with the following properties in the Category ID header:  
  
    Name: TotGrp  
    ControlSource: =Count(\*) 'Returns the number of records per group.
5. Set the Category ID header section's ForceNewPage property to Before Section.
6. Place a text box control bound to the Product Name field with the following properties in the report's detail section:  
  
    Name: [Product Name]  
    ControlSource: [Product Name]
7. Create a line control in the detail section that extends horizontally across the section. Set the line control's Name property to LineControl.
8. Create a new module with the following code:

```
Declarations Section of Module
Option Compare Database 'Use database order for string comparisons.
Option Explicit
Global TotCount As Integer
```

```
'Call the SetCount() function from the group header section's
'OnPrint property using the syntax: =SetCount(Report)
```

```
Function SetCount (R As Report)
 TotCount = 0
 R![Product Name].Visible = True
End Function
```

```
'Call the PrintLines() function from the detail section's OnPrint
'property using the syntax: =PrintLines(Report,[TotGrp])
```

```
Function PrintLines (R As Report, TotGrp)
 TotCount = TotCount + 1
 If TotCount = TotGrp Then
 R.NextRecord = False
 ElseIf TotCount > TotGrp And TotCount < 15 Then
 R.NextRecord = False
 R![Product Name].Visible = False
 End If
End Function
```

This method repeats the last detail line for a data group until the total number of constant lines has been created. In this example, the last detail line is repeated until 15 detail lines have been printed for each group. The Product Name control is hidden in order to not repeat the last value, which creates a constant grid of 15 lines.

Additional reference words: 1.00 1.10 2.00 preprinted pre-printed fixed  
KBCategory: kbusage  
KBSubcategory: RptsLyt

**INF: How to Print a Single Record from a Form in a Report**  
**Article ID: Q98801**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to select a single record on a form and print it in a report.

MORE INFORMATION

=====

The following procedure explains how to print only the record you select from the Customers form in the sample database NWIND.MDB:

1. Create a report called PrintRecordReport based on the Customers table.
2. Create the following macro:

Macro: PrintRecordMacro

-----  
OpenReport

ReportName: PrintRecordReport

View: Print Preview

Where Condition: [Customer ID]=Forms![Customers]![Customer ID]

3. Add a command button to the Customers form and set the following button properties:

Command Button

-----  
ControlName: PrintRecord

Caption: Print Record

OnPush: PrintRecordMacro

When you choose the command button on the form, the report opens and runs the underlying query. The query references the form for the Customer ID and synchronizes that Customer ID to the Customer ID on the report.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsOthr

**INF: How to Print Duplicate Mailing Labels for Each Record**  
**Article ID: Q100146**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to print multiple labels for each customer record. The example that follows uses the Customers table in the sample NWIND.MDB database.

MORE INFORMATION

=====

The following example generates 10 labels for each customer record:

1. In the sample database NWIND.MDB, create a new table called Label Count with one number field. Use the default properties and do not create a primary key. Enter a 1 in the first record, 2 for the second record, and so on.

Table: Label Count  
-----  
FieldName: Count  
    DataType: Number

2. Create a new query based on the Label Count and Customers tables:

Query: LabelQuery  
-----  
Type: Select Query  
Tables: Label Count and Customers  
Join: None

3. From the View menu, choose Totals so that all fields are set to Group By, as follows:

Field: Company Name  
    Table: Customers  
    Total: Group By  
Field: Contact Name  
    Table: Customers  
    Total: Group By  
Field: Address  
    Table: Customers  
    Total: Group By  
Field: City  
    Table: Customers  
    Total: Group By  
Field: Region  
    Table: Customers  
    Total: Group By

Field: Postal Code  
Table: Customers  
Total: Group By  
Field: Count  
Table: Label Count  
Total: Group By  
Criteria: <= [Enter Number of Labels to Print for Each  
Customer]

NOTE: The Count field in the Label Count table can have an optional parameter value specified in the Criteria field to enter the number of labels to print for each record in the Customers table. If this query is used without a parameter value, 10 identical labels are printed for each record in the Customers table.

4. Using the Microsoft Access ReportWizard, create a new mailing label report based on the LabelQuery query. If you specify a parameter value, you are prompted to enter the number of labels to print for each customer record.

Additional reference words: 1.00 1.10 2.00 multiple many labels  
KBCategory: kbusage  
KBSubcategory: RptLabel



## **INF: How to Print Group Footer at Specific Location**

**Article ID: Q119655**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to print a report's group footer information at a specific place on the page. This can be useful when you are printing reports to preprinted forms. While you cannot do this using the group footer's property settings, you can use either of the following two methods described in this article:

- Place the information in the form's page footer, which is always printed at a fixed location.
- Use the MoveLayout, PrintSection, and NextRecord report properties to control where the group footer is printed.

NOTE: This method can also be used to print the Report Footer at a specific location.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

#### Method 1: Using the Page Footer

-----

The page footer is always printed at a fixed location on the report's page, just above the bottom margin. For example, if your form has a bottom margin of 1 inch and the height of the page footer is 1 inch, the page footer begins 2 inches from the bottom of the page, or 9 inches from the top of the page (for an 11 inch long page). Placing the group footer controls in the page footer causes them to be printed at the page footer's fixed location on the page.

NOTE: You cannot use aggregate functions in a control in a page footer. If you need to use aggregate controls in your footer, you may want to use the next method in this article, or refer to the "Using Macros to Print Reports or Transfer Data" chapter of the Microsoft Access "User's Guide" for an example of calculating totals in a page footer.

#### Method 2: Using Report Properties

-----

This method involves using the MoveLayout, PrintSection, and NextRecord

report properties to move the group footer to a specific printing location on the page. With this method, you are not actually specifying the group footer's coordinates on the page, but moving the group footer to various existing printing locations on the report page.

To ensure that the group footer is printed where you want it to be, you must consider the height of each section of the report. Since you cannot specify the location of the group footer section (or any other section) with a property setting, you must use the MoveLayout property to move the section to the next available printing location. For example, if the group footer section's height is 0.5 inches and the Top property (which specifies how far from the top of the report the section is printed) is set to 3.75 inches, then the MoveLayout property moves the group footer section to be printed in the area from 3.75 inches to 4.25 inches from the top of the report (a total of 0.5 inches).

The following example demonstrates how to print the group footer at a specific place on the page by adjusting the section heights, and by using a sample user-defined function called SetGrpFtrLoc() that tests to see if the Top property is set to less than the GrpFtrLoc setting (the location where the group footer will be printed). If it is, the function moves the group footer section to the next printing location. If not, it does not move the group footer.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the sample database NWIND.MDB.
2. Create a new module and enter the following line in the Declarations section of the module:

```
Option Explicit
```

Enter the following function in the module:

```
Function SetGrpFtrLoc(Rpt as Report, GrpFtrLoc as Double)
 GrpFtrLoc=GrpFtrLoc*1440 'Convert from inches to twips.

 If Rpt.top < GrpFtrLoc Then 'Not at location yet, so
 Rpt.moveLayout = True 'move to next print location.
 Rpt.nextRecord = False 'Do not go to next record.
 Rpt.printSection = False 'Do not print the section.
 End If

End Function
```

3. Save and then close the module.
4. Open the Catalog report in Design view.
5. Change the Category Name header section's Height property to:  
1.7 inches
6. Change the detail section's Height property to:

0.3 inches

7. Set the Category Name footer section's properties as follows:

Height: 0.3 inches  
ForceNewPage: After Section  
OnFormat: =SetGrpFtrLoc([Report],7)

Note that the 7 in the SetGrpFtrLoc() function in the footer section's OnFormat property indicates that you want the group footer to begin 7 inches from the top of the page.

8. Create a new text box control with the following properties in the report's Category Name footer section:

Name: Number of Products  
ControlSource: =Count([Product ID])

NOTE: In Microsoft Access version 1.x, the Name property is called the ControlName property.

9. Preview the report.

NOTE: If you do not adjust the section heights as specified, the location of the group footer will vary from page to page.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 23, "Using Macros to Print Reports or Transfer Data," pages 576-579

Microsoft Access "User's Guide," version 1.1, Chapter 23, "Using Macros to Print Reports or Transfer Data," pages 580-583

Microsoft Access "User's Guide," version 2.0, Chapter 26, "Using Macros to Print Reports or Transfer Data," pages 693-696

For more information about the MoveLayout, NextRecord, and PrintSection report properties, search for "MoveLayout" then "MoveLayout, NextRecord, and PrintSection Properties" using the Microsoft Access version 2.0 Help menu.

Additional reference words: 1.00 1.10 2.00 reports

KBCategory: kbusage

KBSubcategory: RptSort

## INF: How to Print Odd or Even Pages of a Report

Article ID: Q101075

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to print separately the odd and even pages of a report for duplex printing or to print reports with different margins for binder storage.

### MORE INFORMATION

=====

The best way to print separately odd and even pages of report is to set up a form that prompts you for the pages to print (odd or even) and then calls a macro to print the report. (Page 559 of the 1.0 "User's Guide" shows a sample form that prompts the user to fill in values on the form before a macro prints the report.) The form in the example below contains a text box control to hold the value "odd" or "even" and a command button to print the report.

NOTE: When you use the following steps to print even pages, all the odd pages generate blank pages (and vice versa). Whenever you implement this routine, your report generates blank pages between each printed page.

1. Create a new form called Print Odd Even Pages:
  - a. Locate an unbound text box with ControlName = PrintPages.
  - b. Locate a command button with Caption = Print Report. From the OnPush property, call the following OddEven.Print macro group.
2. Create a new macro called OddEven with the following macro groups:

NOTE: In the following example, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this example.

| Macro Group Name | Condition                                                                    | Action      |
|------------------|------------------------------------------------------------------------------|-------------|
| Print            |                                                                              | OpenReport  |
| TestOddEven      | Forms![Print Odd Even Pages] _<br>![PrintPages]="Even" And _<br>Page Mod 2=1 | CancelEvent |
|                  | Forms![Print Odd Even Pages] _<br>![PrintPages]="Odd" And _<br>Page Mod 2=0  | CancelEvent |

OpenReport Actions

-----  
Report Name: Customer Mailing Labels  
View: Print Preview

3. From each report section that is printing data, call the macro group OddEven.TestOddEven from the OnPrint event.

For example, in the NWIND.MDB sample database report Customer Mailing Labels, the only section that is printing is the detail section. The page header and page footer in this report do not print any values. For the detail section, set the OnPrint event to:

OddEven.TestOddEven

4. With the form open, enter Odd or Even in the text box. When you choose the Print Report command button, the Customer Mailing Labels report opens. Press OK for all countries. The report then runs its OnPrint events, which call the macro group OddEven.TestOddEven to print only the odd or even pages.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 23, "Using Macros to Print Reports or Transfer Data," pages 556-560

Microsoft Access "User's Guide," version 2.0, Chapter 26, "Using Macros to Print Reports or Transfer Data"

Additional reference words: 1.00 1.10 2.00 every other page left right

KBCategory: kbusage

KBSubcategory: RptLayout

**INF: How to Print Text to Default Printer with Access Basic**  
**Article ID: Q105662**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

In Microsoft Access, there is no standard way of sending text to the default printer without printing a form or report. This article demonstrates a method of using Windows API functions to send text to the printer.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0. This article also assumes that you are familiar with programming in the Windows environment. For more information on programming in the Windows environment using API functions, please refer to the Microsoft Windows Software Development Kit manuals.

MORE INFORMATION

=====

Unlike Microsoft Visual Basic, Microsoft Access does not have a printer object for sending text to the printer. The following example demonstrates how to use Windows API functions to send text to the printer.

Notes

-----

- This article does not describe the API function arguments. For details on the API functions, please refer to the Microsoft Windows Software Development Kit manuals.
- You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code. You should also alias any API function declarations. For more information on using aliases, search for "alias" then "declare statement" using the Microsoft Access Help menu.
- In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'-----
'Declaration Section
'-----
Option Compare Database 'Use database order for string comparisons
```

Option Explicit

Type DOCINFO

    cbSize As Integer  
    lpzDocname As Long  
    lpzOutPut As Long

End Type

Declare Function GetProfileString% Lib "Kernel" (ByVal lpAppName\$, \_  
    ByVal lpkeyname\$, ByVal lpDefault\$, \_  
    ByVal lpReturnedString\$, ByVal nsize%)

Declare Function CreateDC% Lib "GDI" (ByVal lpDriverName\$, \_  
    ByVal lpDeviceName\$, ByVal lpOutput\$, \_  
    lpInitData As Any)

Declare Function DeleteDC% Lib "GDI" (ByVal hDC%)

Declare Function TextOut% Lib "GDI" (ByVal hDC%, ByVal X%, ByVal Y%, \_  
    ByVal lpString\$, ByVal nCount%)

Declare Function Mylstrcpy& Lib "Kernel" Alias "lstrcpy" ( \_  
    ByVal lpString1 As Any, ByVal lpString2 As Any)

Declare Function StartDoc% Lib "GDI" (ByVal hDC%, lpdi As DOCINFO)

Declare Function StartPage% Lib "GDI" (ByVal hDC%)

Declare Function EndPage% Lib "GDI" (ByVal hDC%)

Declare Function EndDocAPI% Lib "GDI" Alias "EndDoc" (ByVal hDC%)

Declare Function Rectangle% Lib "GDI" (ByVal hDC%, ByVal X1%, \_  
    ByVal Y1%, ByVal X2%, ByVal Y2%)

'-----  
'Start of Function  
'-----

Function Printer ()

    Dim lpReturnedString\$  
    Dim MyDoc As DOCINFO  
    Dim MyDocumentname\$  
    Dim nPrinter, nDriver, nDevice  
    Dim szDevice, szDriver, szOutPut  
    Dim hDC%, X%, MyString\$

    MyDocumentname\$ = "My Document"

'-----  
' Retrieve the currently selected printer as  
' establish with the Control panel.  
' Sample string as returned by lpReturnedString\$:  
'  
'    HP LaserJet III Si PostScript, pscript, LPT1:  
'-----

    lpReturnedString\$ = Space\$(128)  
    nPrinter = GetProfileString("windows", "device", ",,,", \_  
        lpReturnedString\$, Len(lpReturnedString\$))

'-----  
' Parse the string of its three components  
'-----

    nDevice = InStr(lpReturnedString\$, ",")  
    nDriver = InStr(nDevice + 2, lpReturnedString\$, ",")

```

szDevice = Mid$(lpReturnedString$, 1, nDevice - 1)
szDriver = Mid$(lpReturnedString$, nDevice + 1, nDriver - nDevice _
- 1)
szOutPut = Mid$(lpReturnedString$, nDriver + 1)

'-----
' Create the DOCINFO structure for StartDoc()
' - lpszDocname is name displayed in PRINTMAN
' - lpszOutPut is not used and set to NULL
'-----
MyDoc.cbSize = Len(MyDoc)
MyDoc.lpszDocname = Mylstrcpy(MyDocumentname$, MyDocumentname$)
MyDoc.lpszOutPut = 0&

'-----
' Create the device context
'-----
hDC% = CreateDC(szDriver, szDevice, szOutPut, 0&)

X% = StartDoc(hDC%, MyDoc)
X% = StartPage(hDC%)

'-----
' Rectangle arguments are X, Y, cX, cY
'-----
X% = Rectangle(hDC%, 10, 10, 1000, 150)
MyString$ = "Derek and Robyn are brother and sister!"

'-----
' Second and third arguments are the X and Y
' coordinates on paper.
'-----
X% = TextOut(hDC%, 30, 20, MyString$, Len(MyString$))

X% = EndPage(hDC%)
X% = EndDocAPI(hDC%)

'-----
' Release the device context when done
'-----
hDC% = DeleteDC(hDC%)

```

End Function

```

'-----
'End of Function
'-----

```

#### REFERENCES

=====

Microsoft Windows Software Development Kit "Guide to Programming,"  
version 3.1, page 254-255

Microsoft Windows Software Development Kit "Programmer's Reference  
Volume 2: Functions," version 3.1



Additional reference words: 1.00 1.10 2.00 ADK  
KBCategory: kbprint  
KBSubcategory: PtrOthr

**INF: How to Programmatically Embed or Link an Object in a Form**  
**Article ID: Q114214**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to programmatically embed or link an object in an OLE object field on a form using the new object frame properties in Microsoft Access version 2.0.

MORE INFORMATION

=====

You can set the object frame Action property at run time to perform a number of operations on an object frame. These operations include the ability to embed and link objects in an object frame, as well as other operations for programmatic access to OLE functionality.

There are other object frame properties that need to be set as prerequisites to setting the Action property. The properties that need to be set are determined by the type of OLE object you are working with and the type of action, using the Action property, you want to perform.

NOTE: The constants for use by the Action property are defined in the file CONSTANT.TXT, which is included with Microsoft Access. To use the constants in your code, you must either declare them yourself or copy them from the CONSTANT.TXT file and paste them into the Declarations section of your Access Basic module.

Linking an OLE Object

-----

To link an object in an object frame on a form, first set the following properties:

- OLETypeAllowed: Set to OLE\_LINKED to indicate the OLE field will contain a linked object.
- SourceDoc: Set to the path and filename of the file to be linked.
- SourceItem: Indicates the data in the source document to link to. This could be a cell or cell range in a Microsoft Excel spreadsheet, or a bookmark in a Microsoft Word for Windows document. This property setting is optional.

Once you set these properties, you can set the Action property to OLE\_CREATE\_LINK to link the object in the object frame.

The following sample code demonstrates how to programmatically link a range of cells from a Microsoft Excel spreadsheet to an OLE object field named Sales Totals bound to an object frame with the same name on a form:

```
' Specify what kind of object can appear in the field.
[Sales Totals].OLETypeAllowed = OLE_LINKED

' Specify the file to be linked.
[Sales Totals].SourceDoc = "C:\EXCEL\SALES.XLS"

' Specify the (optional) cell range to link to.
[Sales Totals].SourceItem = "R1C1:R5C5"

' Create the linked object.
[Sales Totals].Action = OLE_CREATE_LINK
```

#### Embedding an OLE Object

-----

To embed an object in an object frame on a form, first set the following properties:

- OLETypeAllowed: Set to OLE\_EMBEDDED to indicate the OLE field will contain an embedded object.
- SourceDoc: Set to the path and filename of the file containing the information to embed.

Once you set these properties, you can set the Action property to OLE\_CREATE\_EMBED to create an embedded object with a copy of the information in the source document file.

The following sample code demonstrates how to embed a Word for Windows document in an OLE object field named Word Documents bound to an object frame with the same name on a form:

```
' Specify what kind of object can appear in the field.
[Word Documents].OLETypeAllowed = OLE_EMBEDDED

' Specify the file containing the information to embed.
[Word Documents].SourceDoc = "C:\WINWORD\JONES.DOC"

' Create the embedded object.
[Word Documents].Action = OLE_CREATE_EMBED
```

#### Creating an Empty Embedded Object

-----

To create an empty embedded object in an object frame, first set the following properties:

- Class: Set to the type of object you want to create. For a Microsoft Excel worksheet, use Excel.Sheet. For a Word for Windows document, use Word.Document.
- OLETypeAllowed: Set to OLE\_EMBEDDED to indicate the OLE field will contain an embedded object.

Once you set these properties, you can set the Action property to OLE\_CREATE\_EMBED to create an empty embedded object of the type specified

in the Class property. Then, set the Action property again to invoke the application for that object.

The following sample code demonstrates how to programmatically create new Word for Windows documents in an OLE object field named Word Documents bound to an object frame with the same name on a form:

```
' Specify what kind of object can appear in the field.
[Word Documents].Class = "Word.Document"

' Specify what kind of object can appear in the field.
[Word Documents].OLETypeAllowed = OLE_EMBEDDED

' Create the embedded object.
[Word Documents].Action = OLE_CREATE_EMBED

' Invoke Word for Windows to edit the empty embedded object.
[Word Documents].Action = OLE_ACTIVATE
```

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 13,  
"Communication With Other Applications," pages 295-297

For more information about the Action property, search for "Action" then  
"Action Property" using the Microsoft Access Help menu.

Additional reference words: 2.00 automation  
KBCategory: kbole  
KBSubcategory: IntpOle

**INF: How to Query an Oracle Table Without Attaching It**  
**Article ID: Q109824**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to use the SQL IN clause to query an Oracle database without attaching it.

In Microsoft Access version 2.0, this can be accomplished more easily by using a SQL pass-through query. See the "References" section of this article for a reference on this method.

MORE INFORMATION

=====

You can use the IN clause to query only one external database at a time. The IN clause requires two arguments: database name and connect string. When you attach Oracle tables, specify an empty string for the database name.

The connect string consists of the following items:

- Data Source Name (DSN): The name of the ODBC data source.
- Database Name (DATABASE): The Oracle connect string to the server. The Oracle connect string consists of a prefix and a server name. (For TCP/IP users, there may be a prefix, a server name, and an SID.) The prefix is based on the network protocol. The following are the prefixes for networks that support SQL\*Net for Windows:
  - t: for TCP/IP
  - x: for IPX/SPX
  - p: for Named Pipes
- Workstation ID (WSID): The name of the workstation from which the connection is being made (optional).
- User ID (UID): The login ID (optional). If not present, the user will be prompted for the login ID the first time the macro is run.
- Password (PWD): The login password (optional). If not present, the user will be prompted for the login password the first time the macro is run.

The following is a sample connect string to Oracle. Note that it starts with "ODBC," indicating to the system that it should use the ODBC driver (as opposed to using an installable ISAM driver):

NOTE: In the following sample connect string, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating

this connect string.

```
ODBC;DSN=DataSourceName;Database=x:SrvrName;WSID=Wrk1;USID=JaneDoe;_
PWD=ABC123
```

The following example demonstrates how to use the SQL IN clause in a query to query an Oracle database without attaching it. Note that this example assumes that the ODBC Driver Manager and Oracle ODBC driver is installed, that a data source called Corp was set up using the ODBC Driver Manager, and that the table called Employees on the server called Payroll can be successfully attached using the Attach command on the File menu:

1. Open the sample database NWIND.MDB.
2. Create a new, unbound query.
3. From the View menu, choose SQL.
4. Type the following query in the SQL window, and then choose OK:

```
Select *
FROM Employees
IN "" "ODBC;dsn=Corp;database=p:Payroll;uid=JSmith;pwd=KL3d";
```

5. Save and then close the query. Note that if you return to the SQL window after saving the query, Microsoft Access may place additional characters in the IN clause that you will have to remove in order to avoid a syntax error.

#### REFERENCES

=====

For more information on the SQL IN clause, search for "In clause (SQL)" then "IN Clause (SQL)" using the Microsoft Access Help menu.

Microsoft Access "Language Reference," version 1.1, Appendix B, "Microsoft Access SQL," page 513

Microsoft Access "User's Guide," version 2.0, pages 273 and 276

Additional reference words: 1.00 1.10 2.00 attach remote data source  
KBCategory: kbusage  
KBSubcategory: ObcOthr

## **INF: How to Query on Category Keywords**

**Article ID: Q119525**

-----  
The information in this article applies to:

- Microsoft Access version(s) 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Every Knowledge Base (KB) article contains a category and a subcategory. You can query on the categories and subcategories to help you find specific groups of articles. If you are using the KB Help file, you can use the Find button on the toolbar to enter category queries.

The categories and subcategories are abbreviated and concatenated to form unique words that you can query on. For example, FmsCmbo stands for Forms, Combo/List Box, where Forms is the category and Combo/List Box is the subcategory. Note that the first letter of the category and subcategory are capitalized; this does not affect your queries, but makes it easier to recognize the subcategory.

To display all the articles in a category, query on the category code with a wildcard (for example, "Fms\*"). To display all the articles in a specific subcategory, query on the entire code (for example, "FmsCmbo"). You can use the logical operator OR to query on more than one category or subcategory (for example, "FmsCmbo OR FmsProp"). When you are looking for articles on a specific subcategory it is often a good idea to include the Othr subcategory as well--not all articles are put in the best possible subcategory.

See the "References" section later in this article for information about how to find a complete list of categories and subcategories for Microsoft Access KB articles.

### MORE INFORMATION

=====

To query on one category, enter the category code and a wildcard. For example, to display all Forms articles enter:

Fms\*

To query on more than one category, enter each category code with a wildcard. For example, to display all the articles on Forms or Reports enter:

Fms\* or Rpts\* or Fmr\*

To query on one subcategory, enter the complete code. For example, to display all articles on Query, Joins enter:

QryJoin

To query on more than one subcategory, enter all the complete codes. For

example, to display articles on Query, Joins and Query, Make Table enter:

QryJoin or QryMktbl

You can also mix categories and subcategories in one query. For example, to display articles on Reports, Labels and all Printing articles, enter:

RptLabel or Ptr\*

#### REFERENCES

=====

For a complete list of all the categories and subcategories used in Microsoft Access KB articles, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q119526

TITLE : INF: Microsoft Access KB Categories and Subcategories

For more information about how to query using the Find button, click the Find button, then click the Hints button in the Search dialog box.

Additional reference words: 1.00 1.10 2.00 kbkeyword

KBCategory: kbusage

KBSubcategory: KbCodes



**INF: How to Query SQL Server Tables Without Attaching**  
**Article ID: Q101375**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to use the SQL IN clause to query a SQL Server table without attaching it.

MORE INFORMATION

=====

You can use IN to query only one external database at a time.

The IN clause requires two arguments: database name and connect string. When you attach SQL Server tables, you must specify an empty string for the database name. The connect string consists of the following items:

|                          |                                                                      |
|--------------------------|----------------------------------------------------------------------|
| Data Source Name (DSN)   | The name of the SQL Server computer                                  |
| Database Name (DATABASE) | The name of the SQL Server database                                  |
| Workstation ID (WSID)    | The name of the workstation that is making the connection (optional) |
| User ID (UID)            | The login id*                                                        |
| Password (PWD)           | The user password*                                                   |

\* This item is optional within the connect string. If it is not present, the system prompts the user with a dialog box to enter this information the first time the macro is executed each session.

The following is a sample connect string to SQL Server. Note that it starts with "ODBC," which tells the system that it uses the ODBC driver (as opposed to an installable ISAM driver).

```
ODBC;DSN=ServerName;Database=Pubs;WSID=Wrk1;USID=JaneDoe;PWD=ABC123
```

The following example assumes that the ODBC driver manager and SQL Server driver are installed, that a data source called "Corp" was set up using the ODBC driver manager, and that the table "Employees" can be successfully attached using the Attach command on the File menu.

1. Open the sample database NWIND.MDB.
2. Create a new query.
3. From the View menu, choose SQL.

4. Enter the following query:

```
Select *
FROM Employees
IN "" "ODBC;dsn=Corp;database=Payroll;uid=JSmith;pwd=KL3d";
```

5. Choose OK to exit the SQL dialog box and close and save the query.

NOTE: If you return to the View SQL window after saving the query, Microsoft Access may place additional characters in the IN clause that you must remove to avoid a syntax error.

#### REFERENCES

=====

Microsoft Access "Language Reference," version 1.1, Appendix B, "Microsoft Access SQL," page 513

For more information about the IN clause, search for "IN (SQL)" then "IN Clause (SQL)" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 attach remote data source

KBCategory: kbusage

KBSubcategory: ObcOthr

## INF: How to Rank Records Within a Query

Article ID: Q120608

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article contains three examples that demonstrate how to rank records within a query. The first example ranks records from a table. The second example ranks the results of a totals query. The third example demonstrates an alternative method of handling ties in ranking.

All the examples in this article require a field containing a value that can be evaluated using either the greater than (>) or the less than (<) operator, and use a subquery to count the number of records that are greater than or less than the current record.

### MORE INFORMATION

=====

In the first two examples below, records following a tie in ranking are given the same rank number they would have received had there been no tie. The following table demonstrates a sample result with a tie in ranking:

| Salesperson | Sales  | Rank |
|-------------|--------|------|
| Peterson    | \$8000 | 1    |
| Wakita      | \$7000 | 2    |
| Akerley     | \$7000 | 2    |
| Reston      | \$6000 | 4    |

In the third example below, a second query using the DISTINCT clause is used to return a unique list of ranking less than that of the current record. The following table, using the same records as the table above, demonstrates the results of this method:

| Salesperson | Sales  | Rank |
|-------------|--------|------|
| Peterson    | \$8000 | 1    |
| Wakita      | \$7000 | 2    |
| Akerley     | \$7000 | 2    |
| Reston      | \$6000 | 3    |

### Notes:

- In the following examples, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating these examples.
- Following the steps in these examples will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

### Example 1

-----

This example returns results that show the order in which employees were hired:

1. Open the sample database NWIND.MDB and create a new query based on the Employees table.
2. If the property sheet is not being displayed, choose the Properties button on the toolbar.
3. Click the title bar of the Employees field list, and then set the field list's Alias property to Emp1.
4. Drag the Last Name and Hire Date fields from the field list to the query grid.
5. In the Hire Date column's Sort row, select Ascending.
6. In the third column's Field row, enter the following expression:  

```
Seniority: (Select Count(*) from Employees Where _
[Hire Date] < [Emp1].[Hire Date];)
```
7. Run the query. Note that the returned records are ranked from 0 to 14. The subquery returns the number of employees hired before the current employee. To rank the records from 1 to 15, use the following expression in step 6 above:  

```
Seniority: (Select Count(*) from Employees Where _
[Hire Date] < [Emp1].[Hire Date]+1;)
```
8. To see the results of a tie in ranking, change the hire date of any employee to match another (in the Employees table).

### Example 2

-----

This example returns results that rank employees by the number of sales each employee has made:

1. Create a new query based on the Orders table.
2. Drag the Employee ID and Order ID fields from the field list to the query grid.
3. Choose the Totals button on the toolbar.
4. In the Employee ID column's Total row, select Group By. In the Order ID column's Total row, select Count. Save the query as OrdersPlaced.
5. Create a new query based on the OrdersPlaced query.
6. Set the OrdersPlaced field list's Alias property to Ord1.

7. Drag both fields from the field list to the query grid, and then select Descending in the CountofOrder ID column's Sort row.

8. In the third column's Field row, enter the following expression:

```
Ranking: (Select Count(*) from OrdersPlaced Where _
[CountofOrder ID] > [Ord1].[CountofOrder ID]) + 1 _
```

9. Run the query.

### Example 3

-----

This example demonstrates an alternative method of handling ties in ranking:

1. Repeat steps 1-8 in example 2. After step 8, save the query as EmpRanking. Run the query and note that the number of orders placed by employees 3 and 1 differs by 13.
2. Delete from the Orders table 13 of the orders placed by employee 3. Run the EmpRanking query again and note the tie.
3. Create a new query based on the Orders table. Drag the Employee ID and Order ID fields from the field list to the query grid.
4. Choose the Totals button on the toolbar.
5. In the Employee ID column's Total row, select Group By. In the Order ID column's Total row, select Count.
6. In the Employee ID column's Show row, clear the check box.
7. Set the query's UniqueValues property to Yes.
8. Run the query. Note that the query returns a list of the eight different totals of orders placed. Save the query as DistinctCount.
9. View the EmpRanking query in Design view. Replace the expression in the third column's Field row with the following expression and then run the query:

```
Ranking: (Select Count(*) from DistinctCount Where _
[CountofOrder ID] > Ord1.[CountofOrder ID]) + 1 _
```

### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 12, "Advanced Queries," pages 296-297

For more information about subqueries, search for "subqueries" then "SQL Subqueries" using the Microsoft Access Help menu.

Additional reference words: 2.00 top first

KBCategory: kbusage

KBSubcategory: QryHowto



**INF: How to Refer to a Control on a Subform or Subreport**  
**Article ID: Q113352**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to refer to controls on subforms or subreports, and describes some common problems you may encounter when you refer to controls on subforms or subreports.

MORE INFORMATION

=====

To refer to a control on a subform, use the following syntax:

```
Forms![main form name]![subform control name].Form![control name]
```

To refer to a control on a subreport, use the following syntax:

```
Reports![main report name]![subreport control name].Report![control name]
```

NOTE: Only subforms are discussed in the rest of this article, but all the information applies to both subforms and subreports.

It is important to note that you cannot refer to controls on a subform with the following syntax:

```
Forms![subform name]![control name]
```

This is because a subform on a main form is not a form, but is a control just like a text box or a list box. You must refer to a subform as a control rather than a form, and specify the Form identifier following the subform control name reference to access the controls on a subform.

It is also important to note that when you are referring to controls on a subform, you must specify the value contained in the ControlName property of the subform control. The ControlName property should not be confused with the SourceObject property for a subform control. The SourceObject property is used to indicate which subform to use in the subform control. The ControlName property is used to specify the name by which the subform control will be referenced. ControlName is typically set the same as the SourceObject, but it does not have to be.

For example, consider an Order Details subform on an Orders form with the following properties:

```
ControlName: Order Details
SourceObject: Order Details
```

You can refer to a Unit Price control on the Order Details subform

with the following reference:

```
Forms![Orders]![Order Details].Form![Unit Price]
```

If, however, the subform control has the following properties

```
ControlName: Details
SourceObject: Order Details
```

you must refer to the Unit Price control on the Order Details subform with this reference:

```
Forms![Orders]![Details].Form![Unit Price]
```

#### Examples of Referencing Controls on a Subform

---

The following SetValue macro action could be used to increase the Unit Price value on the Orders Subform by 10 percent:

```
SetValue
 Item: Forms![Orders]![Order Details].Form![Unit Price]
 Expression: Forms![Orders]![Order Details].Form![Unit Price]* 1.1
```

If the macro is attached to a button on the Orders form, you can use this expression for the SetValue Expression argument:

```
[Order Details].Form![Unit Price]*1.1
```

If you are referring to a control on a subform from another control on the same subform, you do not have to enter the Form property identifier. For example, to refer to the Unit Price value on the Order Details subform in a macro attached to a button on the Order Details subform, you can enter:

```
[Unit Price]
```

The following expression can be entered as the ControlSource property for the Subtotal control on the Orders main form to display a value calculated in the hidden Order Subtotal control on the Order Details subform:

```
=[Orders Subform].Form![Order Subtotal]
```

To refer to the value of a control on the parent (main) form from a control on a subform, use the Parent property. For example, the following expression entered in a control on a subform refers to the Customer ID field on the parent form.

```
=Parent![Customer ID]
```

To refer to a control on a nested subform (a subform on a subform), the following syntax can be used:

```
Forms![main form name]![subform control name].Form![nested subform control name].Form![control name]
```

#### Common Problems Encountered When Referencing Subform Controls

---



- The following error message occurs:

Invalid reference to form '<subform name>'

This means that you tried to reference a control on a subform with the following syntax:

```
Forms![subform name]![control name]
```

The problem is that the subform is not really a form, but a control that appears on a main form. Refer to the subform as a control rather than a form with the following syntax:

```
Forms![main form name]![subform control name].Form![control name]
```

- "#Name?" appears in a control with an expression referring to a subform control.

This can occur when the ControlName property for the subform control is not what you expect. Open the main form in Design view, select the subform control, then activate the Property window from the View menu. Compare the ControlName property value to the SourceObject property value.

The SourceObject property is used to indicate which subform to use in the subform control. The ControlName property is used to specify the name by which the subform control will be referenced. The ControlName is typically set the same as the SourceObject, but it does not have to be.

- The following error message occurs:

Invalid reference to field '<subform name>'

This error message has the same cause as the problem above.

#### REFERENCES

=====

For more information, search for "subforms: referring to controls on" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ExrObj

## INF: How to Reference Controls or Fields on a Subform

Article ID: Q95007

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Microsoft Access enables a developer to create subform controls that display embedded forms on a form or report or display embedded reports on a report. You can reference a control or field that resides on a subform by using the Form property of the parent form. This article describes several techniques you can use to reference a control or field that resides on a subform or nested subform.

### MORE INFORMATION

=====

In Microsoft Access, a parent form can have many subforms, but a top-level parent form is limited to three levels of nesting for nested subforms.

To reference a control or field on the subform, use the Form property. For more information on the Form property, search for "Form," then "Form, Report Properties" using the Microsoft Access Help menu.

The following step-by-step example shows by example how to reference a field or control on a subform or nested subform:

1. Create three tables giving them each two fields as shown below:

Table: Main Table

-----

ID: Counter

Field1: Text

Table: Sub Form Table

-----

ID: Counter

Field2: Text

Table: Nested SubForm Table

-----

ID: Counter

Field3: Text

2. Create three forms based on the three tables and set the form properties as shown below. Note that forms have RecordSources--these become ControlSources when you make each form a subform.

Form: Nested Sub

-----

ControlSource: Nested Subfrom Table

Field: ID

Field: Field3

Form: Sub

-----

ControlSource: Sub Form Table

Field: ID

Field: Field2

Control: Nested Sub

Form: Main

-----

ControlSource: Main Table

Field: ID

Field: Field1

Control: Sub

3. Create a new module, and add the following code to it:

```
Function DoIt ()
 Dim MyForm As Form
 Dim MySubForm As Form
 Dim MyControl As Control

 '=====
 'SUBFORM - The next four examples show how to set the value of
 'Field2 on the Sub subform. This example shows four ways to
 'reference the control:
 '
 ' 1. Straight reference using the Forms object.
 ' 2. Setting the Forms object to a form variable.
 ' 3. Setting the subform to a form variable.
 ' 4. Setting the field on the subform to a control variable.
 '
 'NOTE: Each example below is based on the previous example.
 'Some require that the object on the right side of the
 'equation be already initialized. Although variables are
 'used, you can substitute this with the proper Forms object.
 'The example of nested forms uses the Forms object in all the
 'examples.
 '=====

 'straight reference
 '-----
 Forms![Main]![Sub].Form!Field2 = "Hello"

 'referenced using form property
 '-----
 Set MyForm = Forms![Main]
 MyForm.[Sub].Form.Field2 = "Hello"

 'setting subform to a form object
 '-----
 Set MySubForm = MyForm.[Sub].Form
 MySubForm.Field2 = "Hello"

 'reference item on subform through control object
 '-----
```

```

Set MyControl = MySubForm.Field2
MyControl = "Hello"

'=====
'NESTED SUBFORM - The next four examples show the same ideas
'as shown above but for nested subforms.
'=====

'straight reference
'-----
Forms![Main]![Sub].Form![Nested Sub].Form!Field3 = "Hello"

'referenced using form property
'-----
Set MyForm = Forms![Main]![Sub].Form
MyForm.[Nested Sub].Form.Field3 = "Hello"

'setting subform to a form object
'-----
Set MySubForm = Forms![Main]![Sub].Form.[Nested Sub].Form
MySubForm.Field3 = "Hello"

'reference item on subform through control object
'-----
Set MyControl =
 Forms![Main]![Sub].Form![Nested Sub].Form!Field3
MyControl = "Hello"

```

End Function

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbusage  
KBSubcategory: FmsHowto

## **INF: How to Repair a Damaged Database**

**Article ID: Q109953**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

The Repair and Compact utilities in Microsoft Access are useful tools for recovering and optimizing Microsoft Access database files.

However, these utilities affect only tables, queries, and indexes. They cannot affect forms, reports, macros, or modules.

### MORE INFORMATION

=====

#### The Repair Utility

-----

You can use the Repair utility to help fix databases with certain types of damage. The Repair utility will attempt to repair only the tables, queries, and indexes in the database. The Repair utility will not attempt to repair damaged forms, reports, macros, or modules, but will simply copy them to the new repaired database.

#### Compact Utility

-----

The Compact utility eliminates empty space in an existing database. It does so by creating a temporary database, exporting all the objects from the original database into the temporary database, removing the original database, and then renaming the temporary database to the original database's name. You also have the option to name the temporary database to some other name so that you can retain the original database.

#### Recovering a Damaged Database

-----

The following steps outline a general method for attempting to repair a damaged database:

1. Back up the damaged database (.MDB) file.
2. If the damage is in a table, query, or index, run the Repair utility as follows:
  - a. Close all open databases.
  - b. From the File menu, choose Repair.
  - c. In the File Name box, select the damaged database and choose OK.

If the repair is successful, you will receive the message:

Repair of database '<path><database name>' completed successfully.

If the repair does not succeed you will receive a message stating so. This means either that the damage to a table, query, or index is so severe that it cannot be corrected, or that the damage is in an object that the Repair utility cannot affect.

3. If the damage is in a form or report, the damage can be either in the form or report itself, or in one or more controls on the form or report.
  - If the damage is in the form or report itself, the best approach is to create a new form or report and then copy the controls from the original form or report.
  - If the damage is in a control on the form or report, create a new form or report and then re-create the controls on the new form or report. It is best to re-create all the controls, since there is no way to tell which controls are damaged.
4. If the damage is in a macro or module, the damage can be in the macro or module itself, or in the contents of the macro or module.
  - If the damage is in the macro or module itself, create a new macro or module and then copy the contents of the original macro or module.
  - If the damage is in the contents of the macro or module, you must create a new macro or module and then re-create the contents of the original macro or module.

If you cannot repair the database with these steps, the database is probably damaged beyond repair. If this is the case, you should restore your last backup database, or re-create the database.

#### REFERENCES

=====

Microsoft Access "User's Guide," pages 636-639

Additional reference words: 1.00 1.10 2.00 Repairing Compacting Fixing  
KBCategory: kbusage  
KBSubcategory: GnlCmp

**INF: How to Repeat Subreport Header at Top of Page**  
**Article ID: Q120907**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY  
=====

This article describes how to repeat a subreport's header when the subreport spans multiple pages.

MORE INFORMATION  
=====

The method demonstrated in the following example uses a flag in the page header to indicate whether the subreport spans multiple pages. When the subreport's report header is printed the flag is set to 1. When the subreport's report footer is printed, the flag is set to 0. When the page header is printed, if the flag is set to 1 then the subreport spans multiple pages and the subreport's header should be repeated.

The following example demonstrates how to cause a subreport's header to be repeated if the subreport spans multiple pages:

1. Open the sample database NWIND.MDB.
2. Create the following macro and then save it as RepeatSubHeader:

| Macro Name | Condition | Action      |
|------------|-----------|-------------|
| MainRH     |           | SetValue    |
| MainPH     | [Flag]=0  | CancelEvent |
| SubRH      |           | SetValue    |
| SubRF      |           | SetValue    |

MainRH Actions

-----

SetValue  
Item: Reports![Categories Main Report]![Flag]  
Expression: 0

SubRH Actions

-----

SetValue  
Item: Reports![Categories Main Report]![Flag]  
Expression: 1

SubRF Actions

-----

SetValue  
Item: Reports![Categories Main Report]![Flag]  
Expression: 0

3. Create a new report called Products Sub Report based on the Products table.
4. Place the following three text box controls in the new report's detail section:

```
Text Box:
 ControlName: Product ID
 ControlSource: Product ID
Text Box:
 ControlName: Category ID
 ControlSource: Category ID
Text Box:
 ControlName: Product Name
 ControlSource: Product Name
```

5. From the Format menu, choose Report Header/Footer.
6. Place the following three labels in the report's report header section:

```
Label:
 ControlName: Text1
 Caption: Product ID
Label:
 ControlName: Text2
 Caption: Category ID
Label:
 ControlName: Text3
 Caption: Product Name
```

7. Set the report header section's OnPrint property to the following macro:

```
RepeatSubHeader.SubRH
```

8. Set the report footer section's OnFormat property to the following macro:

```
RepeatSubHeader.SubRF
```

9. Create a new report called Categories Main Report based on the Categories table.
10. Copy the controls from the Product Sub Report report's report header section and paste them into the page header section of the Categories Main Report report.
11. Place the following text box in the Categories Main Report report's page header section:

```
Text Box:
 ControlName: Flag
 Visible: No
```

12. Place the following three text boxes in the Categories Main Report report's detail section:



Text Box:  
 ControlName: Category ID  
 ControlSource: Category ID  
 Text Box:  
 ControlName: Category Name  
 ControlSource: Category Name  
 Text Box:  
 ControlName: Description  
 ControlSource: Description

13. Embed the Products Sub Report report in the Categories Main Report report by dragging the Products Sub Report report from the Database window to the detail section of the Categories Main Report report.
14. From the Format menu, choose Report Header/Footer to add a report header and footer to the main report. If you do not want a main report header or footer, set the header or footer section's height to 0.
15. Set the OnFormat property of the report header section of the Categories Main Report report to the following macro:

```
RepeatSubHeader.MainRH
```

16. Set the OnPrint property of the page header section of the Categories Main Report report to the following macro:

```
RepeatSubHeader.MainPH.
```

17. Preview or print the report. Note that the subreport's header information is repeated when the subreport spans multiple pages.

#### Displaying Main Report Information on Every Page

---

The example above assumes that you do not want anything except the subreport header to be printed in the main report's page header. If you do want to print other information, you will need to hide and unhide each subreport header control in the page header. Change the MainPH macro in step 2 above to:

| Macro Name | Condition | Action                                                 |
|------------|-----------|--------------------------------------------------------|
| MainPH     | [Flag]=0  | SetValue<br>Item: [Text1].visible<br>Expression: False |
| ...        | ...       | SetValue<br>Item: [Text2].visible<br>Expression: False |
| ...        | ...       | SetValue<br>Item: [Text3].visible<br>Expression: False |
|            | [Flag]=1  | SetValue<br>Item: [Text1].visible<br>Expression: True  |
| ...        | ...       | SetValue                                               |

```
Item: [Text2].visible
Expression: True
... SetValue
Item: [Text3].visible
Expression: True
```

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbusage  
KBSubcategory: RptSub

## INF: How to Replace a String with Another String

Article ID: Q109330

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article demonstrates a sample function you can use to replace one string with another string.

### MORE INFORMATION

=====

The following sample function can be used to replace one string with another. This function accepts three parameters that specify:

- the string that contains the string you want to replace
- the specific string to be replaced
- the value of the replacement string

The following steps demonstrate how to create and use the sample function Replace():

1. Create a new module.
2. Enter the Access Basic code below.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'=====
' Declarations section of the module
'=====
Option Explicit

Function Replace(Valuein, WhatToReplace, Replacevalue)
 Dim Temp as String, P
 Temp = Valuein
 P = Instr(Valuein, WhatToReplace)
 Do While P > 0
 Temp=Left(Valuein,P-1) & Replacevalue &_
 Mid(Valuein,P+Len(WhatToReplace))
 P = Instr(P + Len(WhatToReplace),Valuein, WhatToReplace,1)
 Loop
 Replace = Temp
End Function
```

3. To replace the string "Sample" with the string "Test" in the string "This is a Sample," use the following parameters for the Replace() function:

Valuein: "This is a Sample"  
WhatToReplace: "Sample"  
ReplaceValue: "Test"

The function Replace("This is a Sample","Sample","Test") will return:

This is a Test

#### REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 276-277 and  
page 303

Additional reference words: 1.00 1.10 2.00 substitute  
KBCategory: kbprg  
KBSubcategory: PgmOthr

**INF: How to Replace the Default Input Mask Error Message**  
**Article ID: Q120567**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to substitute your own custom message for the generic error message that occurs when you violate an input mask. The generic error message for input mask violation is:

```
The value you entered isn't appropriate for the input mask
'!\(999") "000\ -0000;;_' specified for this field. Press Esc
to cancel your changes
```

This article assumes you are familiar with creating and using input masks in Microsoft Access.

MORE INFORMATION

=====

You can trap input mask violations and substitute your custom error message for the generic error message in the form's Error event. The following sample Access Basic code demonstrates this method:

```
Sub Form_Error (DataErr As Integer, Response As Integer)
 Const INPUTMASK_VIOLATION = 2279
 If DataErr = INPUTMASK_VIOLATION Then
 MsgBox "There was an input mask violation!"
 Response = DATA_ERRCONTINUE
 End If
End Sub
```

Setting the Response parameter to DATA\_ERRCONTINUE lets you ignore the error and continue without displaying the default error message. You can then supply a custom error message in place of the default.

If you have multiple controls on your form with different input masks, you can use Screen.ActiveControl.Name to get the name of the control causing the error. The following sample Access Basic code demonstrates this method for a form with phone number, social security number, and ZIP code fields:

```
Sub Form_Error (DataErr As Integer, Response As Integer)
 Const INPUTMASK_VIOLATION = 2279
 If DataErr = INPUTMASK_VIOLATION Then
 Select Case Screen.ActiveControl.Name
 Case "Phone"
 Beep
 MsgBox "The phone number you entered is invalid!"
 Case "SSN"
 Beep
 MsgBox "The SSN you entered is invalid!"
 End Select
 End If
End Sub
```

```
 Case "Zip"
 Beep
 MsgBox "The ZIP code you entered is invalid!"
 Case Else
 Beep
 Msg = "An input mask violation occurred in control "
 Msg = Msg & Screen.ActiveControl.Name & "!"
 End Select
 Response = DATA_ERRCONTINUE
 End If
End Sub
```

#### REFERENCES

=====

For more information about using the Error event, search for "Error" then "Error Event" using the Microsoft Access Help menu.

For more information about using the Screen object, search for "Screen" then "Screen Object" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: GnlInpt

## INF: How to Reset Toolbar Options

Article ID: Q120970

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

The toolbars in Microsoft Access are fully customizable. This article describes how to reset the toolbars to their original settings, so that only the toolbars that are appropriate for a given view are displayed.

### MORE INFORMATION

=====

The following example demonstrates a macro that you can use to reset the toolbars to their original settings:

1. Create a macro called Reset Toolbars with the following action and arguments:

| Macro Name | Macro Actions |
|------------|---------------|
|------------|---------------|

-----

|                |             |
|----------------|-------------|
| Reset Toolbars | ShowToolbar |
|----------------|-------------|

| Reset Toolbars Actions |
|------------------------|
|------------------------|

-----

|             |
|-------------|
| ShowToolbar |
|-------------|

|                        |
|------------------------|
| Toolbar Name: Database |
|------------------------|

|                         |
|-------------------------|
| Show: Where Appropriate |
|-------------------------|

2. Click the selector (the box to the left of the ShowToolbar action) to select the ShowToolbar action.
3. From the Edit menu, choose Copy.
4. Choose Paste from the Edit menu 18 times, to create a total of 18 ShowToolbar actions in the macro.
5. Change the Toolbar Name argument of the second through eighteenth ShowToolbar actions to match the following table:

| ShowToolbar Action # | Toolbar Name Argument |
|----------------------|-----------------------|
|----------------------|-----------------------|

-----

|    |                 |
|----|-----------------|
| 2  | Relationships   |
| 3  | Table Design    |
| 4  | Table Datasheet |
| 5  | Query Design    |
| 6  | Query Datasheet |
| 7  | Form Design     |
| 8  | Form View       |
| 9  | Filter/Sort     |
| 10 | Report Design   |

|    |               |
|----|---------------|
| 11 | Print Preview |
| 12 | Toolbox       |
| 13 | Palette       |
| 14 | Macro         |
| 15 | Module        |
| 16 | Microsoft     |
| 17 | Utility 1     |
| 18 | Utility 2     |

NOTE: Setting the Microsoft toolbar to Where Appropriate will cause it to be shown in all views, so you may want to set it to No.

6. Run the macro. All the toolbars will be reset to their original settings.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 24, Macro Basics," pages 605-625

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: GnlGlobl



**INF: How to Retrieve the Path of an Attached MS Access Table**  
**Article ID: Q116146**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes a sample user-defined Access Basic function that you can use to retrieve the path and filename of the originating database for an attached Microsoft Access table. The function uses the attached table's Connect property to get this information.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

The following example demonstrates how to create and use the sample GetAttachedDBName() function:

1. Open the sample database NWIND.MDB.
2. Attach the Examples table from the sample database SOLUTIONS.MDB. (The SOLUTIONS.MDB database file is usually in the SAMPAPPS subdirectory.)
3. Create a new module and enter the following code.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

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

Option Explicit

'=====
```

' The GetAttachedDBName() function requires the name of an  
' attached Microsoft Access table, in quotation marks, as an  
' argument. The function returns the full path of the originating  
' database if successful, or returns 0 if unsuccessful.  
'=====

```
Function GetAttachedDBName (TableName As String)
 Dim db As Database, Ret
 On Error GoTo DBNameErr
 Set db = DBEngine.Workspaces(0).Databases(0)
 Ret = db.TableDefs(TableName).Connect
 GetAttachedDBName = Right(Ret, Len(Ret) - (InStr_
```

```
 (1, Ret, "DATABASE=") + 8))
Exit Function
DBNameErr:
 GetAttachedDBName = 0
End Function
```

4. From the View menu, choose Immediate Window.

5. In the Immediate window, type the following line and then press ENTER:

```
? GetAttachedDBName("Examples")
```

The path of the attached table's originating database will be displayed in the Immediate window.

#### REFERENCES

=====

For more information about the Connect property, search for "Connect" then "Connect Property (Data Access)" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbinterop

KBSubcategory: IsmIea

**INF: How to Retrieve Windows For Workgroups User Information**  
**Article ID: Q101676**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Using the Microsoft Windows for Workgroups API, you can retrieve information, such as user name, workgroup, domain, and computer name, about the currently running computer.

MORE INFORMATION

=====

Windows for Workgroups comes with a function called NetWkstaGetInfo() that returns information about a computer. This function is especially useful for multiuser applications.

The following procedure explains how to use Access Basic and several Windows API calls to access the NetWkstaGetInfo() function:

1. Create a new module containing the following declarations and functions:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

NOTE: You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.

NOTE: Make sure to choose the Compile All command from the Run menu to verify that you do not receive any compilation errors.

```
Option Explicit
Declare Function NetWkstaGetInfo% Lib "NetAPI" (ByVal lServer&, _
 ByVal sLevel%, ByVal pbBuffer&, ByVal cbBuffer%, pcbTotalAvail%)
Declare Function GlobalAlloc% Lib "Kernel" (ByVal fFlags%, _
 ByVal nSize&)
Declare Function GlobalLock& Lib "Kernel" (ByVal hMem%)
Declare Function GlobalUnlock% Lib "Kernel" (ByVal hMem%)
Declare Function GlobalFree% Lib "Kernel" (ByVal hMem%)
Declare Sub lstrcpy Lib "Kernel" (ByVal dest As Any, _
 ByVal src As Any)
Declare Sub hmemcpy Lib "Kernel" (ByVal dest As Any, _
 ByVal src As Any, ByVal size As Long)

Function GetBinInt (sobj As String, off As Integer)
 If (Asc(Mid$(sobj, off + 1, 1)) < 128) Then
 GetBinInt = Asc(Mid$(sobj, off, 1)) +_

```

```

 Asc(Mid$(sobj, off + 1, 1)) * 256
Else
 GetBinInt = ((&HFF - Asc(Mid$(sobj, off + 1, 1))) * 256) - _
 Asc(Mid$(sobj, off, 1))
End If
End Function

Function GetWorkstationInfo (sComputer As String, _
 sUserName As String, sWorkgroup As String, sLogonDomain As String)
 Dim nRetSize As Integer, nStruct, ret As Integer, dummy As Integer
 Dim hMem As Integer, lpMem As Long, sTemp As String

 ' Get the size of the return structure.
 ret = NetWkstaGetInfo(0&, 10, 0&, 0, nRetSize)
 If (ret <> 0) And (ret <> 2123) Then
 GetWorkstationInfo = False
 Exit Function
 End If

 ' Allocate memory for the structure.
 hMem = GlobalAlloc(0, CLng(nRetSize))
 nStruct = nRetSize
 If (hMem <> 0) Then
 lpMem = GlobalLock(hMem)

 ' Read workstation information into structure.
 ret = NetWkstaGetInfo(0&, 10, lpMem, nRetSize, nRetSize)
 If (ret = 0) Then
 sTemp = Space(100)
 hmemcpy sTemp, lpMem, nStruct

 ' Retrieve the computer name.
 sComputer = Space(100)
 lpMem = CLng(GetBinInt(sTemp, 1)) + _
 (CLng(GetBinInt(sTemp, 3)) * CLng(&H10000))
 lstrcpy sComputer, lpMem
 sComputer = Mid(sComputer, 1, InStr(sComputer, Chr(0)) - 1)

 ' Retrieve the user name.
 sUserName = Space(100)
 lpMem = CLng(GetBinInt(sTemp, 5)) + _
 (CLng(GetBinInt(sTemp, 7)) * &H10000)
 lstrcpy sUserName, lpMem
 sUserName = Mid(sUserName, 1, InStr(sUserName, Chr(0)) - 1)

 ' Retrieve the workgroup name.
 sWorkgroup = Space(100)
 lpMem = CLng(GetBinInt(sTemp, 9)) + _
 (CLng(GetBinInt(sTemp, 11)) * &H10000)
 lstrcpy sWorkgroup, lpMem
 sWorkgroup = Mid(sWorkgroup, 1, InStr(sWorkgroup, _
 Chr(0)) - 1)

 ' Retrieve the logon domain name.
 sLogonDomain = Space(100)
 lpMem = CLng(GetBinInt(sTemp, 15)) + _
 (CLng(GetBinInt(sTemp, 17)) * &H10000)

```

```

 lstrcpy sLogonDomain, lpMem
 sLogonDomain = Mid(sLogonDomain, 1, InStr(sLogonDomain, _
 Chr(0)) - 1)
 End If
 ' Free the memory allocated.
 dummy = GlobalUnlock(hMem)
 dummy = GlobalFree(hMem)
Else
 ret = -1
End If
GetWorkstationInfo = IIf(ret = 0, True, False)
End Function

```

2. Test this function by creating the following Access Basic function:

```

Sub ShowInfo ()
 Dim a$, b$, c$, d$

 If GetWorkstationInfo(a$, b$, c$, d$) Then
 MsgBox a$ & " " & b$ & " " & c$ & " " & d$
 Else
 MsgBox "Unable to get information."
 End If
End Sub

```

3. In the Immediate window, type the following, and then press ENTER:

```
ShowInfo
```

Either a dialog box displaying information about the computer, or, if you are running this function on a non-Windows for Workgroups computer, an error message stating "Unable to get information" will be displayed.

Additional reference words: 1.00 1.10 2.00 user name multi-user networks

KBCategory: kbprg

KBSubcategory: PgmApi

## **INF: How to Run Access Basic Functions with DDE**

**Article ID: Q109397**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Microsoft Access does not have any facility for running Access Basic functions as a dynamic data exchange (DDE) server. However, Access Basic functions can be run from a DDE client application if they are contained in an SQL statement used to initiate a DDE conversation with Microsoft Access using the SQL topic.

This article describes how to run Access Basic functions from a DDE client application.

### MORE INFORMATION

=====

A DDE client application such as Microsoft Excel or Microsoft Word for Windows can use DDE to run commands in Microsoft Access as a DDE server. When it is used as a DDE server, Microsoft Access recognizes any of the following as a valid command:

- The name of a macro in the currently open database
- Any action that you can run in Access Basic using the DoCmd statement
- The OpenDatabase and CloseDatabase commands, which are used only for DDE operations

Note that Microsoft Access does not have DDE server functionality for running Access Basic functions.

How to run Access Basic functions from a DDE client application

-----

The following examples demonstrate how to run Access Basic functions from DDE client applications. In both examples below, the MyFuncnt() function is run against the None table in the sample database NWIND.MDB (note that the MyFuncnt() function and the None table do not exist in the NWIND database; they are simply names used for illustration):

This is the DDE command issued from Microsoft Excel:

```
chan=INITIATE("MSACCESS","NWIND;SQL SELECT MyFuncnt() FROM None;")
```

This is the DDE command issued from Word for Windows:

```
chan=DDEInitiate("MSACCESS", "NWIND;SQL SELECT MyFuncnt() FROM None;")
```

Note that you can name any table that exists in the database in the From

clause.

The following example demonstrates how to add a new customer to the Customers table in the NWIND database by running an Access Basic function from a DDE client application:

1. Open the sample database NWIND.MDB and create a new table called None with one field of any data type.
2. Open or create a module and enter the following sample Access Basic function:

```
'Declarations section
Option Explicit

Function AddNewCust$ (CustomerID$, CompanyName$)
 Dim MyDB As Database, MyTable As Table
 Set MyDB = CurrentDB()

 Set MyTable = MyDB.OpenTable("Customers") ' Open the table

 MyTable.AddNew ' Prepare new record
 MyTable("Customer ID") = CustomerID$ ' Set record key
 MyTable("Company Name") = CompanyName$ ' Set company name
 MyTable.Update ' Save changes
 MyTable.Close ' Close the table
End Function
```

3. In the DDE client application, initiate a DDE conversation with MSACCESS as the application and an SQL statement that calls the AddNewCust() function you created in step 2 as the topic.

The following example demonstrates how to run the AddNewCust() function from Microsoft Excel:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
AddNewCustomer
chan=INITIATE("MSACCESS","NWIND.MDB;SQL SELECT _
 AddNewCust$("JOHNJ","John's Place") FROM None;")
=TERMINATE(chan)
=RETURN()
```

The following example demonstrates how to run the AddNewCust() function from Word for Windows:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Sub MAIN
 DDETerminateAll
 qt$ = Chr$(34)
 Funct$ = "AddNewCust$(" + qt$ + "JOHNJ" + qt$ + "," + qt$ + _
 "John's Place" + qt$ + ")"
```

```
Chan = DDEInitiate("MSACCESS", _
 "NWIND;SQL SELECT " + Funct$ + "FROM None;")
DDETerminate Chan
End Sub
```

#### REFERENCES

=====

For more information on using Microsoft Access as a DDE server, search for "DDE" then "Using Microsoft Access As a DDE Server" using the Microsoft Access Help menu.

Microsoft Access "Language Reference," version 1.0, pages 118-124

README.TXT file in Microsoft Access version 1.0

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpDde



## **INF: How to Search for Characters or Strings in Records**

**Article ID: Q100135**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article demonstrates three methods of searching for and identifying records containing a particular character or string. In the example below, we search for an asterisk (\*) character.

### MORE INFORMATION

=====

Say, for example, you have a table of names. Some of the names have an asterisk character embedded in them, and you would like a list of all the names containing asterisks.

#### First Method--Using the InStr() Function

-----

One solution is to use the InStr() function. The InStr() function returns the position of the first occurrence of a string within another string. The example below uses the InStr() function to determine whether or not the asterisk character exists in the names in the table. If it does, it becomes a member of the query's dynaset.

1. Start Microsoft Access, open a database, and create a new table called Nametest.
2. Create a text field called Names and enter the following data into the table:

```
Names

*Pat
Pat*
Tom
Ri*ck
Harry
```

3. Create a new query based on the Nametest table.
4. Drag the Names field from the field list box into the query grid.
5. In the Field row of the query grid, next to the Names column, type in the expression:

```
NewField: InStr(1,[Names],"*")
```

6. In the Criteria row of the same column, enter:

>0

7. Uncheck the Show check box of that column.
8. Run the query by choosing Run from the Query menu.

You should get the results:

```
*Pat
Pat*
Ri*ck
```

This method can be used to obtain a dynaset containing any sequence of characters -- just change the asterisk in the expression above to match the string or character you want to search for.

Second Method--Using the Like Operator

-----

The Like Operator returns a dynaset that matches a pattern. The example below uses the Like Operator to determine whether or not the asterisk character exists in the names in the table. If it does, it becomes a member of the query's dynaset.

1. Start Microsoft Access, open a database, and create a new table called Nametest.
2. Create a text field called Names and enter the following data into the table:

```
Names

*Pat
Pat*
Tom
Ri*ck
Harry
```

3. Create a new query based on the Nametest table.
4. Drag the Names field from the field list box onto the query grid.
5. In the Criteria row of the same column, enter:

```
Like "*[*]*"
```

The first and last \* are wildcards, much like the MS-DOS asterisk wildcard. The [\*] tells Microsoft Access to look for the character asterisk and not to use it as a wildcard.

6. Run the query. You should get the results:

```
*Pat
Pat*
Ri*ck
```

This is another method that can be used to obtain a dynaset containing

any sequence of characters -- just change the [\*] in the criteria above to match the string or character you want to search for.

If you are searching for a character that Microsoft Access interprets as a wildcard, such as \*#?[]!-, it must be inside square brackets. For example:

\* should be entered as [\*]  
# should be entered as [#]  
? should be entered as [?]  
[ should be entered as [[  
] should be entered as []]  
! should be entered as [!]  
- should be entered as [-]

### Third Method--Using the Find Command

-----

The Find command searches for the data you specify in the current table or dynaset. The find command is available in the Datasheet view of a table, query, or form and in the Form view of a form. The find dialog box remains open after each search. This enables you to find as many occurrences as you want without having to choose the find command repeatedly. This method finds each name with an asterisk one at a time.

1. Start Microsoft Access, open a database, and create a new table called Nametest.
2. Create a text field called Names and enter the following data into the table:

```
Names

*Pat
Pat*
Tom
Ri*ck
Harry
```

3. Choose the Edit menu and the Find option.
4. Type in [\*] and choose Any Part of Field for Where:.
5. Choose the Find First button and you should get the result:

```
*Pat
```

6. Choose the Find Next button and you should get the result:

```
Pat*
```

7. Choose the Find Next button again and you should get the result:

```
Ri*ck
```

This method can also be used to find any sequence of characters -- just change the [\*] in the Find What: to match the string or character

you want to search for. Please see method two regarding searching for a character that Microsoft Access interprets as a wildcard, such as \*#?[]!-.

#### REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 259, 279-280

Microsoft Access "User's Guide," version 1.0, Chapter 15, "Finding and Sorting Data," pages 370-375

Microsoft Access "User's Guide," version 1.1, Chapter 15, "Finding and Sorting Data," pages 374-379

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryOthr

**INF: How to Select Controls on Subforms with SetFocus Method**  
**Article ID: Q113548**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

You can use the new SetFocus method for forms and form controls in Microsoft Access version 2.0 to select a field or control on an open form, form datasheet, table datasheet, or query dynaset, or to select other forms.

This article describes how to use the SetFocus method to select controls on subforms and nested subforms, and how to select controls on a main form from a control on a subform.

MORE INFORMATION

=====

How to Select a Control on a Subform from a Main Form

-----

To select a control on a subform from a main form, do the following:

1. Select the subform control.
2. Select the appropriate control on the subform.

The following example demonstrates how to select a control on a subform from a main form:

1. Open the sample database NWIND.MDB.
2. Create a new, blank form based on the Customers table.
3. From the View menu, choose Field List to display the field list. Drag the Company Name and Contact Name fields to the detail section of the form.
4. Drag the Orders form from the Database window onto the detail section of the new form. Microsoft Access will embed the Orders form as a subform, linking the Orders form to the new form on the Customer ID field. Note that the Orders form contains its own subform to show order details.
5. Create a command button on the form with the following properties:

Name: GotoEmployeeID  
Caption: Employee ID  
OnClick: [Event Procedure]

To create the event procedure, choose the Build button and then select

the Code Builder. Add the following code to the form module, and then close the module:

```
' Select the Orders subform control.
Me![Orders].SetFocus

' Select the Employee ID control on the Orders subform.
Me![Orders].Form![Employee ID].SetFocus
```

6. Save the new form with the name Customer Orders.
7. View the form in Form view, and then choose the Employee ID button.

#### How to Select a Control on a Nested Subform from a Main Form

---

To select a control on a nested subform (a subform of a subform), do the following:

1. Select the subform control.
2. Select the nested subform control.
3. Select the appropriate control on the nested subform.

The following example demonstrates how to select a control on a nested subform from a main form:

1. Continuing the example above, add the following command button to the Customer Orders form:

```
Name: GotoQuantity
Caption: Quantity
OnClick: [Event Procedure]
```

Create the following code for the OnClick event procedure:

```
' Select the Orders subform control.
Me![Orders].SetFocus

' Select the nested Orders Subform subform control.
Me![Orders].Form![Orders Subform].SetFocus

' Select the Quantity field on the nested subform.
Me![Orders].Form![Orders Subform].Form![Quantity].SetFocus
```

2. View the form in Form view, and then choose the Quantity button.

#### How to Select a Control on a Main Form from a Subform

---

To select a control on a main form from a subform, do the following:

1. Select the main form.
2. Select the appropriate control on the main form.

The following example demonstrates how to select a control on a main form from a subform:

1. Continuing the example above, open the Orders form in Design view. Add the following new command button to the form:

```
Name: GotoContact
Caption: Contact
OnClick: [Event Procedure]
```

Create the following code for the OnClick event procedure:

```
' Select the main form.
Forms![Customer Orders].SetFocus

' Select the Contact Name control on the main form.
Forms![Customer Orders]![Contact Name].SetFocus
```

2. Save and then close the Orders form.
3. Open the Customer Orders form in Form view.
4. Choose the Contact button on the Orders subform.

How to Select a Control on a Subform from a Nested Subform

---

To select a control on a subform from a nested subform, do the following:

1. Select the main form.
2. Select the subform control.
3. Select the appropriate control on the subform.

The following example demonstrates how to select a control on a subform from a nested subform:

1. Continuing the example above, open the Orders Subform form in Design view.
2. Modify the Discount field's OnExit property to be:

```
OnExit: [Event Procedure]
```

Create the following code for the OnExit event procedure:

```
' Select the main form.
Forms![Customer Orders].SetFocus

' Select the Orders subform control.
Forms![Customer Orders]![Orders].SetFocus

' Select the Employee ID control on the Orders subform.
Forms![Customer Orders]![Orders].Form![Employee ID].SetFocus
```

2. Save and close the Orders Subform form.

3. Open the Customer Orders form in Form view. Use the mouse to select the Discount field in the nested Orders Subform subform. Press TAB to exit the Discount field.

#### REFERENCES

=====

For more information about the SetFocus method, search for "SetFocus" then "SetFocus Method" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsOthr



## INF: How to Send Information to the Clipboard

Article ID: Q96900

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

There is no Microsoft Access command to send information to the Clipboard. To post information to the Clipboard, you need to define an Access Basic function that calls several Windows API functions. This article defines a function that copies text to the Clipboard.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

NOTE: In version 2.0, you can use the OLE\_COPY action to copy an OLE field to the clipboard. This article does not address this case.

### MORE INFORMATION

=====

#### Notes

-----

- In the following sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.
- You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive the duplicate procedure name error, remove or comment the Declare statement from your code.

To call the function, you can make a string assignment to the returned value, such as:

```
X = ClipBoard_SetData("This string will go to the Clipboard!")
```

```
'=====
' General Declarations
'=====
```

```
Option Compare Database 'Use database order for string comparisons
Option Explicit
```

```
Declare Function kb_OpenClipBoard% Lib "User" _
 Alias "OpenClipBoard" (_
 ByVal hwnd%)
```

```

Declare Function kb_GlobalAlloc% Lib "Kernel" _
 Alias "GlobalAlloc" (ByVal wFlags%, ByVal wBytes&)
Declare Function kb_GlobalLock& Lib "Kernel" _
 Alias "GlobalLock" (ByVal hMem%)
Declare Function kb_lstrcpy& Lib "Kernel" _
 Alias "lstrcpy" (ByVal lpString1 As Any, _
 ByVal lpString2 As Any)
Declare Function kb_GlobalUnlock% Lib "Kernel" _
 Alias "GlobalUnlock" (ByVal hMem%)
Declare Function kb_CloseClipboard% Lib "User" Alias "CloseClipboard"()
Declare Function kb_EmptyClipboard% Lib "USER" _
 Alias "EmptyClipboard" ()
Declare Function kb_SetClipboardData% Lib "User" _
 Alias "SetClipboardData" (ByVal wFormat%, _
 ByVal hMem%)

```

```

Global Const GHND = &H42
Global Const CF_TEXT = 1
Global Const MAXSIZE = 4096

```

```

Function Clipboard_SetData (MyString$)
 Dim hGlobalMemory%, lpGlobalMemory&, hClipMemory%, X%

 '-----
 ' Allocate moveable global memory.
 '-----
 hGlobalMemory% = kb_GlobalAlloc(GHND, Len(MyString$) + 1)

 '-----
 ' Lock the block to get a far pointer
 ' to this memory.
 '-----
 lpGlobalMemory& = kb_GlobalLock(hGlobalMemory%)

 '-----
 ' Copy the string to this global memory.
 '-----
 lpGlobalMemory& = kb_lstrcpy(lpGlobalMemory&, MyString$)

 '-----
 ' Unlock the memory.
 '-----
 If kb_GlobalUnlock(hGlobalMemory%) <> 0 Then
 MsgBox "Could not unlock memory location. Copy aborted."
 GoTo OutOfHere2
 End If

 '-----
 ' Open the Clipboard to copy data to.
 '-----
 If kb_OpenClipboard(0&) = 0 Then
 MsgBox "Could not open the Clipboard. Copy aborted."
 Exit Function
 End If

 '-----
 ' Clear the Clipboard.

```

```
'-----
X% = kb_EmptyClipboard()

'-----
' Copy the data to the Clipboard.
'-----
hClipMemory% = kb_SetClipboardData(CF_TEXT, hGlobalMemory%)
```

OutOfHere2:

```
If kb_CloseClipboard() = 0 Then
 MsgBox "Could not close Clipboard."
End If
```

End Function

#### REFERENCES

=====

For more information on how to retrieve information from the Clipboard,  
query on the following words here in the Microsoft Knowledge Base:

Clipboard\_GetData and GetClipboardData

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

## **INF: How to Shell to a Windowed MS-DOS Application**

**Article ID: Q116384**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

You can use the Shell() function to launch a Microsoft Windows or MS-DOS application from within Microsoft Access. The function accepts a windowstyle argument that specifies the style of window the application will run in. The windowstyle argument applies only to Microsoft Windows applications. To launch an MS-DOS application in a window, create a program information file (PIF), and specify the PIF filename for the Shell() function's commandstring argument.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The following example demonstrates how to use the Shell() function to launch the MS-DOS EDIT.COM editor in a window from Microsoft Access:

1. In Program Manager, start PIF Editor. From the File menu, choose New.
2. In the Program Filename box, type "C:\DOS\EDIT.COM" (without quotation marks). If your copy of EDIT.COM is in a directory other than C:\DOS, substitute the appropriate directory name.
3. In the Start-up Directory box, type "C:\DOS" (without quotation marks).
4. In the Display Usage area, select the Windowed option button.
5. From the File menu, choose Save As. In the File Name box, type "C:\DOS\EDIT.PIF" (without quotation marks).
6. Start Microsoft Access and open any database. Create a new, blank form.
7. Add a command button to the form.
8. Set the command button's OnClick property to the following event procedure:

```
Dim Ret
Ret = Shell("C:\DOS\EDIT.PIF", 1)
```

9. View the form in Form view, and then click the command button.

REFERENCES

=====

Microsoft Access "Language Reference," version 2.0, page 568

Additional reference words: 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: How to Shrink and Grow a Rectangle in a Report**  
**Article ID: Q124642**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

In a report, a text box can expand vertically if you set its CanGrow and CanShrink properties to Yes. Rectangles and lines, however, do not have these properties and cannot expand. To create a rectangle around a control that may grow or shrink, you should draw the rectangle using the Line method.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The following example demonstrates how to use the Line method to draw a rectangle around a memo field of variable size:

1. Open the sample database NWIND.MDB.
2. Create a blank new report based on the Employees table.
3. Set the report's Width property to 5 in.
4. If the field list is not displayed, choose Field List from the View menu to display it.
5. Drag the Notes field from the field list to the report's Detail section.
6. Delete the Notes text box's label by selecting the label and then pressing the DEL key.
7. Set the Notes text box's properties as follows:

Name: Notes  
ControlSource: Notes  
CanGrow: Yes  
CanShrink: Yes  
Left: 0.2 in  
Top: 0.166  
Width: 4.6042 in  
Height: 0.166 in

8. Select the horizontal Detail bar to select the detail section of the report. Set the detail section's Height property to 0.5 in.

9. Set the detail section's OnPrint property to the following event procedure:

```
Dim X1 As Single, Y1 As Single
Dim X2 As Single, Y2 As Single
Dim Offset As Single

' Specify unit of measurement for coordinates on a page...
Me.ScaleMode = 1 ' ...in twips (1440 twips = 1 inch).

' Define an offset of 1/8 inch from the text box to the rectangle.
Offset = 1440 / 8

' X and Y coordinates for the top left corner of the box.
X1 = Me![Notes].Left - Offset
Y1 = Me![Notes].Top - Offset

' X and Y coordinates for the bottom right corner of the box.
X2 = Me![Notes].Left + Me![Notes].Width + Offset
Y2 = Me![Notes].Top + Me![Notes].Height + Offset

Me.DrawWidth = 3 ' Width of the line (in pixels).
Color = RGB(0, 0, 0) ' Use black line color.

' Draw the rectangle with the Line method.
Me.Line (X1, Y1)-(X2, Y2), Color, B
```

10. In the Object box on the toolbar, select "(general)" and then type the following line in the module's Declarations section:

```
Option Explicit
```

11. From the Run menu, choose Compile Loaded Modules to compile the code.
12. Close the module and then preview the report. Note that the Notes field and the box that surrounds it expand and contract together.

#### REFERENCES

=====

For more information about the Line method, search for "Line" then "Line Method" using the Microsoft Access Help menu.

Additional reference words: 2.00 ab expand adjust can shrink can grow  
KBCategory: kbusage  
KBSubcategory: FmrHowto

**INF: How to Simulate Multiple Record Selections in a Form**  
**Article ID: Q120275**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

In Microsoft Access, you can select multiple records in a form using the record selector at the left side of the form. However, there is no built-in method to determine from a macro or Access Basic code which records have been selected.

This article describes how to simulate the record selector using a Yes/No field, so that you can make multiple, noncontiguous selections, which you cannot do using the standard record selector. This article also describes how to create buttons to select all the records, unselect the records, and delete the selected records. This technique will work in either a single-user or a multiuser environment.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

This method of simulating the record selector uses a Yes/No field in the form's underlying table. You can place the Yes/No field on the form bound to either a toggle button (to look like the built-in record selector) or a check box or option button.

You can determine which records are selected using a select query to find all the records where the Yes/No field is set to Yes, or by checking the form's Recordset to see which records have the Yes/No field set to Yes.

This method is simple and works well in a single-user environment where the table will not be shared. In a multiuser environment, the Yes/No field must be placed in a separate selection table unique to each user. The selection table should include the key fields from the original table in addition to the Yes/No field. The form can then be based on a query that joins the original table with the selection table on the key fields. There must be one record in the selection table for each record in the original table in order to have a one-to-many relationship.

The selection table must be unique for each user on the system, and cannot be shared with other users on the system. In multiuser environments, you should consider dividing the database into two databases, with one database (the application database) containing all the forms, reports, macros, and modules, and the other database (the data database) containing all the tables. You can then attach the tables in the data database to the application database. In such a two-database system, you can put the



selection table in the local application database to be used only by the individual user.

#### How to Create a Multiple Selection, Multiuser Customers Form

-----

The following example demonstrates how to create a toggle selection button on the Customers form in the sample database NWIND.MDB, using a separate selection table to maintain a list of selected records for use in a multiuser environment.

Note that to run best in a multiuser environment, the Customers table should be stored in a different, shared database attached to the NWIND database. For this example, assume that the table is an attached, shared, Microsoft Access table.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the sample database NWIND.MDB.
2. Create the following new table and then save it as Customers Selected:

Table: Customers Selected

-----  
Field Name: Customer ID

Data Type: Text

Field Size: 5

Field Name: Selected

Data Type: Yes/No

Indexes: Customers Selected

-----  
Index Name: PrimaryKey

Field Name: Customer ID

3. Create a new select query based on the Customers and Customers Selected tables. Join the tables on the Customer ID field and include all the fields from the Customers table and the Selected field from the Customers Selected table in the query grid. Save the query as Customer Query.
4. Open the Customers form in Design view and set the form's RecordSource property to the Customer Query query.
5. Create a toggle button with the following properties on the form:  
  
ControlSource: Selected (the name of the Yes/No field)  
DefaultValue: False
6. Set the form's OnOpen property to the following event procedure.

NOTE: In the following sample code, an underscore ( \_ ) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```

Sub Form_Open (Cancel As Integer)
 Dim DB As Database
 Set DB = CurrentDB()

 ' Clean out the Customers Selected table.
 DB.Execute "DELETE * FROM [Customers Selected]"

 ' Create a matching record in the Customers Selected table
 ' for each record in the Customers table.
 DB.Execute "INSERT INTO [Customers Selected] ([Customer ID]) _
 SELECT [Customer ID] FROM [Customers]"

 ' Requery the form to make sure it sees the new Customers
 ' Selected records.
 Me.Requery

End Sub

```

- View the Customers form in Form view. When you open the form, the event procedure specified in the OnOpen property setting will run, ensuring that the Customers Selected table contains a record with the Selected field set to No for each record in the Customers table. You can move among the records, using the toggle button to mark records for selection. If you create a new record, the DefaultValue property setting for the toggle button will cause a new record to be created in the Customers Selected table.

To determine which records are selected, you can create a simple select query to find all the records in the Customers Selected table with the Selected field set to Yes.

How to Create Buttons to Select All the Records, Clear the Selection, and Delete the Selected Records

---

You can create buttons on the form so that the form's users can easily select all the records in the form or clear the current selection. You can also create a button to delete all the selected records. To create these buttons, add three command buttons with the following properties to the Customers form:

Command Button 1

---

Name: btnSelectAll  
Caption: Select All  
OnClick: [Event Procedure]

```

Sub btnSelectAll_Click ()
 Dim DB As Database
 Set DB = CurrentDB()

```

```

On Error Goto Err_btnSelectAll_Click

```

```

' Update the Selected field for all customers to Yes (True).
DB.Execute "UPDATE [Customers Query] SET [Selected] = True;"

```

```

' Refresh the form to display the changes.

```

```
DoCmd DoMenuItem A_FORMBAR, A_RECORDSMENU, A_REFRESH
```

```
Bye_btnSelectAll_Click:
Exit Sub
Err_btnSelectAll_Click:
Beep: MsgBox Error$, 48
Resume Bye_btnSelectAll_Click:
End Sub
```

Command Button 2

---

```
Name: btnUnselectAll
Caption: Unselect All
OnClick: [Event Procedure]
```

```
Sub btnUnselectAll_Click ()
Dim DB As Database
Set DB = CurrentDB()

On Error Goto Err_btnUnselectAll_Click

' Update the Selected field for all customers to No (False).
DB.Execute "UPDATE [Customers Query] SET [Selected] = False;"

' Refresh the form to display the changes.
DoCmd DoMenuItem A_FORMBAR, A_RECORDSMENU, A_REFRESH

Bye_btnUnselectAll_Click:
Exit Sub
Err_btnUnselectAll_Click:
Beep: MsgBox Error$, 48
Resume Bye_btnUnselectAll_Click:
End Sub
```

Command Button 3

---

```
Name: btnDeleteSelected
Caption: Delete Selected
OnClick: [Event Procedure]
```

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Sub btnDeleteSelected_Click ()
Dim DB As Database
Set DB = CurrentDB()

On Error Goto Err_btnDeleteSelected_Click

' If the current record has been edited and is selected, save it.
If Me.Dirty And Me![Selected] Then
DoCmd DoMenuItem A_FORMBAR, A_FILE, A_SAVERECORD
End If

' Delete all customers where the Selected field is marked.
```

```

DB.Execute "DELETE Customers.*, [Customers Selected].Selected _
FROM Customers INNER JOIN [Customers Selected] ON _
Customers.[Customer ID] = [Customers Selected].[Customer ID] _
WHERE [Customers Selected].Selected=True"

' Refresh the form to display the changes.
DoCmd DoMenuItem A_FORMBAR, A_RECORDSMENU, A_REFRESH

Bye_btnDeleteSelected_Click:
Exit Sub
Err_btnDeleteSelected_Click:
Beep: MsgBox Error$, 48
Resume Bye_btnDeleteSelected_Click:
End Sub

```

NOTE: When you use the Delete Selected button, customers who currently have orders in the NWIND database will not be deleted, due to referential integrity. You can modify the relationship between the Customers and the Orders tables to include cascading deletes, so that the removal of a customer will cause the removal of all of that customer's orders. To set up cascading deletes for the Customers and the Orders tables, follow these steps:

1. Select the Database window and then choose Relationships from the Edit menu.
2. Double-click the join line between the Customers and the Orders tables.
3. Select the Cascade Delete Related Records check box.
4. Choose OK.

Additional reference words: 2.00  
KBCategory: kbusage  
KBSubcategory: FmsOthr

**INF: How to Skip Used Mailing Labels and Print Duplicates**  
**Article ID: Q95806**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to print multiple copies of the same mailing label, and how to use a partially used page where only some of the labels are available.

To make use of this article, you need to be familiar with Access Basic and know how to design Microsoft Access reports.

MORE INFORMATION

=====

Printing Multiple Copies of the Same Label

-----

When you choose Print from the File menu, you can choose to print multiple copies of the same report. But when you try to print a single mailing label 20 times, Microsoft Access prints one label on each of 20 pages.

On a dot matrix printer, using single column labels, you can work around this problem by defining each label as a separate page. However, you cannot use this method for laser printers or multiple-column labels. To solve this problem, use the step-by-step procedure described below.

Using Labels that Would Otherwise Be Wasted

-----

After printing labels, you usually end up with a partially used last page. There is no built-in mechanism in Microsoft Access to use the remaining labels on a partially used page. Microsoft Access always starts on a new page. On a dot matrix printer, you can adjust the top of form manually. But you cannot do that on laser printers. To solve this problem, use the step-by-step procedure described below.

Step-by-Step Procedure to Solve Both Problems

-----

The Microsoft Access report generator provides powerful hooks that allow control over the finished product. By calling a function from the OnFormat property of the report's detail section, you can alter the MoveLayout, NextRecord, and PrintSection properties to leave blank spaces or print multiple copies on the same page.

The code listed below is generic. You can attach it to any Mailing Label report to print multiple copies and to skip used labels if needed. To use the example, you need to have a mailing label report called MyLabels.

1. Create a new module, and place the following lines in the declaration section:

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

Option Compare Database
Option Explicit

Dim LabelBlanks&
Dim LabelCopies&
Dim BlankCount&
Dim CopyCount&

'=====
```

```
 ' The following function will cause an inputbox to
 ' display when the report is run that prompts the user
 ' for the number of used labels to skip and how many
 ' copies of each label should be printed.
'=====
```

```
Function LabelSetup ()
 LabelBlanks& = Val(InputBox$("Enter Number of blank labels to skip"))
 LabelCopies& = Val(InputBox$("Enter Number of Copies to Print"))
 If LabelBlanks& < 0 Then LabelBlanks& = 0
 If LabelCopies& < 1 Then LabelCopies& = 1
End Function
```

```
'=====
```

```
 ' The following function sets the variables to a zero
'=====
```

```
Function LabelInitialize ()
 BlankCount& = 0
 CopyCount& = 0
End Function
```

```
'=====
```

```
 ' The following function is the main part of this code
 ' that allows the labels to print as the user desires.
'=====
```

```
Function LabelLayout (R As Report)
 If BlankCount& < LabelBlanks& Then
 R.NextRecord = False
 R.PrintSection = False
 BlankCount& = BlankCount& + 1
 Else
 If CopyCount& < (LabelCopies& - 1) Then
 R.NextRecord = False
 CopyCount& = CopyCount& + 1
 Else
 CopyCount& = 0
 End If
 End If
End Function
```

2. Open the report named MyLabels in Design view and add the following line to the OnPrint property of the Detail section:

```
=LabelLayout(Reports![MyLabels])
```

3. Add the following line to the OnOpen property of the MyLabels Report:

```
=LabelSetup()
```

4. Although typically labels do not have a report header, choose Report Hdr/Ftr from the Layout menu to add a report header and footer. Then Add the following line to the OnFormat property of the report header:

```
=LabelInitialize()
```

5. Set the Height property for both the report header and report footer to 0.

When you print the report, the report calls the LabelSetup() function, which first asks you to enter the number of used labels to skip on the first page (BlankCount) and then asks how many of each label you want printed (CopyCount).

When the report header is formatted, it calls the LabelInitialize() function, so when you switch from preview to print, the BlankCount and CopyCount fields are set to zero. As each label is formatted, the LabelLayout() function adjusts the NextRecord and MoveLayout properties to skip used labels and print the desired duplicates.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptLabel

**INF: How to Specify a Custom Starting Page Number for a Report**  
**Article ID: Q100156**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

In Microsoft Access reports, page numbering always starts with the number 1. However, you may want your report to have page numbering that starts with some number other than 1. This article describes two methods of defining custom starting page numbers. Both methods provide custom page numbers; however, the second method uses Access Basic and can therefore include error checking. This method is more complex, but avoids the possibility that a nonnumeric value will be entered when you are prompted for a starting number.

MORE INFORMATION

=====

Method 1

-----

In Design view of any report with page numbering, change the ControlSource property of the text box for page numbering from

=Page

to

=Page + [Enter a Starting Page Number] -1

This causes Microsoft Access to prompt you to enter a starting page number each time the report is previewed or printed.

For example, in the sample database NWIND.MDB, open the "Alphabetical List of Products" report in Design view. On the right side of the page footer, there is a text box with a ControlSource property of "=Page". Change this property of the text box to "=Page + [Enter a Starting Page Number] -1".

Method 2

-----

This method assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.



1. Open a new module or a previously created module and enter the following code:

```
'*****
'Declarations Section of Module
'*****
Option Explicit
Global PageChoice As Integer

'=====
```

'Create the following GetPageChoice() function in the Module  
'=====

'This function is called in the OnOpen property of the Report.  
Function GetPageChoice ()  
    Dim choice As String  
    Do  
        choice = InputBox("Enter a Starting Page Number:", "\_"  
                          Number Report", "1")  
        If Not (IsNumeric(choice)) Then  
            MsgBox "Value Entered is not a Number."  
        End If  
        Loop While Not (IsNumeric(choice))  
        PageChoice = CInt(choice)  
End Function

'=====

'Create the following ReturnPageChoice() function in the Module  
'=====

'This function is called by the text box that will contain the  
' pagenumber.  
Function ReturnPageChoice (pgnumber As Integer)  
    ReturnPageChoice = PageChoice + pgnumber - 1  
End Function

2. Open your report in Design view.
3. From the Edit menu choose Select Report, then from the View menu choose Properties to display the property sheet. Change the OnOpen property to:

```
OnOpen: =GetPageChoice()
```

4. In the page footer of the report create a text box with the following properties:

```
Text Box: Text1

Name: PageNumber (in version 1.x, ControlName: PageNumber)
ControlSource: =ReturnPageChoice(Page)
```

When you preview or print the report, you will be prompted for a starting page number. Each page will be numbered consecutively starting with this number.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 18, "Designing Reports," pages 422-424 and 440-444, and Chapter 23, "Using Macros to Print Reports or Transfer Data," pages 571-572

Microsoft Access "Language Reference," version 1.0, page 364

Microsoft Access "User's Guide," version 2.0, Chapter 26, "Using Macros to Print Reports or Transfer Data"

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptOthr

**INF: How to Speed Up Iterative Processes in Access Basic**  
**Article ID: Q112724**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

To speed up iterative (looping) processes through large numbers of rows using Access Basic, declare all field references explicitly.

MORE INFORMATION

=====

The following is an example of Access Basic code that does not iterate efficiently:

```
Sub Slow()
 Dim d As Database
 Dim r As Recordset
 Set d = DBEngine.Workspaces(0).Databases(0)
 Set r = d.OpenRecordset("Order Details")
 While Not r.EOF
 r.Edit
 r.Fields("Price") = r.Fields("Qty") * r.Fields("UnitCost")
 r.Update
 r.MoveNext
 Wend
 r.Close
End Sub
```

In the example above, the field variable "lookup" (that is, where Access Basic equates variable names with database fields) for the three field variables Price, Qty, and UnitCost is performed in the same While loop where the calculations are performed. In this configuration, both calculations and lookups must be performed inside the While loop, which is not an efficient design.

The following is an example of Access Basic code that is more efficient:

```
Sub Faster()
 Dim d As Database
 Dim r As Recordset
 Dim Price As Field, Qty As Field, UnitCost As Field
 Set d = DBEngine.Workspaces(0).Databases(0)
 Set r = d.OpenRecordset("Order Detail")
 Set Price = r.Fields("Price")
 Set Qty = r.Fields("Qty")
 Set UnitCost = r.Fields("UnitCost")
 While Not r.EOF
 r.Edit
 Price = Qty * UnitCost
 r.Update
 Wend
```

```
 r.MoveNext
 Wend
 r.Close
End Sub
```

This example runs faster because Access Basic performs the field lookup only once for each field and completes it before executing the loop. A direct reference to each of the three fields is then stored in the three field variables Price, Qty, and UnitCost. As a result, no lookup is required in the While loop, and the field values are accessed and manipulated directly.

NOTE: The code examples above are only for illustrative purposes. In some cases, an Update query can be a faster way to accomplish the task. An example of such a case would be when you need to modify one field based on another field (or fields). Also, speed differences may be slight for small numbers of records.

Performance can also be optimized by using the transaction processing features of Access Basic (BeginTrans, CommitTrans, and Rollback). These features allow Microsoft Access to cache the information and reduce disk I/O.

#### REFERENCES

=====

For more information about transaction processing, search for "Transactions" then "BeginTrans, CommitTrans, Rollback Methods" using the Microsoft Access Help menu.

Additional reference words: 2.00 Queries Programming AB

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: How to Start the AutoDialer Without the Toolbar Button**  
**Article ID: Q114218**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to invoke the AutoDialer without choosing the AutoDialer button on the toolbar.

MORE INFORMATION

=====

You can add the AutoDialer button to a toolbar to automatically dial a telephone number using your computer's modem. To invoke the AutoDialer without using the AutoDialer button, call the `wlib_AutoDial()` function, supplying the function with the telephone number as an argument.

For a form with a field called Phone that holds a telephone number, you could call the `wlib_AutoDial()` function from a command button by setting the button's `OnClick` property to:

```
=wlib_AutoDial([Phone])
```

You can also call the `wlib_AutoDial()` function in any module's Immediate window. For example, to dial the phone number 555-1212, type the following in the Immediate window:

```
? wlib_AutoDial("555-1212")
```

REFERENCES

=====

For more information about the AutoDialer, search for "AutoDialer" then "AutoDialer" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: How to Start the Mail Merge Wizard Without the Toolbar**  
**Article ID: Q118871**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to start the Mail Merge Wizard without using the Merge It button on the toolbar.

MORE INFORMATION

=====

The Mail Merge Wizard helps you merge data from a Microsoft Access table or query with a form letter or other document in Microsoft Word version 6.0 for Windows. You can start the Mail Merge Wizard by clicking the Merge It button on the toolbar.

To start the Mail Merge Wizard without using the Merge It button, call the PM\_Entry() function from within your application. Make sure to supply the function with a valid table or query name as an argument. For example, to merge the data in the Customers table with a Word 6.0 document by clicking a command button on a form, you would specify the following setting for the command button's OnClick property:

```
=PM_Entry("Customers")
```

NOTE: WZLIB.MDA must be loaded in the [Libraries] section of your MSACC20.INI file for this function to be available. If WZLIB.MDA is not loaded as a library, you cannot run the PM\_Entry() function.

REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Appendix D, "Using Other Windows-Based Applications with Microsoft Access," page 739

Additional reference words: 2.00 interoperability

KBCategory: kbprg

KBSubcategory: MdlOthr

## **INF: How to Store Subform Totals in a Main Form Field**

**Article ID: Q119993**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to compute a sum of values in a subform, and store that value in a field on a main form.

This article assumes that you are familiar with constructing and using a main form and subform in a one-to-many relationship.

### MORE INFORMATION

=====

#### Drawbacks to Storing Calculated Values

-----

Although this article demonstrates how to store a subform total in a field on a main form, it should be noted that it is not good database design to store computed values, for the following reasons:

- Storing calculated values occupies additional space in your database.
- You run the risk of violating the integrity of your data. If you open the subform and change one of the values that is summed, this change will not be reflected in the total stored in the main form, making the data inconsistent.

Note that the sample database NWIND.MDB shipped with Microsoft Access version 2.0 does not store the order amount from the Order Details table in the Orders table as earlier versions did. Instead, it computes the totals in a query that is based on the Order Details table. This query is grouped on the Order ID field in the first column, and computes the total in the second column. This query can be joined to the Orders table in another query to see order information just as though the total had been stored in the Orders table. This method is not only the preferred database design, it is also easier to implement than storing the total as described below.

#### How to Store Computed Subform Totals in a Main Form

-----

This section describes how the sample database NWIND.MDB shipped with Microsoft Access version 1.1 computes the order amount in the Order Details subform and stores that value in the Order Amount field in the main form based on the Orders table.

In order to correctly compute and store the order amount, a macro must be run to recompute and store the amount whenever information for the order changes. This is accomplished in the Orders form by running the macro

specified in the AfterUpdate property setting of the Unit Price, Quantity, and Discount fields. A macro is also specified in the AfterUpdate property setting of the Orders Subform form so that when the record is saved, the order amount is recomputed and stored.

The following macro, called Orders.Update Order Amount, is specified in the AfterUpdate property setting of the Unit Price, Quantity, and Discount fields:

| Action     | Description                                                                                                                  |
|------------|------------------------------------------------------------------------------------------------------------------------------|
| DoMenuItem | Update the Order Subtotal control after changing values in any of the following controls: Unit Price, Quantity, or Discount. |
| RunMacro   | Run the Write Order Amount macro.                                                                                            |

Orders.Update Order Amount Actions

---

DoMenuItem  
Menu Bar: Form  
Menu Name: File  
Command: Save Record

RunMacro  
Macro Name: Orders.Write Order Amount

The DoMenuItem action saves the Order Details record with the call to the Orders.Write Order Amount macro so that the new order can be computed.

The Orders.Write Order Amount macro is specified in the AfterUpdate property setting of the Order Details subform and, as seen above, is invoked from the Orders.Update Order Amount macro. The Orders.Write Order Amount macro has the following actions:

| Action   | Description                                                                                  |
|----------|----------------------------------------------------------------------------------------------|
| SetValue | Write the value in the Order Amount control into the Order Amount field in the Orders table. |

Orders.Write Order Amount Actions

---

SetValue  
Item: Forms![Orders]![Order Amount]  
Expression: DSum("[Extended Price]", "[Order Details2]", "[Order ID]=Forms![Orders]![Order ID]")

It should be noted that the description for the SetValue action above is misleading. The action is not really writing the value in the Order Amount control, it is instead computing the total using a DSum() function and writing this value to the Order Amount field.

The DSum() expression computes the sum of line item values by summing the Extended Price field in the Order Details2 query where the Order ID in the Order Details table matches the current Order ID in the Orders form.

Using the DSum() Function Instead of Summing in the Subform Footer

---



A common technique for computing the sum of values in subform fields is to include an expression in the subform footer that uses the Sum() function, and then reference this sum field from the main form. The Orders form demonstrates this technique. The subform footer has a text box called Order Subtotal with the following expression:

```
=Sum([Extended Price])
```

The Extended Price field is computed in the Order Details2 query that the subform is based on. The Extended Price field is computed by multiplying the Unit Price and Quantity fields, and includes a discount. This expression is displayed on the main Orders form in a text box called Subtotal with the following expression:

```
=[Orders Subform].Form![Order Subtotal]
```

It would seem that you could use this value in the Orders.Write Order Amount macro instead of the DSum() expression. However, there are problems associated with doing this. The Orders.Write Order Amount macro in the NWIND database shipped with Microsoft Access version 1.0 does use the subform Sum() expression. In place of the DSum() expression, the SetValue action writes the following:

```
Forms![Orders]![Orders Subform].Form![Order Subtotal]
```

The problem with this method is that the expression may not always immediately reflect the current order amount. The value is computed independently of other actions on the form. When a field that affects this calculation is altered in the subform, the Orders.Write Order Amount macro may be run by the form's AfterUpdate property before Microsoft Access has a chance to recompute the new order amount in the subform. This results in incorrect values being stored.

This problem is described in more detail in the following article in the Microsoft Knowledge Base:

```
ARTICLE-ID: Q101090
TITLE : PRB: NWIND Order Form Updates Order Amount Field with Zero
```

This problem no longer occurs in the NWIND database shipped with Microsoft Access version 1.1 since the DSum() expression is used instead of referencing the Order Subtotal field. This explains why the description for the SetValue action in the NWIND database shipped with Microsoft Access 1.1 incorrectly reads "Write the value in the Order Amount control..." as described earlier.

#### REFERENCES =====

For more information about the DSum() function, search for "DSum" then "DSum Function" using the Microsoft Access Help menu.

For more information about domain aggregate functions such as the DSum() function, please see the following article in the Microsoft Knowledge Base:

```
ARTICLE-ID: Q108098
```

TITLE : INF: DLookup() Usage, Examples, and Troubleshooting Tips

Microsoft Access "User's Guide," version 1.0, Chapter 11, "Using Expressions in Forms," pages 296-298

Microsoft Access "User's Guide," version 1.1, Chapter 11, "Using Expressions in Forms," pages 300-302

Microsoft Access "User's Guide," version 2.0, Chapter 18, "Using Expressions in Forms," pages 457-459

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsSubf

## INF: How to Synchronize Two Combo Boxes on a Form

Article ID: Q97624

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to create two combo boxes so that making a selection in the first combo box limits the list of available items in the second combo box.

### MORE INFORMATION

=====

The following example uses the sample database NWIND.MDB. The first combo box lists the available product categories, and the second combo box lists the available products for the category selected in the first combo box:

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create the following new query based on the Products table, then save the query as Category Combo Query:

Query: Category Combo Query

-----  
Table: Products

Type: Select Query

Field: Product ID

Sort: Ascending

Show: Yes

Field: Product Name

Table: Products

Show: Yes

Field: Category ID

Show: No

Criteria: Forms![Categories and Products]![Categories]

3. Create a new, unbound form with the following combo boxes, then save the form as Categories And Products.

Combo Box 1

-----  
ControlName: Categories

RowSourceType: Table/Query

RowSource: Categories

ColumnCount: 1

BoundColumn: 1

After Update: Refresh Products

Combo Box 2

-----

ControlName: Products  
RowSourceType: Table/Query  
RowSource: Category Combo Query  
ColumnCount: 2  
ColumnWidth: 0  
BoundColumn: 1

NOTE: The BoundColumn property of the first combo box should not be set to the field named in the Criteria row of the above query, otherwise the second combo box displays only the first record.

3. Create the following new macro, then save it as Refresh Products:

```
Requery
 ControlName: Products
```

4. View the Categories And Products form in Form view.

When you select a category in the first combo box, the second combo box is updated to list only the available products for the selected category.

Notes

-----

In the above example, the second combo box is filled with the results of the Category Combo Query query. This query finds all the products that have a Category ID that matches the category selected in the first combo box.

Whenever a category is selected in the first combo box, the AfterUpdate property runs the Refresh Products macro, which forces the second combo box to run the Category Combo Query query again. This refreshes the list of available products in the second combo box. Without this macro, you would have to force the second combo box to refresh itself by pressing the F9 key.

Additional reference words: 1.00 1.10 2.00 listbox combobox

KBCategory: kbusage

KBSubcategory: FmsCmbo

**INF: How to Trap Keystrokes that Leave a Control on a Form**  
**Article ID: Q113547**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to trap the ENTER, TAB, SHIFT+TAB, PAGE UP, and PAGE DOWN keys when they are pressed in a form, and then either replace the key's action with another action, or disable the action.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

The example below demonstrates how to use the GetKeyState() Windows application programming interface (API) call to detect which key or key combination is used to exit a control on a form. This technique can be used to prevent a user from exiting a control, or to remap the key or key combination to a different action:

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Enter the following Declarations and function in an Access Basic module:

```
' **** Declarations Section ****
Option Explicit

Global Const VK_CONTROL = &H11
Global Const VK_SHIFT = &H10

Global Const VK_TAB = &H9
Global Const VK_RETURN = &HD
Global Const VK_PRIOR = &H21 ' PGUP
Global Const VK_NEXT = &H22 ' PGDN

Declare Function GetKeyState% Lib "user.exe" (ByVal nKey%)
' ****

Function CheckKey ()
 Dim sControl As String
 Dim sShift As String

 ' Was the CTRL or SHIFT key used?
 If GetKeyState(VK_CONTROL) < 0 Then sControl = "Ctrl+"

```

```

If GetKeyState(VK_SHIFT) < 0 Then sShift = "Shift+"

' Was the TAB key used?
If GetKeyState(VK_TAB) < 0 Then
 MsgBox "You pressed " & sControl & sShift & "TAB!"
End If

' Was the ENTER or RETURN key used?
If GetKeyState(VK_RETURN) < 0 Then
 MsgBox "You pressed " & sControl & sShift & "ENTER!"
End If

' Was the PAGE DOWN key used?
If GetKeyState(VK_NEXT) < 0 Then
 MsgBox "You pressed " & sControl & sShift & "PGDN!"
End If

' Was the PAGE UP key used?
If GetKeyState(VK_PRIOR) < 0 Then
 MsgBox "You pressed " & sControl & sShift & "PGUP!"
End If

End Function

```

2. Open any form in Design view, and set any control's OnExit property to:

```
=CheckKey()
```

3. View the form in Form view, select the control you modified in step 2, and then try to exit the control.

The next example demonstrates how to use the GetKeyState() Windows API call to modify the ENTER key's action so that it adds lines to a text box, rather than moving to the next control in the tab order:

1. In version 1.x, create the following sample Access Basic function in a module:

```

Option Explicit
Global Const VK_RETURN = &HD
Declare Function GetKeyState% Lib "user.exe" (ByVal nKey%)

Function MakeEnterAddLines()
 If GetKeyState(VK_RETURN) < 0 Then
 DoCmd CancelEvent
 SendKeys "^{ENTER}"
 End If
End Function

```

-OR-

In version 2.0 set the EnterKeyBehavior property to:

```
New Line In Field
```

2. Set the OnExit property of any text box control on any form to:

```
=MakeEnterAddLines()
```

3. View the form in Form view, move to the control you modified, and then press ENTER.

The next example demonstrates how to use the GetKeyState() Windows API call to disable the PAGE UP and PAGE DOWN keys in a form:

1. Create the following sample Access Basic function in a module:

```
Option Explicit
Global Const VK_PRIOR = &H21 ' PGUP
Global Const VK_NEXT = &H22 ' PGDN
Declare Function GetKeyState% Lib "user.exe" (ByVal nKey%)

Function DisablePGUP_PGDN()
 If GetKeyState(VK_PRIOR) < 0 Or GetKeyState(VK_NEXT) < 0 Then
 DoCmd CancelEvent
 End If
End Function
```

2. Set the OnExit properties of all the controls where the PAGE UP and PAGE DOWN keys are to be disabled, on any form, to:

```
=DisablePGUP_PGDN()
```

The next example demonstrates how to use the GetKeyState() Windows API call to cause the TAB key, when it is pressed in the last control in the last record in a subform, to navigate to the next control on the main form:

1. Create the following sample Access Basic function in a module:

```
Option Explicit
Global Const VK_TAB = &H9
Global Const VK_SHIFT = &H10
Declare Function GetKeyState% Lib "user.exe" (ByVal nKey%)

Function TABtoMainForm(F As Form)
 Dim DS As Dynaset
 On Error Goto Bye_TABtoMainForm

 ' Move to last record in form dynaset
 Set DS = F.Dynaset
 DS.MoveLast

 ' See if the form record is the last record
 If DS.Bookmark = F.Bookmark Then
 If GetKeyState(VK_TAB) < 0 And Not _
 (GetKeyState(VK_SHIFT) < 0) Then
 DoCmd CancelEvent
 SendKeys "^{TAB}"
 End If
 End If

Bye_TABtoMainForm:
```

End Function

2. Set the OnExit property of the last control in the subform to:

```
=TABtoMainForm(Form)
```

#### REFERENCES

=====

For more information about using the GetKeyState() call, query on the following word here in the Microsoft Knowledge Base:

GetKeyState

For more information about other ways to disable the PAGE UP and PAGE DOWN keys, query on the following words here in the Microsoft Knowledge Base:

pgup and pgdn

Additional reference words: 1.00 1.10 2.00 trapping

KBCategory: kbusage

KBSubcategory: FmsEvt



## INF: How to Trap SQL ODBC Login Cancel Error

Article ID: Q122250

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

If you choose Cancel when you are logging in to a SQL Server database with Access Basic code, you receive the run-time error message "Operation canceled by user." This article describes how to trap this error.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

The following sample function demonstrates how to trap the error described in the "Summary" section above. To use this function, first create it in a module, and then type "?TryLogin()" (without quotation marks) in the Immediate window.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
'*****
' Declarations section
'*****
Option Explicit

'*****
' Sample function TryLogin()
'*****
Function TryLogin ()
 On Error GoTo LoginError
 DoCmd TransferDatabase A_ATTACH, "<SQL Database>", "ODBC;", _
 a_table, "dbo.authors", "dbo_authors"
 MsgBox "Successful Attachment!"
Exit Function

LoginError:
 If Err = 3059 Then
 MsgBox "You canceled the login."
 Else
 MsgBox Error$
 End If
 Exit Function
End Function
```

REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 10,  
"Handling Run-Time Errors," pages 221-238

Additional reference words: 2.00 odbc

KBCategory: kbusage kberrmsg

KBSubcategory: ObcProb

## **INF: How to Uninstall Microsoft Access Version 2.0**

**Article ID: Q112138**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

Microsoft Access version 2.0 includes commands to uninstall, or remove, many but not all of its components. This article describes how to remove a typical, single-user installation of Microsoft Access 2.0 from your hard disk.

### MORE INFORMATION

=====

Before you uninstall Microsoft Access, you should back up any databases (any .MDB or .MDA files) created in Microsoft Access 2.0. If you implemented any security in your Microsoft Access 2.0 database system, make sure to back up the SYSTEM.MDA file.

You might also want to save your data tables in another format, such as Microsoft Access version 1.x, dBASE IV, or text delimited. The steps for saving Microsoft Access 2.0 tables in version 1.x format are described below. For more information on exporting data to other formats, search on "exporting data: basics" using the Microsoft Access Help menu.

#### Saving Microsoft Access 2.0 tables in 1.x Format

-----

1. In Microsoft Access 2.0, choose Export from the File menu.
2. In the Data Destination box, select Microsoft Access.
3. Select the table to export, then choose OK.
4. Select a Microsoft Access 1.x database to export the table to.
5. Repeat steps 3 and 4 for each of your tables.

#### Uninstalling Microsoft Access 2.0

-----

1. Run Microsoft Access Setup by double-clicking the MS Access Setup icon in your Microsoft Access 2.0 program group in Program manager, or by running SETUP.EXE from the ACCESS\SETUP directory in File Manager.
2. Choose Remove All. When you are prompted "Are you sure you want to remove this application?" choose Yes.
3. If you are prompted to remove shared components for Microsoft Graph and MSInfo, choose Keep if you have other applications, such as Microsoft Word for Windows or Microsoft Excel, that share these

applications.

4. After the uninstall process is completed, use File Manager to remove your Microsoft Access 2.0 directory and any files it contains.
5. Use File Manager to remove these files from the Windows directory:

CCARD200.EXE  
MSACC20.INI

6. Use File Manager to remove these files from the WINDOWS\SYSTEM directory:

CAUTION: Many of these files are shared with other Windows applications. Do not delete these files if you have other applications, such as Microsoft Visual Basic 3.0, Microsoft Excel 5.0, or Microsoft Word 6.0 for Windows, that share these files with Microsoft Access 2.0.

BTRV200.DLL  
CCAPI200.DLL  
DAO2016.DLL  
DBNMP3.DLL  
DRVACC2.HLP  
DRVSSRVR.HLP  
MSABC200.DLL  
MSAEXP20.DLL  
MSAJT200.DLL  
MSJETERR.DLL  
MSJETINT.DLL  
OLE2.DLL  
OLE2.REG  
OLE2CONV.DLL  
OLE2DISP.DLL  
OLE2NLS.DLL  
OLE2PROX.DLL  
PDX200.DLL  
SQLSRVR.DLL  
SQORA.DLL  
STORAGE.DLL  
TYPELIB.DLL  
XBS200.DLL

7. Use File Manager to remove these ODBC files from your Windows directory:

CAUTION: Do not remove ODBC files if other applications, such as Microsoft Visual Basic 3.0, Microsoft Excel 5.0, or Microsoft Word 6.0 for Windows, that share these ODBC files with Microsoft Access 2.0 are installed on your computer.

ODBC.INI  
ODBCINST.INI

8. Use File Manager to remove these ODBC files from the WINDOWS\SYSTEM directory:

CAUTION: Do not remove ODBC files if other applications, such as Microsoft Visual Basic 3.0, Microsoft Excel 5.0, or Microsoft Word 6.0

for Windows, that share these ODBC files with Microsoft Access 2.0 are installed on your computer.

ODBC.DLL  
ODBCADM.EXE  
ODBCADM.HLP  
ODBCINST.DLL  
ODBCINST.HLP  
ODBCJT16.DLL  
ODBCTL16.DLL

9. Remove the ODBC icon from the Windows Control Panel as follows:
  - a. Open the CONTROL.INI file located in the Windows directory with a text editor such as Windows Notepad.
  - b. Under the [MMCPL] section, change the NumApps entry to one less than the current number.
  - c. Also under the [MMCPL] section, delete the ODBC= line.
  - d. Save the CONTROL.INI file and close it.

Additional reference words: 2.00 un-install  
KBCategory: kbsetup  
KBSubcategory: StpOthr

**INF: How to Update a Table Based on Values in a Second Table**  
**Article ID: Q93690**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

To transfer data between two tables, create an additional field in one of the tables to store the common data, then create an update query to accomplish the transfer.

MORE INFORMATION

=====

This process can be illustrated using the sample database NWIND.MDB. The following example adds a Product Name field to the Order Details table, then copies the Product Name field from the Products table based on the Product ID field.

NOTE: Normally you would not duplicate the Product Name field, since duplicate data is not good database design.

1. Open the Order Details table in Design view.
2. Add a Product Name field with the default Text data type.
3. Save and then close the table.
4. Create a new query based on the Order Details and Products tables.

NOTE: These tables are automatically joined on the Product ID field because a relationship has already been defined. If your tables are not joined, join them on the original linked field.

5. From the Query menu, choose Update.
6. From the View menu, choose Table Names. Drag the Product Name field from the Order Details table to the first cell in the Field row of the query grid.
7. In the Update To row, type:

[Products]![Product Name]

8. From the Query menu, choose Run. Close the query.

REFERENCES

=====

For more information about using an update query with two tables, query on the following words here in the Microsoft Knowledge Base:

update and counter and query

Microsoft Access "User's Guide," version 1.0, Chapter 7

Additional reference words: 1.00 1.10 2.00 link join transfer cascade

KBCategory: kbusage

KBSubcategory: QryMktbl

## INF: How to Use a Query to Filter Unique Data

Article ID: Q90809

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### 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 GroupBy() 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.

NOTE: This query will return the unique data. To find duplicate records, use the Find Duplicates Query Wizard in Microsoft Access version 2.0. For more information on finding duplicates using Microsoft Access version 1.x, query on the following words here in the Microsoft Knowledge Base:

find and duplicates

#### REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryTotal

## **INF: How to Use a Secured Database in Word Mail Merge**

**Article ID: Q120810**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Word for Windows, version 6.0
- 

### SUMMARY

=====

To access a secured Microsoft Access database using ODBC or DDE, you must log on using a valid Microsoft Access account. Microsoft Word for Windows will not open a merge document if you do not have Read Data permission on the table or query that is specified as the data source.

This article describes how to use a secured Microsoft Access database in a Microsoft Word version 6.0 for Windows mail merge.

### MORE INFORMATION

=====

This article describes how to complete the following tasks to set up an example of using a secured Microsoft Access database in a Word 6.0 mail merge:

1. Create a secured database in Microsoft Access.
2. Create a merge document in Word, using DDE to access the secured database as the data source.
3. Create a merge document in Word, using ODBC to access the secured database as the data source.

Detailed instructions for these tasks are listed below.

#### 1: Create a Secured Database in Microsoft Access

-----

1. Start Microsoft Access and open any database. If you have never secured a database you will be logged on as the Admin user. If you have implemented security, log on as a member of the Admins group.
2. From the Security menu, choose Users. Choose the New button, then add the following new users:

Name: DB\_Admin  
Personal ID: 9898  
Name: DB\_User  
Personal ID: 7676

3. Add DB\_Admin to the Admins group. Verify that DB\_User is a member of the Users group, and then choose the Close button.
4. If you have not secured a database before, choose Change Password from the Security menu, and then enter "AD1MN2" (without quotation marks) in the New Password and Verify boxes. Choose OK. If you have already

assigned a password to the Admin user, go to step 5.

5. Quit and then restart Microsoft Access. Log on as DB\_Admin.
6. Open any database.
7. From the Security menu, choose Users. Remove the Admin user from the Admins group.
8. Quit and then restart Microsoft Access. This step ensures that the Admin user is not a member of the Admins group when you create your new secured database. This prevents other users from logging on to your secured database as the Admin user.
9. From the File menu, choose New. Create a new database called SEC\_TEST.MDB.
10. From the Security menu, choose Change Password. Enter "DB1AD2" (without quotation marks) in the New Password and Verify boxes, and then choose OK.
11. Use the Table Wizard to create a new table called Mailing List. Add several records to the table.
12. From the Security menu, choose Permissions. Select the Groups option button, then select the Users group in the User/Group Name box. In the Object Name box, select Mailing List and then clear all the Permissions check boxes. Choose the Assign button, then choose the Close button.
13. Quit and then restart Microsoft Access. Log on as DB\_User. Open the SEC\_TEST database.
14. Verify that DB\_User cannot open or delete the Mailing List table.
15. From the Security menu, choose Change Password. Type "DB1US1" (without quotation marks) in the New Password and Verify boxes, then choose OK.

## 2: Create a Merge Document in Word, Using DDE to Access the Secured Database

---

1. Start Word and create a new document. Save the document as TESTMRG.DOC.
2. From the Tools menu, choose Mail Merge. Choose the Create button, select Form Letters, and then choose the Active Window button.
3. Choose the Get Data button, select Open Data Source, and then in the List Files Of Type box, select MS Access Databases (\*.mdb).
4. Locate and select SEC\_TEST.MDB (the secured database you created in the section above). Select the Confirm Conversions check box, and then choose OK.
5. Select MS Access Databases Via DDE, and then choose OK.

6. When you are prompted, enter the user name DB\_Admin and the password DB1AD2, and then choose OK.
7. Select the Mailing List table, and then choose OK.
8. Choose the Edit Main Document button.
9. Choose the Insert Merge Field button on the Mail Merge toolbar. Add several fields to the document.
10. Choose the Merge To New Document button on the Mail Merge toolbar and verify that the merge is successful.
11. Close the new form letter document.
12. Save and then close TESTMRG.DOC.
13. Reopen TESTMRG.DOC. When you are prompted, enter the user name Admin and the password AD1MN2. Verify that the document fails to open.
14. Reopen TESTMRG.DOC. When you are prompted, enter the user name DB\_User and the password DB1US1. The document will fail to open again. Only by using the user name DB\_Admin with the password DB1AD2 can you open the document.

### 3: Create a Merge Document in Word, Using ODBC to Access the Secured Database

---

1. Follow steps 1-4 in section 2 above.
2. Select "MS Access 2.0 Databases via ODBC" and then choose OK.
3. When you receive a message stating that the default login is invalid, choose OK and then enter the user name DB\_Admin and the password DB1AD2. Choose OK.
4. Select the Mailing List table, choose OK, and then choose the Edit Main Document button.
5. Choose the Insert Merge Field button on the Mail Merge toolbar. Add several fields to the document.
6. Choose the Merge To New Document button on the Mail Merge toolbar and verify that the merge is successful.

The "MS Access 2.0 Databases via ODBC" data source is a default ODBC data source that was added when the Microsoft Access 2.0 ODBC driver for Word and Microsoft Excel was installed. You can create new ODBC data sources using the ODBC icon in Control Panel. Creating a new ODBC data source lets you specify the default database and user name, but does not let you specify the password or SYSTEM.MDA.

To avoid being prompted for a user name and password each time you open the document, add this information to the connect string programmatically.

REFERENCES

=====

For information about creating a Word Basic macro to set the connect string, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q110334

TITLE : WordBasic Examples Using ODBC: MailMergeOpenDataSource

For more information about the ODBC Microsoft Access driver, please see the DRVACC2.HLP file.

Additional reference words: 2.00

KBCategory: kbinterop

KBSubcategory: IntpPrtm

**INF: How to Use an Import/Export Spec. in Another Database**  
**Article ID: Q119472**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to use a previously defined import/export specification from one database in another database. Note that the method described in this article involves the manipulation of system tables. While manipulating system tables is neither supported nor encouraged, the only method for using a previously defined import/export specification in another database is to transfer the relevant system tables from one database to another.

MORE INFORMATION

=====

Import/export specifications are stored in the system tables MsysIMEXSpecs and MsysIMEXColumns. These two tables can be imported or exported, but the target database MUST NOT contain an existing import/export specification. If the target database already contains an import/export specification, additional tables that Microsoft Access does not recognize will be created. The best approach is to create a new database for the import/export specification.

How to Import an Import/Export Specification into a New Database

-----

The following example demonstrates how to import an import/export specification into a new database:

1. Open the database containing the import/export specification you want to import into another database.
2. From the View menu, choose Options. In the Category box, select General, and then set the Show System Objects item to Yes. Choose OK.
3. From the File menu, choose New Database. Type a name for the new database in the File Name box, then choose OK.
4. From the File menu, choose Import.
5. In the Data Source box, select Microsoft Access, and then choose OK.
6. In the File Name box, select the database that contains the import/export specification you want to import, and then choose OK.
7. In the Object Type box, select Tables. In the Objects In <Database> box, select MsysIMEXSpecs and then choose the Import button. Choose OK when you are notified that the table was successfully imported.

8. Repeat step 7 using the MsysIMEXColumns table, and then choose the Close button.
9. From the View menu, choose Options. In the Category box, select General, and then set the Show System Objects item to No. Choose OK.

NOTE: You can also export the MsysIMEXSpecs and MsysIMEXColumns tables using the Export command from the File menu.

#### REFERENCES

=====

For more information about import/export specifications, search for "import/export specifications" then "Creating and Editing Import/Export Specifications" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: IsmIea

## INF: How to Use Command Line Switches

Article ID: Q105128

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

When you run Microsoft Access from Program Manager or File Manager, you can use command line switches to modify the start up process. For example, you can open a particular database, execute a macro, or supply a password.

### MORE INFORMATION

=====

Microsoft Access recognizes the following command line switches:

#### Version 1.x Only

-----

    /s                                    Bypasses the Microsoft startup screen.

NOTE: The startup, or splash, screen in Microsoft Access can be also be suppressed using the Microsoft Access version 1.x Distribution Kit (ADK).

In version 2.0, the /s switch no longer applies. Instead, include the StartupScreen entry to display your own bitmap logo rather than the default Microsoft Access logo. Although this is a setting for ADT applications, this entry can also be used without the ADT. Include the following statement in your MSACC20.INI file:

```
[Run-Time Options]
StartupScreen=C:\LOGO\MYBITMAP.BMP
```

#### Versions 1.x and 2.0

-----

|                   |                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <database>        | Opens the specified database.                                                                                                                                                                       |
| /Excl             | Opens the database for exclusive access.                                                                                                                                                            |
| /Ro               | Opens the database for read-only access.                                                                                                                                                            |
| /User <user name> | Starts Access using the specified user name.                                                                                                                                                        |
| /Pwd <password>   | Starts Access using the specified password.                                                                                                                                                         |
| /X <macro>        | Starts Access and runs specified macro.                                                                                                                                                             |
| /Cmd              | Specifies what value to pass to the Command function in an Access Basic procedure that is called by a RunCode action in an AutoExec macro. This option must be the last option on the command line. |

#### Version 2.0 Only

-----



|                            |                                                                                                                                                                                                               |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /Ini <initialization file> | Starts Microsoft Access using the options in the specified initialization file instead of the standard MSACC20.INI file. The initialization file must have the same entries as the standard MSACC20.INI file. |
| /Compact <target database> | Compacts the database specified before the /Compact option and then closes Microsoft Access. To compact to a different name, specify a target database.                                                       |
| /Repair                    | Repairs the specified database and then closes Microsoft Access.                                                                                                                                              |
| /Convert <target database> | Converts a 1.x database to a 2.0 database, with a new name, and then closes Microsoft Access. Specify the source database before the /Convert option.                                                         |

For example, you could type the following command in the Run dialog box (available by choosing Run from the Program Manager File menu):

```
C:\ACCESS\MSACCESS.EXE NWIND.MDB /Excl /X Add Products
```

This sample syntax starts Microsoft Access, opens the NWIND sample database for exclusive use, and runs the Add Products macro.

#### REFERENCES

=====

For more information, search for "command line" then "Starting Microsoft Access with Command-line Options" using the Microsoft Access Help menu.

Microsoft Windows "User's Guide," version 3.1, Chapter 2, "Application Basics," pages 53-54

Additional reference words: 1.00 1.10 2.00

KBCategory: kbui

KBSubcategory: UifOthr

## **INF: How to Use DAO to Assign or View Permissions**

**Article ID: Q112106**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

In Microsoft Access version 2.0, programmers can assign permissions to database objects programmatically using data access objects (DAO). Securing a database involves several steps, including:

1. Creating group accounts.
2. Creating user accounts.
3. Assigning users to groups.
4. Assigning permissions on database objects to users and groups.
5. Encrypting the database, if desired.

This article demonstrates the fourth step, assigning permissions on database objects to users and groups.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual, Chapter 3, "Introducing Access Basic."

### MORE INFORMATION

=====

To assign permissions for an object, you must reference the UserName and Permissions properties of the objects document. All database objects can be treated as documents in a container. For example, given the DB variable assignment

```
Dim DB As Database
Set DB = DBEngine.Workspaces(0).Databases(0)
```

a table called Table1 can be referred to using DAO as follows:

```
DB.Containers("Tables").Documents("Table1")
```

However, it is usually easier to assign and work with object variables:

```
Dim DB as Database, DOC as Document
Set DB = DBEngine.Workspaces(0).Databases(0)
Set DOC = DB.Containers("Tables").Documents("Table1")
```

You can directly reference the UserName and Permissions properties

of the Table1 document like this:

```
UserNameVariable = DOC.UserName
PermissionsVariable = DOC.Permissions
```

-or-

```
DOC.UserName = "MyUser"
DOC.Permissions = DB_SEC_FULLACCESS
```

#### Reading Permissions Information

-----

To read the permissions that a given user has on an object, simply set the UserName property of the document to the name of the user or group you want to inquire on, and then read the value of the Permissions property.

The following example shows how to read the permissions for the user John on form MyMainForm:

```
Dim DB as Database, DOC as Document, HisPermissions as Long
Set DB = DBEngine.Workspaces(0).Databases(0)
Set DOC = DB.Containers("Forms").Documents("MyMainForm")
DOC.UserName = "John"
HisPermissions = DOC.Permissions
```

This example shows how to read the permissions for the group Supervisors on the report SalarySummary:

```
Dim DB as Database, DOC as Document, GroupPermissions as Long
Set DB = DBEngine.Workspaces(0).Databases(0)
Set DOC = DB.Containers("Reports").Documents("SalarySummary")
DOC.UserName = "Supervisors"
GroupPermissions = DOC.Permissions
```

Note that whether you are inquiring on a group or a user, you still set the UserName property. There is no GroupName property for a document.

#### Assigning Permissions Information

-----

To assign permissions for a user on an object, set the UserName property of the document to the name of the user or group, and then set the Permissions property of the document to the appropriate value.

Generally, you should use the predefined constants when you are assigning permissions. For a complete list of the predefined constants, search for "Security: Constants" using the Microsoft Access Help menu.

The following example shows how to assign Full permissions for the user Martha on the table Employees:

```
Dim DB as Database, DOC as Document
Set DB = DBEngine.Workspaces(0).Databases(0)
Set DOC = DB.Containers("Tables").Documents("Employees")
DOC.UserName = "Martha"
```

```
DOC.Permissions = DB_SEC_FULLACCESS
```

The next example shows how to assign Open/Run and Modify Design permissions for the group Developers on the form Customers:

```
Dim DB as Database, DOC as Document
Set DB = DBEngine.Workspaces(0).Databases(0)
Set DOC = DB.Containers("Forms").Documents("Customers")
DOC.UserName = "Developers"
DOC.Permissions = DB_SEC_FRMRPT_EXECUTE+DB_SEC_FRMRPT_WRITEDEF
```

Methods to Programmatically View and Assign Permissions

-----

Method 1:

This method uses a sample subroutine called AssignPerms to update all the objects in a database. To assign permissions for a user on all the objects in a database, you can move through all the documents in all the collections in the database. The following sample subroutine demonstrates one way of moving through all the documents in a database and changing the permissions on them:

```
Sub AssignPerms (GrpUserName as String, NewPerm as Long)
 Dim DB as Database, I as Integer, J as Integer
 Set DB = DBEngine.Workspaces(0).Databases(0)

 For I = 0 to DB.Containers.Count - 1
 For J = 0 to DB.Containers(I).Documents.Count - 1
 DB.Containers(I).Documents(J).UserName = GrpUserName
 DB.Containers(I).Documents(J).Permissions = NewPerm
 Next J
 Next I
End Sub
```

End Sub

You could call this subroutine from within Access Basic code, passing the group or user name and the Permissions value you want to assign.

For example, to revoke all permissions on all objects from the Guests group, call AssignPerms as follows:

```
AssignPerms "Guests", DB_SEC_NOACCESS
```

To assign full permissions on all objects to the Managers account, call AssignPerms like this:

```
AssignPerms "Managers", DB_SEC_FULLACCESS
```

Method 2:

This method uses two sample functions, GetPermissions() and SetPermissions(), to display and assign the permissions for a user or group on the database object you specify.

To create the sample functions, first enter the following text in the Declarations section of a module:

```
Option Compare Database
Option Explicit
Global Const SUCCESS_SUCCESS = 0
```

Then enter the following code in the module:

```
' *****
' FUNCTION: GetPermissions()
'
' Inputs: UserGrpName - name of a user or group account
' ObjClass - name of an object container
' ObjName - name of an object document
'
' Returns: Value of Permissions property or error number
' that was generated.
' *****
Function GetPermissions& (UserGrpName$, ObjClass$, ObjName$)

 On Error GoTo Err_GetPermissions

 ' Set DB to the current database, and set the DOC variable
 ' .. to the object specified in the arguments.
 Dim DB As Database, DOC As Document
 Set DB = DBEngine.Workspaces(0).Databases(0)
 Set DOC = DB.Containers(ObjClass).Documents(ObjName)

 ' Set the UserName property of the document to the
 ' .. user or group you want to obtain the permissions for.
 DOC.UserName = UserGrpName

 ' Get the permissions value.
 GetPermissions = DOC.Permissions

Bye_GetPermissions:
 Exit Function

Err_GetPermissions:
 ' If an error occurs, display the message and terminate the
 ' .. function, returning the error number.
 MsgBox Err & " " & Error$
 GetPermissions = Err
 Resume Bye_GetPermissions

End Function

' *****
' FUNCTION: SetPermissions()
'
' Inputs: UserGrpName - name of a user or group account
' ObjClass - name of an object container
' ObjName - name of an object document
' NewPerm - new Permissions value
'
' NewPerm will typically be set by adding together the constants
' predefined for the security options. For a list of the
' constants search Help on "Permissions Property."
```

```

'
' Returns: SUCCESS_SUCCESS or the error number that was generated.
' *****
Function SetPermissions& (UserGrpName$, ObjClass$, ObjName$, _
 NewPerm&)

 On Error Goto Err_SetPermissions

 ' Set DB to the current database, and set the DOC variable
 ' to the object specified in the arguments.
 Dim DB As Database, DOC As Document
 Set DB = DBEngine.Workspaces(0).Databases(0)
 Set DOC = DB.Containers(ObjClass).Documents(ObjName)

 ' Set the UserName property of the document to the
 ' .. user or group you want to assign the permissions for.
 DOC.UserName = UserGrpName

 ' Set the permissions property to the value passed as
 ' .. an argument to the function.
 DOC.Permissions = NewPerm

 SetPermissions = SUCCESS_SUCCESS

Bye_SetPermissions:
 Exit Function

Err_SetPermissions:
 ' If an error occurs, display the message and terminate the
 ' .. function, returning the error number.
 MsgBox Err & " " & Error$
 SetPermissions = Err
 Resume Bye_SetPermissions

End Function

```

These functions can be called anywhere you can use an expression in Microsoft Access, including from within Access Basic code, in the Field, Criteria, or Update To row of a query, or the ControlSource property of a form or report control.

The following example shows how to call the GetPermissions() function from another Access Basic function:

```

Dim ObjProp as Long
ObjProp = GetPermissions("User1","Tables","Table1")

```

This function call would return the permissions value for User1 on the table object Table1.

The next example shows how to call the GetPermissions() function from the ControlSource property of a control on a form or report with controls that contain the user name, object class, and object name that you want to list the permissions for:

```

ControlName: ObjProp
ControlSource: =GetPermissions(Me!UsrName, Me!ObjClass, Me!ObjName)

```

To assign full permissions for the Admins group to a table called MyTable, you could call the SetPermissions() function as follows:

```
Dim RETVAL as Long
RETVAL = SetPermissions("Admins","Tables","MyTable", DB_SEC_FULLACCESS)
```

#### REFERENCES

=====

For more information about using DAO to create group and user accounts, and about how to assign users to group accounts, query on the following words here in the Microsoft Knowledge Base:

DAO and Security and CreateGroup or CreateUser

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: ScrtOthr

**INF: How to Use DDE to Communicate with a Run-Time Application**  
**Article ID: Q101318**

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

SUMMARY

=====

Run-time applications developed with Microsoft Access have the same dynamic data exchange (DDE) server capabilities as applications in the retail version of Microsoft Access.

However, in order to communicate with a run-time application using DDE, you must use the name of the .MDB file, instead of "MSACCESS", when initiating a DDE conversation.

MORE INFORMATION

=====

The following example demonstrates how to communicate with a run-time application developed with Microsoft Access:

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Choose the Macro button and select the Sample Autoexec macro.
3. From the File menu, choose Rename. In the Macro Name box, type "Autoexec" (without the quotation marks) and then press ENTER.
4. Quit Microsoft Access.
5. From the File menu in the Microsoft Windows Program Manager, choose Run.
6. Type "c:\access\msarn110.exe nwind.mdb" (without the quotation marks), and then press ENTER.
7. Start Microsoft Access, and create or open a database other than NWIND.MDB.
8. Create or open a module and add the following Access Basic function:

Option Explicit

```
Function DDEWithNWIND ()
 Dim chan
 chan = DDEInitiate("NWIND", "NWIND;TABLE Employees")
 MsgBox DDERequest(chan, "FirstRow")
 DDETerminate chan
End Function
```



Note that the first argument specified in the DDEInitiate command, the application name, is "NWIND", which is the name of the .MDB file, and not "MSACCESS."

9. From the View menu, choose Immediate Window. Type the following, and then press ENTER:

```
? DDEWithNWIND()
```

A message box will display the first record from the Employees table in the NWIND.MDB database.

#### REFERENCES

=====

For more information on using Microsoft Access as a DDE server, query on the following words here in the Microsoft Knowledge Base:

DDE and server

Additional reference words: 1.10 2.00 adk adt

KBCategory: kbinterop

KBSubcategory: IntpDde

**INF: How to Use DDE to Pass Information to MS Access 2.0**  
**Article ID: Q113300**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

In Microsoft Access, you cannot poke data into a table using a dynamic data exchange (DDE) channel. However, you can pass information to an Access Basic function using a function with parameters in a DDE channel to the SQL topic. The data can then be processed by the function and added to a table.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

The following example demonstrates how to add a new customer to the Customers table in the sample database NWIND.MDB using data from Microsoft Excel and Microsoft Word for Windows:

1. Create a table called None with one field of any data type, with one record.
2. Open a module and create a function with the parameters that you want to pass. The following Access Basic function will create a new record in the Customers table, and accepts an argument for the Customer ID field and an argument for the Company Name field:

```
Declarations Section
Option Explicit

Function AddNewCust (CustomerID$, CompanyName$)
 Dim MyDB As Database, MyTable As Recordset
 Set MyDB = DBEngine.Workspaces(0).Databases(0)

 ' Trap any error that might occur.
 On Error Resume Next

 ' Open table.
 Set MyTable = MyDB.OpenRecordset("Customers", DB_OPEN_TABLE)

 MyTable.AddNew ' Prepare new record.
 MyTable("Customer ID") = CustomerID$ ' Set record key.
 MyTable("Company Name") = CompanyName$ ' Set company name.
 MyTable.Update ' Save changes.
 MyTable.Close ' Close table.
```

```
' Return the error code.
AddNewCust = Err
```

End Function

This function will return 0 if successful, otherwise a Microsoft Access error number will be returned. If a record in the Customers table already exists with the specified Customer ID, error number 3022 will be returned. This error would return the following error message if it were not trapped:

```
Duplicate value in index, primary key, or relationship.
Changes were unsuccessful.
```

3. In the DDE client application, initiate a DDE link using MSACCESS as the application and a SQL select query as the topic. The topic should be of the form:

```
DatabaseName;SQL Select FunctionName(args) From None;
```

In the sample topic above, None is the name of the empty table, and FunctionName(args) is the call to the Access Basic function that processes the arguments (args) passed to it.

For example, the following Microsoft Excel macro will insert a new record with "JOHNJ" as the Customer ID and "John's Place" as the Company Name into the Customers table:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code.

```
AddNewCustomer
chan=INITIATE("MSACCESS","NWIND;SQL SELECT _
AddNewCust("JOHNJ","John's Place") FROM None;")
=ALERT(REQUEST(chan,"FirstRow"))
=TERMINATE(chan)
=RETURN()
```

The following is the equivalent Microsoft Excel Visual Basic function:

```
Function AddNewCustomer()
Chan = DDEInitiate("MSACCESS", "NWIND;SQL SELECT _
AddNewCust("TADO","Tad's Place") FROM None;")
Result = DDERequest(Chan, "FirstRow")
MsgBox Result(1)
DDETerminate Chan
End Function
```

The following is the equivalent Microsoft Word for Windows Word Basic macro:

```
Sub MAIN
qt$ = Chr$(34) ' create the quote character
Funct$ = "AddNewCust(" + qt$ + "JOHNJ" + qt$ + "," + qt$ + _
```

```
 "John's Place" + qt$ + ")"
Chan = DDEInitiate("MSACCESS", "NWIND;SQL SELECT " + Funct$ + _
 " FROM None;")
Result$ = DDERequest$(Chan, "FirstRow")
MsgBox Result$
DDETerminate Chan
End Sub
```

#### How the SQL Initiate Technique Works

-----

When you initiate a DDE link with Microsoft Access using an SQL statement, each function in the SELECT statement will run one time for each record in the table the query selects from.

In the examples above, the None table has only one record to query against. Since the table has only one record, the AddNewCust() function runs only one time. If the None table had two records, the AddNewCust() function would run twice, trying to add the same record two times.

The query's result will return the results of the function being called in the SELECT statement. The AddNewCust() function returns the error code if an error occurs. This is useful so that the DDE client, such as Microsoft Excel, Word, or Visual Basic, can know whether the function ran successfully or not.

#### REFERENCES

=====

For more information, search for "DDE" then "Using Microsoft Access as a DDE Server" using the Microsoft Access Help menu.

Additional reference words: ddepoke  
KBCategory: kbinterop  
KBSubcategory: IntpOthr

## INF: How to Use Get, WritePrivateProfileString

Article ID: Q90988

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Microsoft Access does not have a function to store or access settings in user-created applications (that is, a log of users, window position settings, and so on). However, you can use the Microsoft Windows 3.1 Application Program Interface (API) through Access Basic to create user-defined functions to store and access settings in a file outside the database.

### MORE INFORMATION

=====

#### 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%              | - Maximum 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  |

#### How to Create the Functions

-----

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Create a new module.

2. Type the following declarations in the Global Declarations section:

```
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$, _
 ByVal lpfilename$)
```

```
Dim GotInfo As String
```

3. Type the following functions in the module:

```
Function GetIni (GetInfo$)
Dim lpfilename$, lpapplicationname$, lpkeyname$, lpDefault$, _
 lpreturnedstring$
Dim nSize%
```

```
 lpfilename$ = "<YOUR.INI>"
 lpapplicationname$ = "<Your Section>"
 lpkeyname$ = GetInfo$
 lpDefault$ = "No Value"
 lpreturnedstring$ = Space$(255)
 nSize% = Len(lpreturnedstring$)
```

```
 GetIni = GetPrivateProfileString(lpapplicationname$, _
 lpkeyname$, lpDefault$, _
 lpreturnedstring$, nSize%, _
 lpfilename$)
```

```
 GotInfo = lpreturnedstring$
End Function
```

```
Function WriteIni (Where$, Info$)
Dim lpapplicationname$, lpkeyname$, lpString$, lpfilename$
 lpapplicationname$ = "<Your Section>"
 lpkeyname$ = Where$
 lpString$ = Info$
 lpfilename$ = "<YOUR.INI>"
```

```
 WriteIni = WritePrivateProfileString(lpapplicationname$, _
 Where$, Info$, lpfilename$)
```

```
End Function
```

How to Use the Functions

- 
1. When used in a macro, the function WriteIni() creates the entry First=<TIME> in the file <YOUR.INI> in the [<Your Section>] 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> in the variable GotInfo as follows:

```
=GetIni("First")
```

The variable GotInfo now holds the value <TIME> and can be used anywhere needed in the application.

NOTE: This example does not have error trapping. Unexpected results may occur if the declarations and the variable types are not correct or your file is not in the location specified or does not exist.

#### REFERENCES

=====

Microsoft Access "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

Additional reference words: 1.00 1.10 2.00

KBCategory: KBPRG kbdocerr

KBSubcategory: PgmApi

## **INF: How to Use Graphic Images as Wallpaper for Forms**

Article ID: Q103428

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

You can use graphic images as "wallpaper," or background, for your forms to make them more attractive.

### MORE INFORMATION

=====

The following example demonstrates how to add wallpaper to a form:

1. Start the Microsoft Windows Paintbrush program (or any graphics program of your choice) and open the WINLOGO.BMP bitmap file (or any bitmap file of your choice).
2. Select the graphics image. If you are using Paintbrush, choose the Pick tool (a pair of open scissors below a dotted rectangle) from the toolbox. Select the bitmap image by using the mouse to place the crosshairs in the upper-left corner of the bitmap, then click and hold down the left mouse button and drag the rectangle to the lower-right corner to encompass the entire image.
3. From the Edit menu, choose Copy. This places a copy of the bitmap image on the Windows Clipboard.
4. Start Microsoft Access, and open the sample database NWIND.MDB.
5. Open the Customers form in Design view. Select the form by clicking the small, white box in the upper-left corner of the form.
6. From the Edit menu, choose Paste. This pastes the bitmap image to an object frame control on the form.

Note that the graphic image does not cover the entire form, but only the upper-left corner of it. This is because this particular graphic image is too small for the form. To cover the entire form, either use a larger graphic image, or use a different graphics program to alter the graphic's size.

7. From the Layout menu, choose Send To Back. This puts the graphic image in the background.
8. On this form, the BackStyle property for all the label controls is set to Clear. In order to make the labels more readable against the new background, change the labels' BackStyle properties to Normal. You can also change the BackColor properties, using the Color Palette.

### REFERENCES



=====

"Inside Microsoft Access," April 1993, Vol. 1, No. 2, pages 1-5

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: GnlApp

## **INF: How to Use In-Place Activation with OLE Objects**

**Article ID: Q112748**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

Microsoft Access version 2.0 supports in-place activation of OLE objects in a form's Form view. In a form or report's Design view, or in a table, query, or form's Datasheet view, activation of an OLE object will cause the object's original application to run in the background as a server application (if the original application is available), so you can edit the object in Microsoft Access.

### MORE INFORMATION

=====

In-place activation of an OLE object allows you to edit or otherwise manipulate the object without switching to the object's original application. For example, you could edit an embedded document that was created in Microsoft Word for Windows without having to start Word for Windows.

When you edit an OLE object in place, the object's border becomes a hatched frame and the menus and menu options supported by the object's original application are added to the standard Microsoft Access menus. Toolbars supported by the object's original application may also be displayed.

The object's original application runs in the background and supplies the functionality it supports for the object, but the application is not visible and does not appear in the Windows Task List.

For an OLE object to be activated in place, the following must be true:

- The object must have been created in an OLE server application that supports in-place activation.
- The object must be displayed in an object frame on a form that is open in Form view.
- The frame's Enabled property must be set to Yes.

If the Locked property is set to Yes, you will be able to edit the object, but any changes you make will be discarded when the object is deactivated.

If the object is linked, you can activate it in place, but you cannot edit it in place.

The AutoActivate property controls how an object is activated. The default value is Double-Click, which allows you to double-click the object to activate it. If you change this property to GetFocus, the object is activated as soon as you select or tab to it. If you set the property to

Manual, you have to select the object, choose <object type> from the Edit menu, and then choose the first verb listed to activate it.

The following is an example of how to create an OLE object that supports in-place activation:

1. Start Microsoft Access and either create a new database or open one of your choosing.
2. Create a new form.
3. Using the Unbound Object Frame tool, create a new frame on the form.
4. In the Object Type box, select a server that supports OLE 2.0. For this example, choose Microsoft Graph 5.0, and then choose OK.
5. From the Microsoft Graph File menu, choose Exit & Return To Form: <Formname>.
6. In Microsoft Access, set the frame's Enabled property to Yes and its Locked property to No.
7. View the form in Form view.
8. Double-click the embedded graph. The graph's border will change to a hatched frame, and several menus and menu options will change. If no Graph 5.0 toolbars are displayed, choose Toolbars from the View menu and select one of the toolbar check boxes.
9. Click anywhere in the form outside of the OLE object to deactivate the graph.

#### REFERENCES

=====

For more information on in-place activation and in-place editing search on "in-place activation" and "in-place editing" using the Microsoft Access Help menu.

For more information on whether or not a specific object application supports in-place activation, refer to that application's documentation.

Additional reference words: 2.00 Interoperability OLE Embedding Embed Linking Link

KBCategory: kbole

KBSubcategory: IntpOle

**INF: How to Use OLE Automation to Modify MS Graph Object**  
**Article ID: Q119469**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to use OLE Automation to modify a Microsoft Graph version 5.0 graph's title and hide or show the graph's legend.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

Changing a Graph's Title

-----

The following example demonstrates how to use OLE Automation to modify a Graph 5.0 graph's title.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the sample database NWIND.MDB.
2. Open the Sales By Product form in Design view.
3. Set the form's DefaultEditing property to Allow Edits.
4. Add the following controls to the form:

Text Box

-----

Name: MyTitle

Command Button

-----

Name: Title

Caption: Show Caption

5. Set the Show Caption command button's OnClick property to the following event procedure.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```

Sub Title_Click ()
Dim The_title As String

'-----
' This function reads the text box. If it is null or blank, it
' supplies a generic title, otherwise it adds a title and sets
' the caption to the text box's contents.
'-----

If IsNull(Me![MyTitle]) Or Me![MyTitle] = "" Then
 The_title = "No Title"
Else
 The_title = Me![MyTitle]
End If

If Me![Title].Caption = "Show Title" Then
 Me![Embedded13].Object.Application.Chart.Hastitle = -1
 Me![Embedded13].Object.Application.Chart.Charttitle.Caption = _
 The_title
 Me![Title].Caption = "Hide Title"
Else
 Me![Embedded13].Object.Application.Chart.Hastitle = 0
 Me![Title].Caption = "Show Title"
End If

End Sub

```

6. Add the following line to the Declarations section of the Sales By Product form's form module:

```
Option Explicit
```

7. View the form in Form view. Type any text in the text box you created in step 4, and then click the Show Caption button. The graph's title will change to the text box's text.

#### Hiding or Showing a Graph's Legend

---

The following example demonstrates how to hide or show a Graph 5.0 graph's legend.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the sample database NWIND.MDB.
2. Open the Sales By Product form in Design view.
3. Add a command button with the following properties to the form:

```
Name: Legend
Caption: Show Legend
```

4. Set the Show Legend command button's OnClick property to the following event procedure:

```

Sub Legend_Click ()

'-----
' If the legend is not present, show it, otherwise hide it.
'-----

If Me![Legend].Caption = "Show Legend" Then
 Me![Embedded13].Object.Application.Chart.Haslegend = -1
 Me![Legend].Caption = "Hide Legend "
Else
 Me![Embedded13].Object.Application.Chart.Haslegend = 0
 Me![Legend].Caption = "Show Legend"
End If
End Sub

```

5. Add the following line to the Declarations section of the Sales By Product form's form module:

```
Option Explicit
```

6. View the form in Form view. Click the Show Legend button to toggle between hiding and showing the form's legend.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 8, "Events," pages 183-201, and Chapter 13, "Communicating with Other Applications," pages 281-311

For more information about OLE Automation, search for "OLE Automation" then "Interoperability with Microsoft Word and Microsoft Excel (Common Questions" using the Microsoft Access Help menu.

Additional reference words: 2.00 link embed

KBCategory: kbtool

KBSubcategory: IntpGrph

**INF: How to Use OLE Automation to Transfer Data to MS Excel**  
**Article ID: Q114836**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

In Microsoft Access version 2.0, you can use code to manipulate data in applications that support OLE Automation, such as Microsoft Word version 6.0 for Windows or Microsoft Excel version 5.0.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

The following example demonstrates how to take data from a text box on a form and place it in a cell on an existing Microsoft Excel spreadsheet, and then format the text as bold:

1. Start Microsoft Excel and create a new spreadsheet. Save the spreadsheet as C:\ACCESS2\OLE\_TEST.XLS.
2. Start Microsoft Access and open any database. Create a new, unbound form. Create a new text box on the form. Set the text box's Name property to ToExcel.
3. Create a command button on the form, without using the Command Button Wizard. If the property sheet is not open, open it by choosing the Properties button on the toolbar.
4. Create an OnClick event procedure for the command button by clicking in the button's OnClick property field, choosing the Build button to the right of the field, selecting Code Builder, and then choosing OK.
5. Insert the following code in the procedure:

```
Dim mysheet As Object

'Set object variable equal to the OLE object.
Set mysheet = GetObject("c:\access2\ole_test.xls", "excel.sheet")

'Put the value of the ToExcel text box into the cell on the
'spreadsheet and make the cell bold.
mysheet.cells(1, 1).value = Me!ToExcel
mysheet.cells(1, 1).font.bold = True

'Set the Visible property of the sheet to True, save the
'sheet, and quit Microsoft Excel.
```

```
mysheet.application.windows("ole_test.xls").visible = True
mysheet.application.activeworkbook.save
mysheet.application.[quit]
```

```
'Clear the object variable.
Set mysheet = Nothing
```

NOTE: The brackets around the word "quit" are necessary to specify that this is a Microsoft Excel method, rather than a Microsoft Access method.

6. Close the form module.

7. View the form in Form view. Type any text in the text box, and then click the command button.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 13, "Communicating with Other Applications"

For more information about OLE Automation, search for "OLE Automation" then "Interoperability with Microsoft Word and Microsoft Excel" using the Microsoft Access Help menu.

For more information about the syntax used in communicating with an application via OLE Automation, see the documentation for that application.

Additional reference words: 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr



## INF: How to Use Seek Method with Multiple-Field Index

Article ID: Q109710

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to use the Seek method when your table's primary key or index includes more than one field. Note that when you use the Seek method on multiple fields, the Seek fields must be in the same order as the fields in the underlying table. If they are not, the Seek will fail.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

When you use the Seek method to find a record using the primary key, you need to supply a value for each field in the primary key. If you cannot supply values for all the fields in the primary key, use ">" instead of "=" for the Comparison argument.

If you set the Index property in your function, refer to the primary key field as PrimaryKey, instead of its actual name. With a non-primary index in Microsoft Access version 1.x, refer to the field as Index1, Index2, Index3, Index4, or Index5, corresponding to the two-field index created in the table.

In Microsoft Access version 2.0, you can rename multiple-field indexes to a custom user-defined name. For example, in version 1.x, a multiple-field index on the First Name and Last Name fields might be assigned to "Index1." In version 2.0, you can create your own assigned name by using the Indexes window in a table's Design view. "Index1" could be changed to "First Name and Last Name." Microsoft Access version 1.x prohibits custom multiple-field index names, while version 2.0 permits them. All versions of Microsoft Access refer to a multiple-field primary key as PrimaryKey.

The following example demonstrates how to use the Seek method on a table with a multiple-field primary key:

1. Open the sample database NWIND.MDB.
2. Create a new module and enter the following sample function:

```
'*****
'Declarations section of the module
'This function uses Seek on a two-field PrimaryKey
'*****
```

Option Explicit

```
Function SeekTwoField1()
 Dim db As Database, tbl As Table
 Set db = CurrentDB()
 Set tbl = db.OpenTable("Order Details")
 tbl.Index = "PrimaryKey" 'Or "Indexn" on a two-field index in
 '1.x or "MyIndex" in 2.0.

 tbl.Seek "=", 10000, 17
 If tbl.NoMatch Then
 MsgBox "Not a record. Try another"
 Else
 MsgBox "The Record is in the table"
 End If
 tbl.Close
End Function
```

```
'*****
'Declarations section of the module
'This function uses Seek on a two-field PrimaryKey,
'when only one field is supplied to the function
'*****
Option Explicit
```

```
Function SeekTwoField2()
 Dim db As Database, tbl As Table
 Set db = CurrentDB()
 Set tbl = db.OpenTable("Order Details")
 tbl.Index = "PrimaryKey" 'Or "Indexn" on a two-field index in
 '1.x or "MyIndex" in 2.0.

 tbl.Seek ">", 10000
 If tbl.NoMatch Then
 MsgBox "Not a record. Try another"
 Else
 MsgBox "The Record is in the table"
 End If
 tbl.Close
End Function
```

3. Type the following in the Immediate window and then press ENTER:

```
? SeekTwoField1()
```

You will receive the message "The Record is in the table."

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.0, Chapter 8,  
"Manipulating Data," page 114

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: How to Use SendKeys in a Macro to Change an OLE Link**  
**Article ID: Q94040**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes two methods to automate the changing of a linked object's source. The first method is to use a macro, and the second method is to use OLE Automation, which is available in Microsoft Access version 2.0.

MORE INFORMATION

=====

Macro Method

-----

The following example demonstrates how to create and use a macro that will automatically change the source of a linked OLE object based on a filename stored in a table.

1. Create a new table with the following fields, and then save it as Linked References:

Table: Linked References

-----

FieldName: Key  
    DataType: Counter  
FieldName: File Name  
    DataType: Text  
FieldName: Linked Object  
    DataType: OLE Object

2. In Datasheet view, add the following records to the table. Note that the bitmaps used in this example are stored in the Microsoft Windows directory (by default, C:\WINDOWS). To embed the objects, open each bitmap in Microsoft Windows Paintbrush, select the bitmap with the selection tool, and then choose Copy from the Edit menu. Switch to Microsoft Access, place the cursor in the Linked Object field, and choose Paste Link from the Edit menu.

| Key       | Filename   | Linked Object         |
|-----------|------------|-----------------------|
| (counter) | ARCADE.BMP | Paste Link ARCADE.BMP |
| (counter) | ARCHES.BMP | Paste Link ARCHES.BMP |
| (counter) | ARGYLE.BMP | Paste Link ARGYLE.BMP |

3. Create a new form based on the table Linked References, and add all three fields to the form.
4. Save the form as OLE Test.

5. Create a new macro with the following actions:

In Microsoft Access version 1.x:

| Action      | Arguments                                             |
|-------------|-------------------------------------------------------|
| GoToControl | Control Name: Linked Object                           |
| SendKeys    | Keystrokes: %eol<br>Wait: No                          |
| SendKeys    | Keystrokes: =Forms![OLE Test]![File Name]<br>Wait: No |
| SendKeys    | Keystrokes: {ENTER}<br>Wait: No                       |

In Microsoft Access version 2.0:

| Action      | Arguments                                             |
|-------------|-------------------------------------------------------|
| GoToControl | Control Name: Linked Object                           |
| SendKeys    | Keystrokes: %ekl%c<br>Wait: No                        |
| SendKeys    | Keystrokes: =Forms![OLE Test]![File Name]<br>Wait: No |
| SendKeys    | Keystrokes: {ENTER}{ENTER}<br>Wait: No                |

6. Save the Macro as Update OLE Link.

7. Open the form OLE Test in Design view and select the control File Name.

8. Set the AfterUpdate property to the macro Update OLE Link.

9. To test the form, switch to Form view. Change the field File Name in the first record to LEAVES.BMP. The OLE object should update to display the bitmap LEAVES.BMP.

OLE Automation

Microsoft Access version 2.0 supports OLE Automation, a feature of the OLE 2.0 standard. You can use OLE Automation to directly access and manipulate another application's objects from outside that application.

1. Create the following subprocedure in the File Name control's AfterUpdate event procedure. The procedure uses the SourceDoc and Action properties to change the source of the linked object.

```
Sub File_Name_AfterUpdate ()
 Const OLE_LINKED = 0
 Const OLE_CREATE_LINK = 1
 Const OLE_SIZE_ZOOM = 3
 [Linked Object].OLETypeAllowed = OLE_LINKED ' Object type allowed
 [Linked Object].SourceDoc = [File Name] ' Specify source file
 [Linked Object].Action = OLE_CREATE_LINK ' Create linked object
 [Linked Object].SizeMode = OLE_SIZE_ZOOM ' Adjust control size
```

End Sub

2. View the Linked References table in Form view.
3. Change the File Name control to REDBRICK.BMP. The OLE object should update to display the REDBRICK.BMP bitmap.

#### REFERENCES

=====

Microsoft Access "User's Guide," versions 1.0 and 1.1, Chapter 13

Microsoft Access "User's Guide," version 2.0, Chapter 19

Microsoft Access "Building Applications," version 2.0, Chapter 13

Additional reference words: 1.00 1.10 2.00 bound

KBCategory: kbole

KBSubcategory: IntpOle

**INF: How to Use the Attributes Property for TableDef Objects**  
**Article ID: Q117536**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY  
=====

You can use the Attributes property of a TableDef object to determine specific table properties. For example, you can use the Attributes property to find whether a table is a system table or an attached table.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION  
=====

The Attributes property of a TableDef object specifies characteristics of the table represented by the TableDef object. The Attributes property is stored as a single Long number and is the sum of the following Long constants:

| Constant           | Description                                                                                                                                                |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DB_ATTACHEXCLUSIVE | For databases that use the Microsoft Jet database engine, indicates the table is an attached table opened for exclusive use.                               |
| DB_ATTACHSAVEPWD   | For databases that use the Jet database engine, indicates the user ID and password for the attached table should be saved with the connection information. |
| DB_SYSTEMOBJECT    | Indicates the table is a system table.                                                                                                                     |
| DB_HIDDENOBJECT    | Indicates the table is a hidden table (for temporary use).                                                                                                 |
| DB_ATTACHEDTABLE   | Indicates the table is an attached table from a non-Open Database Connectivity (ODBC) database, such as Microsoft Access or Paradox.                       |
| DB_ATTACHEDODBC    | Indicates the table is an attached table from an ODBC database, such as Microsoft SQL Server or ORACLE Server.                                             |

For a TableDef object, use of the Attributes property depends on the status of TableDef, as the following table shows:

| TableDef                          | Usage      |
|-----------------------------------|------------|
| Object not appended to collection | Read/write |
| Base table                        | Read-only  |
| Attached table                    | Read-only  |

When checking the setting of this property, you can use the AND operator to test for a specific attribute. For example, to determine whether a table object is a system table, perform a logical comparison of the TableDef Attributes property and the DB\_SYSTEMOBJECT constant.

#### Sample Code

-----

The following user-defined sample Access Basic function loops through all the tables in a database and displays a message box listing each table name and whether or not the table is a system table:

```
Option Compare Database 'Use database order for string comparisons.

Option Explicit

Function ShowTableAttribs()
 Dim DB As Database
 Dim T As TableDef
 Dim TType As String
 Dim TName As String
 Dim Attrib As String
 Dim I As Integer

 Set DB = DbEngine.Workspaces(0).Databases(0)

 For I = 0 To DB.TableDefs.Count - 1
 Set T = DB.TableDefs(I)
 TName = T.Name
 Attrib = (T.Attributes And DB_SYSTEMOBJECT)
 MsgBox TName & iif(Attrib, ": System Table", ": Not System Table")
 Next I
End Function
```

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 7, "Objects and Collections," pages 155-181

For more information about the Attributes property, search for "Attributes" then "Attributes Property" using the Microsoft Access Help menu.

Additional reference words: 2.00 dao

KBCategory: kbprg

KBSubcategory: PgmObj

## INF: How to Use the ColumnWidth Property

Article ID: Q111778

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes the ColumnWidth property, which applies to fields or columns in Datasheet view. In Microsoft Access version 1.1, this property is documented in the "The Secrets of Access Wizards" manual which is included with the Microsoft Access Distribution Kit. In Microsoft Access version 2.0, this property is documented in the online Help system.

### MORE INFORMATION

=====

The ColumnWidth property applies to fields or columns in Datasheet view. It is an integer that represents the column width in twips. You can specify a width, or use one of the following predefined settings:

| Setting | Description                                             |
|---------|---------------------------------------------------------|
| 0       | Hides the column                                        |
| -1      | Sizes the column to the default width (default setting) |
| -2      | Sizes the column to fit the visible text                |

Note that you can also set this property by choosing the Column Width command from the Layout menu in Microsoft Access version 1.x, or from the Format menu in Microsoft Access version 2.0, in Datasheet view.

If you set a field's ColumnWidth property to 0 to hide the field, that field's ColumnHidden property is set to -1. If you set the field's ColumnHidden property to 0 to show the field, the field's ColumnWidth property is restored to the setting it had before the field was hidden. If you hid the field by setting its ColumnWidth property to 0, the ColumnWidth property is set to -1, which is the default width. Note that the ColumnWidth property for a field is not available when the field's ColumnHidden property is set to -1.

Microsoft Access evaluates the properties for a table or query in Datasheet view when the table or query is first opened in Datasheet view. If the table or query is open in Datasheet view when you change a field's ColumnWidth property using Access Basic, the changes will not take effect until you close and then reopen the table or query in Datasheet view.

In Microsoft Access version 1.x, in order to change the ColumnWidth property of a field on a form, the form must be open in Design view. If the form is not open in Design view, the property is read-only. In Microsoft Access version 2.0, the form can be open in any view, and changes to the ColumnWidth property will take effect immediately.

The following example demonstrates how to set the ColumnWidth property for



the Customers form in Datasheet view. This setting will change the Address column's width to 1.5 inches (2160 twips):

```
Forms![Customers]![Address].ColumnWidth = 2160
```

#### REFERENCES

=====

Microsoft Access Distribution Kit, "The Secrets of Access Wizards," version 1.1, page 12

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

## **INF: How to Use the GetSystemMetrics() API Call**

Article ID: Q88922

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

In the Windows environment, various display resolutions may cause screens to appear out of proportion. As a developer, you can obtain the width and height of various elements of the window display using the Windows application programming interface (API) function GetSystemMetrics(). Incorporating this function into a Microsoft Access application gives the developer more information for 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.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The Windows GetSystemMetrics() API function retrieves information about the system metrics (the width and height of various display elements of a particular window). The GetSystemMetrics() function can also return flags that indicate whether a mouse is present or if the meaning of the left and right mouse buttons have been reversed. System metrics are dependent upon the system display, and may vary from display to display.

To use the GetSystemMetrics() functions, do the following:

1. Place the following Declare statement in the global section of the Access Basic Code window:

```
Declare Function GetSystemMetrics% Lib "user" (ByVal nIndex%)
```

2. Depending on what window property you want to determine, you must define the correct constant to pass to the GetSystemMetrics() function. Below are sample declarations of the constants and their 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_MOUSEPRESENT = 19 ' Non-zero if mouse hardware is
 ' installed
Const SM_PENWINDOWS = 41 ' Handle of Pen Windows dynamic link
 ' library if Pen Windows is
 ' installed
Const SM_SWAPBUTTON = 23 ' Non-zero if the left and right
 ' mouse buttons are swapped
```

The following sample call will return the height of a form's caption bar:

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

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

## **INF: How to Use the IIf() (Immediate If) Function**

**Article ID: Q106288**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

You may receive unexpected results when you are using the IIf() (Immediate If) function to evaluate an expression. Your results will depend on the use of double quotation marks in the function's arguments.

### MORE INFORMATION

=====

The following example demonstrates how the use of double quotation marks in the IIf() function's arguments can affect the results:

1. Open the sample database NWIND.MDB.
2. Create the following new query based on the Orders table:

Query: Test

-----  
FieldName: Sum(IIf("[Freight]>50",1,0))  
Total: Expression

3. Run the query. The result will be 1078 records, the total number of orders, not just those with a freight charge greater than 50.
4. View the query in Design view. Remove the double quotation marks so the query matches the following:

Query: Test

-----  
FieldName: Sum(IIf([Freight]>50,1,0))  
Total: Expression

5. Run the query again. The result will be 447 records, which is the number of records with a freight charge greater than 50.

The double quotation marks around the "[Freight] > 50" argument caused Microsoft Access to evaluate the expression as a string instead of as a comparison. A string will always evaluate to true, while a comparison will test for the condition to be met before returning a value.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ExrOthr

## **INF: How to Use the LIKE Operator in Parameter Queries**

**Article ID: Q88671**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0  
-----

### SUMMARY

=====

This article describes how to search for all the records that begin with a certain letter.

### MORE INFORMATION

=====

An easy way to prompt for a character to search for is to create a parameter query. The LIKE operator can be used with the wildcard symbol (\*) to accomplish this task.

First, create a query for the tables you want, and then place the following in the criteria for the field:

```
LIKE "[Enter the first char to search by:]|*"
```

Note that the Parameters dialog box has a display limited of 37 characters.

When you run this query you will be prompted with the message you specified in the LIKE statement. To find records that start with a "C", type "c" (without the quotation marks) and begin the search.

### REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryParm

## INF: How to Use the Query-by-Form (QBF) Technique

Article ID: Q95931

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to use a form to specify the criteria for a query in Microsoft Access. This technique is called query by form (QBF).

### MORE INFORMATION

=====

You use the QBF technique to create a query form in which you enter query criteria. The query form contains blank text boxes, each representing a field in a table that you want to query. You make entries in only the text boxes for which you want to specify search criteria.

The query form is usually very similar to a data entry form, as in the following sample query form:

First Name: \_\_\_\_\_  
Last Name: \_\_\_\_\_  
City: \_\_\_\_\_  
State: \_\_\_\_\_  
Zip Code: \_\_\_\_\_

You can enter any combination of search criteria in the form. You can specify only a City, a City and a State, only a Zip Code, or any other combination of values. Fields that you leave blank on the form are ignored (no search criteria are applied to those fields). When you choose the Search button, a query will be performed using the search criteria from your form.

The following example demonstrates how to create a query form:

1. Open the sample database NWIND.MDB.
2. Create the following new, unbound form and save it as QBF\_Form:

Form: QBF\_Form  
-----  
Text box:  
    ControlName: What Customer ID?  
Text box:  
    ControlName: What Employee ID?  
Command button:  
    Caption: Search  
    OnPush: QBF\_Macro

NOTE: In Microsoft Access version 2.0, the OnPush property is named OnClick.

3. Create the following new macro, and then save it as QBF\_Macro:

```
Macro: QBF_Macro

OpenQuery
 Query Name: QBF_Query
 View: Datasheet
 Data Mode: Edit
```

4. Create the following new query based on the Orders table, and then save it as QBF\_Query:

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code.

```
Query: QBF_Query

Field: Customer ID
 Sort: Ascending
 Criteria: Forms![QBF_Form]![What Customer ID?] Or _
 Forms![QBF_Form]![What Customer ID?] Is Null
Field: Employee ID
 Sort: Ascending
 Criteria: Forms![QBF_Form]![What Employee ID?] Or _
 Forms![QBF_Form]![What Employee ID?] Is Null
Field: Order ID
Field: Order Date
```

5. View the QBF form in Form view. Enter the following combinations of criteria, choosing the Search button after each combination:

| Customer ID | Employee ID | Result                        |
|-------------|-------------|-------------------------------|
| <blank>     | <blank>     | All 1078 orders               |
| AROUT       | <blank>     | 14 orders for AROUT           |
| AROUT       | 4           | 4 AROUT orders for employee 4 |
| <blank>     | 4           | 209 orders for employee 4     |

After you view the dynaset, or set of results, for each query, close the Datasheet window before beginning your next search. Each time you choose the Search button, the parameters in the QBF query filter the data according to the search criteria specified on the QBF query form.

#### Notes on the QBF Parameter Criteria

The sample QBF query above implements a criterion in the query as

```
Forms!FormName!ControlName Or Forms!FormName!ControlName Is Null
```

to filter the data. This criterion returns all matching records. If a criterion is null, all the records are returned for the specified field.

You can specify any of the following alternative criteria to return slightly different results:



NOTE: In the following sample criteria, an underscore (\_) is used as a line- continuation character. Remove the underscore when re-creating these criteria.

- Like "|Forms!FormName!ControlName|\*" Or \_  
Forms!FormName!ControlName Is Null

This criterion is the same as the QBF sample above, except that you can query using a wildcard. For example, if you enter "Jo" in a field using this criterion, the query will return everything in the field that begins with "Jo", including Johnson, Jones, Johanna, and so on, instead of returning only items with an exact match of "Jo".

- Between Forms!FormName!StartDate And Forms!FormName!EndDate Or \_  
Forms!FormName!StartDate Is Null

You can use this criterion to query a date field, using Start Date and End Date text boxes on the query form. Records whose start and end dates fall between the values you specify on the query form will be returned.

- Like Forms!FormName!ControlName & "\*" Or Is Null

This criterion returns both records that match the criterion and records that are null. If the criterion is null, all the records are returned. Note that the asterisk (\*) is considered a parameter, as it is part of a larger Like expression. Since the asterisk is a hard-coded criterion value (for example, Like "\*") records with null values are returned.

- Like IIf(IsNull(Forms!FormName![ControlName]), \_  
"\*,[Forms]![FormName]![ControlName])

This criterion returns all the records that match the criterion. If no criterion is specified in the query form, all records that are NOT null are returned.

- IIf(IsNull(Forms!FormName![ControlName]), \_  
[FieldName],[Forms]![FormName]![ControlName])

This criterion returns all the records that match the criterion. If no criterion is specified in the query form, all records that are NOT null are returned (the same results as the example above).

#### REFERENCES

=====

For more information on complex criteria optimization and returning nulls from parameter queries, query on the following words here in the Microsoft Knowledge Base:

parameter query and like criteria

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryParm



## **INF: How to Use the Status Bar Progress Meter**

**Article ID: Q103404**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

The SysCmd() function in Microsoft Access can be used to create a progress meter in the status bar to give a visual representation of the progress of an operation. When you perform an operation with a known duration or number of steps, you can use the SysCmd() function to visually represent the operation's progress.

The following information documents the use of the SysCmd() function in Microsoft Access to display a progress meter.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

### MORE INFORMATION

=====

The syntax of the SysCmd() is as follows:

```
SysCmd(<action> [, <text>][, <value>])
```

Where

- <action> is a numeric expression identifying the type of action to take. The expression is one of the following:
  - 1 - Initialize the progress meter.
  - 2 - Update the progress meter with the specified value.
  - 3 - Remove the progress meter.
- <text> is a string expression identifying the text that will appear left-aligned in the status bar to the left of the progress meter.
- <value> is a numeric expression that controls the display of the meter. This is required when the action is 1 or 2.

When the action is 1, the value indicates the maximum number the meter should display, with the maximum value for the meter indicating 100 percent.

When the action is 2, the value is used to calculate and update the percentage complete in the progress meter.

NOTE: The SysCmd() function returns NULL, which is not used.

#### Initializing the Progress Meter

-----

When the <action> argument is 1, the <value> argument is the maximum value of the meter, or 100 percent. To display a progress meter with a maximum value of 1000 initialized, type the following in the Immediate window:

```
? SysCmd(1, "This is my meter!", 1000)
```

#### Updating the Progress Meter

-----

When the <action> argument is 2, the <value> argument is used by SysCmd() to calculate the percentage displayed by the meter. To update the progress meter to 25 percent complete, type the following in the Immediate window:

```
? SysCmd(2, 250)
```

#### Removing the Progress Meter

-----

When the <action> argument is 3, the progress meter is removed from the status bar. To remove the meter, type the following in the Immediate window:

```
? SysCmd(3)
```

#### Using SysCmd() in an Access Basic Function

-----

The following sample function will open the Customers table in the sample database NWIND.MDB. The function will display a list of the contact names in the table. As the names are displayed, a meter will progress in the status bar indicating the relative progress of the function.

1. Open the sample database NWIND.MDB.
2. Create a new module called "Meter Test."
3. Place the following statement in the declarations section:

```
Option Explicit
```

4. Create a new function called "Meter ()" as follows:

```
Function Meter ()
 Dim MyDB As Database, MyTable As Table
 Dim Count As Long
 Dim Progress_Amount As Integer, RetVal As Variant

 Set MyDB = CurrentDB()
```

```

Set MyTable = MyDB.OpenTable("Customers")

' Move to last record of the table to get the total
' number of records.
MyTable.MoveLast
Count = MyTable.RecordCount

' Move back to first record.
MyTable.MoveFirst

' Initialize the progress meter.
RetVal = SysCmd(1, "Reading Data...", Count)

' Enumerate through all the records.
For Progress_Amount = 1 To Count
 ' Update the progress meter.
 RetVal = SysCmd(2, Progress_Amount)

 ' Print the contact name in the Immediate window.
 Debug.Print MyTable![contact name]

 ' Goto the next record.
 MyTable.MoveNext
Next Progress_Amount

' Remove the progress meter.
RetVal = SysCmd(3)
End Function

```

To run the function, type the following in the Immediate window:

```
? Meter()
```

#### REFERENCES

=====

Microsoft Access Distribution Kit, "The Secrets of Access Wizards,"  
page 30

Additional reference words: 1.00 1.10 2.00 thermometer statusbar  
KBCategory: kbprg  
KBSubcategory: PgmOthr

## **INF: How to Use the Windows OpenFile Dialog Box**

**Article ID: Q96114**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to use the OpenFile dialog box from Access Basic.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The OpenFile common dialog box can be implemented in Microsoft Access through the use of the Windows GetOpenFileName() application programming interface (API) function. Using this function will simplify programming issues for the developer. This function creates a system-defined dialog box, familiar throughout Windows, that makes it possible for the user to select a file to open. This function will return a valid file to the programmer that is fully qualified with the path name. The developer can customize how the system will handle specific situations, such as specifying that the file must exist when the user wants to save a file through the use of flags.

To use the code below, paste the function and declarations into a module.

### NOTES:

- In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.
- You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.

```
'-----
' Global Declaration Section
'-----
Option Compare Database
Option Explicit

Type tagOPENFILENAME
 lStructSize As Long
```

```

 hwndOwner As Integer
 hInstance As Integer
 lpstrFilter As Long
 lpstrCustomFilter As Long
 nMaxCustFilter As Long
 nFilterIndex As Long
 lpstrFile As Long
 nMaxFile As Long
 lpstrFileName As Long
 nMaxFileName As Long
 lpstrInitialDir As Long
 lpstrTitle As Long
 Flags As Long
 nFileOffset As Integer
 nFileExtension As Integer
 lpstrDefExt As Long
 lCustData As Long
 lpfnHook As Long
 lpTemplateName As Long
End Type

Declare Function GetOpenFileName% Lib "COMMDLG.DLL" (_
 OPENFILENAME As tagOPENFILENAME)
Declare Function GetSaveFileName% Lib "COMMDLG.DLL" (_
 OPENFILENAME As tagOPENFILENAME)
Declare Function lstrcpy& Lib "Kernel" (ByVal lpDestString As Any, _
 ByVal lpSourceString As Any)

Dim OPENFILENAME As tagOPENFILENAME

Global Const OFN_READONLY = &H1
Global Const OFN_OVERWRITEPROMPT = &H2
Global Const OFN_HIDEREADONLY = &H4
Global Const OFN_NOCHANGEDIR = &H8
Global Const OFN_SHOWHELP = &H10
Global Const OFN_ENABLEHOOK = &H20
Global Const OFN_ENABLETEMPLATE = &H40
Global Const OFN_ENABLETEMPLATEHANDLE = &H80
Global Const OFN_NOVALIDATE = &H100
Global Const OFN_ALLOWMULTISELECT = &H200
Global Const OFN_EXTENSIONDIFFERENT = &H400
Global Const OFN_PATHMUSTEXIST = &H800
Global Const OFN_FILEMUSTEXIST = &H1000
Global Const OFN_CREATEPROMPT = &H2000
Global Const OFN_SHAREAWARE = &H4000
Global Const OFN_NOREADONLYRETURN = &H8000
Global Const OFN_NOTESTFILECREATE = &H10000

Global Const OFN_SHAREFALLTHROUGH = 2
Global Const OFN_SHARENOWARN = 1
Global Const OFN_SHAREWARN = 0

'-----
' Open Common Dialog Function
'-----
Function OpenCommDlg ()
 Dim Message$, Filter$, FileName$, FileTitle$, DefExt$

```

```

Dim Title$, szCurDir$, APIResults%

'*Define the filter string and allocate space in the "c" string
Filter$ = "Access (*.mdb)" & Chr$(0) & "*.MDB;*.MDA" & Chr$(0)
Filter$ = Filter$ & "Text (*.txt)" & Chr$(0) & "*.TXT" & Chr$(0)
Filter$ = Filter$ & "Batch (*.bat)" & Chr$(0) & "*.BAT" & Chr$(0)
Filter$ = Filter$ & Chr$(0)

'* Allocate string space for the returned strings.
FileName$ = Chr$(0) & Space$(255) & Chr$(0)
FileTitle$ = Space$(255) & Chr$(0)

'* Give the dialog a caption title.
Title$ = "My File Open Dialog" & Chr$(0)

'* If the user does not specify an extension, append TXT.
DefExt$ = "TXT" & Chr$(0)

'* Set up the default directory
szCurDir$ = CurDir$ & Chr$(0)

'* Set up the data structure before you call the GetOpenFileName

OPENFILENAME.lStructSize = Len(OPENFILENAME)

'If the OpenFile Dialog box is linked to a form use this line.
'It will pass the forms window handle.

OPENFILENAME.hwndOwner = Screen.ActiveForm.hWnd

'If the OpenFile Dialog box is not linked to any form use this line.
'It will pass a null pointer.

OPENFILENAME.hwndOwner = 0&

OPENFILENAME.lpstrFilter = lstrcpy(Filter$, Filter$)
OPENFILENAME.nFilterIndex = 1
OPENFILENAME.lpstrFile = lstrcpy(FileName$, FileName$)
OPENFILENAME.nMaxFile = Len(FileName$)
OPENFILENAME.lpstrFileTitle = lstrcpy(FileTitle$, FileTitle$)
OPENFILENAME.nMaxFileTitle = Len(FileTitle$)
OPENFILENAME.lpstrTitle = lstrcpy(Title$, Title$)
OPENFILENAME.Flags = OFN_FILEMUSTEXIST Or OFN_READONLY
OPENFILENAME.lpstrDefExt = lstrcpy(DefExt$, DefExt$)
OPENFILENAME.hInstance = 0
OPENFILENAME.lpstrCustomFilter = 0
OPENFILENAME.nMaxCustFilter = 0
OPENFILENAME.lpstrInitialDir = lstrcpy(szCurDir$, szCurDir$)
OPENFILENAME.nFileOffset = 0
OPENFILENAME.nFileExtension = 0
OPENFILENAME.lCustData = 0
OPENFILENAME.lpfHook = 0
OPENFILENAME.lpTemplateName = 0

'* This will pass the desired data structure to the Windows API,
'* which will in turn it uses to display the Open Dialog form.

```



```
APIResults% = GetOpenFileName(OPENFILENAME)

If APIResults% <> 0 Then

 '* Note that FileName$ will have an embedded Chr$(0) at the
 '* end. You may wish to strip this character from the string.

 FileName$ = Left$(FileName$, InStr(FileName$, Chr$(0)) -1)

 Message$ = "The file you chose was " + FileName$
Else
 Message$ = "No file was selected"
End If

MsgBox Message$
```

End Function

#### REFERENCES

=====

"Microsoft Windows Software Development Kit Programmer's Reference,  
Volume 2: Functions," version 3.1

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: How to Use TimeValue() to Calculate Difference in Times**  
**Article ID: Q100141**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to use the TimeValue() function to find the difference between two values that have Date/Time data types and that occur within a 24-hour period.

MORE INFORMATION

=====

To set up an example that uses the TimeValue() function, create a form with the following text boxes:

Object: Text Box

-----

ControlName: Start\_Time

Object: Text Box

-----

ControlName: End\_Time

To calculate the difference between the times in these two fields, create a third text box with the following properties:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code.

Object: Text Box

-----

ControlName: Time\_Difference

ControlSource: =Format(TimeValue([End\_Time])\_  
-TimeValue([Start\_Time]),"Short Time")

Note that in the example above, the Format() function is used to format the time difference. Short Time displays the difference in hours and minutes.

REFERENCES

=====

For more information on formatting values, search for "format" then "format, format\$" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr



## **INF: How to Wait for a Shelled Process to Finish**

**Article ID: Q99940**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

When you are using the Shell() function or the RunApp macro action to run another program or process, Microsoft Access does not wait for the shelled process to finish before processing the next line in the macro or function. This causes problems for macros and functions whose subsequent actions depend on the results of the shelled process.

This article includes a function called WaitShell(), which, given an application name or process to run, starts the process and waits for it to terminate.

### MORE INFORMATION

=====

The following Microsoft Access function, WaitShell(), uses GetModuleUsage, a Microsoft Windows API function, to determine if the shelled process has terminated. The Shell() function, if it successfully starts the process, returns a handle to the process module. If the process is no longer running, the handle is invalid and GetModuleUsage returns a value of 0. WaitShell() simply loops until the module is no longer valid.

NOTE: You may have the following GetModuleUsage Windows API function defined in an existing Microsoft Access library. If you receive a duplicate procedure name error message, delete the Declare statement from your code.

Option Explicit

Declare Function GetModuleUsage% Lib "Kernel" (ByVal hModule%)

```
Function WaitShell(AppName as String)
 Dim hMod as Integer
 hMod = Shell(AppName, 1)
 If (Abs(hMod) > 32) then
 While (GetModuleUsage(hMod))
 DoEvents
 Wend
 Else
 MsgBox "Unable to start " & AppName
 End If
End Function
```

To test this function, create and run the following Microsoft Access macro

Action

```

RunCode
MsgBox
```

```
Macro Actions
```

```

RunCode
 Function Name: =WaitShell("Command.com")
MsgBox
 Message: Done!
```

-or-

run the following Microsoft Access function:

```
Function TestWaitShell()
 x=WaitShell("Command.com")
 MsgBox "Done!"
End Function
```

Both examples above launch instances of the DOS prompt, which remain until you terminate them by typing "exit" at the command line and pressing ENTER.

#### REFERENCES

=====

Microsoft Windows "Programmer's Reference, Volume 2: Functions,"  
version 3.1, pages 403-404

Additional reference words: 1.00 1.10 2.00 terminate basic shell  
KBCategory: kbprg  
KBSubcategory: PgmOthr

**INF: Implementing Counter Values on Attached SQL Server Tables**  
**Article ID: Q100166**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

When you create a table in Microsoft Access and port it to SQL Server, some tasks are difficult to implement because SQL Server does not include a Counter data type. This article discusses how to maintain the counter functionality of an attached SQL Server table.

MORE INFORMATION

=====

With a multiuser system, you must take special precautions. Assume that the Counter column in SQL Server is called ID and that ID is the unique index for this table. In most cases, users try to simulate counters with a SQL Server OnInsert trigger that computes the maximum value and adds 1. However, Microsoft Access uses that index (bookmark) to point to the record and is not aware that the trigger is changing the index value. Thus, Microsoft Access loses track of that record until the next time the table is queried.

Since SQL Server triggers do not work and SQL Server does not have a unique Counter data type as does Microsoft Access, you must use an Access Basic function to control the counter value, as in the procedure below:

1. In a single-column, single-row Microsoft Access table, store a value that represents the next highest counter value.

NOTE: You must store the table in Microsoft Access, since the OpenTable action does not work on an attached SQL Server table and, therefore, cannot open the table exclusively. For more information on the OpenTable action, refer to the Microsoft Access "Language Reference" manual.

2. In the data entry form, apply a function to the BeforeUpdate event so that the function completes the following tasks:

- Open the table that contains the counter value
- Grab the value for the current record
- Add 1 to the value in the table
- Close the table

Each user must have exclusive access to the table to ensure that no duplicate values are recorded. You must do error checking when trying to open the table exclusively because another user may have the table open at the same time. If this occurs, the program loops and retries several times before alerting the first user.

3. Once the value has been retrieved, set the ID of the current record equal to that value and commit the record to the SQL Server database.

#### REFERENCES

=====

For more information about, and an example of, creating a unique custom counter, query on the following words here in the Microsoft Knowledge Base:

table and counter and custom and unique

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr

**INF: Imported Microsoft Excel Date Fields Are Off by 4 Years**  
**Article ID: Q103996**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Excel versions 2.0, 3.0, 4.0, and 5.0
- 

SUMMARY

=====

If date fields imported from a Microsoft Excel spreadsheet are off by four years, the 1904 date system was used on the original spreadsheet. This setting can be verified in Microsoft Excel by choosing Calculation from the Options menu. In the Sheet Options group, there is a check box labeled "1904 Date System." If the box is selected, the spreadsheet is based on the 1904 date system. If the box is not selected, the spreadsheet is based on the 1900 date system.

MORE INFORMATION

=====

The date systems used by Microsoft Excel can be based on one of two different dates. By default, a serial number of 1 in Microsoft Excel represents January 1, 1900. The default for the serial number 1 can be changed to represent January 2, 1904. This option was included in Microsoft Excel for Windows to make it compatible with Excel for the Macintosh, which defaults to January 2, 1904.

If the spreadsheet you want to import into Microsoft Access was based on the 1904 date system, complete the following steps before importing the spreadsheet into Microsoft Access:

1. Open the spreadsheet in Microsoft Excel.
2. From the Options menu (or from the Tools menu in version 5.0), choose Calculation.
3. In the Sheet Options group, clear the box labeled "1904 Date System."
4. Save and close the spreadsheet.

The spreadsheet is now ready to be imported into Microsoft Access.

If you do not have Microsoft Excel available to switch the date system before importing the spreadsheet, you can use the following steps to perform an Update Query to correct the dates after the spreadsheet has been imported into Microsoft Access:

1. Create a new query based on the table that was created when you imported the spreadsheet.
2. Drag the DateField name to the Field row in the query grid. For this example, the date field's name is "DateField."



3. From the Query menu, choose Update.
4. In the Update To field, enter the following:

[DateField] + 1462

5. From the Query menu, choose Run.

A message box appears and tells you how many records were updated. Your dates should now appear correctly.

#### REFERENCES

=====

For more information about date serial numbers, search for "NOW function" then "NOW" using the Microsoft Excel for Windows, version 5.0 Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IsmEx15

**INF: Importing an Entire Sheet from a MS Excel 5.0 Workbook**  
**Article ID: Q115190**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Excel for Windows, version 5.x
- 

SUMMARY

=====

This article describes how to use Access Basic code to import an entire worksheet from a Microsoft Excel version 5.0 workbook without specifying a range.

MORE INFORMATION

=====

To import an entire worksheet, refer to the worksheet without a named range, but include an exclamation point (!). For example, the following sample code will import Sheet5 from a workbook named T.XLS.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function ImportXL5 ()
 DoCmd TransferSpreadsheet
 A_IMPORT,5,"TestTable","T.XLS",True,"Sheet5!"
End Function
```

If you do not specify a value for the last argument, Microsoft Access will import the first worksheet that it finds in the workbook. If you specify a range, that range will be imported from the first worksheet in the workbook. To specify a range from a specific worksheet, use the syntax in the following example:

```
Sheet5!R2C1:R15:C5
```

Additional reference words: 2.00

KBCategory: kbinterop

KBSubcategory: IsmExl5

## INF: Importing DataEase Data Files

Article ID: Q94194

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Microsoft Access does not import DataEase data files directly, but you can export your DataEase data to a formatted disk file that Microsoft Access can import. The information in this article applies to DataEase versions 4.2, 4.5, and 4.53. DataEase is a character-based relational database system.

To export DataEase data from DataEase, create a DQL report, specify the Format as export, and set the Print Style to print to a disk file. Select one of the following export formats to create a file Microsoft Access can import:

- Lotus (creates a .WKS file)
- Variable Length (creates a delimited ASCII text file)
- Fixed Length (creates a fixed-width ASCII text file)

### MORE INFORMATION

=====

#### Export Using DataEase

-----

To create a formatted disk file in DataEase 4.x, create a DQL report using the following steps. For more detailed information about creating reports, please refer to the DataEase documentation.

1. Define the report specifying the fields you want to include in the output.
2. Specify the report Format as Export by choosing either Define Format or Modify Format under either DQL Advanced Processing or Query by Example - Quick Reports.

Select 7. Export under Define Format or Modify Format. You should see the following list of the export formats DataEase supports.

- \* 1. Lotus (wks)
- 2. GrafTalk
- 3. MultiMate
- 4. WordPerfect
- 5. Mail-Merge
- \* 6. Variable Length
- \* 7. Fixed Length
- 8. DIF

Choose 1, 2, or 6. Depending on your choice, DataEase may ask you if you

want to include field names in the exported file.

3. Specify the Print Style for the report. Enter the Report Destination as option #3 Disk to print to a disk file. Then enter a valid file name in the Filename field.

#### Import Using Microsoft Access

-----

1. Start Microsoft Access and open your database.
2. From the File menu, choose Import.
3. In the Data Source box, select the import format that matches the export format you chose in DataEase.

#### Choosing Export/Import Format

-----

Here are some things to consider when choosing an export/import format.

Microsoft Access can import Lotus (.WKS) files and create a table with field names if you included the field names when exporting in DataEase. During the import process, Microsoft Access asks you if the first row contains field names. Most DataEase data types should convert to appropriate Microsoft Access field types.

If you choose the Variable Length format when exporting, DataEase will ask you to enter the field separator. Enter a comma (,). Microsoft Access recognizes the comma as the default field separator.

If you choose the Fixed Length format when exporting in DataEase, you will need to create an Import Specification file in Microsoft Access prior to importing the file.

For additional information registered DataEase users can contact DataEase International HotLine Support at 1-203-374-2825.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 4, "Importing, Exporting, and Attaching," pages 72-81

Microsoft Access "User's Guide," version 2.0, Chapter 9, "Importing, Exporting, and Attaching," pages 175-188

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr

**INF: Importing Word Print Merge Data Files to Microsoft Access**  
**Article ID: Q96899**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Word for Windows, versions 1.0, 1.1, 1.1a, 2.0, 2.0a, 2.0a-CD, 2.0b, 2.0c
- 

SUMMARY

=====

This article describes how to import data from a Word for Windows print merge data file for use in Microsoft Access.

MORE INFORMATION

=====

In order to use data from a Word for Windows print merge data file in Microsoft Access, the file must be saved as a comma- or tab-delimited unformatted text file.

Word for Windows can perform print merges on data contained in files that are comma- or tab-delimited, or on data formatted from a table. If the data is already comma- or tab-delimited, you must take care to ensure the file is saved as an unformatted text file.

If the data is in table form, the table must first be converted to a tab-delimited format before saving it as a text file.

NOTE: In Word for Windows, you can create multiple-line information within a table cell that contains carriage returns (paragraph marks). The carriage returns must be removed before converting the table to a tab-delimited format.

Converting Data from a Table to a Tab-Delimited Format

-----

Word for Windows 1.x:

1. Open the data document in Word for Windows.
2. Position the insertion point anywhere inside the table.
3. Press ALT+Keypad 5 to select the entire table (make sure that NUM LOCK is turned off).
4. From the Insert menu, choose Table To Text.
5. In the Convert Table To box, select Tab Delimited, then choose OK.

Word for Windows 2.x:

1. Open the data document in Word for Windows.

2. Position the insertion point anywhere inside the table.
3. From the Table menu, choose Select (or press ALT+Keypad 5).
4. From the Table menu, choose Convert Table To Text.
5. Choose the Tabs button from the Separate Text With group and choose OK.

#### Saving the Data File as an Unformatted Text-Only File

---

Word for Windows 1.x:

1. From the File menu, choose Save As, and enter an unique filename in the Save File Name box.
2. Choose the Options button.
3. From the File Format box, select Text Only, and choose OK.

Word for Windows 2.x:

1. From the File menu, choose Save As, and type a unique filename in the File Name box.
2. In the Save File As Type box in the lower left corner, select Text Only, and choose OK.

#### Removing Unwanted Carriage Returns from a Table-Formatted Data File

---

Word for Windows 1.x:

1. In the Search For box type "^p" (the caret symbol (^) is generated by pressing SHIFT+6).
2. In the Replace With box press SPACEBAR once.
3. Clear the Confirm Changes check box and choose the OK button.

Word for Windows 2.x:

1. From the Edit menu, choose Replace.
2. In the Find What box, type "^p" (the caret symbol (^) is generated by pressing SHIFT+6).
3. In the Replace With box, press SPACEBAR once.
4. Choose the Replace All button.

#### Importing the Text Data File into Microsoft Access

---

1. From the File menu, choose Import.

2. Select Text (Delimited) from the Data Source list and choose OK.
3. Select the Word for Windows text file to import from the file list and choose the Import button.
4. Select First Row Contains Field Names if the first row in the Word data file contains a list of field names.
5. Choose the Options button.
6. Choose {tab} in the Field Separator box.
7. Choose the OK button to begin the import.

Additional reference words: 1.00 1.10 2.00 winword  
KBCategory: kbinterop  
KBSubcategory: IntpPrtm

**INF: Importing, Exporting, and Attaching FoxPro 2.0, 2.5 Data**  
**Article ID: Q100177**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SUMMARY

=====

In Microsoft Access versions 1.1 and 2.0, you can import from, export to, and attach data in FoxPro versions 2.0 and 2.5.

MORE INFORMATION

=====

If you select either Import or Attach Table from the File menu, you can choose FoxPro 2.0 or FoxPro 2.5 from the Data Source list box.

When you attach a FoxPro file, you can also associate the appropriate FoxPro index (.IDX or .CDX file). The procedure to import, export, and attach files is the same for FoxPro files as it is for dBASE and Paradox files.

dBASE and Paradox are manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, chapter 4, "Importing, Exporting, and Attaching," pages 60-65

Additional reference words: 1.10 2.00 Import/Export fox idx cdx  
KBCategory: kb3rdparty  
KBSubcategory: IsmXbase



## **INF: Incrementing the Numeric Portion of a String**

Article ID: Q88169

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a formula that you can use to set the default value of a bound text field in a form to the next higher numeric value (default value = default value + 1).

The sample formula in this article follows these steps:

1. Find the highest numeric value used in a table.
2. Strip any leading text characters.
3. Add 1 to the value found in step 1.
4. Reconnect the leading text characters.

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

### MORE INFORMATION

=====

Use the following sample procedure to increase the default value of a bound text field on a form:

1. Open an existing database, or create a new database.
2. Create a new table.
3. Add a field called Book ID with a Text data type. From the Edit menu, choose Set Primary Key.
4. Add a second field of any data type to the table.
5. Save the table as Increment.
6. Switch to Datasheet view and enter the following values:

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

7. Close the table.
8. Create a new form based on the Increment table.

9. From the View menu, choose Field List.
10. Drag the Book ID field and the second field from the field list to the form.
11. Change the form's DefaultView property to Single Form.
12. Set the DefaultValue property of the Book ID text box to the following:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
"BO-" & Right(DMax("[Book ID]","Increment"),Len(DMax_
 ("[Book ID]","Increment"))-3)+1
```

13. Switch to Form view and enter a new record.

The components of the formula for the DefaultValue property are:

- DMax("[Book ID]","Increment") locates the highest value in the table.
- Len(DMax(...)) finds the length of the value returned by the DMax() function.
- Right(DMax(...),Len(...)-3) strips the three leading text characters.
- +1 increments the result of the Right() function.
- "BO-" concatenates the text "BO-" on to the incremented number.

Note that this example works correctly when the form's DefaultView property is Single View; it may not work correctly in Continuous View. When you move to a new record and begin to enter data, Microsoft Access displays the next empty record. The default values for this record are calculated before the record you are currently editing is committed. Because the DMax() function calculates the maximum value from those records stored in the table, the Book ID of the record you are currently editing is repeated.

If you are working in a multiuser environment, it is possible that more than one user may receive the same calculated Book ID value. Although you can manually change the Book ID, you can also maintain the highest numeric value in a separate table, using a macro or Access Basic code.

#### REFERENCES

=====

For more information on multiuser counters, query on the following words here in the Microsoft Knowledge Base:

multiuser and counter

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbusage

KBSubcategory: TblPriky

**INF: Introduction to Windows Programming for MS-DOS Programmer**  
**Article ID: Q88164**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

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
 OnPush: =CloseProgram()
Text Box: DisplayWindow
```

NOTE: OnPush is a property of Microsoft Access version 1.x command buttons that gives you the ability to invoke an Access Basic procedure or macro. In Microsoft Access 2.0, the property is called OnClick.

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.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: Launching Doc with Windows API ShellExecute() Function**  
**Article ID: Q121157**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can use the Windows API ShellExecute() function to start the application associated with a given document extension without knowing the name of the associated application. For example, you could start the Paintbrush program by passing the filename ARCADE.BMP to the ShellExecute() function.

MORE INFORMATION

=====

The following example demonstrates how to start an application or load a document into its associated application. The Windows API ShellExecute() function is different from the Access Basic Shell() function in that you can pass the ShellExecute() function the name of a document and it will launch the associated application and then pass the filename to the application. You can also specify the working directory for the application.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Create a new module, and then enter the following lines in the module's Declarations section:

```
Option Explicit
Declare Function ShellExecute Lib "SHELL" (ByVal hwnd%, ByVal _
lpszOp$, ByVal lpszFile$, ByVal lpszParams$, ByVal lpszDir$, _
ByVal fsShowCmd%) As Integer
Declare Function GetDesktopWindow Lib "USER" () As Integer
```

2. Create the following procedure in the module:

```
Function StartDoc (DocName As String)
 StartDoc = ShellExecute(GetDesktopWindow(), "Open", DocName, _
 "", "C:\", 1)
End Function
```

3. From the View menu, choose Immediate Window.
4. In the Immediate window, type the following line and then press ENTER:

```
? StartDoc("ARCADE.BMP")
```

The function will start the Paintbrush application, which will then load the ARCADE.BMP file.



## Notes

-----

- The return value for the StartDoc() function is the same as for the Shell() function. It is the Windows instance value of the application that was started.
- The ShellExecute() function returns the value 31 if there is no association for the specified file type or if there is no association for the specified action within the file type. Other error values are:

| Error Value | Meaning                                                                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------|
| 2           | The file was not found.                                                                                           |
| 3           | The path was not found.                                                                                           |
| 8           | There was insufficient memory to start the application.                                                           |
| 10          | The Windows version was incorrect.                                                                                |
| 11          | The executable file was invalid. Either it was not a Windows application or there was an error in the .EXE image. |

## REFERENCES

=====

Microsoft Windows SDK "Programmer's Reference, Volume 2: Functions,"  
pages 901-904

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: Like vs. = with Wildcard Characters in Query Searches**  
**Article ID: Q88166**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

When you are searching a string using wildcards (for example, \* or ?), you must use Like or Not Like in the Criteria field, rather than using = or <>. When you use the = or <> operators, Microsoft Access treats them as actual characters to search for.

MORE INFORMATION

=====

If you have a table with one column called Col\_Name and the following four records

```
Foos
Foosball
Foos ball
Foos*
```

and you make a query based on the table with the criteria

```
"Foos*"
```

your query will return:

```
Foos*
```

But, if your criteria is

```
Like "Foos*"
```

your query will return the following four records:

```
Foos
Foosball
Foos ball
Foos*
```

The SQL statements built by Microsoft Access for these queries are as follows:

The SQL statement using Like is:

```
SELECT DISTINCTROW tablename.Col_Name
FROM [tablename]
WHERE ((tablename.Col_Name Like "foos*"));
```

The SQL statement using = is:

```
SELECT DISTINCTROW tablename.Col_Name
FROM [tablename]
WHERE ((tablename.Col_Name="foos*"));
```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryOthr

## **INF: Limit on AND's in SQL Select Statement**

**Article ID: Q92690**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

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 version 1.x returns the following error message when you exceed this limit:

Out of Memory

Microsoft Access version 2.0 returns the following error message when you exceed this limit:

Query is too complex

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: QryProb

**INF: Limitations on Setting Report Properties at Run Time**  
**Article ID: Q114510**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

Microsoft Access version 2.0 supports using macros or code to set certain properties at run time for reports.

MORE INFORMATION

=====

To prevent possible problems during formatting and printing, the following limitations have been imposed for setting properties at run time for reports:

- The following properties can only be set from the OnOpen property (not from OnFormat, OnPrint, or OnRetreat):
  - "Datasource" properties: RecordSource, ControlSource, LinkChildFields, and LinkMasterFields.
  - The RecordSource property of a SubReport/SubForm embedded in a report can only be set from the OnOpen of the SubReport/SubForm. This property is Read Only from the OnOpen of the Main Report.
  - Any action in any property that would cause the report width to grow. This includes moving a control past the current report width. This is restricted because you could inadvertently increment the number of pages in the middle of a print job.
- The following properties cannot be set at run time at all:
  - The SourceObject property for subforms and subreports.
  - "Event" properties: OnOpen, OnFormat, OnPrint, and so on.
  - RunningSum
  - HideDuplicates
  - CanGrow and Canshrink
  - KeepTogether
  - ForceNewPage
  - RowSource for Graph objects
  - Height property of the Section

- The following property can be set from the OnFormat property, but not the OnPrint property:

- Height property of a control.

All other properties can be set at run time (in a report's OnOpen, OnFormat, or OnPrint property, for example).

NOTE: All the restrictions in this article also apply to form printing.

#### REFERENCES

=====

For more information about designing reports, search for "setting properties" then "Setting Properties Using Code, or search for "report properties" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: RptProp

## **INF: List of Articles Relating to MSACCESS.INI File**

**Article ID: Q102673**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

The following is a list of articles found in the Microsoft Knowledge Base that relate in some way to the MSACCESS.INI file:

| ARTICLE# | TITLE & DESCRIPTION                                                                                                                |
|----------|------------------------------------------------------------------------------------------------------------------------------------|
| -----    | -----                                                                                                                              |
| Q88175   | INF: Creating, Debugging, and Using an Access Library<br>(Refers to the [Libraries] section)                                       |
| Q88648   | INF: Easy Way to Add Items to the Help Menu<br>(Refers to the [Menu Add-Ins] section)                                              |
| Q90105   | INF: Could not Find SYSTEM.MDA or UTILITY.MDA File<br>(Refers to the [Options] section)                                            |
| Q98224   | PRB: Errmsg: Couldn't Open Table MSysAccounts<br>(Refers to the [Options] section)                                                 |
| Q96996   | INF: How to Change the Font for the Zoom Box<br>(Refers to the [Options] section)                                                  |
| Q90170   | INF: How to Make Access Opening Screen Display<br>(Refers to the [Microsoft Access] section, Tutorial<br>setting only)             |
| Q95051   | PRB: Unable to Connect to Attached SQL Server Table,<br>Error Msg<br>(Refers to the [SQLSERVER] section of the WIN.INI<br>file)    |
| Q101323  | INF: Five Parameters to Set in ISAM Section of<br>MSACCESS.INI<br>(Refers to the [ISAM] section)                                   |
| Q95388   | INF: Doc Err: dBASE SET DELETED Characteristic in<br>MS Access<br>(Refers to the [dBASE ISAM] section, Deleted<br>setting only)    |
| Q98805   | PRB: Internal Database Error (-5207) When Attaching<br>Tables<br>(Refers to the [Paradox ISAM] section,<br>CollatingSequence only) |
| Q100176  | INF: Upper Limit on QueryTimeout and LoginTimeout<br>(Refers to the [ODBC] section)                                                |
| Q100972  | INF: Forcing Access to Use Snapshot Method on Attached                                                                             |

Data  
(Refers to the [ODBC] section)

Q95050 PRB: Data Source Not Found, No Default Driver Specified  
Error  
(Refers to the ODBC.INI file)

Q103173 PRACC9307: Options in Form and Report Wizards Do Not  
Display  
(Refers to the [Form Wizards], [Report Wizards] and  
[Libraries] sections)

Additional reference words: 1.00 1.10 2.00 Initialization x-ref

KBCategory: kbsetup

KBSubcategory: StpOthr



**INF: List of Categories and Examples from SOLUTION.MDB**  
**Article ID: Q111782**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

The Solutions sample application (SOLUTION.MDB) included with Microsoft Access version 2.0 contains many examples of common tasks that you can use to help you design your database applications. This article lists all the available categories and examples in the Solutions sample application.

MORE INFORMATION

=====

These are the categories and examples in the Solutions sample application. Each category is followed by all the examples for that category:

Build a Microsoft Windows interface

- 
- Change toolbars in response to events
  - Close all database objects when you exit the application
  - Create a custom menu bar for a form
  - Create a main menu form
  - Create a startup form
  - Create a toolbar that has custom buttons
  - Create an About dialog box
  - Create an informational form that users can turn off
  - Get information from a user in a dialog box

Sample forms

- 
- About Solutions
  - Add All To List
  - Add Category
  - Edit Products
  - Employee Sales Dialog Box
  - Enter or Edit Products
  - Find Customers
  - Limit List Contents
  - Reattach NWIND
  - Sales Totals
  - Show Sales
  - Solutions
  - Solutions Intro
  - Startup

Work with forms and controls

- 
- Capitalize one or more letters in a field automatically
  - Change the color of a control based on the value in the control
  - Change the color of a section and control at run time

- Clear the values in unbound text boxes and a subform
- Control when you go to a new record
- Create a form in which you can edit but not enter records
- Create a form in which you can enter or edit data
- Create a main menu form
- Create a startup form
- Create an About dialog box
- Create an informational form that users can turn off
- Get information from a user in a dialog box
- Remove the minimize and maximize buttons from a form
- Update a list to show current data

Get more mileage from combo boxes, list boxes, subforms and subreports

---

- Add "(all)" to a list
- Add a new record to a list
- Find a record based on a value you select from a list
- Limit the contents of a list based on the value selected from an option group
- Limit the contents of a list based on the value selected in another
- Show totals in a subreport
- Update a list after you change its contents
- Use a custom function in a subform's SELECT statement
- Use criteria from the main form to set the subform's RecordSource property
- Use two subforms on a form

Query by form

---

- Find a record based on a value you select from a list
- Find records based on values in one or more fields
- Show progressively more detail about a subject
- Use a custom function in a subform's SELECT statement
- Use criteria from the main form to set the subform's RecordSource property
- Use part of a value to specify criteria
- Use two subforms on a form

Using queries with forms and reports

---

- Create a crosstab query for a report
- Create a SELECT statement for a form based on user input
- Limit the contents of a list based on the value selected from an option group
- Limit the contents of a list based on the value selected in another
- Use a custom function in a subform's SELECT statement
- Use a parameterized query in an event procedure
- Use a totals query as the record source for a subform
- Use criteria from the main form to set the subform's RecordSource property

Sample reports

---

- Customer Phone List
- Employee Sales

- Invoice
- Sales Averages
- Sales Letter
- Shipping Log

#### Create advanced reports

---

- Create a crosstab report with dynamic column headings
- Create a form letter
- Create a multiple-column report
- Create a multiple-fact crosstab report
- Create a report for a preprinted form

#### Control what you print on reports

---

- Draw a circle around data to accentuate it
- Hide a section when it falls at the top of a page
- Hide and show controls based on the value in another control
- Hide and show sections that print on a preprinted form
- Print "Continued" at the bottom of a preprinted form
- Print report criteria that was entered in a dialog box
- Print the first and last entries on a page in the page header
- Repeat a group name at the top of a column or page

#### Calculate totals on reports

---

- Calculate row and column totals on a crosstab report
- Show totals in a subreport

#### Use multiple databases

---

- Attach tables at startup

Additional reference words: 2.00 Designing

KBCategory: kbusage

KBSubcategory: GnlApp

## INF: List of Constants in Macros and Modules

Article ID: Q112140

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article lists the constant variables that are in the UTILITY.MDA file. This list is for reference only; importing these constants would result in a duplicate definition error.

Additional constants are listed in the CONSTANT.TXT file located in your Microsoft Access directory. This text file is provided for supplementary use and can be imported.

### MORE INFORMATION

=====

AbcConstants Module from UTILITY.MDA  
-----

NOTE: All of the constants listed below are global constants.

|                        |                    |                    |
|------------------------|--------------------|--------------------|
| V_EMPTY = 0            | V_NULL = 1         | V_INTEGER = 2      |
| V_LONG = 3             | V_SINGLE = 4       | V_DOUBLE = 5       |
| V_CURRENCY = 6         | V_DATE = 7         | V_STRING = 8       |
| DB_BINARY = 9          | DB_OLE = 11        | DB_NONULLS = 3     |
| A_TABLE = 0            | A_QUERY = 1        | A_FORM = 2         |
| A_REPORT = 3           | A_MACRO = 4        | A_MODULE = 5       |
| A_FORMBAR = 0          | A_FILE = 0         | A_EDITMENU = 1     |
| A_RECORDSMENU = 3      | A_NEW = 0          | A_SAVEFORM = 2     |
| A_SAVEFORMAS = 3       | A_SAVERECORD = 4   | A_UNDO = 0         |
| A_UNDOFIELD = 1        | A_CUT = 2          | A_COPY = 3         |
| A_PASTE = 4            | A_DELETE = 6       | A_SELECTRECORD = 7 |
| A_SELECTALLRECORDS = 8 | A_OBJECT = 14      | A_REFRESH = 2      |
| A_OBJECTVERB = 0       | A_OBJECTUPDATE = 3 | A_ANYWHERE = 0     |
| A_ENTIRE = 1           | A_START = 2        | A_UP = 0           |
| A_DOWN = 1             | A_CURRENT = 0      | A_ALL = 1          |
| A_PREVIOUS = 0         | A_NEXT = 1         | A_FIRST = 2        |
| A_LAST = 3             | A_GOTO = 4         | A_NEWREC = 5       |
| A_NORMAL = 0           | A_DESIGN = 1       | A_PREVIEW = 2      |
| A_FORMDS = 3           | A_ADD = 0          | A_EDIT = 1         |
| A_READONLY = 2         | A_HIDDEN = 1       | A_ICON = 2         |
| A_DIALOG = 3           | A_PRINTALL = 0     | A_SELECTION = 1    |
| A_PAGES = 2            | A_HIGH = 0         | A_MEDIUM = 1       |
| A_LOW = 2              | A_DRAFT = 3        | A_PROMPT = 0       |
| A_SAVE = 1             | A_EXIT = 2         | A_IMPORT = 0       |
| A_EXPORT = 1           | A_ATTACH = 2       | A_IMPORTDELIM = 0  |
| A_IMPORTFIXED = 1      | A_EXPORTDELIM = 2  | A_EXPORTFIXED = 3  |
| A_EXPORTMERGE = 4      | OLE_CHANGED = 0    | OLE_SAVED = 1      |
| OLE_CLOSED = 2         | OLE_RENAMED = 3    | OLE_RELEASE = 5    |
| SHIFT_MASK = 1         | CTRL_MASK = 2      | ALT_MASK = 4       |

```

LEFT_BUTTON = 1 RIGHT_BUTTON = 2 MIDDLE_BUTTON = 4
DATA_ERRCONTINUE = 0 DATA_ERRDISPLAY = 1 DATA_ERRADDED = 2
LB_INITIALIZE = 0 LB_OPEN = 1 LB_GETROWCOUNT = 3
LB_GETCOLUMNCOUNT = 4 LB_GETCOLUMNWIDTH = 5 LB_GETVALUE = 6
LB_GETFORMAT = 7 LB_CLOSE = 8 LB_END = 9
SYSCMD_INITMETER = 1 SYSCMD_UPDATESMETER = 2 SYSCMD_REMOVEMETER = 3
SYSCMD_SETSTATUS = 4 SYSCMD_CLEARSTATUS = 5 SYSCMD_RUNTIME = 6
SYSCMD_ACCESSVER = 7 SYSCMD_CLEARHELPTOPIC = 11 OBJSTATE_OPEN = 1
OBJSTATE_NEW = 4 OBJSTATE_DIRTY = 2 A_DELETE_V2 = 7
A_SELECTRECORD_V2 = 8 A_SELECTALLRECORDS_V2 = 9 A_TOOLBAR_YES = 0
A_TOOLBAR_WHERE_APPROP = 1 A_TOOLBAR_NO = 2 DELETE_OK = 0
DELETE_CANCEL = 1 DELETE_USER_CANCEL = 2

```

Form, Report, Macro and Module Specific Security Constants

```

DB_SEC_FRMRPT_READDEF = &H4&
DB_SEC_FRMRPT_WRITEDEF = &H8& + DB_SEC_FRMRPT_READDEF + DB_SEC_DELETE
DB_SEC_FRMRPT_EXECUTE = &H100&
DB_SEC_MAC_READDEF = &HA&
DB_SEC_MAC_WRITEDEF = &H6& + DB_SEC_DELETE
DB_SEC_MAC_EXECUTE = &H8&
DB_SEC_MOD_READDEF = &H2&
DB_SEC_MOD_WRITEDEF = &H4& + DB_SEC_MOD_READDEF + DB_SEC_DELETE
A_MENU_VER1X = 11, A_MENU_VER20 = 20
A_FORMATRTF = "Rich Text Format (*.rtf)"
A_FORMATXLS = "Microsoft Excel (*.xls)"
A_FORMATTXT = "MS-DOS Text (*.txt)"

```

DataConstants Module from UTILITY.MDA

```

DB_OPEN_TABLE = 1
DB_OPEN_DYNASET = 2
DB_OPEN_SNAPSHOT = 4
DB_DENYWRITE = &H1
DB_DENYREAD = &H2
DB_READONLY = &H4
DB_APPENDONLY = &H8
DB_INCONSISTENT = &H10
DB_CONSISTENT = &H20
DB_SQLPASSTHROUGH = &H40
DB_FAILONERROR = &H80
DB_FORWARDONLY = &H100
DB_OPTIONINIPATH = 1
DB_FIXEDFIELD = &H1
DB_VARIABLEFIELD = &H2
DB_AUTOINCRFIELD = &H10
DB_UPDATABLEFIELD = &H20
DB_DESCENDING = &H1
DB_BOOLEAN = 1
DB_BYTE = 2
DB_INTEGER = 3
DB_LONG = 4
DB_CURRENCY = 5
DB_SINGLE = 6
DB_DOUBLE = 7

```

DB\_DATE = 8  
DB\_TEXT = 10  
DB\_LONGBINARY = 11  
DB\_MEMO = 12  
DB\_RELATIONUNIQUE = &H1&  
DB\_RELATIONDONTENFORCE = &H2&  
DB\_RELATIONINHERITED = &H4&  
DB\_RELATIONUPDATECASCADE = &H100&  
DB\_RELATIONDELETECASCADE = &H1000&  
DB\_RELATIONLEFT = &H1000000  
DB\_RELATIONRIGHT = &H2000000  
DB\_ATTACHEXCLUSIVE = &H10000  
DB\_ATTACHSAVEPWD = &H20000  
DB\_SYSTEMOBJECT = &H80000002  
DB\_ATTACHEDTABLE = &H40000000  
DB\_ATTACHEDODBC = &H20000000  
DB\_HIDDENOBJECT = &H1&  
DB\_TABLE = 1  
DB\_QUERYDEF = 5  
DB\_QSELECT = &H0  
DB\_QACTION = &HF0  
DB\_QCROSSTAB = &H10  
DB\_QDELETE = &H20  
DB\_QUPDATE = &H30  
DB\_QAPPEND = &H40  
DB\_QMAKETABLE = &H50  
DB\_QDDL = &H60  
DB\_QSQLPASSTHROUGH = &H70  
DB\_QSETOPERATION = &H80  
DB\_QSPTBULK = &H90  
DB\_UNIQUE = 1  
DB\_PRIMARY = 2  
DB\_PROHIBITNULL = 4  
DB\_IGNORENULL = 8  
DB\_LANG\_ARABIC = ";LANGID=0x0401;CP=1256;COUNTRY=0"  
DB\_LANG\_CZECH = ";LANGID=0x0405;CP=1250;COUNTRY=0"  
DB\_LANG\_DUTCH = ";LANGID=0x0413;CP=1252;COUNTRY=0"  
DB\_LANG\_GENERAL = ";LANGID=0x0409;CP=1252;COUNTRY=0"  
DB\_LANG\_GREEK = ";LANGID=0x0408;CP=1253;COUNTRY=0"  
DB\_LANG\_HEBREW = ";LANGID=0x040D;CP=1255;COUNTRY=0"  
DB\_LANG\_HUNGARIAN = ";LANGID=0x040E;CP=1250;COUNTRY=0"  
DB\_LANG\_ICELANDIC = ";LANGID=0x040F;CP=1252;COUNTRY=0"  
DB\_LANG\_NORDIC = ";LANGID=0x041D;CP=1252;COUNTRY=0"  
DB\_LANG\_NORWDAN = ";LANGID=0x0414;CP=1252;COUNTRY=0"  
DB\_LANG\_POLISH = ";LANGID=0x0415;CP=1250;COUNTRY=0"  
DB\_LANG\_CYRILLIC = ";LANGID=0x0419;CP=1251;COUNTRY=0"  
DB\_LANG\_SPANISH = ";LANGID=0x040A;CP=1252;COUNTRY=0"  
DB\_LANG\_SWEDFIN = ";LANGID=0x040B;CP=1252;COUNTRY=0"  
DB\_LANG\_TURKISH = ";LANGID=0x041F;CP=1254;COUNTRY=0"  
DB\_VERSION10 = 1  
DB\_ENCRYPT = 2  
DB\_DECRYPT = 4  
DB\_VERSION11 = 8  
DB\_VERSION20 = 16  
DB\_SORTARABIC = 267  
DB\_SORTCYRILLIC = 263  
DB\_SORTCZECH = 264

DB\_SORTDUTCH = 259  
DB\_SORTGENERAL = 256  
DB\_SORTGREEK = 269  
DB\_SORTHEBREW = 268  
DB\_SORTHUNGARIAN = 265  
DB\_SORTICELANDIC = 262  
DB\_SORTNORWDAN = 261  
DB\_SORTPDXINTL = 4096  
DB\_SORTPDXNOR = 4098  
DB\_SORTPDXSWE = 4097  
DB\_SORTPOLISH = 266  
DB\_SORTSPANISH = 258  
DB\_SORTSWEDFIN = 260  
DB\_SORTTURKISH = 270  
DB\_SORTUNDEFINED = -1  
DB\_FREELOCKS = 1  
DB\_SEC\_NOACCESS = &H0&  
DB\_SEC\_FULLACCESS = &HFFFFFF  
DB\_SEC\_DELETE = &H10000  
DB\_SEC\_READSEC = &H20000  
DB\_SEC\_WRITESEC = &H40000  
DB\_SEC\_WRITEOWNER = &H80000  
DB\_SEC\_DBCREATE = &H1&  
DB\_SEC\_DBOPEN = &H2&  
DB\_SEC\_DBEXCLUSIVE = &H4&  
DB\_SEC\_CREATE = &H1&  
DB\_SEC\_READDEF = &H4&  
DB\_SEC\_WRITEDEF = &H8& Or DB\_SEC\_READDEF Or DB\_SEC\_DELETE  
DB\_SEC\_RETRIEVEDATA = &H10& Or DB\_SEC\_READDEF  
DB\_SEC\_INSERTDATA = &H20&  
DB\_SEC\_REPLACEDATA = &H40&  
DB\_SEC\_DELETEDATA = &H80&

#### REFERENCES

=====

For more information about constants, search for "constants" then "Constants" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbprg

KBSubcategory: MdlLib

**INF: List of New 2.0 Form/Report Events, Methods, & Properties**  
**Article ID: Q112795**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

This article lists the many new events, methods, and properties for both forms and reports in Microsoft Access version 2.0.

MORE INFORMATION

=====

| NAME                   | TYPE                | APPLIES TO REPORTS |
|------------------------|---------------------|--------------------|
| Action                 | Property            | X                  |
| Activate               | Event               | X                  |
| AfterDelConfirm        | Property, PS, Event |                    |
| AfterInsert            | Property, PS, Event |                    |
| Append                 | Method              |                    |
| AutoActivate           | Property, PS        |                    |
| AutoCenter             | Property, PS        |                    |
| AutoExpand             | Property, PS        |                    |
| AutoTab                | Property, PS        |                    |
| BeforeDelConfirm       | Event               |                    |
| BeforeInsert           | Property, PS        |                    |
| BorderStyle            | Property, PS        | X                  |
| Change                 | Event               |                    |
| Class                  | Property, PS        |                    |
| ColumnHidden           | Property, PS        |                    |
| ColumnOrder            | Property            |                    |
| ColumnWidth            | Property            |                    |
| ControlBox             | Property, PS        |                    |
| CreateDatabase         | Method              |                    |
| CreateObject           | Method              |                    |
| CurrentView            | Property            |                    |
| CurrentSectionLeft     | Property            |                    |
| CurrentSectionTop      | Property            |                    |
| CurrentView            | Property            |                    |
| DatasheetFontHeight    | Property            |                    |
| DatasheetFontItalic    | Property            |                    |
| DatasheetFontName      | Property            |                    |
| DatasheetFontUnderline | Property            |                    |
| DatasheetFontWeight    | Property            |                    |
| Deactivate             | Event               | X                  |
| Delete (Recordset)     | Method              |                    |
| Dirty                  | Property            |                    |
| DisplayType            | Property, PS        | X                  |
| Echo                   | Method              |                    |
| EnterKeyBehavior       | Property, PS        |                    |
| Error                  | Event               | X                  |
| EventProcPrefix        | Property            | X                  |



|                                       |              |   |
|---------------------------------------|--------------|---|
| Execute (QueryDef)                    | Method       |   |
| FastLaserPrinting                     | Property, PS | X |
| FieldSize                             | Method       |   |
| FillCache                             | Method       |   |
| FrozenColumns                         | Property     |   |
| GetObject                             | Method       |   |
| GetOption                             | Method       |   |
| GotFocus                              | Event        |   |
| GotoPage                              | Method       |   |
| Idle                                  | Method       |   |
| InputMask                             | Property, PS | X |
| InSelection                           | Property     |   |
| InsertText                            | Method       |   |
| ItemData                              | Method       |   |
| KeepTogether - Groups                 | Property, PS |   |
| KeyDown                               | Event        |   |
| KeyPress                              | Event        |   |
| KeyUp                                 | Event        |   |
| ListCount                             | Property     | X |
| ListIndex                             | Property     | X |
| Load                                  | Event        |   |
| LostFocus                             | Event        |   |
| lpOleObject                           | Property     | X |
| MaxButton                             | Property, PS |   |
| Me                                    | Property     | X |
| MenuBar                               | Property, PS | X |
| MinButton                             | Property, PS |   |
| Module                                | Property     | X |
| MouseDown                             | Event        |   |
| MouseMove                             | Event        |   |
| MouseUp                               | Event        |   |
| Move                                  | Method       |   |
| Name                                  | Property, PS | X |
| NavigationButtons                     | Property, PS | X |
| NewPassword                           | Method       |   |
| NotInList                             | Event        |   |
| Object                                | Property     | X |
| ObjectPalette                         | Property     | X |
| ObjectVerbs                           | Property     | X |
| ObjectVerbsCount                      | Property     | X |
| OleType                               | Property, PS | X |
| OleTypeAllowed                        | Property, PS | X |
| OnActivate                            | Property, PS | X |
| OnChange                              | Property, PS |   |
| OnClick (replaces<br>OnPush property) | Property, PS |   |
| OnDeactivate                          | Property, PS | X |
| OnError                               | Property, PS | X |
| OnGotFocus                            | Property, PS |   |
| OnKeyDown                             | Property, PS |   |
| OnKeyPress                            | Property, PS |   |
| OnKeyUp                               | Property, PS |   |
| OnLoad                                | Property, PS | X |
| OnLostFocus                           | Property, PS |   |
| OnMouseDown                           | Property, PS |   |
| OnMouseMove                           | Property, PS |   |
| OnMouseUp                             | Property, PS |   |

|                          |              |   |
|--------------------------|--------------|---|
| OnNotInList              | Property, PS |   |
| OnResize                 | Property, PS |   |
| OnTimer                  | Property, PS |   |
| OnUnload                 | Property, PS |   |
| OnUpdated                | Property, PS |   |
| OpenArgs                 | Property     |   |
| OpenDatabase             | Method       |   |
| OpenRecordset            | Method       |   |
| OpenRecordset (Database) | Method       |   |
| Pages                    | Property     | X |
| PaintPalette             | Property     | X |
| PaletteSource            | Property, PS | X |
| Quit                     | Method       |   |
| Recalc                   | Method       |   |
| RecordsetClone           | Property     |   |
| Refresh                  | Method       |   |
| Refresh (DAO)            | Method       |   |
| RefreshLink              | Method       |   |
| RegisterDatabase         | Method       |   |
| Repaint                  | Method       |   |
| RepairDatabase           | Method       |   |
| Requery                  | Method       |   |
| Requery (DAO)            | Method       |   |
| Resize                   | Event        |   |
| SelLength                | Property     |   |
| SelStart                 | Property     |   |
| SelText                  | Property     |   |
| SetFocus                 | Method       |   |
| SetOption                | Method       |   |
| ShortcutMenu             | Property     |   |
| SizeMode                 | Property, PS | X |
| SourceDoc                | Property, PS | X |
| SourceItem               | Property, PS | X |
| TabIndex                 | Property, PS |   |
| TabStop                  | Property, PS |   |
| Tag                      | Property, PS | X |
| Text                     | Property     | X |
| Timer                    | Event        |   |
| TimerInterval            | Property, PS |   |
| Updated                  | Event        |   |
| UpdateOptions            | Property, PS | X |
| Value                    | Property     | X |
| Verb                     | Property, PS | X |
| WindowHeight             | Property     | X |
| WindowWidth              | Property     | X |

#### REFERENCES

=====

For more information about form properties, search for "properties: forms" then "Form Properties" using the Microsoft Access Help menu.

For more information about report properties, search for "properties: reports" then "Report Properties" using the Microsoft Access Help menu.

For more information on events, search for "events: order of" then "Order of Events" using the Microsoft Access Help menu.

To see the order in which events occur on a form, open the Show Events form in the NWIND sample database.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsHowto

**INF: List of OLE 2.0 Features Supported in MS Access v. 2.0**  
**Article ID: Q112749**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

Microsoft Access version 2.0 supports the Object Linking and Embedding (OLE) specification version 2.0. The OLE 2.0 specification describes a standard set of interfaces that allow applications to share features. Applications may provide none, some, or all of this functionality.

MORE INFORMATION

=====

When you install Microsoft Access version 2.0, all the files necessary for OLE 2.0 functionality are installed and Microsoft Access is registered in your system as an OLE 2.0 client. If you choose to install Microsoft Graph during the Microsoft Access installation, it is registered as an OLE 2.0 server application.

Embedding and Linking

-----

Objects created in OLE 1.0 and 2.0 servers can be embedded or linked into Microsoft Access forms, reports, and tables. Microsoft Access supports the OLE 2.0 standard menus and dialog boxes for inserting and updating objects.

Visual Editing

-----

Microsoft Access 2.0 supports in-place editing of OLE objects in a form's Form view. To prevent in-place editing of an embedded OLE object, set the Enabled property of the object's frame to No. In-place activation is not available in a form or report's Design view, or in a table, query, or form's Datasheet view.

Drag and Drop

-----

Microsoft Access 2.0 does not support Drag and Drop across applications. Drag and Drop is used in Microsoft Access 2.0 to quickly create macro actions, new controls, and to customize tool bars.

OLE Automation

-----

Microsoft Access 2.0 provides full access to the OLE Automation methods supported by embedded objects through the use of Access Basic.

Context-Sensitive Menus

-----

Context-sensitive menus are used extensively in Microsoft Access 2.0 to provide shortcuts for commonly used commands. This includes access to an OLE object's object-specific verb menu when you click the secondary mouse button on an OLE object. When an OLE object is activated in place, mouse clicks are passed to the OLE object. When the object is not active the mouse clicks are handled by Microsoft Access.

#### OLE Server

-----

Objects created in Microsoft Access 2.0 cannot be embedded or linked into other OLE 2.0 applications; Microsoft Access 2.0 is not an OLE 2.0 server. You can export objects to other Windows applications using a variety of methods including DDE and the Microsoft Access Export command.

#### Conversion and Emulation

-----

Microsoft Access 2.0 does not have a Convert dialog box. Objects that are registered to use AutoConversion or Object Type Emulation (also called "Activate As" or "Treat As") are fully supported by Microsoft Access 2.0.

#### Relative Links

-----

Links in OLE 2.0 maintain both a relative link source reference and an absolute link source reference. You can take advantage of this functionality in Microsoft Access 2.0, or you can continue to use the OLE 1.0 behavior in which only absolute link source reference information is saved. This allows users operating in a heterogeneous environment (where both Microsoft Access 1.x and 2.0 is used) to share databases. This option is controlled through an entry in the MSACC20.INI file called AllowOLE1LinkFormat. The default for this entry is 0, which means that all OLE 2.0 link tracking information (both the relative and absolute link source references) is maintained and stored.

#### REFERENCES

=====

For more information about modifying the MSACC20.INI file, search for "customizing Microsoft Access" then "Customizing MSACC20.INI Settings" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbole

KBSubcategory: IntpOle

**INF: Listing Object Names Using Data Access Objects (DAO)**  
**Article ID: Q119471**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

In Microsoft Access version 2.0, you can use data access objects (DAO) to list the objects in your database. This article describes a method using DAO that you can use to display all the objects in your database for the object type you select.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The following example demonstrates how to use DAO to list all the objects in your database for the object type you select:

1. Start Microsoft Access and open any database.
2. Create a new, blank form.
3. Add an option group to the form. Set the Option group's Name property to "ChooseObject" (without quotation marks).
4. Add seven option buttons with the following properties to the option group:

Option Button 1  
Name: Tables  
OptionValue: 1

Option Button 2  
Name: Queries  
OptionValue: 2

Option Button 3  
Name: Forms  
OptionValue: 3

Option Button 4  
Name: Reports  
OptionValue: 4

Option Button 5  
Name: Macros/Scripts  
OptionValue: 5

Option Button 6  
Name: Modules  
OptionValue: 6

Option Button 7  
Name: All Objects  
OptionValue: 7

5. Set the option group's AfterUpdate property to the following event procedure.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Sub ChooseObject_AfterUpdate ()

Dim DB As Database, I As Integer, j As Integer, ok_cancel As Integer
Dim System_Prefix, Current_TableName, Hidden_Prefix
Dim Ok as Integer, Cancel as Integer

Ok = 1
Cancel = 2
Set db = DbEngine(0) (0)

Select Case Me![ChooseObject]
Case 1
'System tables are excluded from the list.
For I = 0 To db.TableDefs.Count - 1
Current_TableName = db.TableDefs(I).Name
System_Prefix = Left(Current_TableName, 4)
Hidden_Prefix = Left(Current_TableName, 1)
If System_Prefix <>"MSys" And System_Prefix <> "USys" And _
Hidden_Prefix <> "~" Then
ok_cancel = MsgBox(db.TableDefs(I).Name, 65, "TABLE NAMES")
If ok_cancel = cancel Then
Exit Sub
End If
End If
Next I
Case 2
For I = 0 To db.Querydefs.Count - 1
ok_cancel = MsgBox(db.Querydefs(I).Name, 65, "QUERY NAMES")
If ok_cancel = cancel Then
Exit Sub
End If
Next I
Case 3
For I = 0 To db.Containers("Forms").Documents.Count - 1
ok_cancel = MsgBox(db.Containers("Forms").Documents(I).Name, _
65, "FORM NAMES")
If ok_cancel = cancel Then
Exit Sub
End If
Next I
Case 4
```

```

For I = 0 To db.Containers("Reports").Documents.Count - 1
 ok_cancel = MsgBox(db.Containers("Reports").Documents(I).Name, _
65, "REPORT NAMES")
 If ok_cancel = cancel Then
 Exit Sub
 End If
Next I
Case 5
'Scripts are macros.
For I = 0 To db.Containers("Scripts").Documents.Count - 1
 ok_cancel = MsgBox(db.Containers("Scripts").Documents(I).Name, _
65, "MACRO NAMES")
 If ok_cancel = cancel Then
 Exit Sub
 End If
Next I
Case 6
For I = 0 To db.Containers("Modules").Documents.Count - 1
 ok_cancel = MsgBox(db.Containers("Modules").Documents(I).Name, _
65, "MODULE NAMES")
 If ok_cancel = cancel Then
 Exit Sub
 End If
Next I
Case 7
For I = 0 To db.Containers.Count - 1
 For j = 0 To db.Containers(I).Documents.Count - 1
 ok_cancel = MsgBox(db.Containers(I).Name & Chr(13) & Chr(10) _
& db.Containers(I).Documents(j).Name, 65, "ALL OBJECTS")
 If ok_cancel = cancel Then
 Exit Sub
 End If
 Next j
Next I
End Select
End Sub

```

6. View the form in Form view. Select the option button for the type of object whose names you want to list.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 7, "Objects and Collections," pages 155-181

Additional reference words: 2.00 ab

KBCategory: kbusage

KBSubcategory: GnlPrfm



**INF: Local MS Access Installation Required by ADT**  
**Article ID: Q119710**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SUMMARY

=====

To get the full functionality of the Microsoft Access Developer's Toolkit, such as creating distribution disks for shipping your custom database application, you must have a local installation of Microsoft Access version 2.0. There is no way to work around this requirement. This means that you cannot perform an administrative installation of Microsoft Access to a server (with the SETUP /A command), perform workstation installations of Microsoft Access from the server, and then install and use the Microsoft Access Developer's Toolkit (ADT) on the workstations. The ADT does not work correctly with workstation installations of Microsoft Access; a local installation of Microsoft Access is required.

MORE INFORMATION

=====

This requirement is the first item listed in the "Errata and Omissions in Advanced Topics" section of the README.TXT file that is shipped with the ADT.

While there are methods of working around this requirement, these methods are not recommended, as they create more problems than they solve. One reason for this is that the ADT Setup Wizard is written to look for specific files in specific locations, and a local installation of Microsoft Access puts the proper files in the proper places.

The ADT will work with a Typical, Custom/Complete, or Laptop local installation of Microsoft Access. Which local installation option you choose should be determined by your development environment and which tools you need for developing your application. For example, if you are using ISAM drivers in your database application, you should choose the Complete installation option when you install Microsoft Access.

REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 1, "Setting Up Microsoft Access"

Additional reference words: 2.00 installing setup

KBCategory: kbsetup

KBSubcategory: StpGnrl

## INF: Macro to Check for Duplicate Values in a Field

Article ID: Q103135

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article demonstrates how to use a macro using the DLookup() function to check for duplicate values in a field.

### MORE INFORMATION

=====

Microsoft Access provides automatic checking for duplicate values in a field that has been designated as the primary key. However, this check occurs after all fields for a record have been entered, just before the record is committed to the database. The following sample macro is based on the sample database NWIND.MDB and will perform a duplicate check as the data in a field is entered:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code.

1. Open the NWIND.MDB database.
2. Create a new form based on the Customers table.
3. Create a new macro as follows:

| Macro Name | Condition | Action   |
|------------|-----------|----------|
| MyMacro    | 1.        | MsgBox   |
|            | ...       | SendKeys |

```
1. DLookup("[Customer ID]", "Customers", "[Customer ID] = _
Form.[Customer ID]") is Not Null
```

#### MyMacro Actions

##### MsgBox

Message: That value already exists.

##### SendKeys

Keystrokes: +{TAB}

Wait: No

Note that the ellipsis (...) under the condition indicates that the condition controls both macro actions. The SendKeys action will return the cursor to the field just exited.

4. In Design view for the form, attach this macro to the AfterUpdate

property of the Customer ID control.

5. In Form view, make a note of the value of the current Customer ID.
6. From the Records menu, choose Data Entry to add a new record.
7. Enter the value of the earlier noted Customer ID. The message box with your message will appear and the cursor will be returned to the Customer ID field.

#### REFERENCES

=====

Microsoft Access "User's Guide," Chapter 22, and Appendix C

Additional reference words: 1.00 1.10 2.00 validate same validationrule

KBCategory: kbusage

KBSubcategory: McrCond

**INF: Macro to Enter Date Record Was Last Modified**

**Article ID: Q96103**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article explains how to write a macro that will place the current date into a LastModified field in your table each time a record is modified. This method works when you are editing records from a form.

MORE INFORMATION

=====

When a form is used to edit records, you can date stamp the record by attaching a macro to the BeforeUpdate property of the form. For this example, we will use the Customers table in the sample database NWIND.MDB.

1. Add a new field to the Customers table.

Table: Customers

-----

Field Name: DateModified

DataType: Date/Time

2. Create a macro called LastModified as follows:

| Macro Name   | Action               |
|--------------|----------------------|
| -----        | -----                |
| LastModified | SetValue             |
|              | Item: [DateModified] |
|              | Expression: Date()   |

NOTE: There is no equal sign (=) in front of the Date() function. If you put an equal sign in front of the Date() function, the macro will always put the date 12/30/99 into the [DateModified] control even though the macro step window shows the correct date in the argument section.

3. Create a form called Modify Customers based on the Customers table. Place a few of the fields on the form as follows:

Form Name: Modify Customers

-----

ControlSource: DateModified

ControlSource: Customer ID

ControlSource: Company Name

ControlSource: Address

4. Enter the macro name LastModified into the BeforeUpdate property of the Modify Customers form. Switch the form to Form view. Any time

you modify the record, the current date will be placed in the DateModified control.

Additional reference words: 1.00 1.10 2.00 unexpected

KBCategory: kbusage

KBSubcategory: McrActn

**INF: Microsoft Access 2.0 Disk Contents (3.5", 1.44 MB)**

**Article ID: Q115123**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article lists the directory contents of the disks shipped with Microsoft Access version 2.0 (3.5 inch, 1.44 MB).

MORE INFORMATION

=====

Disk 1

-----

Volume Label: DISK1

|            |     |           |          |        |
|------------|-----|-----------|----------|--------|
| ACMSETUP   | EX_ | 106,497   | 04-05-94 | 12:00a |
| ACMSETUP   | HL_ | 12,618    | 04-05-94 | 12:00a |
| ACREADME   | HLP | 84,659    | 04-05-94 | 12:00a |
| ACSETUP    | TT_ | 52,390    | 04-05-94 | 12:00a |
| ADMIN      | IN_ | 2,530     | 04-05-94 | 12:00a |
| BTRV200    | DL_ | 56,473    | 04-05-94 | 12:00a |
| CTL3D      | DL_ | 10,085    | 04-05-94 | 12:00a |
| CTL3DV2    | DL_ | 10,331    | 04-05-94 | 12:00a |
| DBNMP3     | DL_ | 5,909     | 04-05-94 | 12:00a |
| DECOMP     | EXE | 39,044    | 04-05-94 | 12:00a |
| DRVSSRVR   | HL_ | 54,381    | 04-05-94 | 12:00a |
| INSTCAT    | SQ_ | 14,016    | 04-05-94 | 12:00a |
| MSACAHBB   | DL_ | 61,904    | 04-05-94 | 12:00a |
| MSCPDIS    | DL_ | 8,805     | 04-05-94 | 12:00a |
| MSSETUP    | DL_ | 91,812    | 04-05-94 | 12:00a |
| ODBC       | DL_ | 23,412    | 04-05-94 | 12:00a |
| ODBCADM    | EX_ | 3,270     | 04-05-94 | 12:00a |
| ODBCINST   | DL_ | 42,520    | 04-05-94 | 12:00a |
| ODBCINST   | HL_ | 10,369    | 04-05-94 | 12:00a |
| OLE2       | RE_ | 2,749     | 04-05-94 | 12:00a |
| PDX200     | DL_ | 120,351   | 04-05-94 | 12:00a |
| README     | TXT | 696       | 04-05-94 | 12:00a |
| SETUP      | EXE | 39,856    | 04-05-94 | 12:00a |
| SETUP      | INI | 149       | 04-05-94 | 12:00a |
| SETUP      | IN_ | 2,699     | 04-05-94 | 12:00a |
| SETUP      | LST | 578       | 04-05-94 | 12:00a |
| SETUP      | ST_ | 16,136    | 04-05-94 | 12:00a |
| SQLSRVR    | DL_ | 85,111    | 04-05-94 | 12:00a |
| TYPelib    | DL_ | 78,623    | 04-05-94 | 12:00a |
| XBS200     | DL_ | 150,214   | 04-05-94 | 12:00a |
| _MSSETUP   | EX_ | 9,147     | 04-05-94 | 12:00a |
| 31 file(s) |     | 1,197,334 | bytes    |        |

Disk 2

-----

Volume Label: DISK2

|          |     |           |                 |        |
|----------|-----|-----------|-----------------|--------|
| DAO2016  | DL_ | 165,299   | 04-05-94        | 12:00a |
| MSABC200 | DL_ | 190,545   | 04-05-94        | 12:00a |
| MSACCESS | EX_ | 1,031,533 | 04-05-94        | 12:00a |
| MSAJU200 | DL_ | 50,106    | 04-05-94        | 12:00a |
|          |     | 4 file(s) | 1,437,483 bytes |        |

Disk 3

-----

Volume Label: DISK3

|          |     |            |                 |        |
|----------|-----|------------|-----------------|--------|
| COMPOBJ  | DL_ | 47,475     | 04-05-94        | 12:00a |
| MSAEXP20 | DL_ | 42,016     | 04-05-94        | 12:00a |
| MSAIN200 | DL_ | 120,884    | 04-05-94        | 12:00a |
| MSAJT200 | DL_ | 500,727    | 04-05-94        | 12:00a |
| MSAOLE20 | DL_ | 57,065     | 04-05-94        | 12:00a |
| ODBCJT16 | DL_ | 65,459     | 04-05-94        | 12:00a |
| OLE2     | DL_ | 144,252    | 04-05-94        | 12:00a |
| OLE2DISP | DL_ | 44,927     | 04-05-94        | 12:00a |
| STDOLE   | TL_ | 1,881      | 04-05-94        | 12:00a |
| STORAGE  | DL_ | 68,983     | 04-05-94        | 12:00a |
| UTILITY  | MD_ | 52,695     | 04-05-94        | 12:00a |
| WRKGADM  | EX_ | 86,698     | 04-05-94        | 12:00a |
| WZLIB    | MD_ | 158,976    | 04-05-94        | 12:00a |
|          |     | 13 file(s) | 1,392,038 bytes |        |

Disk 4

-----

Volume Label: DISK4

|          |     |            |                 |        |
|----------|-----|------------|-----------------|--------|
| CONSTANT | TX_ | 1,137      | 04-05-94        | 12:00a |
| DRVACC2  | HL_ | 13,960     | 04-05-94        | 12:00a |
| MSACC20  | H1_ | 1,048,730  | 04-05-94        | 12:00a |
| MSACCPSS | HL_ | 15,959     | 04-05-94        | 12:00a |
| MSAFIN   | DL_ | 12,409     | 04-05-94        | 12:00a |
| MSAU200  | DL_ | 11,681     | 04-05-94        | 12:00a |
| MSINFO   | EX_ | 16,713     | 04-05-94        | 12:00a |
| MSJETERR | DL_ | 5,943      | 04-05-94        | 12:00a |
| MSJETINT | DL_ | 10,112     | 04-05-94        | 12:00a |
| MSTOOLBR | DL_ | 5,536      | 04-05-94        | 12:00a |
| ODBCTL16 | DL_ | 36,022     | 04-05-94        | 12:00a |
| OLE2CONV | DL_ | 29,914     | 04-05-94        | 12:00a |
| OLE2NLS  | DL_ | 40,507     | 04-05-94        | 12:00a |
| OLE2PROX | DL_ | 17,642     | 04-05-94        | 12:00a |
| ORDERS   | MD_ | 64,335     | 04-05-94        | 12:00a |
| SOA200   | DL_ | 19,549     | 04-05-94        | 12:00a |
| VER      | DL_ | 5,737      | 04-05-94        | 12:00a |
| VSHARE   | 38_ | 7,241      | 04-05-94        | 12:00a |
|          |     | 18 file(s) | 1,363,127 bytes |        |

Disk 5

-----

Volume Label: DISK5

|           |     |           |          |        |
|-----------|-----|-----------|----------|--------|
| MAPIVIM   | DL_ | 27,676    | 04-05-94 | 12:00a |
| MSACC20   | H2_ | 1,082,248 | 04-05-94 | 12:00a |
| MSACCESS  | CU_ | 310,145   | 04-05-94 | 12:00a |
| ORDERS    | HL_ | 13,029    | 04-05-94 | 12:00a |
| 4 file(s) |     | 1,433,098 | bytes    |        |

Disk 6

-----

Volume Label: DISK6

|           |     |           |          |        |
|-----------|-----|-----------|----------|--------|
| MSACC20   | H3_ | 835,954   | 04-05-94 | 12:00a |
| NWIND     | MD_ | 546,150   | 04-05-94 | 12:00a |
| SDM       | DL_ | 57,337    | 04-05-94 | 12:00a |
| 3 file(s) |     | 1,439,441 | bytes    |        |

Disk 7

-----

Volume Label: DISK7

|           |     |           |          |        |
|-----------|-----|-----------|----------|--------|
| GRAPH5    | EX_ | 879,809   | 04-05-94 | 12:00a |
| MSACCARD  | DL_ | 2,168     | 04-05-94 | 12:00a |
| MSGGRAPH  | HL_ | 340,135   | 04-05-94 | 12:00a |
| WZBLDR    | MD_ | 212,971   | 04-05-94 | 12:00a |
| 4 file(s) |     | 1,435,083 | bytes    |        |

Disk 8

-----

Volume Label: DISK8

|            |     |           |          |        |
|------------|-----|-----------|----------|--------|
| AUTOCONV   | EX_ | 9,390     | 04-05-94 | 12:00a |
| CCAPI200   | DL_ | 23,659    | 04-05-94 | 12:00a |
| CCARD200   | EX_ | 64,797    | 04-05-94 | 12:00a |
| GREN50     | OL_ | 23,011    | 04-05-94 | 12:00a |
| GRINTL     | DL_ | 55,622    | 04-05-94 | 12:00a |
| MCIOLE     | DL_ | 4,818     | 04-05-94 | 12:00a |
| NETWKSTA   | 50_ | 61,578    | 04-05-94 | 12:00a |
| NEWCUST    | DB_ | 1,104     | 04-05-94 | 12:00a |
| SCP        | DL_ | 4,637     | 04-05-94 | 12:00a |
| SOLUTION   | HL_ | 129,280   | 04-05-94 | 12:00a |
| SOLUTION   | MD_ | 119,655   | 04-05-94 | 12:00a |
| WFWNET     | DR_ | 137,584   | 04-05-94 | 12:00a |
| WZFRMRPT   | MD_ | 241,258   | 04-05-94 | 12:00a |
| WZQUERY    | MD_ | 87,570    | 04-05-94 | 12:00a |
| WZTABLE    | MD_ | 70,152    | 04-05-94 | 12:00a |
| WZTBLDAT   | MD_ | 54,328    | 04-05-94 | 12:00a |
| 16 file(s) |     | 1,088,443 | bytes    |        |

Additional reference words: 2.00

KBCategory: kbref

KBSubcategory: StpOthr





## **INF: Microsoft Access 2.0 Readme Help File Contents**

**Article ID: Q113564**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article lists the contents and topics contained in the Microsoft Access version 2.0 Readme Help file. Note that the Readme Help file replaces the text-only README.TXT file included with earlier versions of Microsoft Access.

### MORE INFORMATION

=====

In the items below, each content section is underlined, with its associated topics below.

#### Microsoft Access Product Update

-----

- Microsoft Access Q&A.
- Using SHARE.EXE or VSHARE.386.
- Using DECOMP.EXE to Decompress Microsoft Access Files.
- Renaming, Deleting, or Moving Files and Directories.
- Files That Must Be Installed Locally.
- Warning When Attaching Tables with Unsupported Indexes.
- Scope and Lifetime of Access Basic Variables and Functions.
- International Issues.

#### Conversion and Compatibility Updates

-----

- Conversion from Microsoft Access Version 1.x to 2.0.
- Setup Removes Microsoft Access Version 1.x Files.
- Improving Index Performance.
- Conflicts with New Property Names in Expressions and Access Basic Code.
- Form Control and Section Names.
- Converting a "Front-end/Back-end" Application.
- Version 2.0 Setup and the UTILITY.MDA File.
- Using Visual Basic Version 3.0 Applications.
- Exporting to Version 1.1 Databases.
- DefaultValue Property Setting Includes Quotation Marks in Version 2.0.
- Owner Access Queries.
- Order of Evaluation of Expressions.
- Applications That Use the SQL Pass-through DLL.
- Closing Pop-up Forms and Their Underlying Windows.
- Unused Error Messages.

#### Network Installations

- 
- Designating Network Connections.
  - Setting Up Microsoft Access on a Network File Server.
  - Creating Custom Installations for Workstation Users.

#### Documentation Updates

---

- Book: Microsoft Access Getting Started.
- Book: Microsoft Access User's Guide.
- Book: Microsoft Access Building Applications.
- Help: Microsoft Access Help.

#### Using Microsoft Access Version 2.0 with Other Software

---

- Microsoft Visual Basic.
- Accessing Microsoft Access Data from Microsoft Word and Microsoft Excel.
- Using Microsoft Access with Paradox Databases.
- Running Multiple Data Access Applications.
- Using Microsoft Access to Import, Attach, or Export to Microsoft FoxPro Version 2.5 Files.
- Lotus cc:Mail or Lotus Notes.
- Using Microsoft Access with Microsoft Video for Windows Windows NT.
- Setting Up Microsoft Access on a Novell Network.
- Using Microsoft Access on Novell NetWare Servers and Banyan VINES Servers (Increasing the Number of Record Locks).
- Using Microsoft Access with Microsoft LAN Manager.
- Using Microsoft Access with Microsoft Windows for Workgroups.
- Using SQL Server Data Installed on a Novell NetWare LAN or a Banyan VINES Network.
- Asynchronous Query Execution and ODBC Drivers.
- ORACLE Server Databases and Null Indexes.

#### Using Microsoft Access with Btrieve

---

- General Considerations.
- How Microsoft Access Locates Btrieve Files.
- Using Databases with Xtrieve-style Security.
- Using Compressed Data Files.
- Using Btrieve in a Multiuser Environment.
- Using Btrieve Indexes.
- Setting Btrieve Options.
- Setting Microsoft Access Options for Btrieve.
- Configuring the Server-based Version (BTRIEVE.NLM).

#### Files Included with Microsoft Access

---

- Outdated Version 1.0 and Version 1.1 Files.
- Microsoft Access Version 2.0 Files.

Additional reference words: 2.00 helpfile read.me acreamdme.hlp

KBCategory: kbreadme  
KBSubcategory: DcmOthr

## **INF: Microsoft Access and ORACLE ODBC Drivers**

**Article ID: Q112105**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

Microsoft Access version 2.0 contains an ODBC driver for connecting to SQL Server. However, Microsoft Access 2.0 does not contain the Oracle 6 ODBC driver because customer research showed that relatively few people used the Oracle driver that was shipped with Microsoft Access version 1.1.

If you have the Oracle ODBC driver supplied with Microsoft Access version 1.1, however, you can use it with Microsoft Access version 2.0. The Oracle ODBC driver included with Microsoft Access version 1.1 supports Oracle 6 functionality and Oracle 6 functionality in Oracle 7, but does not support full Oracle 7 functionality. If you want full Oracle 7 functionality, both Oracle and Q+E have Oracle 7 ODBC drivers available that can be used with Microsoft Access version 2.0.

If you do not have Microsoft Access 1.1, you can obtain the Oracle 6 driver by purchasing the Microsoft ODBC Drivers fulfillment kit. This kit contains ODBC drivers for Microsoft SQL Server, Oracle 6, and other drivers as they become available. To order the kit, send \$5.25 + \$5.00 shipping (+ local sales tax, if applicable) to:

Microsoft Access for Windows Supplemental Offers  
PO Box 3022  
Bothell, WA 98041

Additional reference words: 2.00 Importing Exporting Attaching

KBCategory: kbinterop

KBSubcategory: IntpOthr

## **INF: Microsoft Access and Untested Networks**

**Article ID: Q109739**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

### SUMMARY

=====

This article describes some of the anomalies that can occur when you are using Microsoft Access with networks that have not been tested with Microsoft Access. These networks include DEC Pathworks, Sun PC-NFS, IBM PC-LAN Server, and others.

### MORE INFORMATION

=====

Microsoft Access has been tested and verified as supportable on the following networks: Microsoft LAN Manager, Windows for Workgroups, Novell NetWare versions 2.x and 3.x, Artisoft Lantastic, and Banyan VINES.

Many other network types (including IBM PC-LAN Server, DEC Pathworks, and Sun PC-NFS) have experienced erratic behavior and only limited success with Microsoft Access. Irregular behaviors on these network types can include:

- An inability to install Microsoft Access to a network drive
- An inability to run the Workstation Setup after a successful Admin Setup
- An inability to open a multiuser file (non-exclusively)
- Multiple users cannot update a record correctly
- Unexplained database corruption or data loss

Some of these behaviors are quite serious. We want to protect our customers' valuable data, and the best way to do this is to support Microsoft Access only on networks that we have tested it with. We cannot support Microsoft Access on non-tested networks.

However, just because we have not tested Microsoft Access with a particular network does not mean that the network will not work with Microsoft Windows. Microsoft Windows has been tested with far more networks than has Microsoft Access. It is a good idea, before purchasing any product, to check with the product's manufacturer as to the product's compatibility with your network.

### REFERENCES

=====

Microsoft Access "User's Guide," version 1.1, Appendix D

Additional reference words: 1.00 1.10 2.00 unsupported  
KBCategory: kbsetup  
KBSubcategory: StpNet

## **INF: Microsoft Access Does Not Use Record Numbers**

**Article ID: Q95928**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article discusses the reasons why Microsoft Access does not use record numbers, as do some other database products.

### MORE INFORMATION

=====

Many database products, such as Microsoft FoxPro, dBASE IV, and Paradox, use a record number that is physically associated with the record. For example, the first record in a table has record number 1, the second record has record number 2, and so on. The record numbers remain part of the record as long as the record exists. The first record entered into a table is always record number 1, even if the table is sorted and the record appears somewhere other than the first position in the table.

Microsoft Access does not use this approach. When you open a table in Datasheet view, a number appears at the bottom of the form. This number indicates the position of the record in the recordset, not the record's natural position in the table. For example, if a record with the key value "Anderson" is the first record in a recordset, the number 1 appears at the bottom of the form. However, if you sort the same records on a different key, the record may no longer appear as the first record and a different number may appear at the bottom of the form.

When you open a table, form, or report, Microsoft Access creates a dynaset, a dynamic copy of the appropriate recordset, and assigns numbers to the records therein. These numbers are used only by the user interface to provide a visually unique identifier for each record.

Microsoft Access uses this approach because assigning a physical record number to each record is not "relationally correct." According to relational database theory, data may appear in any order at any time. Therefore, the key value, not a random record number, provides the only reliable method to identify a record.

Because no record number is available, Microsoft Access provides another method to identify records in a dynaset. Using the Bookmark property, you can write a procedure to find a target record, store its bookmark value in a variable, move to other records, and return to the original record. For more information about bookmarks, search on "Bookmark" using the Help menu.

dBASE IV and Paradox are manufactured by Borland International, Inc., a



vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00 record selector vcr

KBCategory: kbprg

KBSubcategory: PgmObj

## INF: Microsoft Access KB Categories and Subcategories

Article ID: Q119526

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Every Knowledge Base (KB) article contains a category and a subcategory. You can query on the categories and subcategories to help you find specific groups of articles. The categories and subcategories are abbreviated and concatenated to form unique words that you can query on. For example, FmsCmbo stands for Forms, Combo/List Box, where Forms is the category and Combo/List Box is the subcategory. Note that the first letter of the category and subcategory are capitalized; this does not affect your queries, but makes it easier to recognize the subcategory.

See the "References" section later in this article for information about how to enter queries based on the categories and subcategories listed in this article.

### MORE INFORMATION

=====

The following list contains all the categories and subcategories used in Microsoft Access KB articles:

|                           |                                  |          |
|---------------------------|----------------------------------|----------|
| Conversion/Compatibility  | 2.0 Behaves Differently Than 1.x | CnvDif20 |
| Conversion/Compatibility  | Convert fails on a db            | CnvNodb  |
| Conversion/Compatibility  | ConvertErrors table              | CnvErtbl |
| Conversion/Compatibility  | Errors while running Convert     | CnvErrs  |
| Conversion/Compatibility  | General                          | CnvGnrl  |
| Conversion/Compatibility  | Other                            | CnvOthr  |
| Conversion/Compatibility  | Performance of Convert Utility   | CnvPrfm  |
| Conversion/Compatibility  | Problems with conversion         | CnvProb  |
| Conversion/Compatibility  | Rewriting/Optimizing for 2.0     | CnvOpt20 |
| Conversion/Compatibility  | Running 1.x databases from 2.0   | CnvRun1x |
| Conversion/Compatibility  | Secured Databases                | CnvScrdb |
| Documentation/Sample Apps | Cue Cards                        | DcmCucd  |
| Documentation/Sample Apps | Help                             | DcmHlp   |
| Documentation/Sample Apps | Manuals                          | DcmMnl   |
| Documentation/Sample Apps | NWIND                            | DcmNwd   |
| Documentation/Sample Apps | Orders                           | DcmOrds  |
| Documentation/Sample Apps | Other                            | DcmOthr  |
| Documentation/Sample Apps | Solution                         | DcmSltn  |
| Environ/Config            | GP Fault                         | EvnGpf   |
| Environ/Config            | Hardware                         | EvnHdwr  |
| Environ/Config            | Networks                         | EvnNtw   |
| Environ/Config            | Operating System                 | EvnOs    |
| Environ/Config            | Other                            | EvnOthr  |
| Environ/Config            | Other Drivers                    | EvnOdr   |
| Environ/Config            | System Performance               | EvnPrfm  |
| Environ/Config            | TEMP Directory                   | EvnTemp  |

|                        |                                |          |
|------------------------|--------------------------------|----------|
| Environ/Config         | TSR Problems                   | Evntsr   |
| Environ/Config         | Video Drivers                  | EvnVido  |
| Expressions            | Expression Builder             | ExrBlDr  |
| Expressions            | Functions                      | ExrFnc   |
| Expressions            | General                        | ExrGnrl  |
| Expressions            | Nulls and Zero-length strings  | ExrNls   |
| Expressions            | Other                          | ExrOthr  |
| Expressions            | Referencing Access Objects     | ExrObj   |
| Expressions            | String manipulation            | ExrStrg  |
| Forms (Specific)       | Button Design Builder          | FmsButb  |
| Forms (Specific)       | Combo/List Box                 | FmsCmbo  |
| Forms (Specific)       | Combo/List Box Builder         | FmsCmbob |
| Forms (Specific)       | Custom Help/Help Compiler      | FmsHlp   |
| Forms (Specific)       | Events (Calling Macro/Mods)    | FmsEvt   |
| Forms (Specific)       | Form Properties                | FmsProp  |
| Forms (Specific)       | How To                         | FmsHowto |
| Forms (Specific)       | Option Group Builder           | FmsOptB  |
| Forms (Specific)       | Option Groups                  | FmsOpt   |
| Forms (Specific)       | Other                          | FmsOthr  |
| Forms (Specific)       | Problems                       | FmsProb  |
| Forms (Specific)       | Requery/Refresh                | FmsRqry  |
| Forms (Specific)       | SubForm                        | FmsSubf  |
| Forms (Specific)       | Wizard: AutoForm/Quick Form    | FmsWiza  |
| Forms (Specific)       | Wizard: Main/Subform           | FmsWizm  |
| Forms/Reports (Common) | Code Behind Forms/Reports      | FmrCdbeh |
| Forms/Reports (Common) | Code Builder                   | FmrCodeb |
| Forms/Reports (Common) | Color Builder                  | FmrColrb |
| Forms/Reports (Common) | Colors/Color Palette           | FmrColr  |
| Forms/Reports (Common) | Default Properties/Template    | FmrProp  |
| Forms/Reports (Common) | Field List                     | FmrFld   |
| Forms/Reports (Common) | How To                         | FmrHowto |
| Forms/Reports (Common) | Other                          | FmrOthr  |
| Forms/Reports (Common) | Problems                       | FmrProb  |
| Forms/Reports (Common) | Toolbox                        | FmrTlxb  |
| FastTips Q&A scripts   |                                | Fstqa    |
| General Functionality  | Application Design             | GnlApp   |
| General Functionality  | Compact/Repair                 | GnlCmp   |
| General Functionality  | Corrupted Database             | GnlBaddb |
| General Functionality  | Data Entry/Editing             | GnlDe    |
| General Functionality  | Database Documentor/Print Def  | GnlDbdoc |
| General Functionality  | Encrypt/Decrypt                | GnlEncr  |
| General Functionality  | File Manipulation              | GnlFilem |
| General Functionality  | Find/Replace                   | GnlFnd   |
| General Functionality  | Formatting                     | GnlFrmt  |
| General Functionality  | Global Settings (View Options) | GnlGlobl |
| General Functionality  | Input Masks                    | GnlInpt  |
| General Functionality  | MSACC20.INI                    | GnlAcini |
| General Functionality  | MultiUser/Locking              | GnlMu    |
| General Functionality  | Other                          | GnlOthr  |
| General Functionality  | Performance Overall            | GnlPrfm  |
| General Functionality  | Quick Sort                     | GnlQksrt |
| General Functionality  | Reserved Words                 | GnlRsvd  |
| General Functionality  | Validation                     | GnlValid |
| Interoperability       | Custom Controls                | IntpCstm |
| Interoperability       | DDE                            | IntpDde  |
| Interoperability       | Graph                          | IntpGrph |
| Interoperability       | Mail (Send from File menu)     | IntpMail |

|                  |                                  |          |
|------------------|----------------------------------|----------|
| Interoperability | OLE                              | IntpOle  |
| Interoperability | OLE Automation                   | IntpOlea |
| Interoperability | Other                            | IntpOthr |
| Interoperability | Print Merge/Mail Merge           | IntpPrtm |
| Interoperability | VB Mapping Layer                 | IntpVb   |
| Interoperability | Word 6.0/Merge It                | IntpWrd6 |
| ISAM             | Btrieve                          | IsmBtrv  |
| ISAM             | Excel 2.0-4.0                    | IsmExl4  |
| ISAM             | Excel 5.0                        | IsmExl5  |
| ISAM             | Fox/xBase                        | IsmXbase |
| ISAM             | General                          | IsmGnrl  |
| ISAM             | How To                           | IsmHowto |
| ISAM             | Imp/Exp/Attach other Access MDBs | IsmIea   |
| ISAM             | Import/Export Setup (specs)      | IsmSetup |
| ISAM             | Import/Export Text (delimited)   | IsmTxd   |
| ISAM             | Import/Export Text (fixed width) | IsmTxfx  |
| ISAM             | Lotus                            | IsmLotus |
| ISAM             | Other                            | IsmOthr  |
| ISAM             | Paradox                          | IsmPdox  |
| ISAM             | Problems                         | IsmProb  |
| ISAM             | Works                            | IsmWorks |
| Knowledge Base   | Category Codes                   | KbCodes  |
| Macros           | Action Arguments                 | McrArg   |
| Macros           | Actions                          | McrActn  |
| Macros           | Condition Column                 | McrCond  |
| Macros           | General                          | McrGnrl  |
| Macros           | How To                           | McrHowto |
| Macros           | Macro Builder                    | McrBldr  |
| Macros           | Macro Name                       | McrName  |
| Macros           | Other                            | McrOthr  |
| Macros           | Problems                         | McrProb  |
| Modules          | Adding/deleting records          | MdlAdrec |
| Modules          | DAO                              | MdlDao   |
| Modules          | Declarations/DIM                 | MdlDecl  |
| Modules          | Editing Records                  | MdlEdit  |
| Modules          | Find Methods                     | MdlFind  |
| Modules          | General                          | MdlGnrl  |
| Modules          | Immediate Window                 | MdlImmed |
| Modules          | Libraries (MDA)                  | MdlLib   |
| Modules          | Navigating                       | MdlNav   |
| Modules          | Other                            | MdlOthr  |
| Modules          | Performance                      | MdlPrfm  |
| Modules          | Problems                         | MdlProb  |
| Modules          | Querydefs                        | MdlQry   |
| Modules          | Recordsets                       | MdlRcd   |
| Modules          | Transactions                     | MdlTran  |
| Modules          | Using Bookmarks                  | MdlBkmk  |
| Modules          | VB Compatibility                 | MdlVb    |
| Non Product      | Chatter                          | NpdChat  |
| Non Product      | Compliment/complaint             | NpdCmnt  |
| Non Product      | CompuServe                       | NpdCis   |
| Non Product      | Legal/Licensing issues           | NpdLegal |
| Non Product      | Manufacturing Issues             | NpdMfg   |
| Non Product      | Non MS Product                   | NpdNotms |
| Non Product      | Other                            | NpdOthr  |
| Non Product      | Post-Sales Request               | NpdPstsl |
| Non Product      | PSS Support                      | NpdPss   |

|                   |                                 |           |
|-------------------|---------------------------------|-----------|
| Non Product       | Sales Support                   | NpdSales  |
| ODBC              | Access 2.0 ODBC Driver          | ObcAc20   |
| ODBC              | Driver Pack                     | ObcDrvpk  |
| ODBC              | How To                          | ObcHowto  |
| ODBC              | MSysConf                        | ObcMsys   |
| ODBC              | Non-MS Drivers                  | ObcNonms  |
| ODBC              | Oracle                          | ObcOracl  |
| ODBC              | Other                           | ObcOthr   |
| ODBC              | Problems                        | ObcProb   |
| ODBC              | SQL Server: MS                  | ObcSqlms  |
| ODBC              | SQL Server: Sybase              | ObcSqlsy  |
| Output To         | AutoStart option                | OtpAuto   |
| Output To         | General                         | OtpGnrl   |
| Output To         | Microsoft Excel/Analyze It      | OtpExl    |
| Output To         | MS-DOS Text (*.txt)             | OtpTxt    |
| Output To         | Other                           | OtpOthr   |
| Output To         | Problems                        | OtpProb   |
| Output To         | Rich Text Format (*.rtf)        | OtpRtf    |
| Printing          | Controlling Printing Options    | PtrOpt    |
| Printing          | Other                           | PtrOthr   |
| Printing          | P. Preview doesn't match output | PtrPrev   |
| Printing          | Print Dialog Box                | PtrDialog |
| Printing          | Print Fixed Font Output         | PtrFxfnt  |
| Printing          | Print Setup                     | PtrSetup  |
| Printing          | Printing Graphs/Lines           | PtrGraph  |
| Printing          | Printing Limitations            | PtrLimit  |
| Printing          | Problems/Performance            | PtrProb   |
| Programming/Logic | API/DLL Calls                   | PgmApi    |
| Programming/Logic | Error Handling/Debugging        | PgmErr    |
| Programming/Logic | How To                          | PgmHowto  |
| Programming/Logic | Object Manipulation             | PgmObj    |
| Programming/Logic | Other                           | PgmOthr   |
| Programming/Logic | Parsing Strings and Files       | PgmParse  |
| Programming/Logic | Performance/Optimization        | PgmPrfm   |
| Programming/Logic | Process Control                 | PgmPrs    |
| Queries           | Append                          | QryAppnd  |
| Queries           | Criteria                        | QryCrit   |
| Queries           | Crosstab                        | QryCross  |
| Queries           | Delete                          | QryDel    |
| Queries           | Filters                         | QryFiltr  |
| Queries           | General                         | QryGnrl   |
| Queries           | How To                          | QryHowto  |
| Queries           | Joins                           | QryJoin   |
| Queries           | Make Table                      | QryMktbl  |
| Queries           | Other                           | QryOthr   |
| Queries           | Parameters/Referencing          | QryParm   |
| Queries           | Performance                     | QryPrfm   |
| Queries           | Problems                        | QryProb   |
| Queries           | Properties                      | QryProp   |
| Queries           | Rushmore                        | QryRush   |
| Queries           | SQL View/Syntax questions       | QrySqlvw  |
| Queries           | SQL: Data Definition query      | QrySqldd  |
| Queries           | SQL: Pass-Through query         | QryPass   |
| Queries           | SQL: Union query                | QryUnion  |
| Queries           | Top N                           | QryTopn   |
| Queries           | Totals/Aggregate Functions      | QryTotal  |
| Queries           | Updatability/Row Fix-Up         | QryFixup  |

|                    |                                 |          |
|--------------------|---------------------------------|----------|
| Queries            | Update                          | QryUpdat |
| Queries            | Wizard: Archive                 | QryWiza  |
| Queries            | Wizard: Crosstab                | QryWizc  |
| Queries            | Wizard: Find Duplicates         | QryWizd  |
| Queries            | Wizard: Find Unmatched          | QryWizu  |
| Relationships      | Can't save multiple layouts     | RltNosav |
| Relationships      | How to create relationships     | RltCreat |
| Relationships      | How to delete relationships     | RltDel   |
| Relationships      | How to view relationships       | RltView  |
| Relationships      | Need mouse to join tables       | RltMous  |
| Relationships      | On attached Access tables       | RltAttac |
| Relationships      | Other                           | RltOthr  |
| Relationships      | Referential Integrity           | RltRef   |
| Reports (Specific) | Envelopes                       | RptEnv   |
| Reports (Specific) | Events                          | RptEvent |
| Reports (Specific) | Layout                          | RptLayou |
| Reports (Specific) | Mailing Labels                  | RptLabel |
| Reports (Specific) | Other                           | RptOthr  |
| Reports (Specific) | Page Numbers                    | RptPagen |
| Reports (Specific) | Problems                        | RptProb  |
| Reports (Specific) | Report Properties               | RptProp  |
| Reports (Specific) | Sorting and Grouping            | RptSort  |
| Reports (Specific) | SubReports                      | RptSub   |
| Reports (Specific) | Totaling                        | RptTotal |
| Reports (Specific) | Wizard: AutoReport/Quick Report | RptWiza  |
| Reports (Specific) | Wizard: Groups/Totals           | RptWizg  |
| Reports (Specific) | Wizard: Single-Column           | RptWizsi |
| Reports (Specific) | Wizard: Snaking Columns         | RptWizsn |
| Reports (Specific) | Wizard: Summary                 | RptWizsu |
| Reports (Specific) | Wizard: Tabular                 | RptWizt  |
| Security           | Changing ownership              | ScrtOwnr |
| Security           | Default accounts                | ScrtDef  |
| Security           | Implementing                    | ScrtImpl |
| Security           | Other                           | ScrtOthr |
| Security           | Permissions                     | ScrtPerm |
| Security           | Users/Groups                    | ScrtUsr  |
| Security           | Workgroup Administrator         | ScrtAdm  |
| Setup              | Compliance Checking             | StpComp  |
| Setup              | Errors starting Access          | StpErr   |
| Setup              | How To                          | StpHowto |
| Setup              | Install Options (Typical, etc.) | StpOptio |
| Setup              | Install: General Issues         | StpGnrl  |
| Setup              | Install: Side by side (1.x&2.0) | StpSide  |
| Setup              | Install: Update (2.0 over 1.x)  | StpUpdat |
| Setup              | Maintenance Install             | StpMaint |
| Setup              | Network Install                 | StpNet   |
| Setup              | ODBC Setup                      | StpOdbc  |
| Setup              | Other                           | StpOthr  |
| Setup              | Registration errors (REG.DAT)   | StpReg   |
| Setup              | Setup fails                     | StpFail  |
| Tables             | Counters                        | TblCount |
| Tables             | Data Types                      | TblDatyp |
| Tables             | Database Design                 | TblDsign |
| Tables             | Field Builder                   | TblFldb  |
| Tables             | Field Properties                | TblFldp  |
| Tables             | How To                          | TblHowto |
| Tables             | Modify                          | TblModfy |

|                |                             |           |
|----------------|-----------------------------|-----------|
| Tables         | Other                       | TblOthr   |
| Tables         | Primary Keys/Indexes        | TblPriky  |
| Tables         | Problems                    | TblProb   |
| Tables         | Wizard: Table               | TblWiz    |
| User Interface | Database window (container) | UifDbwin  |
| User Interface | Datasheet view              | UifDatash |
| User Interface | Dialog Boxes                | UifDialog |
| User Interface | Keyboard                    | UifKeybd  |
| User Interface | Menu Builder                | UifMenub  |
| User Interface | Menus                       | UifMenu   |
| User Interface | Other                       | UifOthr   |
| User Interface | Problems                    | UifProb   |
| User Interface | Right mouse menus           | UifRtmous |
| User Interface | Toolbars                    | UifToolbr |
| Wizards        | Add-in Manager              | WzAdmgr   |
| Wizards        | Attachment Manager          | WzAtmgr   |
| Wizards        | Auto Dialer Builder         | WzAutod   |
| Wizards        | General                     | WzGnrl    |
| Wizards        | How To                      | WzHowto   |
| Wizards        | Import Database             | WzImport  |
| Wizards        | Other                       | WzOthr    |
| Wizards        | Problems                    | WzProb    |

#### REFERENCES

=====

For more information about how to query the KB using categories and subcategories, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q119525

TITLE : INF: How to Query on Category Keywords

For more information about how to query using the Find button, click the Find button, then click the Hints button in the Search dialog box.

Additional reference words: 1.00 1.10 2.00 kbkeyword

KBCategory: kbref

KBSubcategory: KbCodes

**INF: Microsoft Access KB Help Files for Versions 1.x and 2.x**  
**Article ID: Q116190**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

SUMMARY

=====

The Microsoft Access Knowledge Base (KB) Help files contain hundreds of technical articles written mostly by Microsoft Product Support Services (PSS) Support Engineers. To search for articles containing a specific word or text string, select the KB Help window and click the Find button on the toolbar.

There are several types of KB articles in the Help files:

- Titles starting with "INF" contain helpful tips and coding examples.
- Titles starting with "PRB" document unexpected behaviors.
- Titles starting with "PRA" document known problems.
- Titles containing "Question & Answer" are from the FastTips service.

There are two KB Help files available, one for Microsoft Access version 1.x and one for version 2.x. To decrease the file sizes, and therefore the time and expense of downloading these files, we exclude from the Help files articles that Support Engineers seldom use when helping customers.

If you are looking for a specific article or want to see what's new, you can browse the KB on CompuServe, GENie, or the Internet. Once you log on to the service with your account information, follow these instructions to find the KB:

- On CompuServe, type "GO MSKB" (without quotation marks)
- On GENie, type "M 505" (without quotation marks)
- On Internet, look in DESKAPPS\ACCESS\KB\README.TXT

You can obtain the Microsoft Access Knowledge Base Help files either by calling the Microsoft FastTips system (described in detail later in this article) and having disks mailed to you, or by downloading the files from one of the services listed below.

MORE INFORMATION

=====

Download Services

-----

Each KB Help file is available as a self-extracting file called either ACCKB1.EXE (for version 1.x) or ACCKB2.EXE (for version 2.x) in the Microsoft Software Library (MSL) on the following services:

- CompuServe



GO MSL  
Search for ACCKB1.EXE or ACCKB2.EXE  
Display results and download

- Microsoft Download Service (MSDL)  
Dial (206) 936-6735 to connect to MSDL  
Download ACCKB1.EXE or ACCKB2.EXE
- Internet (anonymous FTP)  
ftp ftp.microsoft.com  
Change to the softlib/mslfiles directory  
Get ACCKB1.EXE or ACCKB2.EXE

#### Microsoft FastTips

-----

FastTips is an interactive telephone system that you can call 24 hours a day, 7 days a week for information on many Microsoft products. You can order the Microsoft Access KB Help files to be sent by mail for no charge by dialing the FastTips line at 1-800-936-4100 and following these steps:

1. Press 4 for Microsoft Access.
2. Press 1 for version 2.x or press 2 for version 1.x.
3. Press 1 for Express Order Service.
4. When you are prompted, press 2 for a mail delivery method.
5. At the tone, enter the number of the Item ID: 1028 (version 2.x) or 0928 (version 1.x) and then press #.
6. Press # to verify the order is correct.
7. When you are prompted, leave your name and mailing address in the voice mailbox.
8. When you are prompted, press \* to finish your order.

Additional reference words: 1.00 1.10 2.00 knowledgebase  
KBCategory: kbusage kbfile  
KBSubcategory: Softlib

## **INF: Microsoft Access Repair Utility Enhanced**

**Article ID: Q123588**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

The Microsoft Access repair utility has been enhanced so that more databases can be repaired successfully. The enhancement enables the repair utility to traverse a table from beginning to end, and from end to beginning if needed, allowing the utility to recover data pages that could not be recovered before the enhancement. A data page can become damaged if the computer is turned off while a data page contains changes that have not been saved.

### MORE INFORMATION

=====

This enhancement is included in the Microsoft Jet database engine version 2.5, which is available with the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

### REFERENCES

=====

For more information about repairing databases, search for "repairing databases" then "Repairing a Database" using the Microsoft Access Help menu.

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kbusage

KBSubcategory: GnlCmp

**INF: Microsoft Access, Scanners, and Scanning Software**  
**Article ID: Q103631**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article contains general information about Microsoft Access, scanners, and scanning software.

MORE INFORMATION

=====

To scan an image into Microsoft Access, both a scanner and scanning software are required. The following is a list of scanners that are currently supported by Microsoft Windows:

| Scanner      | Manufacturer    | Phone number   |
|--------------|-----------------|----------------|
| ScanJet      | Hewlett-Packard | (208) 323-2551 |
| ScanMan      | Logitech        | (510) 795-0427 |
| Hand Scanner | Marstek         | (714) 833-7740 |

The Hewlett-Packard (HP) ScanJet provides TWAIN compliance, the industry's standard protocol for scanning and acquiring graphics from software applications.

If the software that comes with the scanner saves the image in a format that a Windows graphics program can recognize, Microsoft Access can display the image.

The following is a list of graphics software packages and their currently supported formats:

| Software   | Graphic Formats     | Phone number   |
|------------|---------------------|----------------|
| Image-In   | .BMP .EPS .TIF      | (800) 345-3540 |
| CorelDRAW  | .BMP .EPS .TIF .PCX | (613) 728-8200 |
| Paintbrush | .BMP .PCX .MSP .DIB | (206) 426-9400 |

Image-In is a computer-aided design (CAD) graphics software package that ships with the Marstek Hand Scanner.

The following procedure explains how you can use Paintbrush to bring an image into Microsoft Access from a scanner:

1. Scan the image and save it in .BMP file format using the scanner software.
2. Start Microsoft Access. In the sample database NWIND.MDB, open the Employees form.

3. From the Records menu, choose Data Entry.
4. Select the Photo field.
5. From the Edit menu, choose Insert Object.
6. In the Insert Object dialog box, choose the File button.
7. Select your file from the list.

The scanned image will display in the Photo field.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, pages 316-339

The following products are manufactured by vendors independent of Microsoft: CorelDRAW by Corel Corporation; ScanJet by Hewlett-Packard Company; and ScanMan by Logitech. We make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 1.00 1.10 2.00 WinIss ole twain picture

KBCategory: kbenv

KBSubcategory: EvnOs

**INF: Microsoft ADT Disk Contents (3.5 Inch, 1.44 MB)**

**Article ID: Q120476**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SUMMARY

=====

This article lists the directory contents of the disks shipped with the Microsoft Access Developer's Toolkit version 2.0 (3.5 inch, 1.44 MB), and of the Jet 2.0/Visual Basic 3.0 Compatibility Layer disk (3.5 inch, 1.44 MB).

MORE INFORMATION

=====

Disk 1

-----

|            |     |                 |          |        |
|------------|-----|-----------------|----------|--------|
| ACMSETUP   | EX_ | 106,497         | 04-14-94 | 12:00a |
| ACMSETUP   | HL_ | 12,618          | 04-14-94 | 12:00a |
| BULLET     | BM_ | 58              | 04-14-94 | 12:00a |
| DECOMP     | EXE | 39,044          | 04-14-94 | 12:00a |
| HC31       | ER_ | 2,496           | 04-14-94 | 12:00a |
| MSACAHBB   | DL_ | 64,531          | 04-14-94 | 12:00a |
| MSCPDIS    | DL_ | 8,805           | 04-14-94 | 12:00a |
| MSSETUP    | DL_ | 91,812          | 04-14-94 | 12:00a |
| README     | TXT | 23,081          | 04-14-94 | 12:00a |
| SETUP      | EXE | 39,856          | 04-14-94 | 12:00a |
| SETUP      | INI | 149             | 04-14-94 | 12:00a |
| SETUP      | IN_ | 1,396           | 04-14-94 | 12:00a |
| SETUP      | LST | 510             | 04-14-94 | 12:00a |
| SETUP      | STF | 7,617           | 04-14-94 | 12:00a |
| SETUPWIZ   | MD_ | 175,215         | 04-14-94 | 12:00a |
| SWU2016    | DL_ | 14,682          | 04-14-94 | 12:00a |
| WZBLDR     | MD_ | 286,648         | 04-14-94 | 12:00a |
| WZFRMRPT   | MD_ | 338,291         | 04-14-94 | 12:00a |
| WZQUERY    | MD_ | 132,405         | 04-14-94 | 12:00a |
| _MSSETUP   | EX_ | 9,147           | 04-14-94 | 12:00a |
| 20 file(s) |     | 1,354,858 bytes |          |        |

Disk 2

-----

|           |     |                 |          |        |
|-----------|-----|-----------------|----------|--------|
| ICONWRKS  | BM_ | 482             | 04-14-94 | 12:00a |
| ICONWRKS  | HP_ | 654             | 04-14-94 | 12:00a |
| ICONWRKS  | PH_ | 1,654           | 04-14-94 | 12:00a |
| IWEDIT    | SH_ | 4,262           | 04-14-94 | 12:00a |
| MRBC      | EX_ | 33,380          | 04-14-94 | 12:00a |
| MSARN200  | EX_ | 1,028,509       | 04-14-94 | 12:00a |
| SUMEXMPL  | TX_ | 907             | 04-14-94 | 12:00a |
| WZLIB     | MD_ | 243,903         | 04-14-94 | 12:00a |
| WZTABLE   | MD_ | 95,027          | 04-14-94 | 12:00a |
| 9 file(s) |     | 1,408,778 bytes |          |        |

Disk 3

```

COMPRESS EX_ 32,402 04-14-94 12:00a
GRAPH5 EX_ 879,114 04-14-94 12:00a
HC31 EX_ 100,623 04-14-94 12:00a
ICONWRKS RT_ 22,111 04-14-94 12:00a
MSACAL20 HL_ 17,562 04-14-94 12:00a
MSAREG EX_ 3,179 04-14-94 12:00a
MSASB20 HL_ 13,526 04-14-94 12:00a
OC1016 DL_ 141,005 04-14-94 12:00a
SHED EX_ 43,767 04-14-94 12:00a
SHED HL_ 64,644 04-14-94 12:00a
WIN30API TX_ 25,464 04-14-94 12:00a
WIN31EXT TX_ 7,765 04-14-94 12:00a
WINMMSYS TX_ 12,360 04-14-94 12:00a
 13 file(s) 1,363,522 bytes
```

Disk 4

```

DBSTRUCT MD_ 271,287 04-14-94 12:00a
MSACAL20 OC_ 88,254 04-14-94 12:00a
MSASB20 OC_ 86,935 04-14-94 12:00a
OUTL1016 HL_ 94,843 04-14-94 12:00a
OUTL1016 OC_ 193,357 04-14-94 12:00a
OUTLINE MD_ 102,952 04-14-94 12:00a
WIN31API HL_ 254,982 04-14-94 12:00a
WZOUTL MD_ 42,118 04-14-94 12:00a
 8 file(s) 1,134,728 bytes
```

Jet 2.0/Visual Basic 3.0 Compatibility Layer

```

ACC2COMP TXT 19,512 04-14-94 12:00a
ACMSETUP EX_ 106,497 04-14-94 12:00a
ACMSETUP HL_ 12,618 04-14-94 12:00a
BTRV200 DL_ 56,473 04-14-94 12:00a
CRWACC20 EX_ 8,122 04-14-94 12:00a
MSACAHBB DL_ 64,531 04-14-94 12:00a
MSAJT112 DL_ 8,173 04-14-94 12:00a
MSAJT200 DL_ 500,727 04-14-94 12:00a
MSCPYDIS DL_ 8,805 04-14-94 12:00a
MSSETUP DL_ 91,812 04-14-94 12:00a
PDBJET DL_ 29,735 04-14-94 12:00a
PDCTJET DL_ 18,881 04-14-94 12:00a
PDIRJET DL_ 19,784 04-14-94 12:00a
PDX200 DL_ 120,351 04-14-94 12:00a
SETUP EXE 39,856 04-14-94 12:00a
SETUP INF 1,866 04-14-94 12:00a
SETUP INI 149 04-14-94 12:00a
SETUP LST 507 04-14-94 12:00a
SETUP STF 3,441 04-14-94 12:00a
VBDB300 DL_ 48,229 04-14-94 12:00a
XBS200 DL_ 150,214 04-14-94 12:00a
MSSETUP EX 9,147 04-14-94 12:00a
 22 file(s) 1,319,430 bytes
```

Additional reference words: 2.00  
KBCategory: kbref

KBSubcategory: GnrlOthr

## **INF: Microsoft Graph Changes May Not Be Retained**

Article ID: Q97774

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

When you create a graph on a form or report and later make some changes to the formatting or to the data itself within Microsoft Graph, these changes may not appear in the form or report.

This article discusses the different sources of information that Graph normally uses and which sources take precedence over others in the hierarchy of graph changes.

### MORE INFORMATION

=====

There are several sources of information for a graph: the Microsoft Graph application, the contents of the graph object's RowSource property, and the data in the underlying table or query.

Microsoft Access regenerates the graph each time you print or preview the graph, or switch to Form view. When Microsoft Access regenerates the graph, it reads the SQL statement in the object's RowSource property. The name of the graph and the data to be graphed are obtained from this SQL statement. Microsoft Access obtains the format of the data from the table or query that is the source of the graph's SQL statement.

The precedence for changes is determined as follows:

- Anything that you change in Microsoft Graph can be overwritten by the SQL statement in the graph's RowSource property or by the underlying data itself. (Examples of what you can change in Microsoft Graph include font, font size, font color, and the type of graph selected.)
- Anything that you change in the SQL statement of the graph's RowSource property can be overwritten by the underlying data. (Examples of what you can change in the SQL statement include the title and labels.)
- Anything that you change in the underlying data has precedence over other changes. (An example of what you can change in the underlying data is the actual format of the data. However, if the SQL statement contains a calculated field, such as =Format([Start Date],"MMM"), the format is determined by the result of this calculation, not by the underlying data.)

### REFERENCES

=====



For more information on customizing the title, labels, and data in a graph, query on "Graph and Format and Label" or "Graph and Title" here in the Microsoft Knowledge Base.

Microsoft Graph "User's Guide," version 3.0, Chapter 3, "Working with a Chart", pages 71-101

Additional reference words: 1.00 1.10 2.00 chart

KBCategory: kbtool

KBSubcategory: IntpGrph

**INF: Modified Toolbar Information Stored In SYSTEM.MDA File**  
**Article ID: Q115896**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

When you modify a built-in toolbar, that information is stored in the SYSTEM.MDA file. The Microsoft Access documentation states that when you modify a built-in toolbar, it is changed in all your databases. Actually, it is changed in all your databases in your Workgroup. If you change Workgroups, or use the same database on a different computer with a different SYSTEM.MDA file, you will not see the modified toolbar.

REFERENCES

=====

For more information about modifying built-in toolbars, search for "toolbars: built-in" then "Modifying Built-In Toolbars" using the Microsoft Access Help menu.

Microsoft Access "User's Guide," version 2.0, Appendix B, "Working with Toolbars," page 717

Additional reference words: 2.00 tool bar

KBCategory: kbui

KBSubcategory: UifToolb

**INF: MS Access Version 2.0 Service Pack Questions and Answers**  
**Article ID: Q123823**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

1. Q. Why is Microsoft releasing a Service Pack for Microsoft Access version 2.0?
  - A. We are releasing the Microsoft Access version 2.0 Service Pack for several reasons. The Service Pack fixes a harmless but irritating problem that causes an internal database flag that is used to determine the state of data in the database to be erroneously set, sometimes causing the error message "Couldn't open SYSTEM.MDA." In order to work around this problem, customers have to run the repair utility on the database, a process that can be lengthy for large databases. The problem does not cause damage to the database or data in it, but can be inconvenient. The Service Pack also includes updated OLE dynamic-link library (DLL) files that allow third-party OLE custom controls to run in Microsoft Access 2.0, updated Paradox, Btrieve, and xBASE ISAM drivers, and an updated Setup Wizard.
  
2. Q. Who should install the Service Pack?
  - A. All Microsoft Access version 2.0 users should install the Service Pack, because it corrects problems and enhances certain features. It is particularly important to those who are:
    - Involved in multiuser database work.
    - Using OLE custom controls in their database applications.
    - Working with Btrieve, Paradox, or xBASE data.

The Service Pack will be available in all the languages that Microsoft Access is available in. Please contact the Microsoft sales office in your country for more details.

3. Q. What is included in the Service Pack?
  - A. The Service Pack includes the latest versions of:
    - The Microsoft Jet database engine (version 2.5).
    - The OLE DLLs (version 2.02).
    - The Btrieve, Paradox, and xBASE ISAM drivers.

The Service Pack also includes an updated Setup Wizard, a brief Readme file, and a Setup program that will install the new files only if it finds Microsoft Access version 2.0 and the Microsoft Jet database engine version 2.0.

4. Q. What is the Microsoft Jet database engine version 2.5?
- A. The Microsoft Jet database engine version 2.5 is an updated version of the database engine that ships with Microsoft Access 2.0 and Microsoft Visual Basic version 3.0. The Microsoft Jet database engine has been modified to enhance database repair functionality and to correct the erroneous setting of an internal database flag that in some cases results in the error message "Couldn't open SYSTEM.MDA."

Note that the Visual Basic Compatibility Layer will continue to work seamlessly with this new version of the Microsoft Jet database engine.

5. Q. What is OLE 2.02?
- A. OLE 2.02 is a minor upgrade to the OLE libraries shipped with Microsoft Access 2.0. These new libraries are included primarily because third-party custom controls will expect to find these libraries installed. Developers using third-party controls or controls distributed by other Microsoft products should install these libraries.

6. Q. How do I obtain the Service Pack?

A. You can obtain the Service Pack from the following sources:

- Internet
- CompuServe (MSACCESS forum and MSL)
- Microsoft Download Service (MSDL) (1-206-936-6735)
- FastTips (1-800-936-4100)
- GENie
- America Online

International customers should contact their local subsidiary to obtain the Service Pack. Microsoft cannot mail the Service Pack disks overseas. The Service Pack will be available in all the languages that Microsoft Access is available in.

7. Q. How much does the Service Pack cost?
- A. There is no charge for the Service Pack.
8. Q. Will Microsoft include the Service Pack files in Microsoft Access version 2.0?
- A. There are no plans to include this software in Microsoft Access version 2.0.
9. Q. My company has several workstations that need the Service Pack. Do I have to obtain as many copies of the Service Pack as we have workstations?
- A. The Service Pack license allows you to install the software on any workstation that has a valid license for Microsoft Access. You need only obtain one copy of the Service Pack to install it on these computers.

10. Q. What is the enhanced repair functionality in the Microsoft Jet database engine version 2.5?
- A. The Microsoft Jet database engine version 2.5 can repair or recover information in some cases where it was not possible to do so before. For detailed information, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q123588  
TITLE : INF: Microsoft Access Repair Utility Enhanced

11. Q. What files are affected by the Microsoft Access version 2.0 Service Pack?

- A. The Service Pack updates the following existing files:

- BTRV200.DLL
- COMPOBJ.DLL
- CTL3D.DLL
- MSAJT200.DLL
- MSAJU200.DLL
- MSJETERR.DLL
- MSJETINT.DLL
- OLE2.DLL
- OLE2.REG
- OLE2CONV.DLL
- OLE2DISP.DLL
- OLE2NLS.DLL
- OLE2PROX.DLL
- PDX200.DLL
- STDOLE.TLB
- STORAGE.DLL
- TYPELIB.DLL
- XBS200.DLL

The Service Pack adds the following new files:

- JETVER.TXT
- READSRV.TXT

12. Q. Is my Visual Basic installation affected by the Service Pack?
- A. No. If you are using the Visual Basic Compatibility Layer (VBCL) and you have updated to the Microsoft Jet database engine version 2.0, the Microsoft Jet database engine will be upgraded for you automatically when you install the Service Pack. Your applications will run without change. If you are using Visual Basic 3.0 and have not installed the VBCL, nothing will be updated.
13. Q. Is this a new version of Microsoft Access? Is it 32-bit?
- A. The Service Pack is not a new version of Microsoft Access, and it is not 32-bit. Rather, it is a free upgrade that updates key components of your current Microsoft Access installation to fix certain problems and augment specific functionality.

14. Q. If some of my users install the Service Pack and some do not, will there be problems sharing the same databases or applications?

A. No. Users with the Service Pack will be fully interoperable with users who have not installed the Service Pack. The problem that results in the error message "Couldn't open SYSTEM.MDA" (as described in question 1) may still occur until all users have installed the Service Pack.

15. Q. How can I get a complete list of all the problems fixed and differences in the Service Pack?

A. Search in the Microsoft Access Knowledge Base (KB) for "Jet 2.5." You can obtain the Microsoft Access KB Help files by calling the Microsoft FastTips system at 1-800-936-4100 and having disks mailed to you, or by downloading the files from any of the following services. When you connect to one of the following services with your account information, follow these instructions to find the KB Help files:

- On Internet, look in DESKAPPS\ACCESS\KB\README.TXT.
- On CompuServe, type "GO MSKB" (without quotation marks).
- On GENie, type "M 505" (without quotation marks).

#### REFERENCES

=====

For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kbusage

KBSubcategory: NpdPss

**INF: Network Users Can Open Their Own SYSTEM.MDA Exclusively**  
**Article ID: Q90851**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

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.

Additional reference words: 1.00 1.10 2.00 WinIss jet

KBCategory: kbenv

KBSubcategory: EvnOs

**INF: New Events and Properties in Microsoft Access v. 2.0**  
**Article ID: Q113302**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

Microsoft Access version 2.0 has many new events and properties. The new events and properties are listed below in alphabetical order.

MORE INFORMATION

=====

- A -

Action  
AfterDelConfirm  
AfterInsert  
Alias  
AllowZeroLength  
AllowZeroLength (Data Access)

Attributes  
AutoActivate  
AutoCenter  
AutoExpand  
AutoTab

- B -

BeforeDel  
BeforeInsert

Bookmark (Data Access)  
BorderStyle

- C -

CacheSize  
CacheStart  
Class  
Clustered  
CollatingOrder  
ColumnHeadings  
ColumnHidden  
ColumnOrder  
Connect

Container  
ControlBox  
Count (Data Access)  
CurrentObjectName  
CurrentObjectType  
CurrentSectionLeft  
CurrentSectionTop  
CurrentView

- D -

DatasheetFontHeight  
DatasheetFontItalic  
DatasheetName  
DatasheetUnderline  
DatasheetWeight  
DataUpdatable  
DateGrouping

DefaultValue (Data Access)  
DestConnectStr  
DestinationDB  
DestinationTable  
Dirty  
DisplayType

- E -

EnterKeyBehavior

EventProcPrefix



- F -

|                   |               |
|-------------------|---------------|
| FastLaserPrinting | ForeignTable  |
| Foreign           | FrozenColumns |
| ForeignName       |               |

- G,H,I,J,K -

|                           |                       |
|---------------------------|-----------------------|
| IgnoreNulls               | InputMask             |
| IgnoreNulls (Data Access) | InSelection           |
| Inherited                 | KeepTogether - Groups |

- L -

|              |              |
|--------------|--------------|
| LastModified | LoginTimeout |
| ListCount    | LogMessages  |
| ListIndex    | lpOLEObject  |

- M -

|           |           |
|-----------|-----------|
| MaxButton | MinButton |
| Me        | Module    |
| MenuBar   |           |

- N -

|                    |                   |
|--------------------|-------------------|
| Name               | NavigationButtons |
| Name (Data Access) | NotInList         |

- O -

|                           |                 |
|---------------------------|-----------------|
| Object                    | OnKeyPress      |
| ObjectPalette             | OnKeyUp         |
| ObjectVerbs               | OnLoad          |
| ObjectVerbsCount          | OnLostFocus     |
| ODBCConnectStr            | OnMouseDown     |
| ODBCTimeout               | OnMouseMove     |
| ODBCTimeout (Data Access) | OnMouseUp       |
| OLEType                   | OnNotInList     |
| OLETypeAllowed            | OnResize        |
| OnActivate                | OnRetreat       |
| OnChange                  | OnTimer         |
| OnClick                   | OnUnload        |
| OnDeactivate              | OnUpdated       |
| OnError                   | OpenArgs        |
| OnGotFocus                | OrdinalPosition |
| OnKeyDown                 | OutputAllFields |
| Owner                     |                 |

- P -

|                 |                       |
|-----------------|-----------------------|
| Pages           | PID                   |
| PaintPalette    | PreviousControl       |
| PaletteSource   | Primary               |
| Password        | Primary (Data Access) |
| PercentPosition | PrtDevNames           |

Permissions

- Q,R -

|                        |                              |
|------------------------|------------------------------|
| QueryTimeout           | ReturnsRecords               |
| Required               | ReturnsRecords (Data Access) |
| Required (Data Access) | RowHeight                    |
| Restartable            | RunPermissions               |

- S -

|              |                  |
|--------------|------------------|
| SelLength    | SourceConnectStr |
| SelStart     | SourceDatabase   |
| SelText      | SourceDoc        |
| ShortcutMenu | SourceField      |
| Size         | SourceItem       |
| SizeMode     | SourceTable      |
| Source       | SourceTableName  |

- T -

|          |               |
|----------|---------------|
| TabIndex | Text          |
| Table    | TimerInterval |
| TabStop  | TopValues     |
| Tag      | Type          |

- U -

|                      |               |
|----------------------|---------------|
| Unique               | UniqueValues  |
| Unique (Data Access) | UpdateOptions |
| UniqueRecords        | UserName      |

- V -

|                              |                     |
|------------------------------|---------------------|
| V1xNullBehavior              | Value               |
| ValidateOnSet                | Value (Data Access) |
| ValidationRule (Data Access) | Verb                |
| ValidationText (Data Access) | Version             |

- W,X,Y,Z -

|              |             |
|--------------|-------------|
| WindowHeight | WindowWidth |
|--------------|-------------|

REFERENCES

=====

For more information on new events and properties in Microsoft Access version 2.0, search for "What's New in Microsoft Access version 2.0," and "properties: reference" using the Microsoft Access Help menu.

Additional reference words: 2.00 PtrIss  
KBCategory: kbusage  
KBSubcategory: FrmrptCtlprp

**INF: No Custom Printer Setup for Mailing Labels ReportWizard**  
**Article ID: Q93023**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The Mailing Labels ReportWizard allows you to choose from 66 different labels. However, the ReportWizard does not verify whether the labels you select will correctly fit on the paper size that you have specified.

You may need to manually customize your printer setup or the number of labels per page after using the ReportWizard.

MORE INFORMATION

=====

The need to customize the printer setup is most common with dot-matrix printers. Label sizes other than .5 inch or 1 inch may not fit evenly on the default 11-inch page.

To manually customize your printer setup or the number of labels per page:

1. Preview your report.
2. From the File menu, choose Print Setup.
3. Customize your margins to accommodate the number and width of the labels.
4. Choose the More button in the Print Setup dialog box to customize the number of rows of labels.

Additional reference words: 1.00 1.10 2.00 wizards

KBCategory: kbusage

KBSubcategory: RptLabel

**INF: Nulls Allowed in Foreign Key with Referential Integrity**  
**Article ID: Q98789**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1 and 2.0
- 

SUMMARY

=====

Microsoft Access allows a Null value in a field that is the foreign key in a relationship if referential integrity is enforced.

To prevent users from leaving the foreign key blank in Microsoft Access version 2.0, set its Required property to Yes. To prevent users from leaving the foreign key blank in Microsoft Access version 1.x, add a table level validation rule to the table on the many side of the relationship. See the example below.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. From the Edit menu, choose Relationships.
3. Under Primary Table, select Customers. Under Related Table, select Orders.

Note the following:

- a. The PrimaryKey Fields and Select Matching Fields both display "Customer ID".
- b. The Enforce Referential Integrity check box is checked.
- c. The Add button is unavailable.

These characteristics indicate an existing relationship between the Customers and Orders tables; thus, you will not be able to make an entry in the Customer ID field of the Orders table if that Customer ID does not exist in the Customers table.

4. Close the Relationships dialog box and open the Orders table.
5. Change the Customer ID field of the first order to "XXXX" and press TAB to try to move to the next record. You receive the error message:

Can't add or change record. Referential Integrity rules require a related record in the table "Customers."

6. Choose OK.

7. Delete the entry in the Customer ID field. Make sure the field is completely empty. Press TAB to move to the next record.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, page 54

Additional reference words: 1.00 1.10 2.00 null compact cascade

KBCategory: kbusage

KBSubcategory: GnlApp

**INF: Number of Times a Custom Function Executes in a Query**  
**Article ID: Q98788**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

A custom function called from a query will execute as follows:

- One time per query if the expression does not reference a field
- One time per record if the expression does reference a field
- Two times per record if there are criteria on the expression

MORE INFORMATION

=====

To optimize a query, Microsoft Access will not re-execute a custom function unless the value passed to it changes. If the function accepts a field as a parameter, Microsoft Access must re-execute the custom function for each record, since the data may change from record to record. If criteria are placed on the result of the function, Microsoft Access must re-execute the function when applying the criteria.

The examples below (examples A, B, and C) demonstrate each of the three scenarios, using a custom function designed to number the records in a query.

NOTE: These examples are simplified to demonstrate the results you can expect when you use custom functions in a query. Many variables can affect the number of times Microsoft Access re-executes a custom function. For example, if you first minimize and then maximize Microsoft Access, the process of repainting the screen also includes re-executing any custom function that is part of the query result.

Complete the following steps to use examples A, B, and C:

1. Create a new module called RecordNumbers in the sample database NWIND.MDB.
2. Add the following to the module's Declarations section:

```
Option Explicit
Global RecordNum
```

Example A: An Expression that Calculates Only Once Per Query

-----

1. Add the following function to the module you created in the previous procedure:

```

Function ShouldIncrement ()
 RecordNum = RecordNum + 1
 ShouldIncrement = RecordNum
End Function

```

2. Create a new query based on the Employees table. Drag the Last Name field to the query grid and add a column with the following expression:

```
RecordNumber: ShouldIncrement()
```

3. Run the query. In Microsoft Access version 1.x, you will get the following result:

| Last Name | RecordNumber |
|-----------|--------------|
| Davolio   | 1            |
| Fuller    | 1            |
| Leverling | 1            |
| Peacock   | 1            |
| Buchanan  | 1            |
| Suyama    | 1            |
| King      | 1            |
| Callahan  | 1            |
| Dodsworth | 1            |

In Microsoft Access version 2.0, you will get the following result:

| Last Name | RecordNumber |
|-----------|--------------|
| Davolio   | 1            |
| Fuller    | 1            |
| Leverling | 1            |
| Peacock   | 1            |
| Buchanan  | 1            |
| Suyama    | 1            |
| King      | 1            |
| Callahan  | 1            |
| Dodsworth | 1            |
| Hellstern | 1            |
| Smith     | 1            |
| Patterson | 1            |
| Brid      | 1            |
| Martin    | 1            |
| Pereira   | 1            |

Note that the function IncrementRecNum() is executed only once.

Example B: An Expression that Calculates Once Per Record

1. Add the function DoesIncrement() to the module RecordNumbers you created in the first procedure:

```

Function DoesIncrement (AnyValue)
 RecordNum = RecordNum + 1

```

```
DoesIncrement = RecordNum
End Function
```

2. Create a new query based on the Employees table. Drag the Last Name field to the query grid and add a column with the following expression:

```
RecordNumber: DoesIncrement([Employee ID])
```

3. Run the query. In Microsoft Access 1.x, you will get the following result:

| Last Name | RecordNumber |
|-----------|--------------|
| Davolio   | 1            |
| Fuller    | 2            |
| Leverling | 3            |
| Peacock   | 4            |
| Buchanan  | 5            |
| Suyama    | 6            |
| King      | 7            |
| Callahan  | 8            |
| Dodsworth | 9            |

In Microsoft Access 2.0, you will get the following result:

| Last Name | RecordNumber |
|-----------|--------------|
| Davolio   | 1            |
| Fuller    | 2            |
| Leverling | 3            |
| Peacock   | 4            |
| Buchanan  | 5            |
| Suyama    | 6            |
| King      | 7            |
| Callahan  | 8            |
| Dodsworth | 9            |
| Hellstern | 10           |
| Smith     | 11           |
| Patterson | 12           |
| Brid      | 13           |
| Martin    | 14           |
| Pereira   | 15           |

Example C: An Expression that Calculates More Than Once Per Record

1. Create a new query based on the Employees table. Drag the Last Name field to the query grid and add a column with the following expression:

```
RecordNumber: DoesIncrement([Employee ID])
```

NOTE: You will need to complete step 1 of example B in order to have the DoesIncrement() function available.

2. Open the module RecordNumbers in Design view. From the View menu,



choose Immediate Window. Type the following statement to initialize the variable RecordNum:

```
RecordNum = 0
```

3. Run the query. In Microsoft Access 1.x, you will get the following result:

| Last Name | RecordNumber |
|-----------|--------------|
| Davolio   | 1            |
| Fuller    | 2            |
| Leverling | 3            |
| Peacock   | 4            |
| Buchanan  | 5            |
| Suyama    | 6            |
| King      | 7            |
| Callahan  | 8            |
| Dodsworth | 9            |

In Microsoft Access 2.0, you will get the following result:

| Last Name | RecordNumber |
|-----------|--------------|
| Davolio   | 1            |
| Fuller    | 2            |
| Leverling | 3            |
| Peacock   | 4            |
| Buchanan  | 5            |
| Suyama    | 6            |
| King      | 7            |
| Callahan  | 8            |
| Dodsworth | 9            |
| Hellstern | 10           |
| Smith     | 11           |
| Patterson | 12           |
| Brid      | 13           |
| Martin    | 14           |
| Pereira   | 15           |

4. Repeat Step 2 in this example (example C).
5. Add the following criteria under RecordNumber:

```
>=0
```

6. Run the query. In Microsoft Access 1.x, you will now get the following result:

| Last Name | RecordNumber |
|-----------|--------------|
| Davolio   | 10           |
| Fuller    | 11           |
| Leverling | 12           |
| Peacock   | 13           |
| Buchanan  | 14           |
| Suyama    | 15           |

|           |    |
|-----------|----|
| King      | 16 |
| Callahan  | 17 |
| Dodsworth | 18 |

In Microsoft Access 2.0, you will now get the following result:

| Last Name | RecordNumber |
|-----------|--------------|
| -----     | -----        |
| Davolio   | 16           |
| Fuller    | 17           |
| Leverling | 18           |
| Peacock   | 19           |
| Buchanan  | 20           |
| Suyama    | 21           |
| King      | 22           |
| Callahan  | 23           |
| Dodsworth | 24           |
| Hellstern | 25           |
| Smith     | 26           |
| Patterson | 27           |
| Brid      | 28           |
| Martin    | 29           |
| Pereira   | 30           |

In this case, Microsoft Access runs the expression twice, once to create the dynaset and again to check the criteria you specified.

#### REFERENCES

=====

For more information on creating custom counters for forms and queries, query on the following words here in the Microsoft Knowledge Base:

custom and counter

Additional reference words: 1.00 1.10 2.00 top 20 limit counter  
KBCategory: kbusage  
KBSubcategory: QryTotal

**INF: Obtain Name of Current User Database Using Access Basic**  
**Article ID: Q89681**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Microsoft Access version 1.x does not have a built-in way to determine the name of the database that the user currently has open. This article lists a sample Access Basic function you can use in Microsoft Access 1.x to find the name of the database.

In Microsoft Access version 2.0, you can find the name of the database using data access objects (DAO). With DAO, you can use the Name property of the DBEngine object to find the path and name of the currently open database. For more information on the DBEngine object, search for "DBEngine" then "DBEngine Object" using the Microsoft Access 2.0 Help menu.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

Although you can obtain the object variable of the current user database using the CurrentDB() function, you are unable to obtain the text representation of the database name. The example provided shows how a user-defined function called GetUserDBName() obtains a text representation of the user database name.

The GetUserDBName() function returns the name of the database with no path or extension. You must add your own path or extension string to the result of the GetUserDBName() function when necessary.

NOTE: The code to determine the current database does not work with run-time versions of Microsoft Access (MSARN110.EXE).

1. Enter the sample code listed below into a module.
2. From the Immediate window of the module, type the following, and then press ENTER:

```
?GetDBName()
```

The name of the database will be returned.

Sample Module Code

-----

Notes:

- In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.
- You may have some Microsoft Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive a duplicate procedure name error message, remove or comment out the declarations statement in your code.

```
Option Compare Database 'Use database order for string comparisons
```

```
Option Explicit
```

```
Declare Function GetWindow% Lib "USER" Alias "GetWindow" _
 (ByVal hwnd%, ByVal Relationship%)
Declare Function GetClassName% Lib "USER" Alias "GetClassName" _
 (ByVal hwnd%, ByVal ClassName$, ByVal Size%)
Declare Function GetWindowText% Lib "User" Alias "GetWindowText" _
 (ByVal hwnd%, ByVal StringText$, ByVal wInt%)
Declare Function GetActiveWindow% Lib "USER" Alias _
 "GetActiveWindow" ()
Declare Function GetParent% Lib "USER" Alias "GetParent" (ByVal hwnd%)
```

```
Const GW_HWNDNEXT = 2
```

```
Const GW_CHILD = 5
```

```
Const AccessDBC = "Odb"
```

```
Const AccessMDICLIENT = "MDIClient"
```

```
Const ACCESSMain = "OMain"
```

```
Function GetDBName ()
```

```
 Dim hwnd As Integer
```

```
 Dim ClassName As String, ClassLen As Integer
```

```
 Dim Caption As String, CaptionLen As Integer
```

```
 Dim Start As Integer, fMDIClient As Integer
```

```
 Dim RetVal
```

```
 '
```

```
 ' Get the handle to the Access window
```

```
 '
```

```
 hwnd = GetActiveWindow()
```

```
 ClassName = Space(127)
```

```
 ClassLen = GetClassName(hwnd, ClassName, Len(ClassName))
```

```
 While ((Left$(ClassName$, ClassLen) <> "OMain"))
```

```
 hwnd = GetParent(hwnd)
```

```
 ClassLen = GetClassName(hwnd, ClassName, Len(ClassName))
```

```
 Wend
```

```
 '
```

```
 ' Traverse the children looking for the 'MDIClient' child
```

```
 '
```

```
 hwnd = GetWindow(hwnd, GW_CHILD)
```

```
 Do While hwnd And Not fMDIClient
```

```
 ClassLen = GetClassName(hwnd, ClassName, Len(ClassName))
```

```
 If Left$(ClassName, ClassLen) = "MDIClient" Then
```

```

 fMDIClient = True
 Else
 hwnd = GetWindow(hwnd, GW_HWNDNEXT)
 End If
Loop

'
' Traverse children until we find the Database Window 'ODb' child
' .. which has a title of "Database: <app name>"
'
hwnd = GetWindow(hwnd, GW_CHILD)
Do While hwnd
 ClassLen = GetClassName(hwnd, ClassName, Len(ClassName))
 ClassName = Mid$(ClassName, 1, ClassLen)
 If Left$(ClassName, ClassLen) = "ODb" Then
 Caption = Space$(127)
 RetVal = GetWindowText(hwnd, Caption, Len(Caption))
 Start = InStr(Caption, ":") + 2
 GetDBName = Mid$(Caption, Start)
 Exit Function
 End If
 hwnd = GetWindow(hwnd, GW_HWNDNEXT)
Loop

GetDBName = "Not Found"

End Function

Additional reference words: 1.00 1.10 2.00 DDE dbname
KBCategory: kbprg
KBSubcategory: PgmOthr

```

## **INF: ODBC Setup for Microsoft Access and SQL Server**

**Article ID: Q88173**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft SQL Server versions 1.10, 1.11, and 4.20 for MS-DOS and Microsoft OS/2, and Microsoft SQL Server for Windows NT
  - Microsoft SQL Administrator versions 4.20 and 4.21 for Windows
  - Microsoft ISQL for Windows version 4.21
- 

### SUMMARY

=====

The installation of Microsoft ODBC for Microsoft and Sybase SQL Server requires the use of ISQL command line procedures to properly run the INSTCAT.SQL script.

If INSTCAT.SQL has not been properly implemented in SQL Server, the following error will be displayed while attaching a SQL table in Microsoft Access:

```
[Microsoft][ODBC SQL Server Driver] The ODBC catalog stored
procedures installed on server <server_name> are version xx.xxxx;
version xx.xx.xxxx is required to ensure proper operation. Please
contact your system administrator.
```

### MORE INFORMATION

=====

To properly configure SQL Server for use with Microsoft ODBC, you must run the SQL script file INSTCAT.SQL (shipped on the ODBC disk in Microsoft Access version 1.x and on disk 1 in Microsoft Access version 2.0) to set up the proper stored procedures that provide catalog information used by Microsoft ODBC.

NOTE: Microsoft SAF for MS-DOS and OS/2 is limited to 511 lines of code in a SQL script. INSTCAT.SQL has well over 511 lines of code.

Microsoft SQL Administrator version 4.20 will open the INSTCAT.SQL script, but you will not be able to run it. This is because version 4.20 of the SQL Administrator does not recognize the GO command used in INSTCAT.SQL. This has been corrected in the SQL Administrator version 4.21.

In Microsoft SQL Server version 4.20a or earlier, 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 utility ISQL (Interactive SQL). If you are using Microsoft SQL Server version 4.20b for OS/2 or Microsoft SQL Server for Windows NT, you can run the INSTCAT.SQL script from within either ISQL/w (Interactive SQL for Windows), or Microsoft SQL Administrator for Windows version 4.21.

Note that in order to run the INSTCAT.SQL script, you must connect to your SQL Server as the System Administrator, or SA.

The ISQL facility is run from either the MS-DOS or OS/2 command prompt. The syntax for this procedure is:

NOTE: In the following example, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this example.

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

Below are descriptions of the command line switches:

```
/U The login name for the system administrator
/n Eliminates line numbering and prompting for user input
/P Password used for the system administrator (case sensitive)
/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
```

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, for more information.

NOTE: This information is available in the Microsoft Access README.TXT file.

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbusage kberrmsg  
KBSubcategory: ObcOthr

**INF: OLE 2.02 Required to Use Custom Controls**

**Article ID: Q123593**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

OLE controls created with the OLE Control Developer's Kit shipped with Microsoft Visual C++ version 2.0 require users to install OLE version 2.02.

Trying to register such a control in Microsoft Access will cause the following error message:

Insert Object: <OLE Control> was unable to add itself to your system.  
Install the control using a separate setup utility.

Using a utility such as REGSVR.EXE or MSAREG.EXE will result in the error message:

DllRegisterServer in <OLE Control> failed.

MORE INFORMATION

=====

To use OLE controls created with the OLE Control Developer's Kit with Microsoft Access, install OLE 2.02.

OLE 2.02 is available with the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927  
TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

The following OLE 2.02 files are installed when you install the Service Pack:

| File Name    | Size   | Version    |
|--------------|--------|------------|
| COMPOBJ.DLL  | 108544 | 2.02       |
| OLE2.DLL     | 302592 | 2.02       |
| OLE2CONV.DLL | 57328  | 2.01       |
| OLE2DISP.DLL | 164832 | 2.02       |
| OLE2NLS.DLL  | 150976 | 2.02       |
| OLE2PROX.DLL | 51712  | 2.02       |
| STORAGE.DLL  | 157696 | 2.02       |
| TYPELIB.DLL  | 177216 | 2.02       |
| OLE2.REG     | 27026  | Not marked |
| STDOLE.TLB   | 4304   | Not marked |

There is another OLE file called OC25.DLL that can be distributed by



developers. This file is usually packaged with OLE controls.

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kbinterop kberrmsg

KBSubcategory: IntpCstm

## INF: OpenDatabase and CloseDatabase Pseudo Actions

Article ID: Q100148

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

### SUMMARY

=====

Microsoft Access includes two "pseudo actions," or dynamic data exchange (DDE) commands, called OpenDatabase and CloseDatabase. These commands are available through the DDE SYSTEM topic, and allow you to open and close Microsoft Access databases over a DDE channel.

### MORE INFORMATION

=====

The syntax for the OpenDatabase and CloseDatabase pseudo actions is

OpenDatabase <Database\_Location>

CloseDatabase

where <Database\_Location> is the path and filename of the Microsoft Access database to be opened. The CloseDatabase action does not take a parameter since Microsoft Access can have only one database open at a time.

The following sample Microsoft Excel macro demonstrates how the OpenDatabase and CloseDatabase pseudo actions work.

NOTE: This macro will not work with Microsoft Access version 1.0.

1. Start Microsoft Access (Microsoft Access must already be running in order for this macro to work).
2. Open a new macro sheet in Microsoft Excel. Enter the following macro:

| Cell | Command                                   |
|------|-------------------------------------------|
| A1   | PseudoOpenCloseDDEExample                 |
| A2   |                                           |
| A3   | chan=INITIATE("MSACCESS","SYSTEM")        |
| A4   | =APP.ACTIVATE("Microsoft Access",FALSE)   |
| A5   | =EXECUTE(chan,"[opendatabase nwind.mdb]") |
| A6   | =EXECUTE(chan,"[Sample AutoExec]")        |
| A7   | =EXECUTE(chan,"[closedatabase]")          |
| A8   | =TERMINATE(chan)                          |
| A9   | =APP.ACTIVATE(,FALSE)                     |
| A10  | =RETURN()                                 |

3. Size your Microsoft Access and Microsoft Excel windows so that you

can view both applications at the same time.

4. Select cell A3. From the Macro menu, choose Run, and then choose OK.

The commands in cells A3 and A4 initiate a DDE channel to Microsoft Access on the SYSTEM topic and activate the Microsoft Access application.

The command in cell A5 opens the sample database NWIND.MDB in the current working directory, using the OpenDatabase pseudo action.

The command in cell A6 runs the Sample AutoExec macro in the NWIND.MDB database.

The command in cell A7 closes the currently opened database file using the CloseDatabase pseudo action.

The commands in cells A8-A10 terminate the DDE channel, activate the Microsoft Excel application, and end the macro.

#### REFERENCES

=====

For more information on using macro actions in DDE, query on the following words here in the Microsoft Knowledge Base:

dde and macro and action

Additional reference words: 1.00 1.10 2.00 OLE/DDE server

KBCategory: kbinterop

KBSubcategory: IntpOthr

**INF: Opening Databases as Read-Only Causes Error Message**  
**Article ID: Q88649**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You receive the error message

Database '<databasename>' is read-only. You won't be able to save changes made to data or object definitions in this database"

when you open a database.

MORE INFORMATION

=====

This error occurs if you open a database under the following three conditions:

- The file is read-only, whether or not its file attribute is set to read-only.
- The directory in which the file resides is also read-only.
- You are using the /Ro command line parameter.

Note that when a database is opened as read-only, an .LDB file is never created.

The only workaround currently available is to implement security measures. Instead of physically opening the file as read-only, open all objects in the database as read-only. The procedures to implement database security are outlined in:

- Microsoft Access "User's Guide," version 1.0, chapter 25
- Microsoft Access "User's Guide," version 2.0, chapter 14

Additional reference words: 1.00 1.10 2.00 WinIss kberrmsg

KBCategory: kbenv

KBSubcategory: EvnOs

## **INF: OpenTable Method on Attached Table Causes Error**

**Article ID: Q109711**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Since an attached table does not actually exist in the current database, it cannot be opened with Access Basic code as a table object from the current database. Trying to do so will generate the following error message:

```
Can't perform operation; it is illegal.
```

```
-or-
```

```
Invalid operation.
```

### MORE INFORMATION

=====

If the attached table is a Microsoft Access table and does not use an ISAM driver (such as the dBASE, Paradox, Btrieve, or Fox ISAM) or an ODBC driver, you can first use the OpenDatabase method to open the database that contains the attached table. Then, you can use the OpenTable method to open the table. This technique is demonstrated in the sample code below:

```
Dim MyDB as Database
Dim MyTable as Table
Set MyDB = OpenDatabase("NWIND.MDB")
Set MyTable = MyDB.OpenTable("Employees")
```

If the table uses an ISAM or ODBC driver, you must use the CreateDynaset method instead. The following example demonstrates this method. In this example, dbo\_Authors is an attached SQL Server table:

```
Dim MyDB as Database
Dim MySet as Dynaset
Set MyDB = CurrentDB()
Set MySet = MyDB.CreateDynaset("dbo_Authors")
```

### Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. From the File menu, choose Attach Table.
3. In the Data Source box, select dBASE III, and then choose OK.
4. In the File Name box, select NEWCUST.DBF, and then choose Attach.
5. In the Select Index Files dialog box, choose Close. Choose OK in the

confirmation dialog box. In the Select File dialog box, choose Close.

6. Create a new module with the following sample code:

```
'*****
'Declarations section of the module
'*****

Option Explicit

'=====
```

' This function will set the database to the current database  
' and try to open an attached table.  
'=====

```
Function AttachTest()
 Dim MyDB as Database
 Dim MyTable as Table
 Set MyDB = CurrentDB()
 Set MyTable = MyDB.OpenTable("NEWCUST")
End Function
```

7. Type the following in the module's Immediate window, and press ENTER:

```
? AttachTest()
```

An error message will appear. To correct the problem, use the CreateDynaset method rather than the OpenTable method.

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.0, Chapter 8, "Manipulating Data," pages 97-138

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: Overview of Conversion from Version 1.1 to 2.0 Issues**  
**Article ID: Q112120**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

When you convert your Microsoft Access version 1.x databases to version 2.0 format, you may have to make some modifications to make sure your databases will perform correctly.

MORE INFORMATION

=====

There are some differences in the formats of Microsoft Access version 1.x and version 2.0 databases. These differences may cause unexpected results when you are using converted databases. Details of the database format changes from version 1.x to 2.0 are listed in Appendix C of the Microsoft Access 2.0 "Building Applications" manual.

The Convert Database command on the File menu, available only when you do not have a database open, can be used to convert databases from version 1.x format to version 2.0 format. When you convert a database from version 1.x to version 2.0 format, all database objects are converted. The following sections describe how the various database objects are converted:

Tables

-----

Tables are automatically converted to version 2.0 database format. Tables that were attached in version 1.x can still be used, but you should refresh all attachments to take advantage of Microsoft Access 2.0's ability to send conversion functions (such as CLng(), CStr(), and so on) to the server rather than processing them locally.

Use the Attachment Manager Add-in in Microsoft Access 2.0 to refresh all the attachments, rather than deleting and re-attaching each attachment as in Microsoft Access 1.x. If an attachment is deleted, all the properties set in the current database for that attachment are lost. To use the Attachment Manager Add-in, choose Add-ins from the File menu, then choose Attachment Manager.

Queries

-----

Microsoft Access automatically compiles queries before running them. The first time you run a query in version 2.0, the compiled format of the query will be discarded and it will be recompiled. Then, the new compiled format will automatically be saved.

Forms and Reports

-----

Forms and reports are converted temporarily into memory as they are opened. As a result, you can gain a small performance increase if you open and then resave all of your pre-existing forms and reports in Microsoft Access version 2.0, so that Microsoft Access does not have to convert them as you use them.

#### Macros

-----

Macros are not converted. The format for macros is not changed in Microsoft Access version 2.0.

#### Modules

-----

The internal format of modules is changed in Microsoft Access version 2.0. Version 2.0 automatically converts modules when they are loaded into memory. However, the converted code is not saved until you save the module. For that reason, you are prompted to save a converted module the first time you close it whether you make changes to it or not. If the converted module is not saved, the conversions are lost. This prompt occurs every time an unconverted module is closed until you save it.

#### Default Values and Validation Rules

-----

Some default value expressions and validation rules, valid in version 1.x databases, may not be valid in version 2.0 databases.

In Microsoft Access version 2.0, validation rules are evaluated starting at the table level. Validation rules in forms or elsewhere are evaluated in addition to table-level validation. At the table level, version 2.0 does not allow the following in validation rules:

- User-defined functions
- Microsoft Access domain functions, such as DCount()
- Total (aggregate) functions, such as Sum()
- References to other fields for field validation (for record validation, references to other fields in the table are allowed)
- CurrentUser() or Eval() functions

During the conversion process, a table named ConvertErrors will be created and filled with descriptions of any default value expressions or validation rules that were not converted. It may be necessary to check all data objects for "#Name?" errors.

#### Changes to Reserved Words and Function Names

-----

There are many new reserved words and function names. "Reserved" means that the word or function name has special significance to Microsoft Access version 2.0 and cannot be used as a user-specified control name, function name, and so on. The Microsoft Access Help system and Appendix C of the Microsoft Access 2.0 "Building Applications" manual lists new reserved words. General rules include:



- Form controls cannot have the name "form." Also, a report control cannot have the name "report."
- Controls in a form or report cannot have the same name as a section.
- Object names that conflict with reserved words can be qualified by enclosing the object name with square brackets, such as: "[Move].visible" instead of "Move.visible".

#### SendKeys Action or Statement

-----

If you use the SendKeys statement or action to fill in dialog boxes or choose commands from menus, the changes in some version 2.0 dialog boxes and menus may require you to recode the statement. Because changes are likely to occur with each version of Microsoft Access, it is a good idea to try to avoid using SendKeys when possible. See Appendix C of the "Building Applications" manual for tips on avoiding the use of SendKeys.

#### Security

-----

If a version 1.x database is secured, it will remain secure whether it is opened using version 1.x or version 2.0. However, Microsoft Access 2.0 cannot be used to change or add permissions in the database, even by the administrator, until the database is converted to version 2.0.

When you install Microsoft Access 2.0 it creates its own workgroup file (SYSTEM.MDA). If Microsoft Access version 2.0 is installed in the same directory as version 1.x, the version 1.x SYSTEM.MDA file will be renamed SYSTEM1X.MDA.

To make changes to the security of a converted database, you must use a version 2.0 SYSTEM.MDA that has identical groups and users (and identical PIDs) as the original SYSTEM.MDA.

NOTE: PIDs (Personal IDs) in Microsoft Access version 2.0 are the equivalent of PINs (Personal ID Numbers) in version 1.x.

#### How to Create a Secure Workgroup:

1. Use the Workgroup Administrator tool to create a new workgroup. This is a version 2.0 SYSTEM.MDA file.
2. Re-create all the users and groups using the same names and PIDs that were used in Microsoft Access version 1.x.

#### How to Convert a Secure 1.x Database to 2.0 Format:

NOTE: In a secure workgroup, only users with Modify Design permissions to all of the objects can convert a version 1.x format to version 2.0 format. Also, you must assign Modify Design permissions to the version 1.x database in Microsoft Access version 1.x using the version 1.x workgroup.

1. Make sure that no one is using the version 1.x database.

2. Log on to Microsoft Access 2.0 as a member of the Admins group who is not the Admin user.
3. From the File menu, choose the Convert Database command.
4. Select the version 1.x database you want to convert. You will be prompted for the version 2.0 database name.

NOTE: The Convert Database command will force you to choose a new name for the database. This lets you keep a backup copy of your version 1.x database, as once you have converted a database from version 1.x to version 2.0 you CANNOT convert it back to version 1.x.

5. Have your users join the new version 2.0 workgroup (SYSTEM.MDA) by using the Workgroup Administrator tool.

NOTE: You can also accomplish this by modifying the MSACC20.INI file in your Windows directory. In the [Options] section of the file, change the SystemDB entry to point to the version 2.0 SYSTEM.MDA file. The [Options] section of the file will be similar to the example below:

```
[Options]
SystemDB=<microsoft access path>\SYSTEM.MDA
```

#### Verifying Conversion

-----

To determine if a database was converted from Microsoft Access version 1.x to version 2.0 with the Convert Database command correctly, check the V1xNullBehavior property. This property is not available if the database was converted by Access Basic code. It is available only if the database was converted using the Convert Database command on the File menu.

The sample function below uses the V1xNullBehavior property to determine if a database was converted from Microsoft Access version 1.x to version 2.0 with the Convert Database command. The flag variable IsConverted holds the function's result. If IsConverted is True (-1,) then the database was converted with the Convert Database command.

```
Function IsConverted (checkdb As String)
 On Error GoTo NotFound
 Dim MyDatabase As Database, V1xNull As Integer
 Set MyDatabase = DBEngine.Workspaces(0).OpenDatabase(checkdb)

 V1xNull = MyDatabase.properties.V1xNullBehavior
 IsConverted = True
 MyDatabase.Close
 Exit Function

NotFound:
 If Err = 3024 Then
 MsgBox "Could not find database."
 Else
 IsConverted = False
 MyDatabase.Close
 End If
```

```
Exit Function
End Function
```

To use this function, type the following in a module's Immediate window, and then press ENTER:

```
? IsConverted("<yourdatabasename>.MDB")
```

Note that a native version 2.0 database will return a 0 when used with this function.

To check which version of Microsoft Access a SYSTEM.MDA file was created with, use the following sample Access Basic function:

```
Function CheckSysDbVer (sysdbname As String) As String
 Dim d As Database
 Set d = DBEngine(0).OpenDatabase(sysdbname)
 CheckSysDbVer = d.version
 d.Close
End Function
```

To use this function, type the following in a module's Immediate window and then press ENTER:

```
? CheckSysDbVer("<microsoft access path>\SYSTEM.MDA")
```

#### Conversion Errors

-----

If the process of converting a 1.x database to version 2.0 results in any unexpected errors or problems, such as a general protection (GP) fault or Microsoft Access closing itself, try running the Repair command (from the File menu) in Microsoft Access version 1.x on the version 1.x database. Then, try to convert the database again. Some version 1.x databases may have slight errors that go undetected until you try to convert the database in version 2.0, because version 2.0 is better at detecting database problems.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 1

Microsoft Access "Building Applications," version 2.0, Appendix C

For more information about converting databases, search for "Convert Database" then "Convert Database Command" using the Microsoft Access Help menu.

Additional reference words: 2.00 permission upgrading upgrade

KBCategory: kbsetup

KBSubcategory: StpOthr

**INF: Overview of MS Access Developer's Toolkit (ADT) v. 2.0**  
**Article ID: Q112137**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Developer's Toolkit version 2.0
- 

SUMMARY

=====

The Microsoft Access Developer's Toolkit (ADT) version 2.0 is the upgrade to the Microsoft Access Distribution Kit (ADK) version 1.1, and complements the retail version of Microsoft Access version 2.0. Developers can use the ADT to assist them in developing advanced database applications, creating client/server solutions, and developing Wizards and other add-ins for Microsoft Access.

MORE INFORMATION

=====

The ADT provides a run-time version of Microsoft Access 2.0 and tools to assist the developer with the creation and distribution of applications developed using the retail version of Microsoft Access 2.0. The retail version of Microsoft Access 2.0 is not included with the ADT, but can be purchased separately.

The ADT includes a run-time version of Microsoft Graph for distributing graphs and charts with applications, the Microsoft Windows Help compiler for creating custom Help files, and a Setup Wizard for creating customized Setup programs. Code samples for Wizards, functions, and other utilities are also included and can be copied and pasted directly into applications.

Detailed technical documentation and customization tools are provided, including the Access Basic "Language Reference" manual, and the "Advanced Topics" manual, which addresses client/server development, connecting to external data sources, and using data access objects (DAO). In addition, there are sample OLE custom controls including calendar, scroll bar and data outline controls.

The Microsoft Access Developer's Toolkit version 2.0 is scheduled to be available through resellers or direct from Microsoft in the early spring of 1994 with a suggested retail price of \$495. For the product's first 90 days, current Microsoft Access Distribution Kit 1.1 and Microsoft Visual Basic 3.0 for Windows users will receive \$100 off the suggested retail price with an on-box rebate coupon.

Additional reference words: 2.00 executable runtime

KBCategory: kbother

KBSubcategory: NpdOthr

## **INF: Overview of New Query Features in Version 2.0**

**Article ID: Q112136**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article summarizes the new query features and functionality in Microsoft Access version 2.0. This information can also be found using the Microsoft Access Help menu by searching for "What's New in Microsoft Access Version 2.0."

### MORE INFORMATION

=====

#### Rushmore Query Optimization

-----

Microsoft Access uses an optimization method similar to that used by Microsoft FoxPro to speed up many types of queries.

#### Automatic Joins for Fields with the Same Name

-----

Even if you have not defined relationships between tables, Microsoft Access automatically creates joins in a query between tables containing matching fields.

#### Greater Updatability of Query Fields

-----

When a query includes fields from more than one related table, you can update data on both sides of the join in the same query.

#### New Query Properties and Property Sheet

-----

With the new query properties, you can add input masks and formats to your queries, and you can limit the number of records displayed to a specified value. Query properties can be set in the property sheet.

#### Query Datasheet Formatting

-----

As with tables and forms, you can save your datasheet layout in queries. For example, if you freeze a column in a query datasheet, you can save the layout so it will appear the same way when you run the query again.

#### Subqueries

-----

You can use SQL SELECT statements as subqueries within your expressions in

the Criteria row of the Query window.

#### Union and Data-Definition Queries

-----

Using SQL in Microsoft Access, you can create union queries and queries that create, change, or delete tables and indexes in Microsoft Access databases. Use the SQL Specific command on the Query menu in a query's Design view to create these types of queries.

#### Improved SQL View Window

-----

You can enter any SQL statement in the improved SQL View window, including statements that you cannot create in the query's QBE grid. The menu and the toolbar buttons are also available when you are working in a SQL View window.

#### SQL Pass-Through

-----

You can pass SQL statements directly to SQL databases, providing improved connectivity in client-server relationships. You can use SQL pass-through queries as the source for a query, form, or report.

#### REFERENCES

=====

For more information about queries, search for "queries: <option>" then view each of the topics using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: QryOthr

**INF: Overview of Version 2.0 Add-ins, Builders, and Wizards**  
**Article ID: Q112751**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article is an overview of the add-ins, builders, and Microsoft Access Wizards in Microsoft Access version 2.0. Listed below is the name of each add-in, builder, and Wizard, and a brief description of its purpose.

MORE INFORMATION

=====

| Name                  | Description of Purpose                                                           |
|-----------------------|----------------------------------------------------------------------------------|
| Add-in Manager        | Used to install, uninstall, or customize add-ins, builders, and Wizards.         |
| Add-ins               | File menu command to control any current add-in, builder, or Wizard.             |
| Archive Query Wizard  | Creates a query that will copy records from one table into another, new table.   |
| Attachment Manager    | Used to view or refresh the links to attached tables.                            |
| AutoDialer            | Used by Microsoft Access to dial a selected phone number.                        |
| AutoForm              | Creates a form based on the currently selected table or query.                   |
| AutoReport            | Creates a report based on the currently selected table or query.                 |
| Code Builder          | Displays the Module window, where you create or modify a procedure for an event. |
| Color Builder         | Creates custom colors, or you can select colors in the palette.                  |
| Combo box Wizard      | Creates a combo box on a form or report.                                         |
| Command button Wizard | Creates a command button with built-in bitmaps and code.                         |
| Crosstab Query Wizard | Creates a query that will summarize data in row and column format.               |

|                                  |                                                                                                              |
|----------------------------------|--------------------------------------------------------------------------------------------------------------|
| Database Documentor              | Used to print a list of design elements of your database objects.                                            |
| Expression Builder               | Creates expressions using functions, operators, constants, and so on.                                        |
| Field Builder                    | Adds a new field to a table's design from a list of tables and fields.                                       |
| Find Duplicates Query Wizard     | Creates a query that will find duplicate records in a table or query.                                        |
| Find Unmatched Query Wizard      | Creates a query to find records that do not match in other tables or queries.                                |
| Graph Wizard                     | Creates a form containing a graph or creates a graph in an existing form or report.                          |
| Groups/Totals Report Wizard      | Creates a report that groups and aggregates values in tabular format.                                        |
| Import Database Add-in           | Imports all the objects from another Microsoft Access database.                                              |
| Input Mask Wizard                | Adds an input mask to a field in a table or query or a text box on a form.                                   |
| List box Wizard                  | Creates a list box on a form or report.                                                                      |
| Macro Builder                    | Creates a new macro where you select from 47 actions for an event.                                           |
| Mailing Label Report Wizard      | Creates a report used to make mailing labels.                                                                |
| Main/Subform Wizard              | Creates a form based on one or more tables or queries.                                                       |
| Menu Builder                     | Used to create a menu bar that you can attach to a form or report.                                           |
| Microsoft Word Mail Merge Wizard | Merges Microsoft Access data with a Word for Windows document.                                               |
| ODBC Connection String Builder   | Creates a connection string to gain access to a backend server.                                              |
| Option Group Wizard              | Creates an option group on a form or report.                                                                 |
| Picture Builder                  | Used to add a picture to a command button or toggle button.                                                  |
| Query Builder                    | Design or modify a query that is the source for a form, report, list box, combo box, graph, or object frame. |



|                             |                                                                             |
|-----------------------------|-----------------------------------------------------------------------------|
| Single-Column Form Wizard   | Creates a form that displays the contents of each record in a column.       |
| Single-Column Report Wizard | Creates a report that displays the contents of each record in a column.     |
| Summary Report Wizard       | Similar to the Groups/Totals Report Wizard but without detail lines.        |
| Table Wizard                | Assists you in creating a table by offering over 40 sample tables.          |
| Tabular Form Wizard         | Creates a form that displays record values in columns from left to right.   |
| Tabular Report Wizard       | Creates a report that displays record values in columns from left to right. |
| Wizards Customizer          | Customizes add-ins, builders, and Wizards.                                  |
| Zoom box                    | Zooms the contents of an input area using SHIFT+F2.                         |

#### REFERENCES

=====

For more information about Microsoft Access 2.0 add-ins, builders, and Wizards, search for "Wizards: reference" then "Builders, Wizards, and Add-ins" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: GnlOthr

**INF: Page Header Will Span Full Width Of Multi-Column Report**  
Article ID: Q90870

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

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 box as either Vertical (also called snaking columns,) or Horizontal. This article describes the steps to create a report with 3 across, 2 inch wide snaking columns.

MORE INFORMATION

=====

To create a grouped, multiple-column report where the page header spans the width of the page, follow these steps:

1. In report Design view, drag the right edge of any report section to the desired width of the report. For this example, drag the edge over to the 7 inch mark on the ruler.
2. Choose Sorting and Grouping from the View menu and enter a field or expression in the Field/Expression box to group your data by.
3. In the Group Properties section of the Sorting and Grouping dialog box, enter 'Yes' for Group Header.
4. Using the Toolbox, create a Label and place it in the Page Header section. Enter the text for the label in the Caption property. Size the text box so it is the same width as the Page Header section.
5. Using the Toolbox, create a text box bound to the group field. Place the text box in the Group Header. Size this control 2 inches wide or less, and place the left edge of the box along the left edge of the group section.
6. In the detail section place your bound control(s) within the first 2 inches.
7. From the File menu, choose Print Setup.
8. In the Print Setup dialog box, choose the More button.
9. In the Items Across box, type the number of columns you want in the detail section. For this example, enter 3. Leave row and column spacing as they are.

10. In the Item Size section, clear the Same as Detail check box.
11. In the Width box, type the number of inches you want to allow for each column of data. For this example, enter 2.
12. Under Item Layout, select Vertical. This creates a snaking column report.
13. Choose OK.
14. Click on the group Header Section and display the property sheet. For the "New Row Or Col" property, choose Before Section. This causes the group header to print at the top of the column one time for each new group.
15. Print or Preview the report.

#### REFERENCES

=====

For other articles describing how to print the group header at the top of continued columns, search in this Knowledge Base for:

print and group's and name and subsequent and column and page

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

For additional information on columnar reports, search for "Multiple-column reports" and "NewRowOrCol" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 columns snake

KBCategory: kbusage

KBSubcategory: RptLayou

**INF: Page Locking, No Cluster Indexing Support**  
**Article ID: Q96892**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Cluster indexing and row level locking are not possible with Microsoft Access, unless you are using an attached table.

MORE INFORMATION

=====

These features are not supported because of the variable-length architecture. When using an attached table, Microsoft Access uses the same locking scheme as does the native product. When you are using the installable ISAMs for dBASE, Paradox, and Btrieve, you will do record-level locking. The Paradox ISAM even understands the clustered indexes and takes advantage of these indexes in some queries.

The following products are manufactured by vendors independent of Microsoft: Btrieve by Btrieve Technologies, dBASE by Borland International, Inc., and Paradox by Borland International. We make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: GnlMu

## **INF: Paradox Tables, Indexes, and Multiple Connections**

**Article ID: Q90152**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### 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 in code.

### MORE INFORMATION

=====

If you try to attach 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.

Additional reference words: 1.00 1.10 2.00 index

KBCategory: kb3rdparty

KBSubcategory: IsmPdox

**INF: Part 2 DDE in Visual Basic to Request Data from MS Access**  
**Article ID: Q99405**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This is the second in a series of two articles containing an advanced example that demonstrates how to use Visual Basic to request data from Microsoft Access by using dynamic data exchange (DDE).

MORE INFORMATION

=====

The following Visual Basic example demonstrates how to use DDE to request information by using SQL statements and the rich set of items supported by Microsoft Access to navigate tables. Further, a subroutine that assists in parsing data from the tab-delimited records that are requested is provided.

Specifically, the example does the following:

1. Gets the number of records and fields in the table using the "FirstRow" and "FieldCount" items.
2. Creates an array to store the records when they are retrieved.
3. Requests the data from a table, one record at a time, using the "FirstRow" and "NextRow" items.
4. Requests a list of field names using the "FieldNames" item.
5. Uses a subroutine to parse each tab-delimited record requested from Microsoft Access and places the data in the array mentioned above. (A separate array holds the field names).
6. This step is optional. Once all the data is placed in the array, a Visual Basic grid control will be populated with the array contents. Further, the grid columns will be populated with the names of the fields.

When you run this example, be sure that Microsoft Access is running and the sample database NWIND.MDB is open.

To use this example, create a new Visual Basic program with a text box, a command button, and a grid control with the following properties:

Text Box

-----

Name: Text1

## Command Button

-----

Name: Command1

## Grid

----

Name: Grid1

NOTE: The grid control is available in the Professional Toolkit for Visual Basic version 1.0 and ships with version 2.0 of the Professional Edition of Microsoft Visual Basic for Windows.

Double-click the command button and type the code that follows between the Sub Command1\_Click () and End Sub lines.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Dim CntRows, CntCols As Integer ' Row and column count in the table
Dim Row, Col As Integer ' Current row/column being filled.
Dim Record As String ' Record read from employees table.

' Get the number of records in the Employees Table
Text1.LinkTopic = "MSACCESS|NWIND;SQL Select Count(*) As _
 EmployeeCount From Employees;"

Text1.LinkItem = "FirstRow" ' Setup to get the first row of data
Text1.LinkMode = 2 ' Establish a manual link
Text1.LinkRequest ' Request the data into the Text1
Text1.LinkMode = 0 ' Terminate the link
CntRows = Val(Text1.Text) ' Convert the result to a number

' Establish a DDE Link to the Employees table
Text1.LinkTopic = "MSACCESS|NWIND;SQL Select * From Employees;"
Text1.LinkMode = 2 ' Establish a manual link

' Get the number of columns in the Employees Table
Text1.LinkItem = "FieldCount" ' Setup to get count of fields/cols
Text1.LinkRequest ' Request the data into Text1
CntCols = Val(Text1.Text) ' Convert the result to a number

' Create the arrays to hold employee information and field names.
ReDim Employees(CntRows, CntCols) As String
ReDim FieldNames(1, CntCols) As String

' Get the field names in the Employees table from Access
Text1.LinkItem = "FieldNames" ' Setup to get the field names
Text1.LinkRequest ' Request the data into Text1

' Parse the tab delimited list of field names and copy it into the
' .. FieldNames array. (ParseRecord subroutine is listed latter)
ParseRecord FieldNames(), 1, (Text1.Text)

' Request each tab delimited record of data one at a time.
For Row = 1 To CntRows
 If Row = 1 Then
```

```

 Text1.LinkItem = "FirstRow" ' Setup for the first record
 Else
 Text1.LinkItem = "NextRow" ' Setup for the next record
 End If
 Text1.LinkRequest ' Request the data into Text1

 ' Parse the tab delimited record and store it in Employees array
 ParseRecord Employees(), Row, (Text1.Text)
Next Row

Text1.LinkMode = 0 ' Terminate link, arrays are filled with data

' OPTIONAL: The following code populates a grid control with the
' contents of the Employees and FieldNames arrays.

' Setup the Grid with the correct number of Rows and Cols
Grid1.Rows = CntRows + 1 ' Add 1 for the row selector buttons
Grid1.Cols = CntCols + 1 ' Add 1 for the column header buttons

' Fill the Grid column header with the data in the FieldNames array
Grid.Row = 0 ' Move to the column header row
For Col = 1 To CntCols
 Grid.Col = Col ' Move to the column
 Grid.Text = FieldNames(1, Col) ' .. and fill it with data
Next Col

' Fill the Grid control rows with the data in the Employees array
For Row = 1 To CntRows
 Grid.Row = Row ' Move to the row
 For Col = 1 To CntCols
 Grid.Col = Col ' Move to the column
 Grid.Text = Employees(Row, Col) ' Fill it with array data
 Next Col
Next Row
End Sub

```

The following subroutine, ParseRecord, is used to extract the tab-delimited data from a record and place the data in an array in the specified row. The subroutine takes the following three arguments:

```

Array() - A two-dimensional array of strings to store data into.
Row - The row position in the array to store the parsed data.
Record - A string with tab delimited information to parse.

```

Create a new module in Visual Basic with the following subroutine:

```

Sub ParseRecord (Array() As String, ByVal Row As Integer, _
 ByVal Record As String)
 Dim Start As Integer ' Start position of field in record
 Dim TabStop As Integer ' Position in the record of the next tab.
 Dim CntCols As Integer ' The number of columns in the Array
 Dim Col As Integer ' The current column (field) being parsed

 CntCols = UBound(array, 2) ' Get count of columns in the array
 Start = 1 ' Start parsing with first character

 ' Find the first tab stop (ASCII character 9) in the record.

```



```
TabStop = InStr(Start, Record, Chr$(9), 0)

' Loop until we fill up all but the last column (Col < CntCols)
' .. or until there are no more columns to read (TabStop <> 0 means
' .. that no tab was found)
Col = 1
While Col < CntCols And TabStop <> 0
 ' extract the field from the string and store it in the array
 Array(Row, Col) = Mid$(Record, Start, TabStop - Start)

 Start = TabStop + 1 ' Increment start to next tab stop
 TabStop = InStr(Start, Record, Chr$(9), 0) ' Find next tab stop
 Col = Col + 1 ' Increment to next column in array
Wend
Array(Row, CntCols) = Mid$(Record, Start) ' Get last col of data
End Sub
```

Additional reference words: 1.00 1.10 2.00 part two vb b\_vbasic  
KBCategory: kbinterop  
KBSubcategory: IntpDde

## INF: Passing Values from a Form to a Parameter Query

Article ID: Q93695

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

When you run a parameter query, Microsoft Access prompts you to enter each parameter in a separate dialog box. This article explains how to create a form that will collect all the values for a parameter query.

### MORE INFORMATION

=====

The following step-by-step example uses the Orders table from the sample database NWIND.MDB. The example shows you two things: how to create a form that prompts for a date value, and how to add a command button to the form to execute the query.

1. Open the NWIND database.
2. Create a query based on the Orders table. Add the following field names with the specified criteria and save the query as MyQuery:

| Field       | Criteria                    |
|-------------|-----------------------------|
| Customer ID |                             |
| Order ID    |                             |
| Order Date  | > [Forms]![MyForm]![MyDate] |

3. Create a new macro with the following action argument, and save it as MyMacro.

| Action    | Action Argument | Value   |
|-----------|-----------------|---------|
| OpenQuery | Query Name      | MyQuery |

4. Create an unbound, blank form by leaving the Select A Table/Query box empty and choosing the Blank Form button.
5. Place a text box control on the form and set the following properties:

|                    |
|--------------------|
| Text Box           |
| -----              |
| Name: MyDate       |
| Format: Short Date |

NOTE: In Microsoft Access versions 1.0 and 1.1, the Name property is called ControlName.

6. Select the Database window by pressing F11.

7. Choose the Macro button with the mouse, and drag MyMacro onto the form. This will generate a command button with the OnPush property set to MyMacro.
8. Click in the Form window to select it, 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 MyMacro macro and run the MyQuery query using the date you entered in the MyDate text box as the query parameter.

Additional reference words: 1.00 1.10 2.00 Parameters

KBCategory: kbusage

KBSubcategory: QryParm

**INF: Placing the User Login Name on a Form or in the Title Bar**  
**Article ID: Q94599**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article explains how to use the User() function in Microsoft Access version 1.x, or the CurrentUser() function in Microsoft Access version 2.0, to display the current user name in a control on a form or in the title bar of the Microsoft Access window.

MORE INFORMATION

=====

In Microsoft Access, you can implement security to control user access to the different objects in your database. Then you can use the User() or CurrentUser() function to display the current user's name.

How to Display User Name in a Control on a Form

-----

Use the following procedure to display a user name in a control on a form:

1. Create a form and place an unbound text box control on the form.
2. For Microsoft Access version 1.x, type "=User()" (without quotation marks) in the text box ControlSource property.

For Microsoft Access version 2.0, type "=CurrentUser()" (without quotation marks) in the text box ControlSource property.

3. Switch to Form view to see the current user name displayed in the control.

How to Display User Name in the Title Bar

-----

Use the following procedure to display a user name in the title bar of the Microsoft Access window:

1. Create a new module.
2. Enter the following Access Basic code:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'=====
'Declarations section of the module
'=====
```

Option Explicit

```
Declare Function SendMessage& Lib "User" (_
 ByVal 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 SetDatabaseWindowWindowTitle ()
'USE ONLY ONE OF THE FOLLOWING TWO LINES OF CODE
'DEPENDING ON YOUR VERSION OF MICROSOFT ACCESS
 stTitle$ = "User = " + User() '>>>>>FOR VERSION 1.X
 stTitle$ = "User = " + CurrentUser() '>>>>>FOR VERSION 2.0
 x% = SendMessage(GetActiveWindow(), WM_SETTEXT, 0, ByVal stTitle$)
End Function
```

```
'=====
'End of code section
'=====
```

3. From the Run menu, choose Compile All.
4. Save the module.
5. Create a new macro with the RunCode action, using the following as the action argument:

```
SetDatabaseWindowWindowTitle()
```

Save the macro and name it ShowUser.

6. Run the ShowUser macro. You should see the Database window title bar change from "Microsoft Access" to "User = <username>".

#### REFERENCES

=====

For more information, search for either "User" then "User Function," or "CurrentUser" then "CurrentUser Function" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

## **INF: Print a Sequential Number for Each Record/Group on Report**

**Article ID: Q98790**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

You can place a text box control that prints a sequential number for each record in the Detail section of a report. This number can be reset so that a separate set of numbers is printed for each group of records. This article explains these procedures by modifying a report in the sample database NWIND.MDB.

### MORE INFORMATION

=====

If you want to print an incremental counter for each record in a report group, place an unbound text box control in the Detail section of the report. To begin the sequence with the number 1, set the ControlSource property to =1, then set the RunningSum property to Over Group.

#### Steps to Reproduce Behavior

-----

1. From the Database window, open the "List of Products by Category" report in Design view.
2. Place the following text box control before the Product Name control in the detail section:

Object: Text Box

-----  
Name: MyLineNumber (in version 1.x, Control Name: MyLineNumber)

ControlSource: =1

RunningSum: Over Group

3. Preview the report. Note that each group's records are sequentially numbered, beginning with the number 1.

You can also increment the counter number for each group in the report by changing the RunningSum property to Over All. To do this, replace step 2 above with the following step:

2. Place the following text box control before the Category Name control in the Category Name Header section:

Object: Text Box

-----  
Name: MyLineNumber2 (in version 1.x, Control Name: MyLineNumber2)

ControlSource: =1

RunningSum: Over All

Repeat step 3 above. Note that each individual record now prints a unique sequential number.

#### REFERENCES

=====

For more information, search on "Calculating the Number of Records in a Report" using the Microsoft Access Help menu

Additional reference words: 1.00 1.10 2.00 reports sequential

KBCategory: kbusage

KBSubcategory: RptOthr

**INF: Programmatically Add Counter Field to Table Using DAO**  
**Article ID: Q113546**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to add a counter field to an existing table using data access objects (DAO).

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The example below demonstrates how to add a counter to a table using DAO and Access Basic:

1. Open the sample database NWIND.MDB.
2. Create the following new table:

```
Table: Counterless

Field Name: MyText
 Data Type: Text
Field Name: MyNumber
 Data Type: Number
Field Name: MyDate
 Data Type: Date/Time
```

3. Save the table as Counterless, without creating a primary key.
4. Create a new module and add the following line to the Declarations section:

```
Option Explicit
```

5. Enter the following function in the module:

```
'*****
' FUNCTION: AddCounter()
'
' PURPOSE: Programmatically adds a counter field to an
' existing table.
'
' ARGUMENTS:
'
' TName: The name of the table.
```



```

' FName: The name of the new counter field.
'
' RETURNS: True (error was encountered) or
' False (no error) as an integer.
'
'*****
Function AddCounter (TName As String, FName As String) As Integer
 Dim DB As Database, TDef As TableDef, Fld As Field

 ' Get the current database.
 Set DB = DBEngine.Workspaces(0).Databases(0)

 ' Open the tabledef to which the counter field will be added.
 Set TDef = DB.TableDefs(TName)

 ' Create a new counter field of type LONG
 ' with the counter attribute.
 Set Fld = TDef.CreateField(FName, DB_LONG)
 Fld.Attributes = DB_AUTOINCRFIELD

 ' Trap for any errors.
 On Error Resume Next

 ' Append the new field to the tabledef.
 TDef.fields.Append Fld

 ' Check to see if an error occurred.
 If Err Then
 AddCounter = False
 Else
 AddCounter = True
 End If

 DB.Close

End Function

```

6. Save the module as DAOAddCounter, and then close it.

7. Create the following macro:

```

Action

RunCode

RunCode argument

Function Name: Addcounter("Counterless","NewID")

```

8. Save the macro as DAOAddCounter, and then close it.

9. Run the macro.

When you open the Counterless table, note the new NewID counter field.

#### REFERENCES

=====

For more information about the CreateField method, search for "CreateField" then "CreateField Method (Data Access)" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: TblModify

## **INF: Query by Form (QBF) Using Dynamic QueryDef**

Article ID: Q117544

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article describes how to use a form to specify the criteria for a query that will be dynamically built by an Access Basic function. This technique is called query by form (QBF).

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

It is possible to create a QBF situation that uses various criteria to filter an existing query. However, this method becomes complex as more and more fields are added. One such complication might be when a user of the form does not enter any value in a criteria field on the form.

A more flexible QBF method is to use an Access Basic function to dynamically create the SQL statement for a query that will be deleted and recreated each time the function runs. Using this method, an empty criteria field will simply not be used as part of the query's WHERE condition.

The following example uses the Orders table in the sample database NWIND.MDB to demonstrate how to use the QBF technique to display the records that match a user's criteria:

1. Open the NWIND database and create a new form.
2. Create six text boxes on the form. Give each of the text boxes one of the following Name properties:

Customer ID  
Ship City  
Ship Country  
Employee ID  
Order Start Date  
Order End Date

3. Create a command button on the form. Set the button's Caption property to "Run Query" (without quotation marks).
4. Set the command button's OnClick property to the following event procedure.

NOTE: In the following sample code, an underscore (\_) at the end of a

line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Dim db As Database
Dim QD As QueryDef
Dim where As Variant
```

```
Set db = dbengine.workspaces(0).databases(0)
```

```
'Delete existing dynamic query, trap error if it does not exist.
On Error Resume Next
db.querydefs.Delete ("Dynamic_Query")
On Error GoTo 0
```

```
'Note Single quotes surrounding text fields [Ship Country]
and [Customer ID]
'Note NO Single quotes surrounding Numeric field [Employee Id]
where = Null
where = where & (" AND [Ship Country]= '" + Me![Ship Country] + "'")
where = where & (" AND [Customer Id]= '" + Me![customer id] + "'")
where = where & (" AND [Employee Id]= " + Me![Employee Id])
```

```
'The following Section is for the Ship City and if it determines the
'first or the last characters of the User's criteria are the "*" it
'will use the "LIKE" operator in the SQL statement instead of the "="
'Also Note Single quotes surrounding text field [Ship City]
```

```
If Left(Me![Ship City],1) = "*" Or Right(Me![Ship City],1) = "*" Then
 where = where & (" AND [Ship City] like '" + Me![Ship City] + "'")
Else
 where = where & (" AND [Ship City] = '" + Me![Ship City] + "'")
End If
```

```
'Note the # symbol surrounding Date field [Order Date]
If Not IsNull(Me![order end date]) Then
 where = where & (" AND [order date] between #" + Me![order start _
 date] + "# AND #" & Me![order end date] & "#")
Else
 where = where & (" AND [order date] >= #" + Me![order start date]_
 + " #")
End If
```

```
MsgBox "Select * from orders " & (" where " + Mid(where, 6) & ";")
'Remove previous line to NOT have the SQL statement displayed
Set QD = db.CreateQueryDef("Dynamic_Query", "Select * from _
orders " & (" where " + Mid(where, 6) & ";"))
DoCmd OpenQuery "Dynamic_Query"
```

5. From the View menu, choose Code. Enter the following Sub procedure in the form module's (general) section:

```
Sub Tests (SQLString, Where_Test)
 If Where_Test = 0 Then
 SQLString = SQLString & " Where "
 Where_Test = Where_Test + 1
 Else
 SQLString = SQLString & " AND "
```

End If  
End Sub

6. View the form in Form view.
7. To run the query, enter the appropriate criteria in the text boxes and then choose the command button.

You may want to try the following sample criteria:

- Ship City: SE\*  
Employee ID: 1
- Customer ID: CACTU  
Order Start Date: 1/1/94

The method demonstrated in the above example has the following features:

- Criteria entered in the text boxes will be evaluated with the AND operator. For example, if you enter "BONAP" in the Customer ID box and "1" in the Employee ID box, the resulting query will display records where [Customer ID] = BONAP AND [Employee ID] = 1.
- If you enter an asterisk (\*) at the beginning or end of the Ship City criteria box, the asterisk will be interpreted as a wildcard and the LIKE operator will be used in the SQL statement. For example, if you enter "Sea\*" in the Ship City box, the resulting query will display records where the Ship City is Seattle.
- If you enter a criterion in the Order Start Date box, but not in the Order End Date box, the resulting query will search for everything after the specified start date (instead of searching for a range of dates).
- After you enter your criteria in the form and choose the command button, the query is built and run. You could also use this technique to print a report based on the query.
- The SQL statement that is built for the dynamic query is displayed in a message box.
- If you do not enter any criteria, the resulting query will return all the records in the table.

#### REFERENCES

=====

For more information about the CreateQueryDef method, search for "CreateQueryDef" then "CreateQueryDef Method" using the Microsoft Access Help menu.

For more information about SQL, search for "SQL: basics" then "Retrieving Data Using SQL" using the Microsoft Access Help menu.

For more information about the QBF technique, search for "QBF" then "Query by Form" using the Microsoft Access Help menu. Or, please see the following articles in the Microsoft Knowledge Base:

ARTICLE-ID: Q95931

TITLE : How to Use the Query by Form (QBF) Technique

ARTICLE-ID: Q109334

TITLE : How to Dynamically Create a Filter

Additional reference words: 2.00 qbf

KBCategory: kbprg

KBSubcategory: PgmObj

## INF: Query with Parameters to Evaluate Complex Criteria

Article ID: Q103252

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Microsoft Access allows the use of variable parameters in queries. This article discusses how to construct a query requiring more than one prompt. Each parameter can be used both as criteria and as a field to allow complicated evaluation of the value entered in each parameter.

### MORE INFORMATION

=====

The following parameter query is based on the Orders table in the sample database NWIND.MDB. It selects orders written between two variable dates provided by the user.

If either the Start Date or the End Date is not entered, the query returns all dates greater than or equal to the Start Date, or less than or equal to the End Date. If neither a Start Date nor an End Date is entered, the query returns all orders.

Query: FindOrdersByDate

---

Field: Order ID

Show: Yes

Field: Order Date

Sort: Ascending

Show: Yes

First Criteria Line: Between [Start Date] and [End Date]

Second Criteria Line: <=[End Date]

Third Criteria Line: >=[Start Date]

Fourth Criteria Line:

Field: [Start Date]

Show: No

First Criteria Line: Is Not Null

Second Criteria Line: Is Null

Third Criteria Line: Is Not Null

Fourth Criteria Line: Is Null

Field: [End Date]

Show: No

First Criteria Line: Is Not Null

Second Criteria Line: Is Not Null

Third Criteria Line: Is Null

Fourth Criteria Line: Is Null

### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, pages 178-181

For more information on similar criteria, query on the following phrase here in the Microsoft Knowledge Base:

query by form

Additional reference words: 1.00 1.10 2.00 queries ui parameter query

KBCategory: kbusage

KBSubcategory: QryParm



## **INF: Questions and Answers About .LDB Files**

**Article ID: Q109957**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

Q: What is an .LDB file and what is it used for?

A: The .LDB file maintains lock information about a database, and each user who opens the database has an entry in it. The .LDB file contains the computer and user names of all users currently accessing a database in a multiuser (network) environment. No other information is read from or written to the .LDB file.

Microsoft Access uses this information to tell which records are locked in a database, and who has them locked, to prevent possible file contention errors and database corruption by multiple users.

However, the .LDB file does not give Microsoft Access information on who has a database opened exclusively, since this is handled by SHARE.EXE (or VSHARE.386 in Windows 3.1). Those programs do not return information to Microsoft Access about who has the database open.

Q: Can I find out the structure of the .LDB file?

A: The format of the .LDB file is not published and is subject to change or elimination in future releases of Microsoft Access. Therefore, code written to manipulate it or read its current format may not work correctly if the .LDB file is eliminated or its format changed in a future release.

Q: Is the information in the .LDB file accessible to users or programmers?

A: No. The information in the .LDB file is only used internally by Microsoft Access and is not designed to be used by users or programmers.

Q: Where and when is an .LDB file created?

A: An .LDB file is created when you open a database. The .LDB file must reside in the same directory as that database. If an .LDB file isn't present, and Microsoft Access requires it, it attempts to create one in the same directory as the database.

Normally, the .LDB file must be in a directory where the user has read-write privileges. However, it is possible to open a database in a directory or device where a user has read-only privileges. To do this, select both the Read Only and the Exclusive check boxes in the Open Database dialog box. The Open Database dialog box can be found by selecting Open Database from the File menu. The combination of read-only and exclusive means that users will not be updating any database information, so record-locking isn't

required. Since record-locking isn't required, Microsoft Access does not attempt to open or create an .LDB file.

This technique also works with directories where the user has read-write privileges and will improve performance if database updating isn't required.

Q: If you move a database, do you also need to move its .LDB file?

A: It isn't necessary to move the .LDB file, since Microsoft Access attempts to re-create the .LDB file if it is not present.

A potential problem can occur if the database is placed in a directory where some users have directory read-only privileges and other users have directory read-write privileges.

When a read-only user opens a database, no information is written to the .LDB file, since that user is not allowed to write any information to the database. When a read-write user opens a database, their name and computer information are written to the .LDB file.

If both of these users (read-only and read-write) open a database and read the same record, a file conflict occurs if the read-write user attempts to update the record, because an update action would conflict with the read-only user. When a file conflict occurs, Microsoft Access pulls data from the .LDB file for what it thinks is the read-only user and uses this data in the file conflict error message given to the read-write user. However, since the read-only user's name and computer weren't written to the .LDB file, and since previously used .LDB files contain the names of previous read-write users, Microsoft Access reads old, outdated user information from the .LDB file and subsequently uses it in the error message instead of valid user information.

To protect against this problem, the database should first be opened and then closed by one of the read-write users before any read-only users open the database. This creates the .LDB file for that database.

Q: Can I erase the .LDB file?

A: You can erase the .LDB file if no one currently has it open. However, unless the database is opened as read-only and exclusive, Microsoft Access will re-create the file the next time the database is opened.

In fact, if you get strange user names reported (such as old user names or garbage) when a lock conflict occurs, your .LDB file is most likely damaged. If you suspect this, it is best to erase your old .LDB file and let Microsoft Access automatically create a new one for you.

Q: What is the maximum size the .LDB file can be?

A: The maximum size is 16K, with each active user taking 64 bytes.

Q: When is information written to the .LDB file?

A: Information is only written to the .LDB file once, when a user opens a database and the .LDB file is created, and active user information is written to it. After that point, Microsoft Access only locks portions of the .LDB file and does not read from it or write to it.

Q: How does Microsoft Access use the information in the .LDB files?

A: Microsoft Access uses .LDB file information to determine who has which files (or records in a file) locked. If Access detects a lock conflict with another user, it reads the .LDB file to get the computer and user name of the user who has the file or record locked. However, the state of the information in the .LDB file has no bearing on the state of the database. If the .LDB file is completely corrupt, everything in the database will still work correctly (although user names reported in lock conflicts might look strange).

In most lock conflict situations, you get a generic "Write conflict" error message that allows you to save the record, copy it to the clipboard, or drop the changes you made. In some circumstances, however, you get the error message:

Couldn't lock table <table name>; currently in use by user  
<user name> on computer <computer name>.

Q: How can I find out which users are accessing my database?

A: Microsoft Access does not provide direct functionality to identify users accessing a database. The level and type of functionality in this area differs, and depends largely on the operating system and network software.

For example, Microsoft Lan Manager provides an Audit API function set that can be used to create and manage real-time audit files. These audit files keep track of server operations and the users who perform them. These API functions can be manipulated through Access Basic to provide user information.

Some operating systems (such as Windows for Workgroups, which comes with a tool called NetWatch) give you the ability to identify users logged onto a shared network directory. However, NetWatch cannot indentify users currently accessing a given file.

#### REFERENCES

=====

Microsoft Lan Manager "A Programmer's Guide," version 2.0, Chapter 8, "Information APIs," pages 234-236

For more information about .LDB files, search for "LDB files" then "Locking Information Files" using the Microsoft Access version 2.0 Help menu.

Additional reference words: 1.00 1.10 2.00 Q&A lockfile  
KBCategory: kbusage

KBSubcategory: Gn1Mu

**INF: Reading, Storing, & Writing Binary Large Objects (BLOBs)**  
**Article ID: Q103257**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can store large data objects (such as sound, video, or graphic data) in a field with the OLE Object data type in a Microsoft Access table. Some large binary data objects cannot be represented, however, if they do not have an OLE server that understands the data being stored. You can also store copies of executable program files or other non-OLE data in a Microsoft Access table. This type of data is referred to as a binary large object bitmap (BLOB).

This article discusses how to use an OLE Object field to store and work with large binary objects that may not be OLE data.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

The following example demonstrates two sample user-defined Access Basic functions that you can use to manage large amounts of binary data in a field with the OLE Object data type. The user-defined functions are ReadBLOB() and WriteBLOB().

- The ReadBLOB() function reads a binary disk file and stores it in an OLE Object field.
- The WriteBLOB() function writes binary data stored in an OLE Object field to a disk file.

This example demonstrates how to copy a binary file into an OLE Object field and then write it back out to a new disk file:

1. Create a new module called BLOB and enter the following lines in the module's Declarations section:

```
Option Explicit
Const BlockSize = 32768
```

2. Enter the following code in the module.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```

Function ReadBLOB (ByVal Source As String, T As Table, sField As _
String)
'*****
' FUNCTION: ReadBLOB()
'
' PURPOSE:
' Reads a BLOB from a disk file and stores the contents in the
' specified table and field.
'
' PREREQUISITES:
' The specified table with the memo field to contain the binary
' data must be opened in Access Basic code and the correct record
' navigated to prior to calling the ReadBLOB() function.
'
' ARGUMENTS:
' Source - The path and filename of the binary information
' to be read and stored.
' T - The table object to store the data in.
' Field - The memo field in table T to store the data in.
'
' RETURN:
' The number of bytes read from the Source file.
'*****
 Dim NumBlocks As Integer
 Dim FileLength As Long, LeftOver As Long
 Dim FileData As String
 Dim SourceFile As Integer
 Dim i As Integer, RetVal As Variant

 On Error GoTo Err_ReadBLOB

 ' Open the source file.
 SourceFile = FreeFile
 Open Source For Binary Access Read As SourceFile

 ' Get the length of the file.
 FileLength = LOF(SourceFile)

 If FileLength = 0 Then
 ReadBLOB = 0
 GoTo Bye_ReadBLOB
 End If

 ' Calculate the number of blocks to read and leftover bytes.
 NumBlocks = FileLength \ BlockSize
 LeftOver = FileLength Mod BlockSize

 ' Initialize the status bar meter.
 RetVal = SysCmd(1, "Reading BLOB", FileLength \ 1000)

 ' Put the table in edit mode so data can be entered.
 T.Edit

 ' Read the leftover data, writing it to the table.
 FileData = String$(LeftOver, 32)
 Get SourceFile, , FileData

```

```

T(sField).AppendChunk (FileData)

' Update the status bar meter.
RetVal = SysCmd(2, LeftOver / 1000)

' Read the remaining blocks of data, writing them to the table.
FileData = String$(BlockSize, 32)
For i = 1 To NumBlocks
 Get SourceFile, , FileData
 T(sField).AppendChunk (FileData)

 ' Update the status bar meter.
 RetVal = SysCmd(2, BlockSize * i / 1000)
Next i

' Update the table record.
T.Update

' Remove the status bar meter.
RetVal = SysCmd(3)

Close SourceFile

ReadBLOB = FileLength

Bye_ReadBLOB:

 Exit Function

Err_ReadBLOB:

 ReadBLOB = -1 * Err
 Resume Bye_ReadBLOB

End Function

Function WriteBLOB (T As Table, sField As String, _
ByVal Destination As String)
'*****
' FUNCTION: WriteBLOB()
'
' PURPOSE:
' Writes BLOB information stored in the specified table and field
' to the specified disk file.
'
' PREREQUISITES:
' The specified table with the memo field containing the binary
' data must be opened in Access Basic code and the correct record
' navigated to prior to calling the WriteBLOB() function.
'
' ARGUMENTS:
' T - The table object containing the binary information.
' sField - The memo field in table T containing the binary
' information to write.
' Destination - The path and filename to write the binary
' information to.
'

```

```

' RETURN:
' The number of bytes written to the destination file.
'*****
 Dim NumBlocks As Integer
 Dim FileLength As Long, LeftOver As Long
 Dim FileData As String
 Dim i As Integer, RetVal
 Dim DestFile As Integer

 On Error GoTo Err_WriteBLOB

 ' Get the size of the field.
 FileLength = T(sField).FieldSize()

 If FileLength = 0 Then
 WriteBLOB = 0
 GoTo Bye_WriteBLOB
 End If

 ' Calculate number of blocks to write and leftover bytes.
 NumBlocks = FileLength \ BlockSize
 LeftOver = FileLength Mod BlockSize

 ' Remove any existing destination file.
 DestFile = FreeFile
 Open Destination For Output As DestFile
 Close DestFile

 ' Open the destination file.
 Open Destination For Binary As DestFile

 ' Initialize the status bar meter.
 RetVal = SysCmd(1, "Writing BLOB", FileLength / 1000)

 ' Write the leftover data to the output file.
 FileData = T(sField).GetChunk(0, LeftOver)
 Put DestFile, , FileData

 ' Update the status bar meter.
 RetVal = SysCmd(2, LeftOver / 1000)

 ' Write the remaining blocks of data to the output file.
 For i = 1 To NumBlocks
 ' Clear the buffer and read a chunk.
 FileData = ""
 FileData = T(sField).GetChunk((i - 1) * BlockSize _
 + LeftOver, BlockSize)

 ' Write the chunk to the output file.
 Put DestFile, , FileData

 ' Update the status bar meter.
 RetVal = SysCmd(2, ((i - 1) * BlockSize + LeftOver) / 1000)
 Next i

 ' Remove the status bar meter.
 RetVal = SysCmd(3)

```



```

 Close DestFile

 WriteBLOB = FileLength

Bye_WriteBLOB:

 Exit Function

Err_WriteBLOB:

 WriteBLOB = -1 * Err
 Resume Bye_WriteBLOB

End Function

Sub CopyFile (ByVal Source As String, ByVal Destination As String)
'*****
' SUB: CopyFile
'
' PURPOSE:
' Demonstrates how to use ReadBLOB() and WriteBLOB().
'
' PREREQUISITES:
' A table called BLOB that contains an OLE Object field called
' Blob.
'
' ARGUMENTS:
' Source - The path and filename of the information to copy.
' Destination - The path and filename of the file to write
' the binary information to.
'
' EXAMPLE:
' CopyFile "c:\access\msaccess.hlp", "c:\access\msaccess2.hlp"
'*****
 Dim BytesRead, BytesWritten
 Dim Msg As String
 Dim db As Database, T As Table

 ' Open the BLOB table.
 Set db = CurrentDB()
 Set T = db.OpenTable("BLOB")

 ' Create a new record and move to it.
 T.AddNew
 T.Update
 T.MoveLast

 BytesRead = ReadBLOB(Source, T, "Blob")

 Msg = "Finished reading "" & Source & """"
 Msg = Msg & Chr$(13) & ".. " & BytesRead & " bytes read."
 MsgBox Msg, 64, "Copy File"

 BytesWritten = WriteBLOB(T, "Blob", Destination)

 Msg = "Finished writing "" & Destination & """"

```

```
Msg = Msg & Chr$(13) & ".. " & BytesWritten & " bytes written."
MsgBox Msg, 64, "Copy File"
```

End Sub

3. Create the following new table and then save it as BLOB:

```
Table: BLOB

Field Name: Blob
Data Type: OLE Object
```

4. With the BLOB module open in Design view, choose Immediate Window from the View menu.
5. Type the following line in the Immediate window and then press ENTER:

```
CopyFile "c:\access\msaccess.hlp", "c:\access\msaccess2.hlp"
```

The ReadBLOB() and WriteBLOB() functions copy the Microsoft Access Help file, which is over 3 MB in size, to the Blob field in the BLOB table and then from there to a disk file called MSACCES2.HLP.

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.1, Chapter 2, "Writing a New Function," pages 8-14, and Chapter 8, "Manipulating Data," pages 98-117

Microsoft Access Distribution Kit "The Secrets of AccessWizards," version 1.1, Chapter 2, "Incantations: AccessWizard Functions and Properties," pages 30-32

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: README.TXT File and Novell Locking Issues**

**Article ID: Q109400**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The README.TXT file shipped with Microsoft Access version 1.1 contains a section titled "Using Microsoft Access on Novell NetWare Servers and Banyan VINES Servers (Increasing the Number of Record Locks)". This information does not apply to Novell NetWare286 (version 2.x). It only applies to Novell NetWare386 (version 3.x) and Banyan VINES.

The README.TXT file does not state that the information does not apply to Novell NetWare286.

MORE INFORMATION

=====

The following information is taken from the README.TXT file:

-----  
Using Microsoft Access on Novell NetWare Servers and Banyan  
VINES Servers (Increasing the Number of Record Locks)  
-----

On Novell NetWare Servers:

If you are attempting to use large Microsoft Access databases on a Novell NetWare server, set the "maximum record locks per connection" to the largest allowable value (10,000). The execution of large bulk operations in Microsoft Access (such as a large update query) can cause the server to run out of record lock connections. If you do not set the number of record locks to the maximum value while using Microsoft Access, it could result in the loss of data due to a server crash.

See your Novell NetWare documentation for a description of how to set the number of record locks.

-----

NOTE: NetWare286 does not have definable configurations for its locking schemes. A server supervisor can use the NetWare console command FCONSOLE to view the locks on the network.

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbsetup  
KBSubcategory: StpNet

**INF: Referencing Fields in Previous or Next Record in a Form**  
**Article ID: Q120489**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to reference a value in a field in the previous or next record in a form for use in a calculation or expression.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The following example demonstrates how to create and use two sample user-defined functions to retrieve the value in a field from the previous or next record on a form:

1. Create a new module and enter the following line in the Declarations section:

Option Explicit

2. Create the following procedures in the module.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```

' FUNCTION: PrevRecVal()
' PURPOSE: Retrieve a value from a field in the previous form
' record.
' PARAMETERS:
' F - The form to get the previous value from.
' KeyName - The name of the form's unique key field.
' KeyValue - The current record's key value.
' fieldNameToGet - The name of the field in the previous
' record to retrieve the value from.
' RETURNS: The value in the field fieldNameToGet from the
' previous form record.
' EXAMPLE:
' =PrevRecVal(Form,"ID",[ID],"OdometerReading")

Function PrevRecVal (F As Form, KeyName As String, KeyValue, _
 fieldNameToGet As String)
 Dim RS As Recordset
```

```

On Error GoTo Err_PrevRecVal

' The default value is zero.
PrevRecVal = 0

' Get the form recordset.
Set RS = F.RecordsetClone

' Find the current record.
Select Case RS.Fields(KeyName).Type
' Find using numeric data type key value?
Case DB_INTEGER, DB_LONG, DB_CURRENCY, DB_SINGLE, _
DB_DOUBLE, DB_BYTE
 RS.FindFirst "[" & KeyName & "] = " & KeyValue
' Find using date data type key value?
Case DB_DATE
 RS.FindFirst "[" & KeyName & "] = #" & KeyValue & "#"
' Find using text data type key value?
Case DB_TEXT
 RS.FindFirst "[" & KeyName & "] = '" & KeyValue & "'"
Case Else
 MsgBox "ERROR: Invalid key field data type!"
 Exit Function
End Select

' Move to the previous record.
RS.MovePrevious

' Return the result.
PrevRecVal = RS(FieldNamesToGet)

Bye_PrevRecVal:
Exit Function
Err_PrevRecVal:
Resume Bye_PrevRecVal
End Function

'*****
' FUNCTION: NextRecVal()
' PURPOSE: Retrieve a value from a field in the next form
' record.
'*****
Function NextRecVal (F As Form, KeyName As String, KeyVal, _
FieldNameToGet As String)
Dim RS As Recordset

On Error GoTo Err_NextRecVal

' The default value is zero.
NextRecVal = 0

' Get the form recordset.
Set RS = F.RecordsetClone

' Find the current record.
Select Case RS.Fields(KeyName).Type
' Find using numeric data type key value?

```

```

Case DB_INTEGER, DB_LONG, DB_CURRENCY, DB_SINGLE, _
DB_DOUBLE, DB_BYTE
 RS.FindFirst "[" & KeyName & "]" = " & KeyValue
' Find using date data type key value?
Case DB_DATE
 RS.FindFirst "[" & KeyName & "]" = #" & KeyValue & "#"
' Find using text data type key value?
Case DB_TEXT
 RS.FindFirst "[" & KeyName & "]" = ' & KeyValue & "'"
Case Else
 MsgBox "ERROR: Invalid key field data type!"
 Exit Function
End Select

' Move to the next record.
RS.MoveNext

' Return the result.
NextRecVal = RS(FieldNamesToGet)

Bye_NextRecVal:
Exit Function
Err_NextRecVal:
Resume Bye_NextRecVal
End Function

```

3. Create the following new table and then save it as Mileage Log:

```

Table: Mileage Log

Field Name: ID
 Data Type: Counter
 Indexed: Yes (No Duplicates)
Field Name: Date
 Data Type: Date/Time
Field Name: Odometer
 Data Type: Number
Field Name: Gallons
 Data Type: Number

Primary Key: ID

```

4. View the Mileage Log table in Datasheet view and enter the following sample data:

| ID | Date    | Odometer | Gallons |
|----|---------|----------|---------|
| 1  | 6/21/94 | 77917.8  | 10.2    |
| 2  | 6/25/94 | 78254.7  | 9.6     |
| 3  | 6/30/94 | 78582.3  | 10      |
| 4  | 7/5/94  | 78918.4  | 10.4    |
| 5  | 7/10/94 | 79223.4  | 9.4     |

5. Use the Form Wizard to create a new tabular form based on the Mileage Log table. Include all the fields except the ID field on the form.
6. View the form in Design view and add the following three text box

controls to the form:

Text Box 1

-----  
Name: PrevOdometer  
ControlSource: =PrevRecVal(Form,"ID",[ID],"Odometer")  
Format: Fixed

Text Box 2

-----  
Name: MilesDriven  
ControlSource: =iif([PrevOdometer]=0,0,[Odometer]-[PrevOdometer])  
Format: Fixed

Text Box 3

-----  
Name: MPG  
ControlSource: =[MilesDriven]/[Gallons]  
Format: Fixed

7. View the form in Form view. The following information will be displayed:

| Date    | Odometer | Gallons | PrevOdometer | MilesDriven | MPG   |
|---------|----------|---------|--------------|-------------|-------|
| 6/21/94 | 77917.8  | 10.2    | 0            | 0           | 0     |
| 6/25/94 | 78254.7  | 9.6     | 77917.80     | 336.90      | 35.09 |
| 6/30/94 | 78582.3  | 10.0    | 78254.70     | 327.60      | 32.76 |
| 7/05/94 | 78918.4  | 10.4    | 78582.30     | 336.10      | 32.32 |
| 7/10/94 | 79223.4  | 9.4     | 78918.40     | 305.00      | 32.45 |

#### REFERENCES

=====

For other methods of referring to a previous record, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q101081

TITLE : Two Methods to Refer to a Field in a Previous Record

Additional reference words: 2.00 running sum

KBCategory: kbusage

KBSubcategory: FmsHowto

## **INF: Relationship Between Btrieve and Microsoft Access**

**Article ID: Q113928**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

Btrieve is not a complete database program, such as Microsoft Access or Microsoft FoxPro. Btrieve is a database engine that provides database functionality to applications developed with programming languages such as Microsoft Visual Basic, Microsoft Visual C++, Pascal, and Assembly Language.

Microsoft Access has the ability to attach, import, and export Btrieve files.

### MORE INFORMATION

=====

Btrieve Technologies defines Btrieve as a record manager, not as a complete stand-alone database application. Btrieve has no user interface; it provides only back-end functionality.

In order to retrieve data using Btrieve, a front end, or user interface, must first be written, usually in a programming language. Microsoft Access has been designed to provide such front end functionality.

The relationship between Btrieve and its front end is similar to the relationship between the Microsoft Jet database engine and a Visual Basic front end application, although the Jet database engine provides greater functionality.

To provide front end functionality for Btrieve, Btrieve Technologies created Xtrieve. Where Btrieve standards and Xtrieve standards differ, Microsoft Access follows the Xtrieve standard.

Many programs have been written to use Btrieve. Several major accounting applications use it to perform database functions. Customers often use Microsoft Access in conjunction with these programs to provide report generation capability. By attaching the Btrieve files, Microsoft Access can read them while the accounting application is still actively using them.

Btrieve is provided by Btrieve Technologies as a dynamic-link library (DLL) file named WBTRCALL.DLL. This DLL file provides the disk I/O and indexing for the Btrieve files. Data Dictionary (DDF) files are used by Xtrieve and Microsoft Access, since Btrieve files do not contain any field information.

The relationship between Microsoft Access and Btrieve is shown in this diagram:

Microsoft Access



|                     |                                                                              |
|---------------------|------------------------------------------------------------------------------|
|                     |                                                                              |
| Jet database engine | Database engine used by Microsoft Access.                                    |
|                     |                                                                              |
| BTRV200.DLL         | Microsoft Access Btrieve ISAM DLL file.                                      |
|                     |                                                                              |
| WBTRCALL.DLL        | This file is provided by Btrieve Technologies.                               |
|                     |                                                                              |
| DDF files           | Data Dictionary files.                                                       |
|                     |                                                                              |
| Hard disk<br>file   | This is the Btrieve file that can be used<br>as a table in Microsoft Access. |

Btrieve and Xtrieve are manufactured by Btrieve Technologies, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 2.00 importing exporting attaching  
 KBCategory: kb3rdparty  
 KBSubcategory: IsmBtrv

**INF: Repeat Group Name at the Top of New Column or Page**  
**Article ID: Q93927**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to create a report that prints "continued" or "Group Name continued" at the top of each new column or page when grouped information flows from one column or page to another.

MORE INFORMATION

=====

Microsoft Access prints a single group header at the beginning of a group to identify the beginning of the group. It does not repeat the header at the top of subsequent columns or pages. However, you can use the procedure described below to force a report to repeat the group name if the records in a group span several columns or pages. This article uses the report List Of Products By Category from the sample database NWIND.MDB.

Three-step Process

-----

The report uses macros to identify the group. It uses the Top property to determine whether a detail section is being formatted at the top of a column or page. If it is, the report shows the contents of the text box that contains the group name and "continued," and it hides the other controls in the detail section. It is a three-step process.

Step One: Create a New Macro Group Containing Three Macros

-----

You need a macro group (Continued) containing three macros (Page Reset, Set Top1, and Print) to check for certain conditions and perform specific actions based on those conditions. By setting conditions, you give the macro the branching characteristics of a program without using a programming language.

The Page Reset macro sets the value of the Top1 text box to the value of the Top property if the current value of Top1 is 0 when Microsoft Access formats the group header.

The Set Top1 macro sets the value of the Top1 text box to 0 when Microsoft Access formats the page header.

The Print macro sets the value of the Top1 text box to the value of the Top property if the value of Top1 is 0 when Microsoft Access formats a detail section. Or, if the Top property is equal to the value of the Top1 text box, it sets the Visible property of the Continued text box to Yes and the Visible property for each of the controls in the detail section to No. In the List Of Products By Category report,

there are four controls to hide in the detail section. It also sets the NextRecord property to False. Otherwise, it hides the message and prints the detail record.

To create the macro group:

1. Create a new macro.
2. From the View menu, choose Macro Names or choose the Macro Names button on the toolbar.
3. From the View menu, choose Conditions or choose the Conditions button on the toolbar.

Microsoft Access displays the complete grid in the Macro window, including the Macro Name, Condition, and Action columns.

4. Enter the macro names, conditions, actions, and action arguments using the following two tables as a guide. Use the first table to fill the grid and use the second to add action arguments to each action. Do not enter the reference letters: [a], [b], [c], and so on. These reference letters are there to help you associate the actions shown in the first table with the action arguments shown in the second table.

| Macro Name | Condition  | Action    | Reference             |
|------------|------------|-----------|-----------------------|
| Page Reset |            | SetValue  | [a]                   |
| Set Top1   | [Top1]=0   | SetValue  | [b]                   |
| Print      | [Top1]=0   | SetValue  | [c]                   |
|            | Top=[Top1] | SetValue  | [d]                   |
|            | ...        | SetValue  | [e] -- repeat for     |
|            | ...        | SetValue  | [e] -- each control   |
|            | ...        | SetValue  | [e] -- in the         |
|            | ...        | SetValue  | [e] -- detail section |
|            | ...        | SetValue  | [f]                   |
|            | ...        | StopMacro | [g]                   |
|            |            | SetValue  | [h]                   |
|            |            | SetValue  | [i] -- repeat for     |
|            |            | SetValue  | [i] -- each control   |
|            |            | SetValue  | [i] -- in the         |
|            |            | SetValue  | [i] -- detail section |

The ellipses in the Condition column are required for lines e, f, and g; enter them exactly as shown. Microsoft Access evaluates macro conditions as true or false. If the expression is true, Microsoft Access performs the action. If it is false, Microsoft Access ignores the action. The ellipses in the Condition column force Microsoft Access to perform the action on that line if the condition on the preceding line is true.

Here is the second table. It gives the action arguments for each action:

| Reference | Action   | Action Arguments |
|-----------|----------|------------------|
| [a]       | SetValue | Item: [Top1]     |
|           |          | Expression: 0    |
| [b]       | SetValue | Item: [Top1]     |
|           |          | Expression: Top  |

|     |           |                                                      |
|-----|-----------|------------------------------------------------------|
| [c] | SetValue  | Item: [Top1]<br>Expression: Top                      |
| [d] | SetValue  | Item: [Continued].Visible<br>Expression: Yes         |
| [e] | SetValue  | Item: [Product Name].Visible<br>Expression: No       |
| [e] | SetValue  | Item: [Product Id].Visible<br>Expression: No         |
| [e] | SetValue  | Item: [Quantity Per Unit].Visible<br>Expression: No  |
| [e] | SetValue  | Item: [Units In Stock].Visible<br>Expression: No     |
| [f] | SetValue  | Item: NextRecord<br>Expression: False                |
| [g] | StopMacro |                                                      |
| [h] | SetValue  | Item: [Continued].Visible<br>Expression: No          |
| [i] | SetValue  | Item: [Product Name].Visible<br>Expression: Yes      |
| [i] | SetValue  | Item: [Product Id].Visible<br>Expression: Yes        |
| [i] | SetValue  | Item: [Quantity Per Unit].Visible<br>Expression: Yes |
| [i] | SetValue  | Item: [Units In Stock].Visible<br>Expression: Yes    |

- The SetValue actions in lines e and i are used to hide the controls in the detail section of your report. The List Of Products By Category report has 4 controls in the detail section: Product Name, Product Id, Quantity Per Unit, and Units In Stock. You should repeat lines e and i for the number of controls in your detail section. For example, if you have only one control in the detail section, you need only one e and one i.
- Save the macro group and specify Continued as the name. This makes Continued the name of the macro group. It now appears in the list of macros and macro groups in the Database window. Use the following syntax to specify each macro in the macro group when you attach a macro to a property on the report:

macrogroupname.macroname

For example, the following specifies the Print macro: continued.print.

#### Step Two: Attach the Macros to the Report

---

- Open the report in Design view.
- Set the following properties for the specified sections by selecting each section with the mouse and viewing its properties:

| Section Name         | Property | Expression           |
|----------------------|----------|----------------------|
| Page Header          | OnFormat | Continued.Page Reset |
| Category Name Header | OnFormat | Continued.Set Top1   |
| Detail               | OnFormat | Continued.Print      |

|                      |           |                    |
|----------------------|-----------|--------------------|
|                      | CanShrink | Yes                |
| Category Name Footer | OnFormat  | Continued.Set Top1 |

This sets the Top1 control to the current value of the Top expression to keep track of the current twip value (position) of the top.

Step Three: Add Controls to the Report and Set Properties

---

1. Open the report in Design view.
2. Create an unbound text box control named Top1.
3. Place the Top1 text box control in the Category Name header section.
4. Assign the following property values to the Top1 text box control:

| Property     | Expression |
|--------------|------------|
| ControlName: | Top1       |
| Visible:     | No         |

5. In the detail section, create a text box bound to an expression. Locate it near the top in the same vertical location as that of the group header label. Assign the following property values to the text box:

| Property       | Expression                      |
|----------------|---------------------------------|
| ControlName:   | Continued                       |
| ControlSource: | =[Category Name] & " continued" |
| Visible:       | No                              |
| CanShrink:     | Yes                             |

NOTE: Replace Category Name with the name of the field being grouped.

6. Set the CanShrink property for the other text boxes in the detail section to Yes.
7. Save and run the report.

The approach described here works well for both single-column and multiple-column reports, but there is a simpler method you can use for a single-column, multiple-page report.

For additional information about the simpler method, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q88156  
 TITLE : INF: "Continued..." Header for Groups That Span Multiple Pages

REFERENCES  
 =====

For additional information about how to make the page header span the full width of a multiple-column report, please see the following article in the

Microsoft Knowledge Base:

ARTICLE-ID: Q90870

TITLE : INF: Page Header Will Span Full Width of Multi-Column  
Report

For information about how to create multiple-page reports with snaking columns, search for "multiple-column reports" then "Creating Snaking Columns on a Report" using the Microsoft Access Help menu.

You can find information, instructions, and examples in the Solutions sample application (SOLUTION.MDB) included with Microsoft Access version 2.0. For more information about printing a conditional page header, open the SOLUTION.MDB database usually located in the ACCESS\SAMPAPPS directory. In the Select A Category Of Examples box, select "Control what you print on reports," then in the Select An Example box, select "Repeat a group name at the top of a column or page."

Additional reference words: 1.00 1.10 2.00 multicolumn snaking-column

KBCategory: kbusage

KBSubcategory: RptSort

## **INF: Requery, Repaint, and Records Refresh**

**Article ID: Q96104**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

The Requery, Repaint, and Records Refresh items are different; however, they are all used to ensure that what you see is updated.

### MORE INFORMATION

=====

#### Requery

-----

The Requery macro action updates controls, such as list boxes and combo boxes. It works for the form on which those controls exist only if you don't include arguments.

#### Repaint

-----

Repaint is not related to data, it merely redraws the object or control. If there is data that has been updated in the underlying dynaset but is not reflected on the form, Repaint cannot be used.

#### Records Refresh

-----

Records Refresh requeries the form itself, as opposed to any control on the form.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsRqry

## **INF: Requirements to Connect to SYBASE SQL Server**

**Article ID: Q96468**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

The SQL Server ODBC driver that ships with Microsoft Access is designed to work with both Microsoft SQL Server and SYBASE SQL Server. This article describes requirements for running Microsoft Access versions 1.0, 1.1, and 2.0 with SYBASE SQL Server.

### MORE INFORMATION

=====

The SQL Server ODBC driver works with both Microsoft and SYBASE SQL Servers. However, the Microsoft applications that shipped with the SQL Server driver (Visual Basic version 2.0 and Microsoft Access) were tested and verified with Microsoft SQL Server only.

The following sections discuss the issues associated with using the Microsoft SQL Server ODBC driver with SYBASE SQL Server.

#### ODBC Driver Requires Catalog Stored Procedures

-----

The ODBC driver requires the catalog stored procedures that ship with SQL Server version 4.2a. Until the 4.9.1 SYBASE release, these catalog procedures did not ship with SYBASE servers. To install the procedures, you must run the INSTCAT.SQL script against the SYBASE server.

NOTE: Sybase changed the system catalogs in their 4.8 release. Therefore, SYBASE SQL Server versions 4.8 and later require a modified script called INSTCAT.48. This revised script is available from Microsoft Product Support and CompuServe. This script is required with Microsoft Access version 1.0. With Microsoft Access 1.1 and 2.0, the INSTCAT.SQL file applies to all versions of Microsoft SQL Server, as well as to all versions of SYBASE SQL Server.

For more information on the INSTCAT.48 script, query on the following words here in the Microsoft Knowledge Base:

access and sybase and instcat.48

#### ODBC Driver Comes with Named Pipe Network Library (Net-Library)

-----

The SQL Server ODBC driver that ships with Microsoft Access and Visual Basic 2.0 comes with the Named Pipe Net-Library. To connect to SYBASE, you need to use the SQL Bridge product or buy the Net-Libraries from Sybase for each client machine. (The only Sybase Net-Library that has



gone through interoperability testing is the Microsoft TCP/IP Net-Library.)

Both Microsoft Access and Visual Basic 2.0 use the network in a way that is atypical of Windows applications; they use multiple simultaneous connections and use asynchronous calls. For this reason, Sybase Net-Libraries may fail with these products in some network configurations. Microsoft and Sybase are discussing how best to perform interoperability testing with the various network configurations supported by Sybase (FTP, Wollongong, Novell and other TCP/IP, DECnet, and so forth).

The following is a list of Net-Libraries that Microsoft has confirmed to work with Microsoft Access/ODBC and SYBASE SQL Server:

NOTE: An EBF is an Emergency Bug Fix, available from Sybase.

| Net-Library                                       | EBF #          |
|---------------------------------------------------|----------------|
| Named Pipe Net-Libraries 1.0                      |                |
| Novell LAN WorkPlace TCP 4.0 with 1.0 Net-Library |                |
| Netware (SPX) 1.0 Net-Library                     |                |
| FTP 2.05 with Net-Libraries 1.0 and 1.0.1         |                |
| Microsoft TCP/IP 1.0 with Net-Library 1.0.1       |                |
| FTP 2.1 with Net-Library 1.0.1                    | 1285           |
| FTP 2.2 with Net-Library 1.02                     | 2300           |
| NetMangae TCP/IP 3.06 Net-Library 1.0             | 1326           |
| PC-NFS Net-Library 1.0                            | 1334           |
| Wollongong Pathway Net-Library 1.0                | 1327*          |
| DECNET (WDBDECDE.DLL)                             | 1561, May 1993 |

\*NOTE: To use Wollongong Pathway, the files C37603.EXE and C39489.EXE are required in addition to EBF #1327. This is required by Wollongong Pathway and is not specific to running ODBC or Microsoft Access. These files are available from Wollongong, either through their BBS or through CompuServe. Contact Wollongong for more information.

The two additional files are for the latest version of Wollongong's Pathway Access, version 2.1.1. (The version number for the run-time software is 1.2.) The run-time software is shipped with Pathway Access, but can also be ordered separately from Wollongong. The run-time software is all that is needed to run Net-Library.

Some of the products mentioned above are manufactured by vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 1.00 1.10 2.00 net-library net library  
KBCategory: kbusage  
KBSubcategory: ObcOthr

## **INF: Returning Case-Sensitive Match in Queries**

**Article ID: Q95605**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Microsoft Access can differentiate between case sensitive text strings using the Asc() function. Using this function in a select query, Microsoft Access can locate an exact case sensitive match.

### MORE INFORMATION

=====

This technique is useful in a situation where the user needs to find all the records in a field that contain lowercase text rather than uppercase text for the first character. An example of this might be in the Employees table of the NWIND.MDB sample database. Please note that the example below will find only the first character. If you need something more complex, you must call an Access Basic function to return the results.

Suppose the text field called First Name contains values such as the following:

```
First Name

andrew
Andrew
Nancy
```

If you need to find all the records that contain the lower case "andrew" as compared to "Andrew," use the Asc() function. To do this, create a new select query:

```
Query: QueryTest

Field name: First Name
 Show: True
Field name: Asc([First name])
 Show: True
 First Criteria Line: Asc("a")
```

This query will return only the record for "andrew," not "Andrew."

### REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, page 40

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

Additional reference words: 1.00 1.10 2.00 chr chr\$  
KBCategory: kbusage  
KSubcategory: QryParm

**INF: Round or Truncate Values to Desired Number of Decimals**  
**Article ID: Q97524**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The Format property of a control can round a number or currency field to the desired number of decimal places. However, this does not change the underlying data, which may contain additional digits that the control does not display. If you add the values in this control, the sum is based on the actual values and not the displayed values, which can cause problem with perceived accuracy.

This article describes four Access Basic functions that round or truncate data to two decimal places so the displayed and formatted value and the actual numeric or currency data are the same.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

The functions are presented in two styles. The first style is appropriate for the AfterUpdate property of a form control, to ensure that the data entered matches the data displayed. The second style is for use in expressions and calculated controls.

To round or truncate numbers to two decimal places, create a new module in Microsoft Access, create an appropriate Declarations section, and add the following functions.

```
'*****
'Declarations section of the module
'*****
```

```
Option Explicit
Const Factor = 100
```

```
'=====
```

```
' RoundAU and TruncaU are designed to be added to the
' AfterUpdate property on a form control.
'=====
```

```
Function RoundAU(X)
 X = Int(X * Factor + .5) / Factor
End Function
```

```
Function TruncaU(X)
```

```

 X = Int(X * Factor) / Factor
End Function

'=====
' RoundCC and TruncCC are designed to be used in
' expressions and calculated controls on forms and reports.
'=====
Function RoundCC(X)
 RoundCC = Int (X * Factor + 0.5) / Factor
End Function

Function TruncCC(X)
 TruncCC = Int (X * Factor) / Factor
End Function

```

#### Examples of Using the Round and Truncate Functions

---

The following examples use the NWIND.MDB sample database provided with Microsoft Access.

- Add the TruncAU() function to the AfterUpdate property of the Unit Price field in the Products form, as follows:

```

Form: Products

ControlName: Unit Price
 AfterUpdate: =TruncAU([Unit Price])

```

If the user accidentally enters \$23.055 instead of \$23.05, the TruncAu() function catches the mistake and changes the value to \$23.05. Instead, if you install RoundAu(), the function changes the value to \$23.06. If you use neither function, the value displays as \$23.06 but the entered value, \$23.055, is used in calculations.

- Add TruncCC() or RoundCC() to expressions in a report or a SetValue macro action. The following example demonstrates using RoundCC() in a report footer:

```

Report: Summary of Sales By Year

ControlName: Total Sales for Quarter
 ControlSource: =Sum(RoundCC([Order Amount]))

ControlName: Total Sales for Year
 ControlSource: =Sum(RoundCC([Order Amount]))

```

If you use RoundCC(), the report sums the values displayed in the report, even though the actual values may contain hidden digits.

NOTE: To change the number of decimal places used, change the value of the global constant, Factor, as follows:

```

 10 = 1 decimal place
 100 = 2 decimal places
 1000 = 3 decimal places, etc.

```

## Limitations

-----

These functions should only be used with Currency data. If used with Double or Single numbers, you can still get minor round-off errors. The reason for this is that Single and Double numbers are floating point. They cannot store an exact binary representation of decimal fractions. Therefore there will always be some error. However, Currency values are scaled integers and can store an exact binary representation of fractions to 4 decimal places.

Additional reference words: 1.00 1.10 2.00 int trunc precision

KBCategory: kbusage

KBSubcategory: FmsHowto

**INF: Row Fix-up in the NWIND.MDB Order Form**  
**Article ID: Q95643**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Microsoft Access provides an automatic "row fix-up" feature that uses links between tables to update (fix up) records in queries and forms. You can use this feature to enter a single value, and Microsoft Access automatically looks up and displays other values for that record.

Row fix-up allows you to update forms and queries without using macros or Access Basic code. This article uses the Orders form in the sample database NWIND.MDB to demonstrate how to set up and use row fix-up.

MORE INFORMATION

=====

To see an example of the row fix-up technique, follow these steps:

1. Open the NWIND.MDB database.
2. Open the Orders form in Form view.
3. Open the Order Information query in Design view.

Note the following about the Orders form:

1. The RecordSource property of the Orders form is set to the Order Information query.
2. In Microsoft Access version 1.x, the Order Information query is based on three tables: Orders, Customers, and Employees. In version 2.0, the query is based on the Orders and Customers tables.
3. The Join property between Customers and Orders tables is:

|                                                  |      |
|--------------------------------------------------|------|
| One                                              | Many |
| -----                                            |      |
| CUSTOMERS.[customer id] ==> ORDERS.[customer id] |      |

4. The Order ID field in the query grid comes from the Orders table, which is on the many side of the relationship, and not from the Customers table, which is on the one side of the relationship. (See point 3, above.)
5. The Bill To combo box is based on the Customer List query. Customer List has the following characteristics:
  - a. The query is based on the Customers table.
  - b. The query returns two columns: Company Name and Company ID.

- c. The bound column is Customer ID.
- d. The visible column is Company Name.

6. Microsoft Access select queries are updatable. This means that when you make a change to a query's dynaset, Microsoft Access can update the source tables with your changes. For more information on this, search for "updating underlying tables," then "When is Query Updatable" using the Microsoft Access Help menu.

The Bill To combo box presents you with a list of company names from the visible column of the Customer List query.

When you make a selection, the combo box stores the bound column--the Customer ID field from the Customer List query--in the Customer ID field in the Orders table. (See points 4 and 5d, above.) It is important to note that the selection was stored in the many-side table. (See point 3a, above.)

After the combo box updates the Customer ID field in the Orders table (the many-side table), the field is changed so it no longer points to the same record in the Customers table (the one-side table). (See point 6, above.) Note that the Orders table is updated because the query's dynaset was changed.

Microsoft Access recognizes that the relationship no longer matches, so it automatically updates the relationship by linking the changed record in the Orders table with the appropriate record in the Customers table.

All the fields in the Orders form that come from the Customers table (the one-side table) are updated to show the values in the record for the newly formed link.

#### Example

-----

When you make a change to the Bill To combo box, Microsoft Access uses row fix-up to update the following list of fields on the Orders form:

- [Address]
- [City]
- [Region]
- [Postal Code]
- [Country]

The following list of fields on the Orders form are not updated by Row fix-up. However they are updated by the Orders Fill In Ship To macro on the AfterUpdate property of the Bill To combo box:

- [Ship Name]
- [Ship Address]
- [Ship City]
- [Ship Region]
- [Ship Postal]
- [Ship Country]

#### REFERENCES



=====

The row fix-up technique is referred to as AutoLookup in Microsoft Access version 2.0 and as dynamic lookup in previous versions. For more information on row fix-up, search for "dynamic lookup" using the Help menu in Microsoft Access version 1.x or 2.0.

For more information, query on the following words here in the Microsoft Knowledge Base:

row and fix-up

Additional reference words: 1.00 1.10 2.00 dynamic lookup

KBCategory: kbusage

KBSubcategory: QryJoin

**INF: Running a MS Access Macro from Visual Basic Using DDE**  
**Article ID: Q97776**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The following information provides some examples of how to use Microsoft Visual Basic to run macros in Microsoft Access using dynamic data exchange (DDE).

MORE INFORMATION

=====

The following code demonstrates how to run a macro called "MyMacro" in Microsoft Access from a Visual Basic application.

```
Sub Command1_Click()
 'the character between MSACCESS and System is the pipe symbol
 Text1.LinkTopic = "MSACCESS|System"

 ' Establish a manual DDE link to Access
 Text1.LinkMode = 2

 ' Run the macro called "MyMacro"
 Text1.LinkExecute "MyMacro"

 ' Terminate the DDE link to Access
 Text1.LinkMode = 0
End Sub
```

To use this code, create a new Visual Basic program with a text box and a command button with the following properties:

```
Text box

Name: Text1

Command Button

Name: Command1
```

Double-click the command button and enter the code example listed above between the "Sub Command1\_Click ()" and "End Sub" lines.

When you run this example, be sure that Microsoft Access is running and that a database containing a macro called "MyMacro" is open. For testing purposes, you may want to create a macro called "MyMacro" that contains a single MsgBox action.

NOTE: While the above example uses the Microsoft Access system Topic listed in the LinkTopic line, Microsoft Access is capable of executing

a macro with virtually every valid topic it supports. For a list of topics that Microsoft Access supports, query on the following words here in the Microsoft Knowledge Base:

b\_waccess and dde and server and topics

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpDde

**INF: Running MS-DOS SHARE with Windows for Workgroups**  
**Article ID: Q88914**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

If you are running a regular version of Microsoft Windows 3.x (not Windows for Workgroups or Windows NT) during the installation of Microsoft Access version 1.0 or 1.1, Setup automatically inserts the following MS-DOS command in the AUTOEXEC.BAT file:

```
<MS-DOS directory>\SHARE.EXE /L:500
```

If you are running Windows for Workgroups, we recommend that you not load this SHARE command.

MORE INFORMATION

=====

Microsoft Access versions 1.0 and 1.1

-----

Setup automatically detects Windows for Workgroups and does not install SHARE if Windows for Workgroups is found.

Windows for Workgroups has its own sharing mechanism (VSHARE.386), which is loaded with Windows by a SYSTEM.INI file entry. VSHARE.386 is intended as a replacement for SHARE.EXE.

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

Because VSHARE.386 is a 386 enhanced mode virtual device driver (VxD), it contains features that improve file sharing when running in 386 enhanced mode. Because it was designed to work with Windows for Workgroups, VSHARE.386 has greater usability features than SHARE.EXE, especially regarding access control.

If SHARE.EXE is loaded and you start Windows for Workgroups in 386 enhanced mode, VSHARE.386 takes over file sharing control until you exit Windows. If you usually run Windows for Workgroups in 386 enhanced mode, you may not need to run SHARE.EXE at all, saving you 5K or more of conventional memory. The number of locks available is especially important if you are using your Windows for Workgroups machine as a network server.

Microsoft Access version 2.0

-----

Microsoft Access version 2.0 will install VSHARE.386 instead of modifying

the AUTOEXEC.BAT file to load SHARE.EXE. If you are already loading SHARE.EXE in the AUTOEXEC.BAT file, Microsoft Access will leave the line in the AUTOEXEC.BAT file alone, but will still install VSHARE.386. When you enter 386 enhanced mode, VSHARE.386 will take over from SHARE.EXE.

Additional reference words: 1.00 1.10 2.00 WFW

KBCategory: kbsetup

KBSubcategory: StpNet

**INF: Sample Code for Running Temporary SQL Pass-Through Query**  
**Article ID: Q124391**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

This article lists sample code that you can use to call and run a temporary SQL pass-through query.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

You can use the following sample Access Basic code to call and run a temporary SQL pass-through query:

```
Function ExecuteSPT (sqltext As String, connectstring As String)

' Purpose: Run a temporary pass-through query.
' Accepts: sqltext: SQL string to run.
' connectstring: Connection string, which must be at least
' "ODBC;".
' Returns: nothing.

Dim mydb As Database, myq As QueryDef
Set mydb = DBEngine.Workspaces(0).Databases(0)

' Create a temporary QueryDef object that is not saved.
Set myq = mydb.CreateQueryDef("")

' Set the ReturnsRecords property to False in order to use the
' Execute method.
myq.ReturnsRecords = False

myq.Connect = connectstring
myq.SQL = sqltext

myq.Execute
myq.Close

End Function
```

To run a stored procedure called sp\_example on your SQL Server, you could type the following line in the Immediate window:

```
?ExecuteSPT("sp_example","ODBC;")
```

Note that this code accepts SQL statements that are specific to the server you are using. Please refer to your server's documentation to determine valid SQL statements for your server.

#### REFERENCES

=====

For more information about pass-through queries, search for "pass-through query" then "Creating a Pass-Through Query" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr

**INF: Sample DDE Macro Communicates with Microsoft Excel 4.0**  
**Article ID: Q97522**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Excel for Windows, version 4.0
- 

SUMMARY

=====

The text below demonstrates communication between two applications through dynamic data exchange (DDE). Specifically, one example shows a macro developed for Microsoft Excel version 4.0 for Windows that starts Microsoft Access, loads a database and runs a macro. A second example shows an Access Basic function that loads Microsoft Excel and runs a macro.

MORE INFORMATION

=====

Running a Microsoft Access Macro from Microsoft Excel

-----

Perform the following three steps to demonstrate this behavior. These steps assume that a "MESSAGE" macro already exists in Microsoft Access.

1. In Microsoft Excel, choose New from the File menu, select Macro Sheet and choose OK.
2. Enter the following macro into the macro sheet. Substitute the appropriate location for files on your computer.

```
=EXEC("c:\access\msaccess.exe c:\access\db4.mdb")
Chan=INITIATE("MSACCESS","system")
=APP.ACTIVATE("Microsoft Access")
=EXECUTE(Chan,"MESSAGE")
=TERMINATE(Chan)
=RETURN()
```

3. To run the macro, select the first cell, select Run from the Macro menu and choose OK.

The EXEC function in the macro loads Microsoft Access minimized and loads the DB4.MDB database. The macro initiates a DDE channel and assigns it to the variable Chan. The APP.ACTIVATE switches to the Microsoft Access window to show the macro actions running.

The EXECUTE function runs a macro named "MESSAGE," as follows.

Name Summary

-----  
Application Name: "MSACCESS"  
Window Title: "Microsoft Access"



Topic: "System"

Item: "Message"

## Running a Microsoft Excel Macro from Microsoft Access

---

Perform the following two steps to demonstrate this behavior. These steps assume that a macro named "Message" already exists in a Microsoft Excel macro sheet named "MACRO1.XLM."

1. In Microsoft Access, create a new function that contains the following code. Substitute the appropriate location for files on your computer.

```
Function CallExcel ()
 Dim Chan
 x = Shell("c:\excel\excel.exe c:\excel\macro1.xlm", 1)
 Chan = DDEInitiate("Excel", "System")
 DDEExecute Chan, "[Run(""macro1.xlm!Message"")] "
 DDETerminate Chan
End Function
```

2. Create a new blank form and place a command button on it.
3. Select the button.
4. From the View menu, choose Properties.
5. Set the button's OnPush property to the following:

```
= CallExcel()
```

Note that the OnPush property is renamed OnClick in Microsoft Access version 2.0.

6. View the form in Form view.
7. Choose the command button to start the function.

The Shell function loads Microsoft Excel and the "MACRO1.XLM" macro sheet full screen and leaves the focus on Microsoft Excel. The macro initiates a DDE channel and assigns it to the variable Chan.

The EXECUTE function runs a macro named "Message."

### Name Summary

---

```
Application Name: "Excel"
Window Title: "Microsoft Excel"
Topic: "System"
Item: "[Run(""macro1.xlm!Message"")] "
```

The syntax for these two macros demonstrates that the correct syntax for the "Item" depends on the target application.

### REFERENCES

=====

"Microsoft Excel Function Reference", version 4.0, pages 14 - 15

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpDde

**INF: Sample Expressions to Extract Portion of Text String**  
**Article ID: Q115915**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article lists sample expressions that you can use in Microsoft Access to extract a portion of a text string. These expressions are commonly used in the Update To line of an update query to place a portion of a larger field into a new field. These expressions can be adapted for use with other common formats. For example, the expression used to extract "Doe" from "Doe, John" can be used to extract "Seattle" from "Seattle, WA."

MORE INFORMATION

=====

The following expressions are sample expressions that you can use in Microsoft Access to extract a portion of a text string.

NOTE: In the following sample expressions, an underscore ( ) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating the expression.

| Original Entry                                                                                                                | Returned by Expression |
|-------------------------------------------------------------------------------------------------------------------------------|------------------------|
| [Names] = "John Doe"                                                                                                          | John                   |
| Expr: Left ([Names], InStr (1, [Names], " ") - 1)                                                                             |                        |
| [Names] = "John Doe"                                                                                                          | Doe                    |
| Expr: Right (Trim ([Names]), Len (Trim ([Names])) - InStr (1, [Names], " "))                                                  |                        |
| [Names] = "Doe, John"                                                                                                         | Doe                    |
| Expr: Left ([Names], InStr (1, [Names], ",", " ") - 1)                                                                        |                        |
| [Names] = "Doe, John"                                                                                                         | John                   |
| Expr: Right (Trim ([Names]), Len (Trim ([Names])) - InStr (1, [Names], " "))                                                  |                        |
| [Names] = "John P. Doe"                                                                                                       | Doe                    |
| Expr: Right (Trim ([Names]), Len (Trim ([Names])) - InStr (InStr (1, [Names], " ") + 1, [Names], " "))                        |                        |
| [Names] = "John P. Doe"                                                                                                       | John                   |
| Expr: Left ([Names], InStr (1, [Names], " ") - 1)                                                                             |                        |
| [Names] = "John P. Doe"                                                                                                       | P.                     |
| Expr: Mid ([Names], InStr (1, [Names], " ") + 1, InStr (InStr (1, [Names], " ") + 1, [Names], " ") - InStr (1, [Names], " ")) |                        |
| [Names] = "Doe, John P."                                                                                                      | P.                     |
| Expr: Right (Trim ([Names]), Len (Trim ([Names])) - InStr (InStr (1, [Names], " ") + 1, [Names], " "))                        |                        |

[ZipCode] = "98052-6399"                    98052  
Expr: Left([ZipCode],5)

[Phone] = (206) 635-7050                    206  
Expr: Mid([Phone],2,3)

[Phone] = (206) 635-7050                    635-7050  
Expr: Right(Trim([Phone]),8)

Additional reference words: 1.00 1.10 2.00 parse name first last parsing  
KBCategory: kbusage  
KBSubcategory: ExrOthr

## INF: Sample Function CopyFile() to Copy Disk Files

Article ID: Q102671

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Access Basic does not have a command, such as the MS-DOS COPY command, to copy a disk file.

This article includes two variations of an Access Basic function called CopyFile(), both of which allow you to copy disk files.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

### MORE INFORMATION

=====

Two variations of the Access Basic CopyFile() function are provided below. The first version uses only Microsoft Access Basic code; the second leverages some function calls to the Windows application programming interface (API) to optimize performance, particularly with larger files.

NOTE: You may have some Windows API functions defined in an existing Microsoft Access library; therefore, your declarations may be duplicates. If you receive the duplicate procedure name error message when you compile or run your code, remove or comment out the appropriate declarations statement from your code.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

#### Method 1: CopyFile() Function Using Standard Access Basic Code

-----

```
'*****
' DECLARATION SECTION
'*****
Option Explicit

'*****
' FUNCTION: CopyFile()
' PURPOSE:
' Facilitates copying a disk file.
' ARGUMENTS:
```

```

' Source - The path\filename of the file to copy from.
' Destination - The path\filename of the file to copy to.
' RETURN:
' The length of the file copied.
'*****
Function CopyFile (ByVal Source$, ByVal Destination$) As Long
 Dim Index1 As Integer, NumBlocks As Integer
 Dim FileLength As Long, LeftOver As Long, AmountCopied As Long
 Dim SourceFile As Integer, DestFile As Integer
 Dim FileData As String
 Dim RetVal As Variant
 Const BlockSize = 32768

 On Error GoTo Err_CopyFile

 ' Remove the destination file.
 DestFile = FreeFile
 Open Destination For Output As DestFile
 Close DestFile

 ' Open the source file to read from.
 SourceFile = FreeFile
 Open Source For Binary Access Read As FreeFile

 ' Open the destination file to write to.
 DestFile = FreeFile
 Open Destination For Binary As DestFile

 ' Get the length of the source file.
 FileLength = LOF(SourceFile)

 ' Calculate the number of blocks in the file and left over.
 NumBlocks = FileLength \ BlockSize
 LeftOver = FileLength Mod BlockSize

 ' Create a buffer for the leftover amount.
 FileData = String$(LeftOver, 32)

 ' Read and write the leftover amount.
 Get SourceFile, , FileData
 Put DestFile, , FileData

 ' Create a buffer for a block to be read.
 FileData = String$(BlockSize, 32)

 ' Read and write the remaining blocks of data.
 For Index1 = 1 To NumBlocks
 ' Read and write one block of data.
 Get SourceFile, , FileData
 Put DestFile, , FileData
 Next Index1

 Close SourceFile, DestFile
 CopyFile = AmountCopied

Bye_CopyFile:
 Exit Function

```

```
Err_CopyFile:
 CopyFile = -1 * Err
 Resume Bye_CopyFile
```

```
End Function
```

```
Method 2: CopyFile() Function Using Standard Access Basic Code
```

```

'*****
' DECLARATION SECTION
'*****
Declare Function fWrite Lib "kernel" Alias "_lwrite" (ByVal hFile%,
 ByVal lpBuff&, ByVal nBuff%) As Long
Declare Function fRead Lib "kernel" Alias "_lread" (ByVal hFile%,
 ByVal lpBuff&, ByVal nBuff%) As Long
Declare Function GlobalAlloc Lib "Kernel" (ByVal wFlags%,
 ByVal dwBytes&) As Integer
Declare Function GGlobalFree Lib "kernel" (ByVal hMem%) As Long
Declare Function GlobalLock Lib "Kernel" (ByVal hMem%) As Long
Declare Function GGlobalUnlock Lib "kernel" (ByVal hMem%) As Long

'*****
' FUNCTION: CopyFile()
' PURPOSE:
' Facilitates copying a disk file.
' ARGUMENTS:
' Source - The path\filename of the file to copy from.
' Destination - The path\filename of the file to copy to.
' RETURN:
' The length of the file copied, if successful.
'*****
Function CopyFile (ByVal Source$, ByVal Destination$) As Long
 Dim FileLength As Long, AmountCopied As Long
 Dim RetVal As Variant, lpBuff As Long
 Dim DestFile As Integer, SourceFile As Integer
 Dim DestDOS As Integer, SourceDOS As Integer
 Dim ApiErr As Integer, AmtRead As Integer
 Dim hMem As Integer
 Const nBuff = 32767
 Const wFlags = &H20

 On Error GoTo Err_CopyFile

 ' Get the size of the file.
 SourceFile = FreeFile
 Open Source For Binary Access Read As SourceFile
 FileLength = LOF(SourceFile)
 Close SourceFile

 ' Allocate and lock memory to buffer file contents.
 hMem = GlobalAlloc(wFlags, nBuff)
 lpBuff = GlobalLock(hMem)

 ' Open the source file to read from.
 SourceFile = FreeFile
```

```

Open Source For Input As SourceFile Len = 1

' Open the destination file to write to.
DestFile = FreeFile
Open Destination For Output As DestFile Len = 1

' Get the operating system file handles.
DestDOS = FileAttr(DestFile, 2)
SourceDOS = FileAttr(SourceFile, 2)

Do
 ' Read and write nBuff worth of data.
 AmtRead = fRead(SourceDOS, ByVal lpBuff, nBuff)
 ApiErr = fWrite(DestDOS, ByVal lpBuff, AmtRead)
 AmountCopied = AmountCopied + AmtRead
Loop Until AmtRead = 0

Close SourceFile, DestFile

' Unlock and free the file buffer memory.
lpBuff = GlobalUnlock(hMem)
hMem = GlobalFree(hMem)

CopyFile = AmountCopied

```

```

Bye_CopyFile:
 Exit Function

```

```

Err_CopyFile:
 CopyFile = -1 * Err
 Resume Bye_CopyFile
End Function

```

The CopyFile() function can be tested from the Immediate window, as follows:

1. Create or open a module in Microsoft Access.
2. From the View menu, choose Immediate Window.
3. Type the following and press ENTER:

```
? CopyFile(C:\WINDOWS\CHESS.BMP", "C:\CHESS.BMP")
```

The CopyFile() function can be invoked from an Access Basic function

```
Dim RetVal As Long
RetVal = CopyFile("C:\WINDOWS\CHESS.BMP", "C:\CHESS.BMP")
```

or by using the RunCode macro action:

```
RunCode
Function Name: CopyFile("C:\WINDOWS\CHESS.BMP", "C:\CHESS.BMP")
```

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbprg



KBSubcategory: PgmOthr

## INF: Sample Function for Summing Date/Time Field

Article ID: Q123282

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Items with a Date/Time data type cannot be summed directly because items with this data type are treated as time in a 24-hour day. You can, however, perform time calculations if the hours, minutes, and seconds are stored separately in Text fields or variables. This article demonstrates how to perform time calculations in this manner.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The following sample user-defined function demonstrates how to split a Date/Time field into separate variables containing hours, minutes, and seconds so that you can manipulate the fields arithmetically. This sample function assumes a table called TimeCard containing a Date/Time field called Timespent.

NOTE: In the following sample code, an underscore ( \_ ) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

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

'*****
' The Addtimes() Function
'*****
Function Addtimes ()

Dim Totalhrs, Totalmin, Totalsec, Minfrmsec, Hrsfrmmin As Integer
Dim Db As Database, Tb As Recordset

Set Db = DBEngine(0) (0)
Set Tb = Db.OpenRecordset("TimeCard")

Totalhrs = 0
Totalmin = 0
Totalsec = 0
```

```

While Not Tb.EOF
 Totalhrs = Totalhrs + Hour(CVDate(tb!Timespent))
 Totalmin = Totalmin + Minute(CVDate(tb!Timespent))
 Totalsec = Totalsec + Second(CVDate(tb!Timespent))
 Tb.MoveNext
Wend

Minfrmsec = Totalsec \ 60
Totalsec = Totalsec Mod 60
Totalmin = Totalmin + minfrmsec

Hrsfrmmin = Totalmin \ 60
Totalmin = Totalmin Mod 60
Totalhrs = Totalhrs + Hrsfrmmin

Debug.Print "Total Time: " & Format(Totalhrs, "00") & ":" & _
Format(Totalmin, "00") & ":" & Format(Totalsec, "00")

End Function

```

To test this function, follow these steps:

1. Create a table called TimeCard with a Date/Time field called Timespent and then save the table.
2. Enter the following data in the Timespent field:

```

9:00
8:15
8:23
4:25
8:12

```

3. In any module's Immediate window, type the following line and then press ENTER:

```
? Addtimes()
```

The total time (38:15) is displayed.

#### REFERENCES

=====

For information about calculating elapsed time with Date/Time fields, please see the following article in the Microsoft Knowledge Base:

```

ARTICLE-ID: Q88657
TITLE : Using Date/Time Data Type and Calculating Elapsed Time

```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: TblDatyp

## INF: Sample Function to Capitalize Words and Phrases

Article ID: Q110391

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes a sample user-defined function called Proper() that you can use to capitalize the first character of a word or set of words.

This article contains two versions of the Proper() function:

- One version is used in the AfterUpdate property of a control on a form, primarily for data entry.
- The other version is used in calculated query fields, calculated controls on forms and reports, macro SetValue expressions, and Access Basic expressions.

NOTE: Microsoft Access version 2.0 includes a sample Proper() function in the sample database SOLUTION.MDB. The Proper() function is located in the Proper module.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual.

### MORE INFORMATION

=====

To create the Proper() function, enter the following code in a module:

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

'The following function is designed for use in the AfterUpdate  
' property of form controls.  
'Features:  
' - Leading spaces do not affect the function's performance.  
' - "O'Brian" and "Wilson-Smythe" will be properly capitalized.  
'Limitations:  
' - It will change "MacDonald" to "Macdonald."  
' - It will change "van Buren" to "Van Buren."  
' - It will change "John Jones III" to "John Jones Iii."  
'=====

```
Function Proper (AnyValue As Variant) As Variant
Dim ptr As Integer
Dim TheString As String
Dim currChar As String, prevChar As String
```

```

If IsNull(AnyValue) Then
 Exit Function
End If

TheString = CStr(AnyValue)
For ptr = 1 To Len(TheString) 'Go through each char. in string.
 currChar = Mid$(TheString, ptr, 1) 'Get the current character.

 Select Case prevChar 'If previous char. is a letter,
 'this char. should be lowercase.
 Case "A" To "Z", "a" To "z"
 Mid(TheString, ptr, 1) = LCase(currChar)

 Case Else
 Mid(TheString, ptr, 1) = UCase(currChar)

 End Select
 prevChar = currChar
Next ptr
AnyValue = CVar(theString)
End Function

```

How to Use This Function in the AfterUpdate Property of a Control

---

1. Create a form based on a table or query.
2. Type the following in the AfterUpdate property of a text box control:

```
=Proper(<control name>)
```

For example, if the ControlName is [Full Name], type the following:

```
=Proper([Full Name])
```

How to Modify This Function for Expressions and Calculated Controls

---

1. Enter the code shown above to create the Proper() function.
2. Modify the following section of code

```

AnyValue=CVar(theString)
End Function

```

To read as follows:

```

Proper=CVar(theString)
End Function

```

Examples

---

The examples below all require the second version of the Proper() function.

Using Proper() in a Query:

```
Field: Full Name:Proper([Last Name] & " " & [First Name])
```

This will concatenate the first and last names and capitalize the first letter of each.

Using Proper() in a Calculated Control on a Form or Report:

```
TextBox
```

```

ControlName: AddressP
ControlSource: =Proper([Address])
```

NOTE: In calculated fields, the ControlName must be unique.

Using Proper() in a Macro:

```
Action: SetValue
Item: Screen.ActiveControl
```

```
Expression: Proper(Screen.ActiveControl)
```

NOTE: You can call this macro from the AfterUpdate property of a control on a form. This has the same affect as the first version of the Proper() function.

#### REFERENCES

=====

For more information about methods of calling a function, please see the following article(s) in the Microsoft Knowledge Base:

```
ARTICLE-ID: Q97514
TITLE : INF: Writing Functions Called from Events or Expressions
```

For more information about manipulating text strings, please see the following article(s) in the Microsoft Knowledge Base:

```
ARTICLE-ID: Q115915
TITLE : INF: Sample Expressions to Extract Portion of Text String
```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

## **INF: Sample Function to Create a Running Sum on a Form**

**Article ID: Q121509**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article describes how to use a sample user-defined Access Basic function to compute a running sum on a form.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

One way to compute a running sum on a form is to use the DSum() function. For information about how to use the DSum() function to compute a running sum on a form, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q103183

TITLE : INF: How to Create a Running Sum on a Form

Using the method demonstrated in this article has the following advantages over using the DSum() function:

- The method demonstrated in this article does not require a sequential ID field. A required sequential ID field limits you to computing a running sum using a single ordering of your records.
- Using the method demonstrated in this article is significantly faster than using the DSum() function to compute a running sum.

### How to Create and Use the RunningSum() Function

-----

1. Open the sample database NWIND.MDB and create a new module with the following line in the Declarations section:

Option Explicit

2. Enter the following code in the module.

NOTE: In the following sample code, an underscore ( \_ ) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Function RunningSum (F As Form, KeyName As String, _
 KeyValue, FieldToSum As String)
```

```

'*****
' FUNCTION: RunningSum()
' PURPOSE: Compute a running sum on a form.
' PARAMETERS:
' F - The form containing the previous value to
' retrieve.
' KeyName - The name of the form's unique key field.
' KeyValue - The current record's key value.
' FieldToSum - The name of the field in the previous
' record containing the value to retrieve.
' RETURNS: A running sum of the field FieldToSum.
' EXAMPLE: =RunningSum(Form,"ID",[ID],"Amount")
'*****

Dim RS As Recordset
Dim Result

On Error GoTo Err_RunningSum

' Get the form Recordset.
Set RS = F.RecordsetClone

' Find the current record.
Select Case RS.Fields(KeyName).Type
' Find using numeric data type key value?
Case DB_INTEGER, DB_LONG, DB_CURRENCY, _
 DB_SINGLE, DB_DOUBLE, DB_BYTE
 RS.FindFirst "[" & KeyName & "] = " & KeyValue
' Find using date data type key value?
Case DB_DATE
 RS.FindFirst "[" & KeyName & "] = #" & KeyValue & "#"
' Find using text data type key value?
Case DB_TEXT
 RS.FindFirst "[" & KeyName & "] = '" & KeyValue & "'"
Case Else
 MsgBox "ERROR: Invalid key field data type!"
 GoTo Bye_RunningSum
End Select

' Compute the running sum.
Do Until RS.EOF
 Result = Result + RS(FieldToSum)

 ' Move to the previous record.
 RS.MovePrevious
Loop

Bye_RunningSum:
RunningSum = Result
Exit Function

Err_RunningSum:
Resume Bye_RunningSum

End Function

```

3. Create the following new query based on the Orders table and the Order Subtotals query. Save the query as Orders Query.



Query: Orders Query

---

Type: select query

Join: Orders.[Order ID] <-> [Order Subtotals].[Order ID]

Field: Order ID

Table: Orders

Sort: Ascending

Field: Subtotal

Table: Order Subtotals

4. Create a new form based on the Orders Query query using the Tabular Form Wizard. Include both fields on the form.

5. Add a text box with the following properties to the new form:

ControlSource: =RunningSum([Form],"Order ID",[Order ID],"Subtotal")

Format: Currency

6. View the form in Form view.

Additional reference words: 2.00 total

KBCategory: kbusage

KBSubcategory: FmsHowto

**INF: Sample Function to Determine Current Page of a Form**  
**Article ID: Q121669**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes a sample, user-defined Access Basic function that you can use on a multiple-page form that contains page breaks to determine which page of the form is current.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

The following steps demonstrate how to create and use the sample function WhichPage():

1. Open the sample database NWIND.MDB.
2. Open the Employees form in Design view.
3. Set the Employee ID control's OnDblClick property to:

```
=WhichPage(Form,[Employee ID])
```

4. Set the Address control's OnDblClick property to:

```
=WhichPage(Form,[Address])
```

Save and then close the form.

5. Create a new module and enter the following lines in the module's Declarations section:

```
Option Compare Database
Option Explicit
```

6. Enter the following sample code in the module:

```
Function WhichPage (F As Form, MyControl As Control)
Dim Page, I As Integer
Dim C As Control
Page = 1
```

```
For I = 1 To F.Count - 1
Set C = F(I)
```

```
 If TypeOf C Is PageBreak Then
 If C.Top < MyControl.Top Then Page = Page + 1
 End If
 Next I
```

```
 MsgBox "Page= " & Page
End Function
```

7. Save the module and then close it.
8. Open the Employees form in Form view. Double-click the Employee ID field, and then the Address field. Note that the function displays the number of the page containing the control.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 4, "Using Forms to Collect, Display, and Filter Information," pages 79-109

Microsoft Access "Introduction to Programming," version 1.1, Chapter 7, "Coding with Forms and Reports," pages 79-96

Additional reference words: 1.00 1.10 2.00 ab

KBCategory: kbusage

KBSubcategory: FrmsOthr

**INF: Sample Function to Determine the Access Startup Directory**  
**Article ID: Q88174**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Microsoft Access does not have a built-in command to return the Microsoft Access startup directory. However, you can accomplish this by making Windows application programming interface (API) calls in Access Basic.

This article describes the steps involved in creating a function that returns the Microsoft Access startup directory.

MORE INFORMATION

=====

The Access Basic CurDir\$() function returns the current directory. Because Microsoft Access can be started from a directory other than the current directory or the current directory can be changed with the ChDir statement, the CurDir\$() function cannot be used to determine the startup directory.

The following sample 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.

Creating the Module

-----

The following steps demonstrate how to create the sample Access Basic function that returns the startup directory:

1. From the File menu, choose New, and select Module.
2. In the Declarations section of the module, enter the following Declare statements:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Option Explicit
Declare Function GetModuleHandle% Lib "kernel" (ByVal FileName$)
Declare Function GetModuleFileName% Lib "kernel" (ByVal hModule%, _
 ByVal FileName$, ByVal nSize%)
```

3. Enter the following to create the function:

```
Function StartUp_Dir ()
Dim hModule%, Buffer$, Length%, Msg$
 hModule% = GetModuleHandle("MSACCESS.EXE")
```

```

 Buffer$ = Space$(255)
 Length% = GetModuleFileName(hModule%, Buffer$, Len(Buffer$))
 Buffer$ = Left$(Buffer$, Length%)
 Msg$ = "Startup path and filename: " & Buffer$
 MsgBox Msg$
End Function

```

4. From the Run menu, choose Compile Loaded Modules in Microsoft Access version 1.x, or choose Compile All in version 2.0.
5. Save the module as Startup Directory.
6. From the View menu, choose Immediate window. Type the following in the Immediate window, and then press ENTER:

```
? StartUp_Dir()
```

The Microsoft Access startup directory should be displayed in the Immediate window.

#### Uses and Variations

The Startup directory is displayed when the following command is issued in the Immediate window:

```
? StartUp_Dir()
```

This function can also be incorporated into other program modules, and can be used in expressions. For example, entering =Startup\_Dir() as the OnPush or OnClick property of a button on a form returns the startup directory of Microsoft Access whenever the button is chosen.

NOTE: You can change the MSACCESS.EXE argument for the Windows API GetModuleHandle() function so that the function returns the startup directory of another program started in the Windows environment. Furthermore, you can pass a program name as a variable to the Windows API function, giving even more flexibility to the function.

#### REFERENCES

=====

For more information on Windows API calls, query on the following words here in the Microsoft Knowledge Base:

handle and api and dll and kernel and user and gdi

"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

Additional reference words: 1.00 1.10 2.00 On Push  
KBCategory: kbprg

KBSubcategory: PgmApi

**INF: Sample Function to Determine the Existence of a Query**  
**Article ID: Q88170**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

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

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic," in version 2.0.

MORE INFORMATION

=====

Microsoft Access does not have a function to determine the existence of a specific query. Using Access Basic code, you can determine if a query exists in the database.

You can use the following sample function to return the existence of a query:

1. Open the sample database NWIND.MDB and create a new module with the following function:

```
'*****
'Declarations section of the module
'*****
```

```
Option Compare Database
Option Explicit
```

```
'=====
```

' The following function is named IsQuery(). It accepts a string  
' argument and returns either True(-1) or False(0).  
'=====

```
Function IsQuery (QueryName As String) As Integer
 Dim DB As Database
 Dim Snap As Snapshot
 Set DB = CurrentDB()
 Set Snap = DB.ListTables()
 Snap.MoveFirst
 IsQuery = False
 Do Until Snap.eof
 If Snap![Name] = QueryName And Snap![TableType] = DB_QUERYDEF Then
 IsQuery = True
```

```
 End If
 Snap.MoveNext
 Loop
 Snap.Close
End Function
```

2. From the View menu, choose Immediate Window. Enter the following command, and then press ENTER:

```
?IsQuery("Order Information")
```

3. "-1" should be displayed in the Immediate window (-1 represents True in Microsoft Access).
4. Experiment by calling the function with a nonexistent query. This should display "0" (which represents False).

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr



## INF: Sample Function to Find File Associations

Article ID: Q109931

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article demonstrates a sample Access Basic function that you can use to obtain the executable filename associated with a document filename by calling the Microsoft Windows API function FindExecutable().

### MORE INFORMATION

=====

The following sample function requires a path string parameter and a filename string parameter. The function returns the path of the associated file.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
' *** Declarations Section ***
Option Explicit

Declare Function FindExecutable% Lib "Shell" (ByVal lpszFile$, _
 ByVal lpszDir$, ByVal lpszResult$)

Function GetAssociation (Path As String, FileName As String)
 Dim Result As String, X As Integer
 Result = Space$(256)
 X = FindExecutable(FileName, Path, Result)
 GetAssociation = Result
End Function
```

To find the path and name of the executable file associated with a file called MYDB.MDB that is located in the root directory on the D drive you could type the following in an Immediate window:

```
? GetAssociation ("D:\", "MYDB.MDB")
```

The function would return

```
D:\ACCESS.100\MSACCESS.EXE
```

if the MYDB.MDB file was associated with the MSACCESS.EXE executable file located in the ACCESS.100 directory on the D drive.

### REFERENCES

=====

Microsoft Windows Software Development Kit "Programmer's Reference,

Volume 2: Functions," Version 3.1, pages 303-305

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: Sample Function to Generate a Random Temporary File Name**  
**Article ID: Q88929**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article demonstrates how to use a sample function to generate a random file name that can be used for a temporary file.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic," in version 2.0.

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 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 temporary file is placed in the same directory Windows uses to write its temporary files. This is handy for those who perform periodic maintenance on their hard drive and delete any leftover temporary files in the Windows TEMP directory.

You can create the sample function MakeTempFileName() by entering the following in a module:

```
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
 Loop
```

```
Close #FHandle%
MakeTempFileName = TF$
End Function
```

To test the function, type the following in the Immediate window. The MakeTempFileName() function will return a name that is used as a text file. The text file will fill with some text, and then be deleted:

```
TempFileName$ = MakeTempFileName("TMP")
Open TempFileName$ For Output As #1
Print #1, "This is a line of text"
Close #1
Kill TempFileName$
```

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbprg  
KBSubcategory: PgmOthr

**INF: Sample Function to Make ENTER Add Lines in a Text Box**  
**Article ID: Q89590**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

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 next field or control is selected. This article describes a sample function that you can use to modify the action that results when you press ENTER so that a line is added to the current field instead of selecting the next field.

MORE INFORMATION

=====

The following sample function changes the behavior of the ENTER key in text boxes to allow adding lines instead of selecting a different field or control. This code should be attached to the BeforeUpdate event of the text box:

In the Declarations section of a module, enter the following:

```
Option Explicit
Declare Function GetKeyState% Lib "user.exe" (ByVal nKey%)
Const VK_RETURN = &HD
```

The following is the sample function MakeEnterWork():

```
Function MakeEnterWork()
 If GetKeyState(VK_RETURN) < 0 Then
 DoCmd CancelEvent
 SendKeys "^{ENTER}"
 End If
End Function
```

NOTE: Some change must be made to the text box before this function will work because Microsoft Access evaluates controls only after they have been modified.

Additional reference words: 1.00 1.10 2.00 carriage return memo

KBCategory: kbusage

KBSubcategory: GnlGlobl

## **INF: Sample Function to Obtain the Last Word in a String**

**Article ID: Q117914**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article demonstrates a sample user-defined Access Basic function that you can use to return the last word in any text string.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

You can use the LastWord() function to return the last word in a string. For example, the LastWord() function will return "Doe" from the string "Mr. and Mrs. John Doe."

You can also use the LastWord() function in your application to sort a list of names by last name, if the first and last names are in the same field, or if the middle names are included with the first names in a field.

### The LastWord() Function

-----

```
Function LastWord (anystring)
 If Not InStr(anystring, Chr(32)) = 0 Then
 Dim FindSpace As String, Pos As Integer, StrLen As Integer
 FindSpace = "Z"
 StrLen = Len(anystring)
 Pos = StrLen
 Do Until FindSpace = Chr(32)
 Pos = Pos - 1
 FindSpace = Mid(anystring, Pos, 1)
 Loop
 LastWord = Mid(anystring, Pos + 1, StrLen - Pos + 1)
 End If
End Function
```

### REFERENCES

=====

For additional information about parsing strings, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q99405

TITLE : INF: Part 2 DDE in Visual Basic to Request Data from MS Access

Additional reference words: 2.00 programming parsing  
KBCategory: kbprg  
KSubcategory: PgmParse

**INF: Sample Function to Open a Form to a Particular Record**  
**Article ID: Q109818**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes a sample Access Basic function you can use to open a form to a particular record using the InputBox() function.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic," in version 2.0.

MORE INFORMATION

=====

To create the sample function OpenFormWithInput(), enter the following Declarations and function in a module:

```
'=====
'Declarations section
'=====

Option Compare Database 'Use database order for string comparisons
Option Explicit

'=====
'Code section
'=====

Function OpenFormWithInput ()
Dim Msg, Title, Defvalue, Answer
 Msg = "Enter a Customer ID (ALWAO, BABUJ, EASTC, FOODI, etc.)."
 Title = "OPEN CUSTOMERS FORM"
 Defvalue = "ALWAO"
 Answer = InputBox(Msg, Title, Defvalue)
 If Answer <> "" Then
 DoCmd OpenForm "Customers", , , "[Customer ID]='" & Answer & "' "
 Else
 DoCmd OpenForm "Customers"
 End If
End Function
```

The OpenFormWithInput() function will open the form with the record whose Customer ID matches the value entered in the input box as the only record in its dynaset. If you change "=" to ">=" in the DoCmd OpenForm line in the function, the form will open to the record you want and display subsequent records to the chosen one as well. For example, if "BABUJ" was the



customer entered in the input box, Customer ID "ALWAO" would not be available, but "WOLVH" would.

To call the function, attach it to a command button's OnPush property, or attach it to the form's OnOpen property. Choosing the input box's Cancel button will cause the form to open at the default record with all of the records available.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 22, "Using Macros with Forms," pages 532-534, version 1.1, pages 536-537

Microsoft Access "Language Reference," version 1.0, "Input, Input\$ Functions," pages 257-258, "OpenForm Action," pages 348-351

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: Sample Function to Provide Password for Attached Tables**  
**Article ID: Q109829**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SUMMARY

=====

When you first attach an external table you have the option to store the User ID and password for the table locally. If you do not store the ID and password locally you will be prompted later for the ID and password when the table is opened (either manually or programmatically).

This article describes a sample function you can use in Microsoft Access versions 1.1 and 2.0 to eliminate the login prompt when the table is opened.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

The following steps demonstrate how to eliminate the login prompt when you open an attached table:

1. You must first provide your login information. To do this, use one of the four standard ODBC dialog boxes, or create your own custom login box.
2. Use the sample function below (or one similar to it) to establish the connection.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Function Establish_Connection ()
 Dim db As Database, Dim dbcurr As Database, Dim ds As Dynaset
 Set dbcurr = CurrentDB()
 Set db = OpenDatabase("", False, False, "ODBC;DSN=
 <servername>;UID=<UserID>;PWD=<Password>;DATABASE=
 <database name>;")
 Set ds = db.CreateDynaset("<any table existing in the
 database defined by DATABASE above>")
End Function
```

Substitute your own server name, password, database name, and table name

in the function above.

When this function is run Microsoft Access will establish a connection with the server specified in the function. The Connection Manager will then store the connection information, enabling you to open other attached tables on that server without being prompted for the login information each time. The Connection Manager will retain the login information until you quit Microsoft Access.

Additional reference words: 1.00 1.10 2.00 Attaching

KBCategory: kbusage

KBSubcategory: ObcOthr

**INF: Sample Function to Remove Alphas from a Numeric Field**  
**Article ID: Q99938**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to create a sample Microsoft Access function called RemoveAlphas() to remove alpha, or nonnumeric, characters from an alphanumeric string. All nonnumeric characters will be removed.

A common use for the RemoveAlphas() function is to remove the parentheses, dashes, and spaces from a telephone number or social security number field. The following strings contain parentheses, dashes, and spaces:

```
"(206) 635-7050"
"206-635-7050"
"535-87-4529"
```

After the RemoveAlphas() function is run, these strings will be:

```
"2066357050"
"2066357050"
"535874529"
```

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

The following example demonstrates how to create the RemoveAlphas() function, and how to use it either while data is being entered, or in an update query to remove nonnumeric characters from phone numbers in a table.

How to Create the RemoveAlphas() Function

- 
1. Create a new module.
  2. In the Module window, type "Option Explicit" (without the quotation marks).
  3. From the Edit menu, choose New Procedure.
  4. Type "RemoveAlphas" (without the quotation marks) in the Name box, and select Function in the Type box. Choose OK.

5. Type the following function in the Module window:

```
Function RemoveAlphas (AlphaNum)
 Dim Clean As String
 Dim Pos, A_Char$

 Pos = 1

 If IsNull(AlphaNum) Then Exit Function

 For Pos = 1 To Len(AlphaNum)
 A_Char$ = Mid(AlphaNum, Pos, 1)
 If A_Char$ >= "0" And A_Char$ <= "9" Then
 Clean$ = Clean$ + A_Char$
 End If
 Next Pos

 RemoveAlphas = Clean$
 AlphaNum = Clean$
End Function
```

6. Save the module with any unique name.

How to Use the RemoveAlphas() Function in Data Entry

---

You can use the RemoveAlphas() function to remove dashes or parentheses while phone numbers are being typed into a field.

For a field called Phone, add the following AfterUpdate property to the Phone text box on a form:

```
=RemoveAlphas([Phone])
```

How to Use the RemoveAlphas() Function in an Update Query

---

1. Create a new query based on the table with the phone number field.
2. Place the phone number field in the first column of the query grid.
3. From the Query menu, choose Update.
4. In the Update To row, enter the following for a field named Phone:

```
RemoveAlphas([Phone])
```

5. Run the query.

Additional reference words: 1.00 1.10 2.00 ab ssn hyphens

KBCategory: kbprg

KBSubcategory: PgmOthr

## INF: Sample Function to Replace Special Characters

Article ID: Q109825

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

When you import a text file that contains tabs or other special characters into Microsoft Access, the special characters are converted and displayed as boxes. You cannot use the Find and Replace commands to find these converted characters. This article describes a sample Access Basic function you can use to find and replace the converted tabs and other special characters.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The following steps can be used to find and replace converted special characters:

1. Create a new module and add the following code:

```
'*****
'Declarations section of the module
'*****

Option Compare Binary
 'Otherwise the function will replace spaces with percent signs
Option Explicit

'=====
```

'The following function will:  
' - Find the tabs in a text or memo field  
' - Call another function to replace the tabs  
'=====

```
Function FindTabs (WhichField As String) as String
 Dim x As Integer, Text As String
 Dim start As Integer
 start = 1
 x = 1
 Text = WhichField

 Do Until x = 0
 x = InStr(start, Text, Chr(9))
 start = x + 1
```

```

 If x > 0 And Not IsNull(x) Then
 Text = ReplaceTabs(x, Text)
 End If
 Loop

 FindTabs = Text
End Function

'=====
'The following function is called from the FindTabs() function. It
'accepts two arguments, text and start. The function replaces tabs
'with %. It returns the updated text.
'=====

Function ReplaceTabs(start As Integer, text As String) As String
 Mid(text, start, 1) = "%"
 ReplaceTabs = text
End Function

```

2. Create a query based on the table to which the text file was imported.
3. Add the field containing the special characters to the QBE grid.
4. Convert the query to an update query by choosing Update from the Query menu.
5. Add the following to the Update To line for the field(s) containing the special characters:

```
FindTabs([fieldname])
```

6. Run the query.

#### REFERENCES

=====

For more information about special characters, search for "ANSI Character" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 find/replace recognize

KBCategory: kbusage

KBSubcategory: GnlFnd

**INF: Sample Function to Retrieve File's Date and Time Stamp**  
**Article ID: Q124392**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

This article demonstrates a sample user-defined Access Basic function called GetFileDateTime() that you can use to retrieve a file's date and time stamp.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

The following example demonstrates how to create and use the GetFileDateTime() function.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Create a new module and enter the following lines in the Declarations section:

```
Option Explicit
```

```
Type OFSTRUCT
```

```
 cBytes As String * 1
 fFixedDisk As String * 1
 nErrCode As Integer
 szReserved As String * 4
 szPath As String * 128
```

```
End Type
```

```
Global Const OF_EXIST = &H4000
```

```
Declare Function WinOpenFile Lib "KERNEL.EXE" Alias _
"OpenFile" (ByVal szFileName As String, _
OpenBuff As OFSTRUCT, ByVal Flag As Integer) As Integer
```

2. Enter the following code in the module:

```
Function GetFileDateTime (ByVal FileName As String) As Variant
 Dim ofs As OFSTRUCT
 Dim iDate As Long
 Dim iTime As Long
```



```

Const DAY_MASK = &H1F
Const MONTH_MASK = &H1E0
Const YEAR_MASK = &HFE00

Const SECOND_MASK = &H1F
Const MINUTE_MASK = &H7E0
Const HOUR_MASK = &HF800

If WinOpenFile(fileName, ofs, OF_EXIST) <> -1 Then
 iDate = Asc(Mid$(ofs.szReserved, 2, 1)) * 256& _
 + Asc(Mid$(ofs.szReserved, 1, 1))
 iTime = Asc(Mid$(ofs.szReserved, 4, 1)) * 256& _
 + Asc(Mid$(ofs.szReserved, 3, 1))
 GetFileDateTime = DateSerial(((iDate And YEAR_MASK) \ _
 &H200) + 1980, (iDate And MONTH_MASK) \ &H20, (iDate And _
 DAY_MASK)) + TimeSerial((iTime And HOUR_MASK) \ &H800, _
 (iTime And MINUTE_MASK) \ &H20, (iTime And SECOND_MASK) * 2)
Else
 GetFileDateTime = Null
End If
End Function

```

3. From the View menu, choose Immediate Window.

4. Type the following line in the Immediate window and then press ENTER:

```
?GetFileDateTime("c:\autoexec.bat")
```

The date and time stamp of the AUTOEXEC.BAT file is returned in the Immediate window.

Additional reference words: 2.00

KBCategory: kbprg

KBSubcategory: MdlOthr

## **INF: Sample Function to Return a Random Record**

**Article ID: Q108435**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

Microsoft Access does not contain a built-in mechanism for returning a random record from a set of records. This article describes a sample user-defined Access Basic function you can use to return a random record.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

### MORE INFORMATION

=====

The following sample function will return a random record using the recordset name and field name that you supply. To create this function, add the following to the Global Declarations section

Option Explicit

and then enter the following in the module:

```
Function FindRandom (RecordSetName$, Fieldname$)

 Dim MyDB As Database
 Dim MyDyna As Dynaset
 Dim SpecificRecord&, i&, NumOfRecords&

 Set MyDB = CurrentDB()
 Set MyDyna = MyDB.CreateDynaset(RecordSetName$)

 On Error GoTo NoRecords

 MyDyna.MoveLast
 NumOfRecords& = MyDyna.RecordCount

 SpecificRecord& = Int(NumOfRecords * Rnd)

 If SpecificRecord& = NumOfRecords Then
 SpecificRecord& = SpecificRecord& - 1
 End If

 MyDyna.MoveFirst
 For i = 1 To SpecificRecord&
 MyDyna.MoveNext
 Next i
```

```
FindRandom = MyDyna(Fieldname$)
Exit Function
```

NoRecords:

```
If Err = 3021 Then
 MsgBox "There Are No Records In The Dynaset", 16, "Error"
Else
 MsgBox "Error - " & Err & Chr$(13) & Chr$(10) & Error, _
 16, "Error"
End If
```

```
FindRandom = "No Records"
Exit Function
```

End Function

To test the function, type the following in the module's Immediate window:

```
? FindRandom("<RecordSetName>", "<FieldName>")
```

Substitute the name of your recordset for <RecordSetName> and the name of a field in your recordset for <FieldName>.

Each time you run the function, a different record will be returned.

#### REFERENCES

=====

For more information on returning random records using a query, query on the following words here in the Microsoft Knowledge Base:

return and random and record and rnd

For more information on the Int() function or the Rnd() function, search for "Int" or "Rnd" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 ADK

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: Sample Function to Use the COMMDLG Color Dialog Box**  
**Article ID: Q109390**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The Palette in Microsoft Access provides a limited selection of colors for database objects and does not allow you to create custom colors. However, the BackColor, ForeColor, and BorderColor properties of forms and reports will accept any valid RGB color value.

This article lists a sample Access Basic function called ChooseColor() that uses the Microsoft Windows standard Color dialog box so that you can choose from a greater selection of colors or create your own custom color.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

MORE INFORMATION

=====

To set up the sample function ChooseColor(), create a new module with the following Declarations section and function:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****
' MODULE DECLARATION SECTION
'*****
Option Explicit

'
' Required COMMDLG Declarations
'
Type ChooseColor
 lStructSize As Long
 hwndOwner As Integer
 hInstance As Integer
 rgbResult As Long
 lpCustColors As Long
 Flags As Long
 lCustData As Long
 lpfnHook As Long
 lpTemplateName As Long
End Type
```

```

Global Const CC_RGBINIT = &H1
Global Const CC_FULLOPEN = &H2

Declare Function ChooseColor_API Lib "COMMDLG.DLL" Alias _
 "ChooseColor" (pCHOOSECOLOR As ChooseColor) As Integer
Declare Function CommDlgExtendedError Lib "COMMDLG.DLL" () As Long

'
' Global Memory Declarations
'
Declare Function GlobalAlloc Lib "Kernel" _
 (ByVal wFlags As Integer, ByVal dwBytes As Long) As Integer
Declare Function GlobalFree Lib "Kernel" _
 (ByVal hMem As Integer) As Integer
Declare Function GlobalLock Lib "Kernel" _
 (ByVal hMem As Integer) As Long
Declare Function GlobalUnlock Lib "Kernel" _
 (ByVal hMem As Integer) As Integer

Global Const GMEM_MOVEABLE = &H2
Global Const GMEM_ZEROINIT = &H40
Global Const GHND = (GMEM_MOVEABLE Or GMEM_ZEROINIT)

Declare Sub hmemcpy Lib "Kernel" _
 (lpDest As Any, lpSource As Any, ByVal dwBytes As Long)

'*****
' FUNCTION: ChooseColor
'
' PURPOSE:
' Uses the standard Windows Color dialog box in COMMDLG.DLL to get
' a 256-color RGB value for use in Microsoft Access. The returned
' value can be used in the BackColor, ForeColor, or BorderColor
' properties.
'
' ARGUMENTS:
' DefaultColor - The default RGB color to be selected.
' Black, 0, is usually the default.
'
' RETURN VALUE:
' >=0 The user-selected RGB value
' -1 Couldn't allocate global memory
' -2 Couldn't lock global memory
' -3 COMMDLG error occurred. A message box will display the
' number prior to this function exiting.
'
'*****
Function ChooseColor (ByVal DefaultColor As Long) As Long
 Dim C As ChooseColor
 Dim MemHandle As Long
 Dim Result As Integer, i As Integer

 ' Define CustomColor array, address, size variables
 ReDim CustomColors(15) As Long
 Dim CustomColorsAddress As Long
 Dim CustomColorsSize As Integer

```

```

' Fill custom colors array with all white
For i = 0 To UBound(CustomColors)
 CustomColors(i) = &HFFFFFF
Next

' Get size of global memory needed for custom colors
CustomColorsSize = Len(CustomColors(0)) * 16

' Get a global memory block to hold a copy of the custom colors
MemHandle = GlobalAlloc(GHND, CustomColorsSize)
If MemHandle = 0 Then
 ChooseColor = -1
 Exit Function
End If

' Lock the custom color's global memory block
CustomColorsAddress = GlobalLock(MemHandle)
If CustomColorsAddress = 0 Then
 ChooseColor = -2
 Exit Function
End If

' Copy custom colors to the global memory block
Call hmemcpy(ByVal CustomColorsAddress, _
 CustomColors(0), CustomColorsSize)

' Initialize Choose Color structure
C.lStructSize = Len(C)
C.hwndOwner = 0&
C.lpCustColors = CustomColorsAddress
C.RgbResult = DefaultColor
C.Flags = CC_RGBINIT Or CC_FULLOPEN

' Call the Choose Color COMMDLG routine
Result = ChooseColor_API(C)

' Did an error occur?
If Result = 0 And CommDlgExtendedError() <> 0 Then
 ChooseColor = -3
 MsgBox Str$(CommDlgExtendedError()), 16, "Choose Color Error"
End If

' Copy the new custom colors to the CustomColors address locally
' .. CustomColor array will now contain list of new custom colors
Call hmemcpy(CustomColors(0), _
 ByVal CustomColorsAddress, CustomColorsSize)

' Unlock and free the global memory block
Result = GlobalUnlock(MemHandle)
Result = GlobalFree(MemHandle)

' Return the selected color
ChooseColor = C.RgbResult

End Function

```

## How to Use the ChooseColor() Function

---

The following sample Access Basic function demonstrates how to use the ChooseColor() function:

```
Function GetColor()
 MsgBox Str$(ChooseColor(0))
End Function
```

The GetColor() function calls the ChooseColor() function and selects black as the default color. A message box will display the color selected in the Color dialog box.

The GetColor() function can be called from a command button by entering the following for the button's OnPush property:

```
=GetColor()
```

Additional reference words: 1.00 1.10 2.00 commdlg.dll

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: Sample Functions to Check User, Group Information (2.0)**  
**Article ID: Q123079**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article contains several sample user-defined functions. You can use these functions to:

- Return a list of users in the current system database.
- Return a list of groups in the current system database.
- Return a list of users in a specified group.
- Return a list of groups to which a specified user belongs.
- Determine if the current user belongs to a specified group.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

You can use the following sample functions to return user and group information in the current system database. Note that each function assumes there is a user called Developer who is a member of the Admins group, and that the Developer account has no password.

The Sample Functions

-----

```
'*****
'Declarations section of the module
'*****

Option Compare Database
Option Explicit

Function ListUsersInSystem ()
'*****
'Purpose: Lists users in the current system database.
'Accepts: No arguments.
'Returns: A list of users in the current system database.
'Assumes: The existence of a user called Developer in the Admins
' group, with no password.
'*****

On Error GoTo err_ListUsersInSystem

Dim MyWorkSpace As WorkSpace, i As Integer
```



```

'Create a new workspace as a member of the Admins group.
Set MyWorkSpace = DBEngine.CreateWorkspace("SPECIAL", "Developer", "")

For i = 0 To MyWorkSpace.Users.count - 1
 Debug.Print MyWorkSpace.Users(i).Name
Next i

MyWorkSpace.Close
Exit Function

err_ListUsersInSystem:
If Err = 3029 Then
 MsgBox "The account used to create the workspace does not exist"
Else MsgBox Error(Err)
End If

MyWorkSpace.Close
Exit Function

End Function

Function ListGroupsInSystem ()
'*****
'Purpose: Lists groups in the current system database.
'Accepts: No arguments.
'Returns: A list of groups in the current system database.
'Assumes: The existence of a user called Developer in the Admins
' group, with no password.
'*****

On Error GoTo err_ListGroupsInSystem

Dim MyWorkSpace As WorkSpace, i As Integer

'Create a new workspace as a member of the Admins group.
Set MyWorkSpace = DBEngine.CreateWorkspace("SPECIAL", "Developer", "")
For i = 0 To MyWorkSpace.Groups.count - 1
 Debug.Print MyWorkSpace.Groups(i).Name
Next i

MyWorkSpace.Close

Exit Function

err_ListGroupsInSystem:
If Err = 3029 Then
 MsgBox "The account used to create the workspace does not exist"
Else MsgBox Error(Err)
End If

MyWorkSpace.Close
Exit Function

End Function

Function ListUsersOfGroup (GroupName As String)
'*****

```

```

'Purpose: Lists users who are members of the specified group in
' the current system database.
'Accepts: The name of a group.
'Returns: A list of users in the specified group.
'Assumes: The existence of a user called Developer in the Admins
' group, with no password.
'*****

On Error GoTo err_ListUsersOfGroup

Dim MyWorkSpace As WorkSpace, i As Integer
Dim MyGroup As Group

'Create a new workspace as a member of the Admins group.
Set MyWorkSpace = DBEngine.CreateWorkspace("SPECIAL", "Developer", "")

Set MyGroup = MyWorkSpace.Groups (GroupName)

For i = 0 To MyGroup.Users.count - 1
 Debug.Print MyGroup.Users(i).Name
Next i

MyWorkSpace.Close
Exit Function

err_ListUsersOfGroup:
If Err = 3265 Then
 MsgBox UCase(GroupName) & " isn't a valid group name", 16, "Error"
ElseIf Err = 3029 Then
 MsgBox "The account used to create the workspace does not exist"
Else MsgBox Error(Err)
End If

MyWorkSpace.Close
Exit Function

End Function

Function ListGroupsOfUser (UserName As String)
'*****
'Purpose: Lists the groups to which a specified user belongs.
'Accepts: The name of a user.
'Returns: A list of groups for the specified user.
'Assumes: The existence of a user called Developer in the Admins
' group, with no password.
'*****

On Error GoTo err_ListGroupsOfUser

Dim MyWorkSpace As WorkSpace, i As Integer
Dim MyUser As User

'Create a new workspace as a member of the Admins group.
Set MyWorkSpace = DBEngine.CreateWorkspace("SPECIAL", "Developer", "")

Set MyUser = MyWorkSpace.Users (UserName)

```

```

For i = 0 To MyUser.Groups.count - 1
 Debug.Print MyUser.Groups(i).Name
Next i

MyWorkSpace.Close
Exit Function

err_ListGroupsOfUser:
If Err = 3265 Then
 MsgBox UCase(UserName) & " isn't a valid user name", 16, "Error"
ElseIf Err = 3029 Then
 MsgBox "The account used to create the workspace does not exist"
Else MsgBox Error(Err)
End If

MyWorkSpace.Close
Exit Function

End Function

Function CurrentUserInGroup (GroupName As String)
'*****
'Purpose: Determines if the current user belongs to the specified
' group.
'Accepts: The name of a group.
'Returns: True if the current user is a member of the specified
' group, False if the current user is not a member of
' the group.
'Assumes: The existence of a user called Developer in the Admins
' group, with no password.
'*****

On Error GoTo err_CurrentUserInGroup

Dim MyWorkSpace As Workspace, i As Integer
Dim MyGroup As Group, MyUser As User

'Create a new workspace as a member of the Admins group.
Set MyWorkSpace = DBEngine.CreateWorkspace("SPECIAL", "Developer", "")

Set MyGroup = MyWorkSpace.Groups(GroupName)
Set MyUser = MyWorkSpace.Users(CurrentUser())
For i = 0 To MyGroup.Users.count - 1
 If MyGroup.Users(i).Name = MyUser.Name Then
 CurrentUserInGroup = True
 Exit Function
 End If
Next i

CurrentUserInGroup = False
MyWorkSpace.Close
Exit Function

err_CurrentUserInGroup:
If Err = 3265 Then
 MsgBox UCase(GroupName) & " isn't a valid group name", 16, "Error"
 CurrentUserInGroup = False

```

```
ElseIf Err = 3029 Then
 MsgBox "The account used to create the workspace does not exist"
Else MsgBox Error(Error)
End If

MyWorkSpace.Close
Exit Function

End Function
```

To test these functions, run them in the Immediate window. For example, to test the ListGroupsOfUser() function, follow these steps:

1. Open the sample database NWIND.MDB.
2. Create a new module and enter the sample functions above.
3. From the View menu, choose Immediate Window.
4. In the Immediate window, type the following line and then press ENTER:

```
? ListGroupsOfUser("Admin")
```

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 7, "Objects and Collections," pages 155-181, and Chapter 13, "Securing Your Application," pages 313-344

Additional reference words: 2.00 security dao

KBCategory: kbusage

KBSubcategory: ScrUsr

**INF: Sample Functions to Send MCI Commands from Access Basic**  
**Article ID: Q109371**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article demonstrates how to create sample Access Basic functions that you can use to make calls to the media control interface (MCI) using the `mciSendString()` function to run a CD player application from Microsoft Access.

NOTE: These sample functions require that you use Windows version 3.1 or Multimedia Windows version 3.0, and a CD-ROM drive that supports MCI.

MORE INFORMATION

=====

The sample program below uses several functions to control basic CD-ROM drive features such as stop, play, and eject. The functions can be used in any CD-ROM application you create. The sample program below is not a complete application, but is rather a basis you can use to create a full-featured application.

The simplest way to use these functions is to create a form with five command buttons on it. Have each of the buttons call one of the five sample functions below with its `OnPush` property.

To create these functions, create a new Access Basic module and enter the following Declarations and functions:

NOTE: In the following sample code, an underscore (`_`) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'***** Declarations Section *****

Option Compare Database
Option Explicit

Declare Function mciSendString% Lib "MMSYSTEM" (ByVal Sound$, _
 ByVal RtnString$, ByVal RtnLength%, ByVal Hndl%)

Dim RetInt As Integer
Dim RetStr As String * 64
Dim mciStatement As String

Global gTrackCurrent As Integer ' stores current track on CD.
Global gTrackCount As Integer ' stores count of tracks on CD.

'***** Play Function *****
```

```

Function CDPlay () As Integer
 ' This function initializes the audio device
 ' with the first call to mciSendString(), and
 ' then starts the CD playing with the second call.

 RetInt = mciSendString("Open CDAudio", "", 0, 0)
 CDPlay = mciSendString("Play CDAudio", "", 0, 0)
End Function

'***** Stop Function *****

Function CDStop () As Integer
 ' This function stops the CD with the first call
 ' to mciSendString(), and closes the audio device
 ' with the second call.
 RetInt = mciSendString("Stop CDAudio", "", 0, 0)
 CDStop = mciSendString("Close CDAudio", "", 0, 0)
End Function

'***** Previous Track Function *****

Function CDTrackPrevious ()
 Dim TrackPrevious As Integer

 ' Set time format to Tracks, minutes, seconds, and frames.
 RetInt = mciSendString("Set CDAudio Time Format TMSF", "", 0, 0)

 ' Retrieve the track number that is currently playing, and
 ' the total number of tracks on the CD. Store these values
 ' in global variables so that they are available to the form.

 mciStatement = "Status CDAudio Current Track"
 RetInt = mciSendString(mciStatement, RetStr, 63, 0)
 gTrackCurrent = Val(RetStr)
 TrackPrevious = gTrackCurrent - 1
 mciStatement = "Status CDAudio Number of Tracks"
 RetInt = mciSendString(mciStatement, RetStr, 63, 0)
 gTrackCount = Val(RetStr)

 ' Check to see if the current track is the first track. If it
 ' is seek to the beginning of the current track and play it.
 ' Otherwise, move to the previous track and play from there to
 ' the end of the CD.

 If TrackPrevious > 0 Then
 mciStatement = "Play CDAudio from " & TrackPrevious
 mciStatement = mciStatement & " To " & gTrackCount
 CDTrackPrevious = mciSendString(mciStatement, "", 0, 0)
 Else
 mciStatement = "Seek CDAudio to " & TrackPrevious
 RetInt = mciSendString(mciStatement, "", 0, 0)
 CDTrackPrevious = mciSendString("Play CDAudio", "", 0, 0)
 End If
End Function

'***** Next Track Function *****

```

```

Function CDTrackNext () As Integer
 Dim TrackNext As Integer

 ' Set time format to Tracks, minutes, seconds, and frames.
 RetInt = mciSendString("Set CDAudio Time Format TMSF", "", 0, 0)

 ' Retrieve the track number that is currently playing, and
 ' the total number of tracks on the CD. Store these values
 ' in global variables so that they are available to the form.

 mciStatement = "Status CDAudio Current Track"
 RetInt = mciSendString(mciStatement, RetStr, 63, 0)
 gTrackCurrent = Val(RetStr)
 TrackNext = gTrackCurrent + 1
 mciStatement = "Status CDAudio Number of Tracks"
 RetInt = mciSendString(mciStatement, RetStr, 63, 0)
 gTrackCount = Val(RetStr)

 ' Check to see if the current track is the last track. If it
 ' is seek to the beginning of the last track and play it.
 ' Otherwise, move to the next track and play from there to the end
 ' of the CD.

 If (TrackNext) < gTrackCount Then
 mciStatement = "Play CDAudio from " & TrackNext
 mciStatement = mciStatement & " To " & gTrackCount
 CDTrackNext = mciSendString(mciStatement, "", 0, 0)
 Else
 mciStatement = "Seek CDAudio to " & gTrackCurrent
 RetInt = mciSendString(mciStatement, "", 0, 0)
 CDTrackNext = mciSendString("Play CDAudio", "", 0, 0)
 End If
End Function

'***** Eject Function *****

Function CDEject () As Integer
 ' This function is not supported by all CD devices.
 CDEject = mciSendString("Set CDAudio Door Open", "", 0, 0)
End Function

'***** Pause Function *****

Function CDPause () As Integer
 CDPause = mciSendString("Stop CDAudio", "", 0, 0)
End Function

```

Note that these functions are only a sample; there are many other functions that can be implemented with `mciSendString()`. For example, the `Position` command returns the current track number that is being played. By using the `Position` command with the timer function described in article Q95924, "How to Implement a Timer," you can create a dynamically updated readout of the current track number being played.

#### REFERENCES

=====

Microsoft Windows Software Development Kit "Multimedia Programmer's Reference," Chapter 7, "MCI Command Strings"

Additional reference words: 1.00 1.10 2.00 CDROM

KBCategory: kbprg

KBSubcategory: PgmApi



## INF: Sample Macro for Repair/Compact Operations

Article ID: Q100923

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article demonstrates a sample macro that repairs and compacts a database.

### MORE INFORMATION

=====

To manually repair and compact a database from the File menu, you must close the database and perform the compact and repair steps individually.

The sample macro in this article automates this process. It repairs a database called MYDB and compacts it to a new database called NewDB. The macro will only prompt for information if the name of the database it is compacting to already exists.

NOTE: The macro must be run from a different database than the one you want to repair and compact.

### Sample Macro

-----

| MacroName | Action      | Defined Below |
|-----------|-------------|---------------|
| Test      | SetWarnings | 1             |
|           | SendKeys    | 2             |
|           | DoMenuItem  | 3             |
|           | SendKeys    | 4             |
|           | DoMenuItem  | 5             |
|           | SetWarnings | 6             |

### Test Actions

- 
1. SetWarnings  
WarningsOn: No
  2. SendKeys  
Keystrokes: MYDB.mdb{enter}  
Wait: No
  3. DoMenuItem  
Menu Bar: Startup (in Microsoft Access version 1.x = Init)  
Menu Name: File  
Command: Repair Database
  4. SendKeys  
Keystrokes: MYDB.mdb{enter}NewDB{enter}  
Wait: No
  5. DoMenuItem  
Menu Bar: Startup (in Microsoft Access version 1.x = Init)

Menu Name: File  
Command: Compact Database

6. SetWarnings  
Warnings On: Yes

#### Notes

-----

- In most cases this macro works fine. However, remember that keystrokes sent by the SendKeys action are buffered temporarily and that results may vary. A large database may take extra time to compact or repair; in this case, the macro may not exhibit expected behavior.
- In Microsoft Access version 2.0, you can use new command-line options to create an icon in Program Manager to compact or repair your database. You can also use the new data access objects (DAO) methods CompactDatabase and RepairDatabase from Access Basic. For more information about these methods, search for "command line," "CompactDatabase," and "RepairDatabase" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: GnlCmp

**INF: Sample Macro to Reset Page Number on Group Level Report**  
**Article ID: Q104760**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

It is possible, using macro and report section properties, to design a report that breaks the page for each new entry in a group and resets the report page number. For example, the "List of Products by Category" report in the sample database NWIND.MDB could be redesigned to break the page for each new category and reset the page number to 1.

MORE INFORMATION

=====

1. Create the following macro:

| Macro Name | Action   |
|------------|----------|
| Pager      | SetValue |

Macro Action Arguments

|               |
|---------------|
| SetValue      |
| Item: Page    |
| Expression: 1 |

2. In the sample database NWIND.MDB, open the "List of Products by Category" report in Design view.
3. Select the Category Name header section and assign the Pager macro to the OnFormat property.
4. Select the Category Name Footer section and set the ForceNewPage property to After Section.

Each category will now begin on a new page and the numbering of these new pages will all begin with the number 1. In the Products table, there currently are not enough records in each category to require more than one page. However, if the Detail section of the "List of Products by Category" report was enlarged to 1 inch, several of the categories would require two pages and their page numbers would continue to increase until the next category occurs.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 23, "Using Macros to Print Reports or Transfer Data," pages 571-572

Microsoft Access "User's Guide," version 2.0, Chapter 26, "Using Macros to

Print Reports or Transfer Data"

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptLayou

## INF: Sample MSACC20.INI File for MS Access Version 2.0

Article ID: Q122246

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article contains the contents of a generic MSACC20.INI file for Microsoft Access version 2.0. The MSACC20.INI file is created by Microsoft Access Setup. If the MSACC20.INI file in your Windows directory is damaged or removed, you can run Setup again, or you can create a new file by re-creating the generic file below. Note that the MSACC20.INI file is a text file, and can be created with any text editor.

### MORE INFORMATION

=====

To create a new MSACC20.INI file, re-create the following file using any text editor and then save the file in your Windows directory as MSACC20.INI. Note that the following file is an example only; you must use appropriate paths and filenames for your installation in the [Options] section of the file.

#### Generic MSACC20.INI File Contents

-----

NOTE: In the following sample file, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this file.

```
;=====
; Generic MSACC20.INI file for version 2.0
;=====
```

#### [Options]

```
SystemDB=C:\ACCESS\system.mda ;path may differ
UtilityDB=C:\ACCESS\utility.mda ;path may differ
AllowCustomControls=1
AllowOLE1LinkFormat=0
```

#### [Clipboard Formats]

```
Microsoft Excel (*.xls)=soa200.dll,1,xls
Rich Text Format (*.rtf)=soa200.dll,2,rtf
MS-DOS Text (*.txt)=soa200.dll,3,txt
```

#### [Report Formats]

```
Microsoft Excel (*.xls)=xls,SOA_RptToBIFF,Biff3,Microsoft Excel (*.xls)
Rich Text Format (*.rtf)=rtf,SOA_RptToRTF,Rich Text Format,Rich Text _
 Format (*.rtf)
MS-DOS Text (*.txt)=txt,SOA_RptToAscii,1,MS-DOS Text (*.txt)
```

#### [Microsoft Access]

Filter=Microsoft Access (\*.mdb)|\*.mdb|All Files (\*.\*)|\*.\*|  
Extension=mdb  
OneTablePerFile=No  
IndexDialog=No  
Maximized=1  
CreateDbOnExport=No  
Tutorial=1

[Delimited Text]  
Filter=Delimited Text (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*|  
Extension=txt  
OneTablePerFile=Yes  
IndexDialog=No  
CreateDbOnExport=No

[Fixed Width Text]  
Filter=Fixed Width Text (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*|  
Extension=txt  
OneTablePerFile=Yes  
IndexDialog=No  
CreateDbOnExport=No

[Word for Windows Merge]  
Filter=Text (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*|  
Extension=txt  
OneTablePerFile=Yes  
IndexDialog=No  
CreateDbOnExport=No

[Microsoft Excel 2.0-4.0]  
Filter=Microsoft Excel (\*.xls)|\*.xls|All Files (\*.\*)|\*.\*|  
Extension=xls  
OneTablePerFile=Yes  
IndexDialog=No  
CreateDbOnExport=No

[Microsoft Excel 5.0]  
Filter=Microsoft Excel 5 (\*.xls)|\*.xls|All Files (\*.\*)|\*.\*|  
Extension=xls  
OneTablePerFile=Yes  
IndexDialog=No  
CreateDbOnExport=Yes

[Lotus WKS]  
Filter=Lotus 1-2-3 (\*.wks)|\*.wks|All Files (\*.\*)|\*.\*|  
Extension=wks  
OneTablePerFile=Yes  
IndexDialog=No  
CreateDbOnExport=No

[Lotus WK1]  
Filter=Lotus 1-2-3 (\*.wk1)|\*.wk1|All Files (\*.\*)|\*.\*|  
Extension=wk1  
OneTablePerFile=Yes  
IndexDialog=No  
CreateDbOnExport=No

```
[Lotus WK3]
Filter=Lotus 1-2-3 (*.wk3)|*.wk3|All Files (*.*)|*.*|
Extension=wk3
OneTablePerFile=Yes
IndexDialog=No
CreateDbOnExport=No
```

```
[Menu Add-Ins]
&Add-in Manager==Wm_Entry()
&Database Documentor==Doc_PrintDataBase()
A&ttachment Manager==Am_Entry()
I&mport Database...==Doc_ImportDatabase()
&Menu Builder==CustomMenuBuilder()
```

```
[Libraries]
wzlib.mda=rw
wzTable.mda=rw
wzQuery.mda=rw
wzfrmrpt.mda=rw
wzblldr.mda=rw
```

```
[Table Wizards]
Table=TW_Entry,,{This wizard creates a new table to store data.}
```

```
[Table Wizard Data Files]
wztblat.mdt=Standard Microsoft Sample Fields
```

```
[Query Wizards]
Crosstab Query=Xtq_Entry,,{This wizard creates a crosstab query that _
 displays data in a compact, spreadsheet-like format.}
Find Duplicates Query=dup_Entry,,{This wizard creates a query that _
 finds duplicate records (rows) in a single table or query.}
Find Unmatched Query=dwz_Entry,,{This wizard creates a query that finds _
 records (rows) in one table that have no related records in another _
 table.}
Archive Query=arc_Entry,,{This wizard creates a query that copies _
 records from an existing table into a new table.}
```

```
[Form Wizards]
Single-Column=zwForm, 1,{This wizard creates a form that displays fields _
 in a single column.}
Tabular=zwForm, 2,{This wizard creates a form that displays each record _
 as a row of fields.}
Graph=zwGraph,, {This wizard creates a form that displays a graph.}
Main/Subform=zwMainSub,, {This wizard creates a form that contains _
 another form.}
AutoForm=zwAutoForm,, {AutoForm automatically creates a simple form _
 based on the selected table or query.}
```

```
[Report Wizards]
Single-Column=zwReport, 3, {This wizard creates a report that displays _
 fields in a single column.}
Groups/Totals=zwReport, 4, {This wizard creates a report that groups _
 records together and displays totals for each group.}
Mailing Label=zwMailingLabel,, {This wizard creates standard Avery _
 mailing labels.}
Summary=zwReport, 12, {This wizard creates a report that displays totals _
```

for groups of records.)  
Tabular=zwReport, 11, {This wizard creates a report that displays each \_  
record as a row of fields.}  
AutoReport=zwAutoReport,, {AutoReport automatically creates a simple \_  
report based on the selected table or query.}  
MS Word Mail Merge=PM\_Entry,, {This wizard links your data to a \_  
Microsoft Word document, so that you can print form letters or \_  
address envelopes.}

[Control Wizards]

MSListBoxWizard=ListBox, list box Builder, LST\_ENTRY,w  
MSComboBoxWizard=ComboBox, combo box Wizard, CMB\_ENTRY,w  
MSOptionGroupWizard=OptionGroup, OptionGroup Wizard, OGrp\_ENTRY,w  
MSCommandButtonWizard=CommandButton, command button Wizard, BW\_ENTRY,w

[Property Wizards]

MSMenuBarBuilder=Menubar, MenuBar Builder, MB\_ENTRY,rw  
MSInputMaskWizard=InputMask, Input Mask Builder, IM\_ENTRY,rw  
MSForeColorBuilder=ForeColor, ForeColor Builder, CP\_ENTRY,rw  
MSBackColorBuilder=BackColor, BackColor Builder, CP\_ENTRY,rw  
MSBorderColorBuilder=BorderColor, BorderColor Builder, CP\_ENTRY,rw  
MSPictureBuilder=Picture, Picture Builder, PP\_ENTRY,rw  
MSODBCConnectStrBuilder=ODBCConnectStr,ODBC ConnectString  
Builder,zwBuilderODBCConnectString,rw  
MSFieldBuilder=FieldName,Field Builder,tw\_FieldEntry, rw

[ISAM]

PageTimeout=5  
LockedPageTimeout=5  
CursorTimeout=10  
LockRetry=20  
CommitLockRetry=20  
MaxBufferSize=512  
ReadAheadPages=16  
IdleFrequency=10

[Paradox 3.X]

Filter=Paradox (\*.db)|\*.db|All Files (\*.\*)|\*.\*|  
Extension=db  
OneTablePerFile=Yes  
IndexDialog=No  
CreateDbOnExport=No

[Paradox 4.X]

Filter=Paradox (\*.db)|\*.db|All Files (\*.\*)|\*.\*|  
Extension=db  
OneTablePerFile=Yes  
IndexDialog=No  
CreateDbOnExport=No

[Installable ISAMs]

Paradox 3.X=C:\WINDOWS\SYSTEM\pdx200.dll  
Paradox 4.X=C:\WINDOWS\SYSTEM\pdx200.dll  
FoxPro 2.5=C:\WINDOWS\SYSTEM\xbs200.dll  
FoxPro 2.6=C:\WINDOWS\SYSTEM\xbs200.dll  
FoxPro 2.0=C:\WINDOWS\SYSTEM\xbs200.dll  
dBASE III=C:\WINDOWS\SYSTEM\xbs200.dll



dBASE IV=C:\WINDOWS\SYSTEM\xbs200.dll

[Paradox ISAM]

ParadoxUserName=rms  
ParadoxNetPath=C:\ACCESS  
ParadoxNetStyle=3.x  
CollatingSequence=Ascii

[FoxPro 2.0]

Filter=FoxPro 2.0 (\*.dbf)|\*.dbf|All Files (\*.\*)|\*.\*|  
Extension=dbf  
OneTablePerFile=Yes  
IndexDialog=Yes  
IndexFilter=FoxPro Index (\*.idx;\*.cdx)|\*.idx;\*.cdx|All Files (\*.\*)|\*.\*|  
CreateDbOnExport=No

[FoxPro 2.5]

Filter=FoxPro 2.5 (\*.dbf)|\*.dbf|All Files (\*.\*)|\*.\*|  
Extension=dbf  
OneTablePerFile=Yes  
IndexDialog=Yes  
IndexFilter=FoxPro Index (\*.idx;\*.cdx)|\*.idx;\*.cdx|All Files (\*.\*)|\*.\*|  
CreateDbOnExport=No

[FoxPro 2.6]

Filter=FoxPro 2.6 (\*.dbf)|\*.dbf|All Files (\*.\*)|\*.\*|  
Extension=dbf  
OneTablePerFile=Yes  
IndexDialog=Yes  
IndexFilter=FoxPro Index (\*.idx;\*.cdx)|\*.idx;\*.cdx|All Files (\*.\*)|\*.\*|  
CreateDbOnExport=No

[dBASE III]

Filter=dBASE III (\*.dbf)|\*.dbf|All Files (\*.\*)|\*.\*|  
Extension=dbf  
OneTablePerFile=Yes  
IndexDialog=Yes  
IndexFilter=dBASE Index (\*.ndx)|\*.ndx|All Files (\*.\*)|\*.\*|  
IndexExtension=ndx  
CreateDbOnExport=No

[dBASE IV]

Filter=dBASE IV (\*.dbf)|\*.dbf|All Files (\*.\*)|\*.\*|  
Extension=dbf  
OneTablePerFile=Yes  
IndexDialog=Yes  
IndexFilter=dBASE Index (\*.ndx;\*.mdx)|\*.ndx;\*.mdx|All Files (\*.\*)|\*.\*|  
CreateDbOnExport=No

[dBase ISAM]

CollatingSequence=Ascii  
Century=Off  
Date=American  
Mark=47  
Deleted=On

[ODBC]

QueryTimeout=60  
LoginTimeout=20

#### REFERENCES

=====

For additional information about the MSACC20.INI file, search for "MSACC20.INI" then "Customizing MSACC20.INI Settings" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbsetup

KBSubcategory: StpHowto

**INF: Sample OLE Automation for MS Word and MS Excel**  
**Article ID: Q123859**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article demonstrates how to activate Microsoft Word version 6.0 for Windows or Microsoft Excel version 5.0 for Windows from a command button on a Microsoft Access form.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

MORE INFORMATION

=====

How to Activate Word

-----

The following example assumes a Microsoft Access table with an OLE object field called MyOle that contains embedded (not linked) Word 6.0 objects. The example demonstrates how to create a command button that will activate Word 6.0 in Normal view with the current Word 6.0 object:

1. Create a form based on the table containing the MyOle field.
2. Add a bound object frame with the following properties to the form:

Name: MyOle  
ControlSource: MyOle

3. Add a command button named Button2 to the form. Set the button's OnClick property to the following event procedure:

```
Sub Button2_Click ()
 Dim Word_Obj As Object
 MyOle.verb = -2
 MyOle.action = 7
 Set Word_Obj = MyOle.object.application.wordbasic
 Word_Obj.ViewNormal
End Sub
```

How to Activate Microsoft Excel

-----

The following example assumes a Microsoft Access table with an OLE object field called My\_excel\_ole that contains embedded (not linked) Microsoft Excel 5.0 objects. The example demonstrates how to create a command button that will activate Microsoft Excel with the current Microsoft Excel

object:

1. Create a form based on the table containing the My\_excel\_ole field.
2. Add a bound object frame with the following properties to the form:

```
Name: My_excel_ole
ControlSource: My_excel_ole
```

3. Add a command button named Button3 to the form. Set the button's OnClick property to the following event procedure:

```
Sub Button3_Click ()
 My_excel_ole.verb = -2
 My_excel_ole.action = 7
End Sub
```

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 13,  
"Communicating with Other Applications," pages 282-297

For additional information about OLE Automation, search for "OLE  
Automation" then "Interoperability with Microsoft Word and Microsoft Excel  
(Common Questions)" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbole

KBSubcategory: IntpOlea

**INF: Sample Table Design to Support Questionnaire Applications**  
**Article ID: Q101675**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes a table design suitable for an application that tallies results from questionnaires and surveys.

MORE INFORMATION

=====

The following examples outline nonrelational table design commonly used for questionnaires and surveys and suggests an improved, relational table design.

Nonrelational Table Design

-----

When designing tables for questionnaire applications, many users begin with a design that resembles the table below. Each record contains multiple fields, called Question1 through Question<n>, that contain responses to the questions.

Table: Table1 (old)

-----

FieldName: Respondent ID [Primary Key]

FieldName: Question1

FieldName: Question2

FieldName: Question3

.

.

.

FieldName: Question<n>

Problems occur when you want to perform crosstab queries to summarize and/or graph the questionnaire results.

Relational Table Design

-----

A relational table design better supports summary queries, reports, and graphs. In the table below, the Question ID field identifies the question and the Response field contains the answer.

Table: Table2 (new)

-----

Field: Respondent ID [Primary Key]

Field: Question ID [Primary Key]

Field: Response

## How to Switch to Relational Database Design

---

To convert data that has been entered in Table1 format, complete the following six steps:

1. Create Table2 according to the above structure, using the following data type criteria:
  - All Respondent ID fields must be the same data type.
  - The Question ID field must be a Text data type.
  - The Response field must be the same data type as the Question<x> data types.
2. Create a new query based on Table1.
3. From the Query menu, choose Append. Select Table2 as the table to which you want to append the data. Design the query as follows:

```
Query: Query1

Field: Respondent ID
 Append To: Respondent ID
Field: Question1
 Append To: Response
Field: "Question1"
 Append To: Question ID
```

4. Run Query1 to append to Table2 each participant's responses to Question1.
5. Repeat steps 2-4 above, replacing Question1 with Question2, and "Question1" with "Question2". You must re-create or modify this query for each question in Table1.
6. After running all the append queries, the result is a table (Table2) that can easily summarize your results in a crosstab query:

```
Query: CrossTabExample

Field: Question ID
 Total: Group By
 Crosstab: Row Heading
Field: Response
 Total: Group By
 Crosstab: Column Heading
Field: Response
 Total: Count
 Crosstab: Value
```

## REFERENCES

---

Microsoft Access "User's Guide," version 1.0, Chapters 1, 6, and 7

Additional reference words: 1.00 1.10 2.00 cross tab append questionnaire  
KBCategory: kbusage  
KSubcategory: TblDsign

**INF: Sending a MAPI Mail Message Using Access Basic**  
**Article ID: Q99403**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY  
=====

The following example includes a Microsoft Access module that demonstrates how to send a Mail Application Programming Interface (MAPI) message to multiple TO recipients and carbon copies (CC).

MORE INFORMATION  
=====

The following procedure explains how to use this sample module:

1. Create a form with the following controls:

```
Form: Test

Caption: TestForm
Text box: Subject
 ControlName: Subject
Text box: To
 ControlName: To
Text box: CC
 ControlName: CC
Text box: Attach
 ControlName: Attach
Text box: Message
 ControlName: Message
Command button: Button0
 Caption: Send Message
 OnPush: =Mail()
```

2. Open either a new module or a previously created module and enter the following code:

NOTE: In the following sample code, an underscore (\_) is used as a line continuation character for easier reading. Remove the underscore when re-creating this code in Access Basic.

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

```
Option Explicit
Option Compare Database 'Use database order for string comparisons
```

```
Type MAPIMessage
 Reserved As Long
 Subject As String
```



```
NoteText As String
MessageType As String
DateReceived As String
ConversationID As String
Flags As Long
RecipCount As Long
FileCount As Long
End Type
```

```
Type MapiRecip
 Reserved As Long
 RecipClass As Long
 Name As String
 Address As String
 EIDSize As Long
 EntryID As String
End Type
```

```
Type MapiFile
 Reserved As Long
 Flags As Long
 Position As Long
 PathName As String
 FileName As String
 FileType As String
End Type
```

```
Declare Function MAPISendMail Lib "MAPI.DLL" Alias "BMAPISendMail" _
 (ByVal Session&, ByVal UIParam&, Message As MAPIMessage, Recipient _
 As MapiRecip, File As MapiFile, ByVal Flags&, ByVal Reserved&) As Long
```

```
Global Const SUCCESS_SUCCESS = 0
Global Const MAPI_TO = 1
Global Const MAPI_CC = 2
```

```
Global Const MAPI_LOGON_UI = &H1
```

```
'*****
' End of declarations section
'*****

'*****
' FUNCTION NAME: Mail
'
' PURPOSE:
' Passes information on the active forms To, Subject, CC,
' Attach, and Message text boxes to the SendMail function.
' It ensures that each box does not have a NULL value. It also
' displays an error message if SendMail fails.
' This function is called from the OnPush property of the form.
'
' INPUT PARAMETERS:
' None
'
' RETURN
' None
'*****
```

```

Function Mail ()
 Dim F As Form, Result
 Set F = Screen.ActiveForm

 ' Make sure user has something in the To: box
 If IsNull(F!To) Or F!To = "" Then Exit Function

 ' Make sure no Null values are in the other boxes
 If IsNull(F!Subject) Then F!Subject = ""
 If IsNull(F!CC) Then F!CC = ""
 If IsNull(F!Attach) Then F!Attach = ""
 If IsNull(F!Message) Then F!Message = ""

 ' Send the message, passing information from the form
 Result = SendMail((F!Subject), (F!To), (F!CC), _
 (F!Attach), (F!Message))

 ' Test the result for any errors
 If Result <> SUCCESS_SUCCESS Then
 MsgBox "Error sending mail: " & Result, 16, "Mail"
 Else
 MsgBox "Message sent successfully!", 64, "Mail"
 End If
End Function

```

```

'*****
' FUNCTION NAME: SendMail
'
' PURPOSE:
' This is the front-end function to the MAPI SendMail function. You
' pass a semicolon-delimited list of To and CC recipients, a
' subject, a message, and a delimited list of file attachments.
' This function prepares MapiRecip and MapiFile structures with the
' data parsed from the information provided using the ParseRecord
' sub. Once the structures are prepared, the MAPI SendMail function
' is called to send the message.
'
' INPUT PARAMETERS:
' sSubject: The text to appear in the subject line of the message
' sTo: Semicolon-delimited list of names to receive the
' message
' sCC: Semicolon-delimited list of names to be CC'd
' sAttach: Semicolon-delimited list of files to attach to
' the message
' RETURN
' SUCCESS_SUCCESS if successful, or a MAPI error if not.
'*****
Function SendMail (sSubject As String, sTo As String, sCC As String, _
 sAttach As String, sMessage As String)
 Dim i, cTo, cCC, cAttach ' variables holding counts
 Dim MAPI_Message As MAPIMessage

 ' Count the number of items in each piece of the mail message
 cTo = CountTokens(sTo, ";")
 cCC = CountTokens(sCC, ";")
 cAttach = CountTokens(sAttach, ";")

```

```

' Create arrays to store the semicolon delimited mailing
' .. information after it is parsed
ReDim rTo(0 To cTo) As String
ReDim rCC(0 To cCC) As String
ReDim rAttach(0 To cAttach) As String

' Parse the semicolon delimited information into the arrays.
ParseTokens rTo(), sTo, ";"
ParseTokens rCC(), sCC, ";"
ParseTokens rAttach(), sAttach, ";"

' Create the MAPI Recip structure to store all the To and CC
' .. information to be passed to the MAPI_SendMail function
ReDim MAPI_Recip(0 To cTo + cCC - 1) As MapiRecip

' Setup the "TO:" recipient structures
For i = 0 To cTo - 1
 MAPI_Recip(i).Name = rTo(i)
 MAPI_Recip(i).RecipClass = MAPI_TO
Next i

' Setup the "CC:" recipient structures
For i = 0 To cCC - 1
 MAPI_Recip(cTo + i).Name = rCC(i)
 MAPI_Recip(cTo + i).RecipClass = MAPI_CC
Next i

' Create the MAPI File structure to store all the file attachment
' .. information to be passed to the MAPI_SendMail function
ReDim MAPI_File(0 To cAttach) As MapiFile

' Setup the file attachment structures
MAPI_Message.FileCount = cAttach
For i = 0 To cAttach - 1
 MAPI_File(i).Position = -1
 MAPI_File(i).PathName = rAttach(i)
Next i

' Set the mail message fields
MAPI_Message.Subject = sSubject
MAPI_Message.NoteText = sMessage
MAPI_Message.RecipCount = cTo + cCC

' Send the mail message
SendMail = MAPI_SendMail(0&, 0&, MAPI_Message, MAPI_Recip(0), _
 MAPI_File(0), MAPI_LOGON_UI, 0&)
End Function

'*****
' FUNCTION NAME: CountTokens
'
' PURPOSE:
' Given a string of delimited items and the delimiter, the number
' of tokens in the string will be returned. This function is useful
' for dimensioning an array to store the delimited items prior to
' calling ParseTokens.

```

```

'
' INPUT PARAMETERS:
' sSource: A delimited list of tokens
' sDelim: The delimiter used to delimit sSource
'
' RETURN
' The number of tokens in sSource, which is the number of delimiters
' plus 1. If sSource is empty, 0 is returned.
'*****
Function CountTokens (ByVal sSource As String, ByVal sDelim As String)
 Dim iDelimPos As Integer
 Dim iCount As Integer

 ' Number of tokens = 0 if the source string is empty
 If sSource = "" Then
 CountTokens = 0

 ' Otherwise number of tokens = number of delimiters + 1
 Else
 iDelimPos = InStr(1, sSource, sDelim)
 Do Until iDelimPos = 0
 iCount = iCount + 1
 iDelimPos = InStr(iDelimPos + 1, sSource, sDelim)
 Loop
 CountTokens = iCount + 1
 End If
End Function

```

```

'*****
' FUNCTION NAME: GetToken
'
' PURPOSE:
' Given a string of delimited items, the first item will be
' removed from the list and returned.
'
' INPUT PARAMETERS:
' sSource: A delimited list of tokens
' sDelim: The delimiter used to delimit sSource
'
' RETURN
' sSource will have the first token removed. The function
' returns the token removed from sSource.
'*****
Function GetToken (sSource As String, ByVal sDelim As String) _
As String
 Dim iDelimPos As Integer

 ' Find the first delimiter
 iDelimPos = InStr(1, sSource, sDelim)

 ' If no delimiter was found, return the existing string and set
 ' .. the source to an empty string.
 If (iDelimPos = 0) Then
 GetToken = Trim$(sSource)
 sSource = ""

 ' Otherwise, return everything to the left of the delimiter and

```

```

' .. return the source string with it removed.
Else
 GetToken = Trim$(Left$(sSource, iDelimPos - 1))
 sSource = Mid$(sSource, iDelimPos + 1)
End If
End Function

'*****
' SUB NAME: ParseTokens
'
' PURPOSE:
' Extracts information from a delimited list of items and places
' it in an array.
'
' INPUT PARAMETERS:
' Array(): A one-dimensional array of strings in which the parsed tokens
' will be place
' sTokens: A delimited list of tokens
' sDelim: The delimiter used to delimit sTokens
'
' RETURN
' None
'*****
Sub ParseTokens (Array() As String, ByVal sTokens As String, ByVal _
 sDelim As String)
 Dim i As Integer
 For i = LBound(Array) To UBound(Array)
 Array(i) = GetToken(sTokens, sDelim)
 Next
End Sub

```

REFERENCES

=====

Microsoft Mail "Technical Reference," Chapter 4

Additional reference words: 1.00 1.10 2.00 mail application programming

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: Set Focus to a Subform Control Using GoToControl Action**  
**Article ID: Q95014**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can create a macro to move the focus to a control on a subform by using the GoToControl macro action to move first to the subform, which is a type of a control. Use the GoToControl action again to move to a particular control on the subform.

MORE INFORMATION

=====

The macro action GoToControl does not allow you to enter the full syntax for the ControlName argument as:

```
Forms![orders]![orders subform].form![Product ID]
```

If you enter this syntax, you will receive the following error:

```
There is no control named 'Forms![orders]![orders
subform].Form!Product id'.
```

To get around this, you need to set up a macro that first sets the focus to the subform control, followed by a second action that sets the focus to a specific control.

In the sample database Northwind Traders (NWIND.MDB), the Orders and Orders Subform forms can be used as an example where you can have a macro set the focus to the Product Id on the Orders Subform.

Create a new macro with two separate GoToControl actions, then attach the macro to a button on the Orders form.

Actions

```

GoToControl
 Control Name: Orders Subform
GoToControl
 Control Name: Product Id
```

When the button is clicked, the focus will move to the Product Id control.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: McrActn

**INF: Setup Wizard Cannot Create Compact and Repair Icons**  
**Article ID: Q123586**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SUMMARY

=====

The Setup Wizard shipped with the Microsoft Access Developer's Toolkit (ADT) does not allow for the creation of Compact and Repair icons for a custom application. This article describes three methods of creating Compact and Repair icons for a custom application.

MORE INFORMATION

=====

The Setup Wizard allows only one set of parameters for a custom application. In almost all cases, the parameters are used to launch the custom application, and cannot therefore be used for a Compact or Repair icon. If you add the /compact and /repair parameters, the Setup Wizard will ignore them.

The following three methods describe how to work around this behavior.

Method 1

-----

NOTE: Method 1 is the only method in this article that is supported by Microsoft Product Support Services.

For information about how to create additional program groups and items with the Setup Wizard, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q119724  
TITLE : INF: Creating Additional ADT Application Programs Groups, Items

Method 2

-----

Download the file EDCSET.ZIP from Lib 16 of the MSACCESS forum on CompuServe. This file contains a modified version of the Setup Wizard. This version of the Setup Wizard was modified by a 3rd party and is not supported by Microsoft Product Support Services.

Method 3

-----

WARNING: ANY USE BY YOU OF METHOD 3 IS AT YOUR OWN RISK. Microsoft provides this information "as is" without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and/or fitness for a particular purpose.

Edit the SETUP.STF file on Disk 1 of the installation disks created by the Setup Wizard to create Compact and Repair icons. Follow these steps to use this method:

1. Make a backup copy of the SETUP.STF file on Disk 1 of the installation disks created by the Setup Wizard.
2. Open the SETUP.STF file in a spreadsheet application such as Microsoft Excel.
3. Near the bottom of the file, locate the text "AddProgManItem" in column E.
4. Create a new line by increasing the Object ID counter in column A by one.
5. Leave column B blank.
6. In column C, add the label "USER\_UserPM\_x" (without quotation marks) where x is one greater than any other USER\_UserPM\_y label above.
7. Leave column D blank.
8. In column E, enter the text "AddProgManItem" (without quotation marks).
9. In column F, add the /compact and /repair parameters. The second entry is the icon description, and must be changed to something other than the original entry (you cannot have two icons with the same description). The third entry is the command-line parameter for the icon (this is where you add the /compact and /repair parameters). The following line is an example of a valid line in column F.

NOTE: In the following sample line, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this line.

```
"My Application Group", "MyApp Maintenance", "%26\msarn200.exe _
%321\myapp.mdb /ini %323\myapp.ini /compact /repair", _
"%322\myapp.ico"
```

10. Find the label "---- User ProgMan Items ----" in column C. In that row, in column F, add the Object ID number for the new line you created to the end of the list of Object ID numbers.
11. In cell B16, increase the Maximum Object ID variable by one.
12. Save and then close the SETUP.STF file.

Additional reference words: 2.00

KBCategory: kbsetup

KBSubcategory: StpHowto



**INF: Setup Wizard OLE Default Dependent on MSARN200 Choice**  
**Article ID: Q123798**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SUMMARY

=====

When you are running the Microsoft Access Developer's Toolkit (ADT) Setup Wizard, the OLE 2.01 (or OLE 2.02 if you have installed the Microsoft Access version 2.0 Service Pack) option is selected by default only if you select the Run With MSARN200 check box in the Add File dialog box.

MORE INFORMATION

=====

If you do not choose to include the Microsoft Access run-time executable file MSARN200.EXE with your custom application, users of your application will need to have a retail version of Microsoft Access to run your application. The retail version of Microsoft Access includes the OLE dynamic-link library (DLL) files, so they do not need to be included by the Setup Wizard in your custom application's Setup.

If you do choose to include the run-time executable file with your custom application, the Setup Wizard will generate a warning if you choose not to include the OLE files. If you choose OK in response to this warning, a flag is set to prevent future warnings about the OLE files. This flag is saved with your template.

REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0, Chapter 2, "Creating a Custom Setup Program," pages 19-37

For information about how to obtain the Microsoft Access version 2.0 Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

Additional reference words: 2.00 runtime jet25 jet 2.5

KBCategory: kbenv

KBSubcategory: EvnOs

## **INF: Shortcut Method for Creating an SQL Union Query**

Article ID: Q114731

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article demonstrates a shortcut method for creating an SQL union query using a second "scratchpad" query to create the second half of the SQL UNION statement.

### MORE INFORMATION

=====

The following example demonstrates how to create a union query that returns all the customers and suppliers in London in the sample database NWIND.MDB:

1. Start Microsoft Access and open NWIND.MDB.
2. Create a new query and add the Customers table.
3. Add the Company Name, Contact Name, and City fields to the query grid.
4. In the Criteria row for the City column, enter "London" (without quotation marks).
5. From the View menu, choose SQL. This creates the first half of the union.
6. Leave the first query open and create a second new query. Add the Suppliers table. This will be the "scratchpad" query.
7. Repeat steps 3 and 4.
8. From the View menu, choose SQL. Select the entire SQL statement except for the ending semicolon (;).
9. Press CTRL+C to copy the selected text to the Clipboard.
10. Close the scratchpad query without saving it, and then switch back to the original query.
11. Position the insertion point before the semicolon (;) at the end of the SQL statement. Type the keyword "UNION" (without quotation marks) with a space before and after it.
12. Position the insertion point before the semicolon (;) at the end of the statement. Paste the text from the Clipboard by pressing CTRL+V. This creates the second half of the union.
13. Run the query. Note that the title bar indicates that it is a union query, and that the Design view button on the toolbar is disabled.

## Tips

----

To be able to tell which table the data comes from, switch back to the query's SQL view and add the table name to the end of each SELECT statement, as in the following example:

```
SELECT DISTINCTROW Customers.[Company Name], Customers.[Contact
Name], Customers.City, "Customers"
FROM Customers
WHERE ((Customers.City="London")) UNION SELECT DISTINCTROW
Suppliers.[Company Name], Suppliers.[Contact Name], Suppliers.City,
"Suppliers"
FROM Suppliers
WHERE ((Suppliers.City="London"));
```

When you run the query, a fourth column, "Expr1003," will be displayed. To give the new column a more meaningful name, add an alias to it. Add the alias to the first SELECT statement as follows:

```
SELECT DISTINCTROW Customers.[Company Name], Customers.[Contact
Name], Customers.City, "Customers" AS [Source Table]
FROM Customers
WHERE ((Customers.City="London")) UNION SELECT DISTINCTROW
Suppliers.[Company Name], Suppliers.[Contact Name], Suppliers.City,
"Suppliers"
FROM Suppliers
WHERE ((Suppliers.City="London"));
```

To sort your query, add an ORDER BY statement to the last SELECT clause in the UNION statement, as in the following example:

```
SELECT DISTINCTROW Customers.[Company Name], Customers.[Contact
Name], Customers.City, "Customers" AS [Source Table]
FROM Customers
WHERE ((Customers.City="London")) UNION SELECT DISTINCTROW
Suppliers.[Company Name], Suppliers.[Contact Name], Suppliers.City,
"Suppliers"
FROM Suppliers
WHERE ((Suppliers.City="London")) ORDER BY [Contact Name];
```

## REFERENCES

=====

For more information about union queries, search for "UNION (SQL)" then "UNION Operation (SQL)" using the Microsoft Access Help menu.

Microsoft Access "User's Guide," version 2.0, Chapter 12 "Advanced Queries," pages 288-291

Additional reference words: 2.00 Queries SQL-Specific Short Cut

KBCategory: kbusage

KBSubcategory: ObcOthr

**INF: Showing All Records (Including Null) in a Parameter Query**  
**Article ID: Q103181**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

When you are running a query that takes its parameters from a form, no records are returned by the query if you leave the field blank. If you type an asterisk (\*) in the field, only records with non-Null values are returned.

This article describes a method for returning all records, including those with Null values, when you leave the parameter blank.

MORE INFORMATION

=====

The following example is based on the sample database NWIND.MDB and assumes that you are familiar with Microsoft Access form, macro, and query design.

1. Create the following unbound form:

Form: Pick Employees

-----

Control: Textbox  
ControlName: Region  
Control: Command Button  
Caption: Run Query  
OnPush: Run Employee Query

Note: In Access 2.0 use OnClick instead of OnPush

2. Create the following macro:

Macro: Run Employee Query

-----

Action: OpenQuery  
Query Name: Employee Query  
View: Datasheet  
Data Mode: Edit

3. Create the following query based on the Employees table:

Query: Employee Query

-----

Field: First Name  
Show: True  
Field: Last Name  
Show: True  
Field: Region

Show: True  
Criteria: Like Forms![Pick Employees]!Region & "\*"   
Or: <leave blank>  
Field: Forms![Pick Employees]!Region  
Show: False  
Criteria: <leave blank>  
Or: Is Null

4. Open your query in Design view. From the Query menu, choose Parameters. Type "Forms![Pick Employees]!Region" (without the quotation marks) as the parameter name, with Text as the data type.
5. Open the Pick Employees form, type "WA" (without the quotation marks) in the Region field, and choose the Run Query button. The dynaset will contain five employee names.
6. Open the Pick Employees form, clear the Region field, and choose the Run Query button again. The dynaset now contains nine employee names, four with blank region codes.

By adding the parameter as a field, we can test the parameter and control the other criteria. The equivalent SQL Where condition is the following:

Where Region Like Forms![Pick Employees]!Region & "\*"   
Or Forms![Pick Employees]!Region Is Null

Additional reference words: 1.00 1.10 2.00 SQL Queries query  
KBCategory: kbusage  
KBSubcategory: QryParm

## **INF: Shrinking Empty or Null OLE Object on Report**

Article ID: Q90855

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### 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.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information about Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

To illustrate how this is done, we will use the sample database NWIND.MDB:

1. Start Microsoft 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. Create a new module with the following in the Declaration section:

Option Explicit

4. Create the following procedure:

```
Function HideNullOle (ctl As Control)
 On Error Resume Next
 ctl.visible = Not IsNull(ctl)
 If Err = 2427 Then ctl.visible = True
End Function
```

The function checks to see if the OLE Object is null, if it is the control is hidden. If it is not null, the control is not hidden. Some Ole Objects take longer to retrieve data and will generate the error # 2427, Object has no value. When this error is generated, the OLE Object is not null and the control is not hidden.

5. From the View menu, choose Report.
6. In the Database window, choose the New button.
7. When prompted, base the report on the Employees table and select Blank Report instead of the Report Wizard.

8. Drag Employee ID, Last Name, First Name, and Photo from the list of fields and place them in the detail section of the report.
9. If the property sheet is not showing, choose Properties from the View menu, or click the Properties button on the toolbar.
10. Click the detail section bar (the bar labeled "Detail"). On the property sheet, change the following properties:

OnFormat: =ShrinkNullOle([photo])

11. 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.
12. For this unbound text box, change the following properties:

Visible: NO  
CanShrink: YES

13. From the Layout menu, choose Send To Back. The OLE object should now be on top of the unbound text box.
14. Save the report as Report1.
15. 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.

Additional reference words: 1.00 1.10 2.00 container remove delete erase  
can  
KBCategory: kbusage  
KBSubcategory: RptProp

## INF: Sorting String Values Based on their Numeric Values

Article ID: Q96112

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

To preface this article, I have a database that contains a product code column. It is a text data type but I store numbers from 1 to 999 and each code can have a single letter for a suffix. For instance, I might have the following codes: 10a, 1d, 100b, 24c, 24a, 1, 89b, and 14. I want to view this data so that it is sorted on the code. It should appear like this: 1, 1d, 10a, 14, 24a, 24c, 89b, and 100b.

This article discusses one method for doing this. It requires basic knowledge of Microsoft Access queries and functions.

### MORE INFORMATION

=====

You need to use a query to view your data so that it is sorted on the code. This will work as long as the numbers do not exceed 1000 (if they do, you can add to this solution to accept larger numbers), there can be only one character digit after the number (that is, a, b, c; not aa, bc, cde), and numbers cannot have leading zeros (that is, 01, 02, 03, and so forth).

The query grid appears as follows:

Query: Query1

-----  
Field: ID

Show: Yes

Field: PRE: Val([ID])

Sort: Ascending

Field: SUF: IIF(Val([ID])<10, Mid([ID], 2, 1),  
IIF(Val([ID]) between 10 and 99,  
MID([ID], 3, 1), Mid([ID], 4, 1)))

Sort: Ascending

This separates the numeric value (PRE) and the string value (SUF) into two separate (temporary) fields in the query. The query sorts on the individual pieces but shows only the true ID field (assuming your field name is ID).

Additional reference words: 1.00 1.10 2.00 alphanumeric sort

KBCategory: kbusage

KBSubcategory: QryOthr



## **INF: Sorting Titles Without Leading Articles (The, An, etc.)**

Article ID: Q98665

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article explains how to sort titles in text fields without including the leading articles "the," "an," or "a."

### MORE INFORMATION

=====

The following is a list of book titles:

Aztec Indians  
The Ant People  
Beast Man  
The Baseball Club  
An Attempt At Fun  
Apple Valley

If you sort the above list using a custom Microsoft Access function in a query, the following new list results:

The Ant People  
Apple Valley  
An Attempt At Fun  
Aztec Indians  
The Baseball Club  
Beast Man

Create the following Access Basic function that returns a book title without the leading article:

Option Explicit

Function Gettitle (Titles) As String

    sp = InStr(1, Titles, Chr\$(32))

    If sp > 0 Then

        Select Case Left(Titles, sp - 1)

            Case "An", "A", "The"     ' add any other articles here!

                Gettitle = Mid(Titles, sp + 1)

            Case Else

                Gettitle = Titles

        End Select

    Else

        Gettitle = Titles

    End If

End Function

To sort a table on the Titles field, create a query with the following

two columns in the query grid:

```
Query

Field: Sorted List: Gettitle([Titles])
 Sort: Ascending
 Show: Yes
Field: [Titles]
 Show: No
```

The first column is a calculated field that extracts the text of the Titles field, but omits the leading article. The results of this calculation are sorted in ascending order. The second Show check box is cleared so that the query result is not displayed.

#### REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 276-277 and 404

Additional reference words: 1.00 1.10 2.00 titles sort strip left

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: Sorting/Grouping a Query Using a Portion of Field's Value**  
**Article ID: Q88927**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can sort or group a query using only a portion of a field's value. For example, in the sample database NWIND.MDB, you can create a query based on the Employees table that uses the Mid() function to sort by six characters, beginning with the fourth character in the Last Name field. You can use the Left() and Right() functions to return the first and last characters of a string argument.

MORE INFORMATION

=====

The following procedure explains how to use the Mid() function to sort data using only a portion of a field's value:

1. Enter the following in the Field row in the first column of the query grid:

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

2. Select the Sort cell in the Last Name column and choose Ascending or Descending.
3. Choose the Totals button on the toolbar. Group By now appears in the Totals row of the query grid.
4. Switch to Datasheet view. "Sort String" appears as the name of the column, but you can change this header to any other name.

Note that the employee records are sorted by the fourth letters of the employees' last names.

This technique is useful if a field contains values with a set number of leading or trailing characters that are insignificant to your sorting task. An example of this is a part number field called PartNo that contains the following values:

```
PartNo

A-453-34567
A-123-45675
B-234-75658
B-645-65759 and so forth
```

To sort these part numbers by the third through fifth characters only, use the following expression:

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

#### REFERENCES

=====

For more information on string functions and related functions, search for "String," "Left," "Right," and "Len" using the Help menu.

Additional reference words: 1.00 1.10 2.00 string function

KBCategory: kbusage

KBSubcategory: QryOthr

## **INF: SQL Server Views Read-Only When Attached**

**Article ID: Q98786**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft SQL Server versions 1.x and 4.2
- 

Microsoft SQL Server supports updatable views. Microsoft Access can only attach these views as read-only because SQL Server views do not have indexes. Microsoft Access requires a unique index on the attached object for update/delete/insert capability.

Please refer to Knowledge Base article Q90100 for more information on Microsoft Access and updating attached SQL Server tables.

Additional reference words: 1.00 1.10 2.00 SQL Server attach table view

KBCategory: kbusage

KBSubcategory: ObcOthr

**INF: Storing Calculated Values Using ControlSource Property**  
**Article ID: Q109704**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes the ControlSource property as it applies to controls, and demonstrates how to use the SetValue macro action to store calculated values in a control's underlying table.

MORE INFORMATION

=====

A control can be classified as bound, unbound, or calculated depending on the contents of its ControlSource property. A bound control on a form or report is linked to a field in an underlying table or query. This type of control is used to display, enter, and update data in a field. For example, the following control is a bound control:

Name: Last Name  
ControlSource: Last Name

NOTE: In Microsoft Access version 1.x, the Name property is called the ControlName property.

This control displays the data from the Last Name field, and stores any changes made in the control to the Last Name field in the form's underlying table.

A calculated control displays a value derived from data in one or more fields from the underlying table or query or from other controls. The calculation is the result of an expression assigned to the ControlSource property for that control. A calculated control can display data that is the result of an expression but cannot save data. For example, the following control is a calculated control:

Name: MyControl  
ControlSource: =[Last Name] & ", " & [First Name]

The MyControl control displays a full name derived from an expression concatenating the last name and first name. The full name is not stored in any table. Calculated controls are often used to display calculations on a form that do not need to be stored in the underlying table.

The ControlSource property for an unbound control is blank, meaning that it is not connected to either a field from the underlying table or query or an expression. An unbound control can display data, but cannot save data. The data contained in an unbound control is temporary and is lost when the form or report is closed.

Storing Calculations in a Field

-----  
The following example demonstrates how to use the SetValue macro action to store calculated values in a control's underlying table.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the sample database NWIND.MDB.
2. Open the Order Details table in Design view.
3. Add the following field to the table:

Field Name: Total  
Data Type: Currency

4. Save and then close the table.
5. Create a new, blank form based on the Order Details table.
6. From the View menu, choose Field List.
7. Drag the Quantity, Unit Price, and Total fields from the field list to the form. These are bound controls on the form. Note that each control's ControlSource property contains the name of the control's underlying field.
8. Create the following new macro and then save it as Test1:

| Macro Name | Action   |
|------------|----------|
| -----      | -----    |
| Test1      | SetValue |

Test1 Actions

-----  
SetValue  
Item: [Total]  
Expression: Ccur([Unit Price]\*[Quantity])

Note that this macro is equivalent to the expression:

[Total]= Ccur([Unit Price]\*[Quantity])

When this macro runs, it will fill the Total field with the calculation's results.

9. Set the form's OnCurrent property to the Test1 macro. This causes the macro to run when the form is opened, and every time you move from one record to another.
10. Set the Quantity text box's AfterUpdate property to the Test1 macro. This causes the Test1 macro to run whenever the contents of the Quantity control are modified.
11. Set the Unit Price text box's AfterUpdate property to the Test1

macro. This causes the Test1 macro to run whenever the contents of the Unit Price control are modified.

12. View the form in Form view.

Note that the Test1 macro runs every time you move from one record to another, or whenever you modify a value in the Quantity or Unit Price fields.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 15, "Designing Forms," page 342, and Chapter 25, "Using Macros with Forms," page 642

For more information about the ControlSource property, search for "ControlSource" then "ControlSource Property" using the Microsoft Access Help menu.

For more information about binding controls to fields, search for "binding controls" then "Binding a Control to a Field" using the Microsoft Access Help menu.

For more information about creating calculated controls, search for "calculated controls" then "Creating a Calculated Control on a Form or Report" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsHowto



**INF: Storing SQL Database Login IDs and Passwords Locally**  
**Article ID: Q101084**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Storing SQL database login IDs and passwords locally requires that the database administrator create a unique table on the server.

MORE INFORMATION

=====

In Microsoft Access version 1.0, if you attach a SQL database table, you can choose whether you want Microsoft Access to store your login ID and password locally. If you do not, Microsoft Access prompts you for your login ID and password each time you connect to the SQL database containing the table.

If you want Microsoft Access to store the connection information in your Microsoft Access database so you do not have to type it each time, you can select the Save Login ID and Password Locally check box in the Attach Tables dialog box when you attach the SQL database table.

This feature is also present in Microsoft Access version 1.1. For Microsoft SQL Server, Sybase SQL Server, and ORACLE Server databases, however, your SQL database administrator can now choose to disable this feature, requiring all users to enter their login IDs and passwords each time they connect to a SQL database.

To disable the ability to store login IDs and passwords locally, your SQL database administrator must create a table called MSysConf in the SQL database. When a user connects to the SQL database, Microsoft Access looks for this table in the database and, if it finds it, queries the table. If the values in the table correctly specify that local storing of login IDs and passwords should be disabled, Microsoft Access does so regardless of whether the Save Login ID and Password Locally check box is selected. If the table is not present or does not specify disabling of the feature, users can store login IDs and passwords locally.

The SQL database table MSysConf should have the following structure.

| Column name | Data type                                        | Allows Null? |
|-------------|--------------------------------------------------|--------------|
| Config      | A data type that corresponds to a 2-byte integer | No           |
| chValue     | VARCHAR(255)                                     | Yes          |
| nValue      | A data type that corresponds to a 4-byte integer | Yes          |



**INF: Summary List of Known Issues for Version 2.0**

**Article ID: Q118369**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article contains a list of confirmed, documented problems with Microsoft Access version 2.0 as of July 14, 1994. This list contains identification numbers (Q numbers), titles, and brief summaries from the articles in the Microsoft Knowledge Base that fully document the problems.

MORE INFORMATION

=====

Q115877 PRA: RunningSum Text Box in Report Not Accumulating Values

When a text box in a report that uses the RunningSum property is based on another control in the report, it does not accumulate the running sum. Instead, it reflects only the value of the control it references.

-----  
Q115124 PRA: Mailing Label Wizard Creates One-Page Report

A mailing label report created with the Mailing Label Wizard contains only one page, no matter how many records are in the underlying table or query. This problem occurs if your default printer is a laser printer, and you are using any of the following Avery mailing labels:

4145, 4162, 4163, 4249, 4250, 4251, 4253, 4254, or 5615

-----  
Q116144 PRA: Find Does Not Locate Existing Item in dBASE Database

When you search for a string in an attached dBASE database, the string is not found, even though the string does exist in the dBASE database.

-----  
Q117169 PRB: Installing MS Access with NT Creates New Program Group

When you install Microsoft Access version 2.0 on a computer running Microsoft Windows NT version 3.1, and you choose to install the Microsoft Access icons in an existing common group such as Applications (Common), Microsoft Access Setup creates a new group called Applications (Common) (Common) and places the Microsoft Access icons in this group, instead of in the group you chose.

-----  
Q113562 PRA: Report Section with Subreport Not Included In Output File

When you use the Output To command to output a report to a file, any

section of a report that includes a subreport is not included in the output file.

---

Q117168 PRA: Ownership of 1.x Objects Can Be Changed with 2.0 DAO

You can use data access objects (DAO) in Microsoft Access version 2.0 to change the ownership of database objects (such as tables, forms, and reports) in unprotected Microsoft Access version 1.x databases, even though you cannot view or change the ownership of these objects using Microsoft Access version 1.x.

---

Q109313 PRB: Changing Show Ruler Option Changes Existing Forms

Changing the Show Ruler option in the Form & Report Design section of the Options dialog box overrides this setting in existing forms and reports.

---

Q114514 PRA: Changes to Print Setup Not Saved with Form or Report

After you change your form or report to print in a different paper orientation (such as landscape), it does not print in the orientation you chose. When you check the Print Setup dialog box, you find that none of the changes you made were saved.

---

Q113929 PRA: Code Called from OnFormat Creates Endless Formatting

A report that calls code from its OnFormat property may not ever finish formatting, becoming stuck in an internal page formatting loop. This problem may be more common in databases that have been converted from Microsoft Access version 1.x to version 2.0.

---

Q116061 PRA: Extra White Space Between Last Section and Report Footer

Extra white space is printed on your report after the last detail record or section of the report, before the report footer section. The extra white space is as tall as the last section printed before the report footer. Note that this problem will not occur on reports that do not have a report footer section.

---

Q117609 PRA: Paradox Password Protection Bypassed by Multiple Users

When you attach a password-protected Paradox table nonexclusively to a Microsoft Access version 2.0 database, the table behaves normally, allowing you to view the table's contents after you supply the correct password. However, once the password has been supplied by a user, any other user can then open the attached table and view its contents without supplying a password.

---

Q117614 PRA: Error Adding Zero-Length String to SQL NOT NULL Column

When you attach a SQL Server table that was created with a NOT NULL restriction on a column, then try to enter a zero-length string in the table, you receive the error message:

```
[Microsoft][ODBC SQL Server Driver][SQL Server] The column <column name> in table <table name> may not be null. (#233)
```

---

Q117616 PRA: Error Deleting Value from Attached SQL Server Table

When you attach an SQL Server table that has a column with a NOT NULL restriction to Microsoft Access, add a value to the restricted column in the attached table, then delete the value and attempt to commit the record, you receive the error message:

```
[Microsoft][ODBC SQL Server Driver][SQL Server] The column <column name> in table <table name> may not be null. (#233)
```

---

Q117610 PRA: GPF Closing Data Outline Control Properties Box

When you add a data outline control to a form or report, then edit the control's properties, the Properties dialog box is normally modal, meaning that you cannot select anything outside the dialog box.

If, however, you click the property sheet for the form or report containing the data outline control, the property sheet gets the focus, and the modality of the data outline control's Properties dialog box is lost. You can then select anything in the database.

When the modality of the control's Properties dialog box is broken as outlined above, your mouse pointer may disappear, and you may receive a general protection (GP) fault.

---

Q117612 PRA: TransferDatabase Fails in Code, Not in Immediate Window

When you use the TransferDatabase action within a transaction in an Access Basic function, you receive the error message:

```
Couldn't update, locked by another user on this system.
```

However, if you set a breakpoint in the function and single-step through it, you do not receive the error message.

---

Q117615 PRA: GPF Using List Box Fill Function with Requery Method

When you fill a list box using an Access Basic function, and the function returns a static variable upon receiving the close code (8), and you then use the Requery method to requery the list box, you receive a general protection (GP) fault.

---

Q114549 PRA: Empty Recordset Report Hangs with Group KeepTogether

Microsoft Access stops while you are previewing a report, and may mark the database as corrupt. Running Repair Database on the database will eliminate the error message that says that the database is corrupt.

---

Q117611 PRA: GPF Using GetChunk Method as Argument for Put Statement

When you use the GetChunk method in the third parameter of a Put statement, you get a general protection (GP) fault.

---

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: GnlPrfm

## **INF: Three Methods for Testing the VBCL**

**Article ID: Q114679**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

### SUMMARY

=====

Once you have installed the Visual Basic Compatibility Layer (VBCL), there are several test methods you can use in Visual Basic version 3.0 to verify that it is working correctly:

- Using the Data Manager window. (This is the simplest method.)
- Using the Data Access control.
- Using the Visual Data (VisData) sample application (Visual Basic Professional Edition only).

### MORE INFORMATION

=====

#### Using The Data Manager Window

-----

1. Start Visual Basic and choose Data Manager from the Window menu.
2. From the File menu, choose Open Database, then choose Microsoft Access.
3. Select the sample database NWIND.MDB, and then choose OK.

If the VBCL is installed correctly, you will see a list of all the tables in NWIND.MDB. If the VBCL is not installed, or is not installed correctly, you will receive the error message "Incompatible database version."

#### Using the Data Access Control

-----

1. Start Visual Basic and place a data access control and a text box control on a new form.
2. Set the following properties for the data access control:

    DatabaseName: C:\ACCESS\SAMPAPPS\NWIND.MDB  
    RecordSource: Employees

3. Set the following properties for the text box control:

    DataSource: Data1  
    DataField: Last Name

4. From the Run menu, choose Start.

If the VBCL is installed correctly, you will see the name "Davolio" in the text box. Also, you will be able to step back and forth through the names using the arrows on either side of the data access control.

If the VBCL is not installed, or is not installed correctly, you will receive the error message "Incompatible database version."

Using the Visual Data (VisData) Sample Application  
-----

1. Start Visual Basic and open the VISDATA.MAK file in the SAMPLES\VISDATA subdirectory.
2. From the Run menu, choose Start.
3. From the File menu, choose Open Database, then choose Microsoft Access.
4. Select NWIND.MDB.

If the VBCL is installed correctly, you will see a list of tables and queries from the NWIND database. Otherwise, you will receive the error message:

```
Incompatible database version
Number: 3041
```

To test the VBCL further, enter the following statement in the SQL window:

```
SELECT * FROM Employees
```

If the VBCL is installed correctly, you will see a window titled "Dynaset: Employees" with the first employee's data showing.

Additional reference words: 2.00 adt attaching testing  
KBCategory: kbinterop  
KBSubcategory: IntpVb



## **INF: Tips for Debugging Access Basic Code**

**Article ID: Q108438**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article contains tips to help you debug your Access Basic code. This article assumes that your code compiles correctly, but does not perform as you expect.

### MORE INFORMATION

=====

The general procedure for debugging Access Basic code is to break larger procedures apart into smaller sub-procedures, verify that the individual steps in each sub-procedure work correctly, and then combine the sub-procedures one by one until the code works correctly.

#### Using The Immediate Window

-----

You can use the Immediate window to execute individual lines of Access Basic code, or to check the values of variables. Therefore, you can use the Immediate window to:

- Test and debug Function and Sub procedures.
- Check the value of a field, control, property setting, variable, or expression.
- Display the result of an expression when the code is running.

The following steps demonstrate how to use the Immediate window to debug your Access Basic functions:

1. Open the module containing your function in Design view.
2. From the View menu, choose Immediate Window.
3. Run your function in the Immediate window by typing "?<FunctionName>()" (without the quotation marks) and then pressing ENTER. Substitute the name of your function for <FunctionName>, and be sure to place all arguments expected by your function inside the parentheses.

For more information on the Immediate window, search for "immediate window" then "Immediate Window" using the Microsoft Access Help menu.

#### Using Breakpoints with the Immediate Window

-----

A breakpoint suspends execution of the code, allowing you to check the

values of variables. You can use the Immediate window with breakpoints to pinpoint code problems quickly.

For example, if the first part of your code runs correctly, but other parts do not, you can follow the sample steps below to find the malfunctioning part:

1. Set a breakpoint right after the part that you know works correctly. To set a breakpoint:
  - a. Move your cursor to the line of code where you want to set the breakpoint.
  - b. Click the Breakpoint button (the five-fingered hand) on the toolbar. Or, press F9, or choose Toggle Breakpoint from the Run menu. The line will appear in bold type, indicating that there is a breakpoint set at that line.
2. Run the function in the Immediate window (see "Using The Immediate Window" above). When code execution reaches the breakpoint, processing is suspended and the breakpoint line is selected. You can then check the value of variables in the code.

For example, to check the value of a variable called MyName, type the following in the Immediate window and then press ENTER:

```
?MyName
```

The value of MyName is displayed in the Immediate window. If the value is correct, check other variables. If the value is incorrect, check the previous lines of code to see where this variable is assigned an incorrect value.

#### Using the Debug.Print Statement

-----

You can use the Debug.Print statement in your code to display the values of your variables in the Immediate window. Good places to include Debug.Print statements include:

- The beginnings of functions, to check the arguments passed to the function. For example, to check the values of two arguments passed to the DoSomeCalcs() function, place the following sample Debug.Print statement at the beginning of the function:

```
Function DoSomeCalcs(Arg1, Arg2)
 Debug.Print "Function DoSomeCalcs " & Arg1 & " " & Arg2
```

When the DoSomeCalcs() function is run, the text "Function DoSomeCalcs" and the values of the two arguments Arg1 and Arg2 will be displayed in the Immediate window. If the values of the arguments are not correct, the problem is occurring before the function is run.

You can also use the Debug.Print statement at the beginning of each function if you have a complex application and you are not sure which function might be causing the problem. This allows you to check the Immediate window to see the order in which functions were called, and

determine which was the last function called. In Access 2.0, you can also select Calls from the View menu to list the current function call stack, and show the order that functions were called.

- Decision structures, to check that you are using the correct logic for the function. For example, the sample code on page 23 of the Microsoft Access "Introduction to Programming" manual uses a Select Case statement. You can add a Debug.Print statement to this sample code, as demonstrated below, to verify that the code is doing what you expect:

```
Function DueDate (anyDate)
Debug.Print "Function DueDate " & anyDate
Dim Result
 If Not IsNull(anyDate) Then
 Result = DateSerial(Year(anyDate), Month(anyDate) + 1, 1)
 Debug.Print "Result: " & Result
 Debug.Print "Weekday(Result): " & Weekday(Result)
 Select Case Weekday(Result)
 Case 1: 'Sunday
 Debug.Print "Case 1"
 DueDate = Result + 1
 Case 7: 'Saturday
 Debug.Print "Case 7"
 DueDate = Result + 2
 Case 6: 'Friday
 Debug.Print "Case 6"
 DueDate = Result - 1
 Case Else
 Debug.Print "Case Else"
 DueDate = Result
 End Select
 Else
 Result = Null
 End If
End Function
```

Run this function in the Immediate window by typing the following and then pressing ENTER:

```
?DueDate(#10/1/93#)
```

The following results will be displayed in the Immediate window:

```
Function DueDate 10/1/93
Result: 11/1/93
Weekday(Result): 2
Case Else
11/1/93
```

These results show that you are using the correct logic for this function. If you had received different results (the wrong case, for example) you could then check the values of other values and expressions to determine where the problem is.

- In loops, to check the values of variables. If you use a breakpoint to check the value of a variable in a loop, you cannot see how the

value of the variable changes as the loop executes. However, if you include the statement "Debug.Print MyName" (without the quotation marks) in your loop structure, the value of variable MyName will be displayed in the Immediate window each time the function cycles through the loop.

This method is also a good way to verify that the loop is executing the number of times you expect. If the loop executes four times, you should see four values for the value in the Immediate window.

- Where clauses, to check the values of criteria in SQL statements. For example, the following sample function creates a dynaset from a SQL statement. If there is a problem in a SQL statement (such as in this function), it can be difficult to locate the problem. However, this function uses the Debug.Print statement to display the SQL statement as Microsoft Access sees it and uses it:

```
Function TestMe()
 Dim db As Database, ds As Dynaset
 Dim empnum As Long
 Dim sql As String
 Set db = CurrentDB()
 empnum = 5
 sql = "select * from orders where [employee id]=empnum;"
 Debug.Print sql
 Set ds = db.CreateDynaset(sql)
End Function
```

Run this function in the Immediate window by typing the following and then pressing ENTER:

```
?TestMe()
```

The following result is displayed:

```
select * from orders where [employee id]=empnum;
```

Note that the Where condition shows [employee id] = empnum, not [employee id] = 5, as you assigned it. To fix this particular problem, change the Where condition to concatenate the empnum variable, as follows:

```
Function testme()
 Dim db As Database, ds As Dynaset
 Dim empnum As Long
 Dim sql As String
 Set db = CurrentDB()
 empnum = 5
 sql = "select * from orders where [employee id]=" & empnum & ";"
 Debug.Print sql
 Set ds = db.CreateDynaset(sql)
End Function
```

When you run the corrected function, the following statement will be displayed:

```
select * from orders where [employee id]=5;
```

## Stepping Through the Code

---

Stepping through your code means executing the code one line at a time. This is sometimes referred to as tracing. Access Basic has two ways of stepping through code: single stepping, and procedure stepping.

Single stepping executes each step of a called procedure, stopping at each line in the procedure. Procedure stepping executes a called procedure all at once, then returns to the next line in the current procedure.

To step through your code:

1. Set a breakpoint in your code at the line where you want to begin stepping. (See "Using Breakpoints with the Immediate Window", above). To step through an entire function, set the breakpoint on the first line of the function.
2. Start the function. When program execution reaches the breakpoint, the line with the breakpoint will be selected in the function's module.
3. To single step, click the Single Step button (single foot) on the toolbar. Or, press F8, or choose Single Step from the Run menu. The selected line of code will be executed, and then the next line of code will be selected. You can continue through your code one line at a time in this fashion.

If your code calls other functions or procedures you will single step through them as well. If you do not want to single step through these other procedures, click the Procedure Step button (double foot) on the toolbar. Or, press SHIFT+F8, or choose Procedure Step from the Run menu.

Note that as you step through your code, you can check the values of variables in the Immediate window (see "Using the Immediate Window", above).

If you want to continue running your code normally (without single or procedure stepping), click the Run button (exclamation point) on the toolbar. Or, press F5, or choose Continue from the Run menu.

## REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.x, Chapter 4, "Debugging Your Access Basic Code"

Microsoft Access "Building Applications," version 2.0, Chapter 9, "Debugging"

Additional reference words: 1.00 1.10 2.00 AB trouble shoot

KBCategory: kbprg

KBSubcategory: PgmOthr

## **INF: Tips for Improving Combo and List Box Performance**

**Article ID: Q112745**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article lists several things you can do to improve the speed and performance of combo and list boxes in your Microsoft Access version 2.0 applications.

### MORE INFORMATION

=====

To improve combo and list box performance:

- Include only fields from the record source that are absolutely necessary. Extra fields can decrease combo or list box performance.
- Index the first field that is displayed in the combo or list box.
- Index any other fields used for criteria (such as when a combo or list box is based on a criteria query).
- In combo boxes, set the AutoExpand property to No if it is not needed.
- Do not hide the combo box's bound column by setting its width to 0 in the ColumnWidth property.
- Create a default value for combo boxes. Combo boxes try to match whatever is entered in them, so if there is no default value for the combo box, the combo box tries to match a null value when it is first opened. An example of a default value for a combo box is:

```
= [combobox].ItemData(0)
```

In the example above, [combobox] is the name of the combo box, and ItemData(0) refers to the first row of the combo box. The default value of the combo box becomes the value in its first row.

- Use unbound subforms to display data when there is a large number of records.
- The first non-hidden column in your combo box should have a text data type, not numeric. In order to find a match in the list, Microsoft Access will have to convert the numeric value to text to do the character by character match. If the data type is text, Microsoft Access does not have to do this conversion.

Additional reference words: 2.00 optimize

KBCategory: kbusage

KBSubcategory: FmsCmbo



## **INF: Tips for Improving Subform Performance**

**Article ID: Q112747**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article lists several things you can do to improve the speed and performance of subforms in your Microsoft Access version 2.0 applications.

### MORE INFORMATION

=====

To improve subform performance:

- If you can, base your subforms on queries rather than tables. Include only fields from the record source that are absolutely necessary. Extra fields can decrease subform performance.
- Index all the fields on the subform that are linked to the main form. Indexes help speed the search process to find the matching subform records.
- Index any fields used for criteria (such as when a subform is based on a criteria query).
- If you are linking on multiple fields, add a calculated field to the main form that concatenates the fields. Then, create a calculated column in the subform's RecordSource property query with the same expression. For example, to link to the subform on an Employee ID field and an Order ID field, create a text box on the main form with the following properties:

```
Name: EmployeeIDOrderID
ControlSource: =[Employee ID] & [Order ID]
```

Next, add the following field to the query that the subform is based on:

```
EmployeeIDOrderID: [Employee ID] & [Order ID]
```

Then, link the main form and the subform on the concatenated field rather than on the two individual fields. The subform properties might look like:

```
LinkChildFields: EmployeeIDOrderID
LinkMasterFields: EmployeeIDOrderID
```

Since Microsoft Access only has to compare one criteria to return the subform's recordset, the subform's performance should be improved.

- Set the subform's DefaultEditing property to ReadOnly if the records



in the subform are not going to be edited.

#### REFERENCES

=====

For general performance and indexing recommendations, search for "performance" and "Rushmore technology" using the Microsoft Access Help menu.

Additional reference words: 2.00 Forms Subforms Speeding Slow

KBCategory: kbusage

KBSubcategory: FmsSubf

**INF: Tips for Optimizing Queries on Attached SQL Tables**  
**Article ID: Q99321**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Special considerations must be made for performance optimization of queries built on attached SQL database tables. An SQL database, for this article, is defined as any client/server database that supports some level of ANSI SQL as an intrinsic part of the database's programming language. Optimal query strategy for these queries is to ensure that all query operations are performed on the server. This article outlines tips on how to ensure that queries against attached SQL database tables are performed on the server.

This article assumes a basic understanding of client/server computing environments and architectures.

MORE INFORMATION

=====

The key to improving query performance on attached SQL database tables is to ensure that no data is filtered on the client. Filtering data on the client increases network traffic and does not allow for leveraging of advanced server hardware, essentially turning a client/server system into a file server system. To this end, keeping the query evaluation on the server reduces overhead and keeps an application running as fast as possible.

Generic query optimization techniques should not be ignored when you are using attached SQL database tables. "WHERE clause" restrictions, such as equality or range comparisons, and sorting should still be performed on indexed fields. For more information on query optimization, call Microsoft Sales Information Center at 800) 426-9400 to order a "white paper" on SQL Server optimization. Other SQL database vendors may provide the same information for their customers.

Use care when implementing intrinsic or user defined functions (UDFs) in query fields or when using criteria that are not supported on the server. Generally, SQL databases have functionality that corresponds to most standard Microsoft Access functions, but each server will be different.

Many intrinsic Microsoft Access functions have direct back-end correspondents. Microsoft Access asks the ODBC driver about intrinsic function support and performs the appropriate mappings.

You can use UDFs and Microsoft Access intrinsic functions without server equivalents when they are accompanied by server-capable restrictions that restrict the data. For example, the following query

Query1a: SELECT \* FROM MillionRowTable WHERE Funk1(coll) = 10

returns the whole table and evaluates Funk1(coll) = 10 locally, whereas the following query

Query1b: SELECT \* FROM MillionRowTable WHERE Funk1(coll) = 10  
AND LastName BETWEEN 'g' AND 'h'

sends the BETWEEN 'g' AND 'h' restriction to the server, returns the qualifying rows, and evaluates Funk1(coll) on only those rows.

Non-remote-capable SELECT list items do not force a query to be executed locally, unless they are used with unique values or a totals query (DISTINCT/GROUP BY). For example, the following query

Query2a: SELECT \* FROM MyTable WHERE Format(coll, ...) = 10

returns the whole table and causes the WHERE clause to be evaluated locally. However, the following query

Query2b: SELECT Format(coll,...) FROM MyTable WHERE col2 = 10

sends "SELECT col1 FROM MyTable WHERE col2 = 10" to the server, presumably returning far less data over the network. It then locally evaluates Format() on the coll values returned.

Of the following two queries, Query3a is sent completely to the server. Query3b sends "SELECT col1 FROM MyTable" and performs the Format() function, and therefore the DISTINCT clause, locally.

Query3a: SELECT DISTINCT col1 FROM MyTable  
Query3b: SELECT DISTINCT Format(coll,...) FROM MyTable

The following two queries are performed as follows: Query4a is sent completely to the server. Query4b sends "SELECT col1 FROM MyTable" and performs the StdDev() aggregate function locally, since it's not a SQL standard function.

Query4a: SELECT Sum(coll) FROM MyTable  
Query4b: SELECT StdDev(coll) FROM MyTable

Crosstab queries present unique restrictions, some pertinent to all queries, some to crosstab queries only:

1. Only standard aggregate functions, such as Count(), Sum(), Min(), Max(), and Avg(), can be used.
2. Aggregate functions cannot be used as row or column headers.
3. Only one aggregate function can be used in the "value."
4. Nothing can be sorted.
5. If a fixed-value list of column headers is supplied, the value cannot be embedded in an expression.

Restriction #1 applies to all queries against SQL databases from

Microsoft Access. Except for #4, the others are uncommon and are crosstab/SQL database specific. Crosstab queries are not SQL standard, and are not generally supported on SQL database servers. However, if the rules above are not violated (which is the case for most simple crosstab queries), Microsoft Access can reformulate the query as a standard GROUP BY query, send it remotely, return only the aggregation result, and transform it into a crosstab locally.

Open-ended restrictions do not use indexes on SQL databases. Typically, a SQL WHERE clause reading

```
WHERE coll > 1000
```

will be slower than

```
WHERE coll between 1000 and 1000000000
```

This is a server problem, not a Microsoft Access problem, but it can affect Microsoft Access performance when a SQL database is used as a back end.

When you are using wildcard characters, take special care to ensure that the correct wildcards are used. SQL Server supports the use of % and \_ rather than ? and \* for wildcards. For more information, query on the following words in the Microsoft Knowledge Base:

```
parameter and queries and wildcards and SQL
```

Additional reference words: 1.00 1.10 2.00 SQL server attach query performance

KBCategory: kbusage

KBSubcategory: ObcOthr

**INF: Tips on How to Troubleshoot Microsoft Access Macros**  
**Article ID: Q89610**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

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:

- 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.
- 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.
- 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.
- 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.
- Immediate Window - This tool can be used to display any values that have been declared. The Immediate window is associated with module objects. To display the Immediate window, open an existing module, or create a new module, then choose Immediate window from the View menu. To display a value, type a question mark, then the variable name, and then press the ENTER key.

For information on displaying the Immediate window without displaying the Module window, search on the following words in the Microsoft Knowledge Base:

immediate and findwindow and showwindow

REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00 debugging

KBCategory: kbusage

KBSubcategory: McrOthr

**INF: Topics Supported by Microsoft Access as a DDE Server**  
**Article ID: Q89586**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Microsoft Access supports dynamic data exchange (DDE) as both a client (destination) and a server (source) application. This article lists the DDE topics that Microsoft Access supports as a DDE server, and the valid DDE items for each topic.

Notes

-----

- In this article, an asterisk (\*) indicates a feature that is not available in Microsoft Access version 1.0.
- All the examples below require that you have the sample database NWIND.MDB open.

MORE INFORMATION

=====

As a DDE server, Microsoft Access supports the following topics:

- The System topic
- The name of a database, table, or query
- A Microsoft Access SQL statement

The System Topic

-----

The System topic is a standard topic for all Microsoft Windows-based DDE server applications, and returns information about the topics supported by the application. The System topic supports the following Microsoft Access data items:

- SysItems: A list of items supported by the System topic in Microsoft Access
- Formats: A list of formats Microsoft Access can copy to the Clipboard
- Status: "Busy" or "Ready"
- Topics: A list of all open databases

Note that any information returned from any item used with the System topic is tab delimited.

The following sample Microsoft Word for Windows WordBasic macro demonstrates how to use the System topic to get information on available topics from Microsoft Access:

```
Chan = DDEInitiate("MSAccess", "System")
Topics$ = DDERequest$(Chan, "Topics")
DDETerminate Chan
MsgBox Topics$, "Topics", 64
```

#### The Database Topic

---

The database topic is the filename of an existing database. After you initiate a DDE conversation with the database, you can request a list of objects in that database. This list of information is tab delimited. Note that you cannot query the SYSTEM.MDB database using DDE.

The database topic supports the following items:

- TableList
- QueryList
- FormList
- ReportList
- MacroList
- ModuleList

The following sample Word for Windows WordBasic macro demonstrates how to retrieve a list of table names in the NWIND database:

```
Chan = DDEInitiate("MSAccess", "NWIND")
TableList$ = DDERequest$(Chan, "TableList")
DDETerminate Chan
```

#### The TABLE TableName, QUERY QueryName, and SQL SQLString Topics

---

The TABLE, QUERY, and SQL DDE topics are used to retrieve information from Microsoft Access tables. The returned information is tab delimited. The syntax for these topics is as follows:

<DatabaseName>; TABLE <TableName>

<DatabaseName>; QUERY <QueryName>

<DatabaseName>; SQL <SQLString>

#### Description of Syntax:

<DatabaseName> The name of the database to which the table or query belongs or the SQL statement applies, followed by a semicolon (;). The database name can be either the base name only (for example, NWIND), or its full path and .MDB extension (for example, C:\ACCESS\NWIND.MDB).

<TableName> The name of an existing table.

<QueryName> The name of an existing query.



<SQLString> A valid SQL SELECT statement of up to 255 characters, ending with a semicolon (;). Note that you can exchange more than 255 characters by omitting the <SQLString> statement and using successive DDEPoke() statements to build an SQL statement. For example, the following WordBasic code uses the DDEPoke() function to build an SQL statement and request the results of the query:

```
Chan = DDEInitiate("MSAccess", "NWIND;SQL")
DDEPoke Chan, "SQLText", "SELECT * FROM Orders "
DDEPoke Chan, "SQLText", "WHERE [Order Amount] > 1000;"
Results = DDERequest$(Chan, "Data")
DDETerminate Chan
```

Note that you cannot initiate a DDE link with Microsoft Access using an action query. Action queries, such as Append (INSERT INTO), Update (UPDATE...SET), Delete (DELETE...FROM), and Make-table (SELECT...INTO) perform an action on data. A Select query, which finds and returns data, can be used as the topic of a DDE conversation.

\*You can perform SQL action queries using DDE by running the RunSQL macro action using the DDEExecute() function.

The following is a list of valid DDE items for the TABLE TableName, QUERY QueryName, and SQL SQLString DDE topics:

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| All          | All data in the table, including field names.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Data         | All rows of data, without field names.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| *FieldNames  | A single-row list of field names.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| FieldNames;T | Two data records, the first a list of field names, and the second a list of data types. The data types are:<br><br>0 Invalid<br>1 True/False (non-NULL)<br>2 Unsigned byte (Byte)<br>3 2-byte signed integer (Integer)<br>4 4-byte signed integer (Long)<br>5 8-byte signed integer (Currency)<br>6 4-byte single-precision floating point (Single)<br>7 8-byte double-precision floating point (Double)<br>8 Date/Time (date is integer, time is fraction)<br>9 Binary data, 255-byte maximum<br>10 ANSI text, not case sensitive, 255-byte maximum (Text)<br>11 Long binary (OLE Object)<br>12 Long text (Memo) |
| NextRow      | The data in the next row in the table or query. When you first open a channel, NextRow returns the data in the first row. If the current row is the last record and you execute NextRow, the request fails.                                                                                                                                                                                                                                                                                                                                                                                                       |

PrevRow           The data in the previous row in the table or query. If PrevRow is the first request on a new channel, the data in the last row of the table or query is returned. If the first record is the current row, the request fails.

FirstRow           The data in the first row of the table or query.

LastRow            The data in the last row of the table or query.

FieldCount         The number of fields in the table or query.

\*SQLText           An SQL statement representing the table or query. For tables, this item returns an SQL statement in the format "SELECT \* FROM table;".

\*SQLText;<n>       An SQL statement in <n>-character chunks that represents the table or query, where <n> is an integer lower than 255. For example, for a query represented by the SQL statement "SELECT \* FROM Orders;" the item "SQLText;7" would return the following tab-delimited chunks:

```

SELECT
* FROM
Orders;

```

The following sample WordBasic macro demonstrates how to get information from the Employees table in NWIND.MDB:

```

Chan1 = DDEInitiate("MSAccess", "NWIND;TABLE Employees")
' Get a count of the number of Employee records.
SQL$ = "SELECT Count([Employee ID]) AS [CountOfEmployees] "
SQL$ = SQL$ + "FROM Employees;"
Chan2 = DDEInitiate("MSAccess", "NWIND;SQL " + SQL$)
EmployeeCount = Val(DDERequest$(Chan2, "FirstRow"))
DDETerminate Chan2

' Quit if there are no records.
If EmployeeCount <> 0 Then
 Msg$ = "NWIND Employee Information:"
 Msg$ = Msg$ + " Record Count:" + Str$(EmployeeCount)
 MsgBox Msg$

 ' Request the first row of data from the Employees table.
 Data$ = DDERequest$(Chan1, "FirstRow")

 ' Display the records.
 For i = 1 To EmployeeCount
 MsgBox Data$
 ' Get the next row of data, if not at the end.
 If i <> EmployeeCount Then
 Data$ = DDERequest$(Chan1, "NextRow")
 End If
 Next i
End If

```

## Executing Macros and Commands in Microsoft Access Using DDE

---

When you are using Microsoft Access as a DDE server, you can use the DDEExecute() function to instruct your application to execute a command. Microsoft Access recognizes any of the following valid commands:

- The name of a macro in the currently open database. A macro can be executed on a channel with any of the five topics described above.
- \*- Any action that you can execute in Access Basic using the DoCmd() statement. You cannot execute the following macro actions: AddMenu, MsgBox, RunApp, RunCode, SendKeys, SetValue, StopAllMacros, and StopMacro.
- \*- The OpenDatabase and CloseDatabase actions, executed only for DDE operations using the System topic.

Note that when you specify an action as a DDEExecute command, the action and any arguments follow the DoCmd() syntax and must be enclosed in brackets ([]). However, applications that support DDE do not recognize intrinsic constants, such as A\_NORMAL, in DDE operations. Therefore, you must use the actual number as an argument. Also, string arguments must be enclosed in quotation marks only if the string contains a comma. Otherwise, quotation marks are not required.

The following sample macro opens the Categories form, first minimized and then restored:

```
Chan = DDEInitiate("MSACCESS", "System")
AppActivate "Microsoft Access"
DDEExecute Chan, "[OpenForm Categories,,,,,2]"
DDEExecute Chan, "[OpenForm Categories]"
DDETerminate Chan
```

### \*Using the OpenDatabase and CloseDatabase Commands

---

The following commands can be executed on a channel opened to the System topic. These commands facilitate the remote opening and closing of databases in Microsoft Access from the client application.

- OpenDatabase <DatabaseName> [, Exclusive[, ReadOnly]]
- CloseDatabase

<DatabaseName> A string expression that is the name of an existing database. This can include the fully qualified MS-DOS path.

[Exclusive] A Boolean value that is True (-1) if the database is to be opened with exclusive (nonshared) access and False (0) if the database is to be opened with shared access. The default is shared access.

[ReadOnly]           A Boolean value that is True if the database is to be opened with read-only access and False if it is to be opened with read/write access. The default is read/write access.

The following sample WordBasic macro demonstrates how to use the OpenDatabase actions to remotely open NWIND.MDB and then open the Employees form:

```
Chan = DDEInitiate("MSACCESS", "System")
DDEExecute Chan, "[OpenDatabase NWIND.MDB]"
DDEExecute Chan, "[OpenForm Employees,0,,1,0]"
DDETerminate Chan
```

#### REFERENCES

=====

For more information on using Microsoft Access as a DDE server, search for "DDE" then "Using Microsoft Access As a DDE Server" using the Microsoft Access Help menu.

For more information on using DDE with Microsoft Access, query on the following words here in the Microsoft Knowledge Base

dde and <the name of the other application in the conversation>

Microsoft Access "User's Guide," Chapters 9 and 13

The README.TXT file that is shipped with Microsoft Access version 1.0

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpDde

**INF: TraceSQLMode Setting Helps Debug SQL Queries to ODBC**  
**Article ID: Q113918**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SUMMARY

=====

You can create a log file of all the SQL queries sent to ODBC for processing by setting the TraceSQLMode setting to 1 in your MSACCESS.INI file (Microsoft Access version 1.1) or in your MSACC20.INI file (Microsoft Access version 2.0).

MORE INFORMATION

=====

To create a log file of all the SQL queries sent to ODBC for processing, set the TraceSQLMode setting in the [ODBC] section of your .INI file to 1. The line will be:

```
[ODBC]
TraceSQLMode=1
```

NOTE: After setting this option, you must restart Microsoft Access for the change to take effect.

Microsoft Access will record all SQL commands in a file called SQLOUT.TXT in the Microsoft Access directory. SQL commands will continue to be recorded in the file as long as the TraceSQLMode setting is set to 1.

To turn the TraceSQLMode option off, change the setting from 1 to 0. The line will be:

```
[ODBC]
TraceSQLMode=0
```

NOTE: After setting this option, you must restart Microsoft Access for the change to take effect.

The following is an example of an SQLOUT.TXT file:

```
===== Open the attached dbo.authors =====
SQLExecDirect: SELECT dbo.authors.au_id FROM dbo.authors
SQLPrepare: SELECT au_id ,au_lname ,au_fname ,phone ,address ,city
,state ,zip ,contract FROM dbo.authors WHERE au_id = ?
SQLExecute: (GOTO BOOKMARK)
SQLPrepare: SELECT au_id ,au_lname ,au_fname ,phone ,address ,city
,state ,zip ,contract FROM dbo.authors WHERE au_id = ? OR au_id =
? OR au_id = ? OR au_id = ? OR au_id = ? OR au_id = ?
SQLExecute: (MULTI-ROW FETCH)
SQLExecute: (MULTI-ROW FETCH)
SQLExecute: (MULTI-ROW FETCH)
```

---

## REFERENCES

Search for "Setting ODBC options in the MSACCESS.INI file" in the README.TXT file in Microsoft Access version 1.1.

Search for "ODBC Settings" then "Customizing MSACC20.INI Settings" using the Microsoft Access version 2.0 Help menu.

Additional reference words: 1.10 2.00 queries tracing sqltrace  
troubleshoot sql

KBCategory: kbusage

KBSubcategory: ObcOthr

**INF: Trapping SQL Server RAISERROR() Function Values**  
**Article ID: Q101678**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SUMMARY

=====

Microsoft SQL Server users often create triggers or stored procedures that perform specific functions that are fired during certain events. Often, a custom-generated error value is desired to indicate the status of such events. This article describes a way to trap that value in Microsoft Access.

NOTE: This article assumes that the user is running Microsoft Access version 1.1, since that version corrects a problem to allow the SQL Server RAISERROR() function to return a value to Microsoft Access.

MORE INFORMATION

=====

1. Create the following stored procedure based on the sample database Pubs in Microsoft SQL Server:

```
CREATE PROCEDURE TestProc
AS
 RAISERROR 25000 'This is a test error'
```

2. Create a trigger on the table from which you want this stored procedure to run:

NOTE: Stored procedures cannot be executed directly from Microsoft Access 1.1, unless they are triggered from a SQL Server trigger or the SQL pass-thru .DLL file (SPT110.DLL) is used. However, when using version 2.0, you can invoke a stored procedure using an SQL pass-through query. For this example, you will use a trigger that fires when a record is updated.

```
CREATE TRIGGER TestTrig
ON Authors
FOR UPDATE
AS
 EXECUTE TestProc
```

3. Add the following code to an Access Basic module. The subprocedure updates the Authors table (the attached table dbo\_authors) in some way, and traps the error value that is passed by the RAISERROR() function:

```
Sub TrapIt ()
 Dim db As Database, Mydyna As Dynaset
 Dim Xerr As String, Xval As Integer,
 Dim Xstart As Integer, Xlen As Integer
```

```
On Error GoTo ErrorHandler
Set db = CurrentDB()
Set Mydyna = db.CreateDynaset("select * from dbo_authors;")
Mydyna.Edit
Mydyna!au_fname = Mydyna!au_fname
Mydyna.Update
Mydyna.MoveNext
Exit Sub
```

ErrorHandler:

```
'This routine parses the error string returned from ODBC and
'extracts only the error value you assigned with the RAISERROR()
'function in SQL Server.
```

```
'traps the error message
 xerror = Error$
'finds start of error value
 Xstart = InStr(1, xerror, "#") + 1
'finds length of error value
 Xlen = InStr(Xstart, xerror, ")") - Xstart
'extracts error value from string
 Xval = Mid(xerror, Xstart, Xlen)
 MsgBox ("You have encountered error #" & CStr(Xval))
 Resume Next
End Sub
```

Additional reference words: 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmPdox



## **INF: Troubleshooting Setup and Installation Problems**

**Article ID: Q96109**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article explains the following troubleshooting techniques that you can use to resolve Microsoft Access Setup and installation problems:

- Checking the installation disks
- Checking the TEMP directory
- Checking what loads when you start Microsoft Windows
- Clean booting your system

Note that you can use many of these techniques to troubleshoot problems that occur after installation.

### MORE INFORMATION

=====

#### Checking the Installation Disks

-----

The following steps help determine whether your problem is related to a bad installation disk. To check a specific disk:

1. Exit completely from Microsoft Windows.
2. At the MS-DOS command prompt, type the following:

```
md \disktest
```

3. Put the suspect disk in the floppy disk drive, then type the following:

```
copy <drive>:*.* c:\disktest
```

where <drive> is the appropriate floppy disk drive letter.

4. If the disk is copied successfully, the MS-DOS command prompt will return and no error message will appear. This means that this disk is not the problem, and you should continue troubleshooting.

If an error message is displayed, call Microsoft Sales Information Center at (800) 426-9400 and request a replacement disk.

5. Delete the temporary DISKTEST directory and any files in it by typing:

```
erase c:\disktest*.*
```

Answer "Y" (without quotation marks) when you are asked if you really want to erase everything.

Then type

```
rd c:\disktest
```

to remove the directory.

#### Checking the TEMP Directory

1. Exit completely from Windows.
2. At the MS-DOS command prompt, type the following:

```
set
```

3. Change to the directory pointed to by the TEMP variable. For example, if you are at the root of your C drive when you type SET and "TEMP=C:\WINDOWS\TEMP" is displayed, type the following command:

```
cd \windows\temp
```

If an "invalid directory" message is displayed when you change to the TEMP directory, you must create the TEMP directory using the MS-DOS Make Directory (MD) command.

4. There should be no \*.TMP files in your TEMP directory when Windows is not running. If there are \*.TMP files in this directory, you should erase them. To see whether there are any .TMP files in your TEMP directory, type the following command:

```
dir *.tmp
```

If there are \*.TMP files in the TEMP directory, erase them by typing

```
erase *.tmp
```

For more information on the TEMP directory, query on the following words here in the Microsoft Knowledge Base:

```
windows and temporary and files and definition
```

5. Use the MS-DOS CHKDSK command to check your hard drive. At the command prompt, type:

```
chkdsk
```

NOTE: Refer to your Microsoft MS-DOS "User's Guide and Reference" if you have lost allocation units, lost clusters, or cross-linked files.

6. Check the amount of free space on your hard drive by looking at the line that says "bytes available on disk." If this number is less than 4000000 bytes (4 MB), free up some additional hard drive space by removing unnecessary files.

## Checking What Loads When You Start Windows

---

For troubleshooting purposes, it is important that you have a clean Windows configuration before you boot from a clean, bootable floppy disk (refer to "Clean Booting Your System" below). The following steps ensure a clean Windows configuration:

1. Using a text editor, such as the MS-DOS Editor or Windows Notepad, open the WIN.INI file. Check the following two lines:

```
load=
run=
```

If there is anything to the right of the equal sign (=) on either of these two lines, place a semicolon (;) at the beginning of line. For example:

```
;load=c:\mydir\myprogram.exe
```

The semicolon causes Windows to ignore that line.

2. Save your changes and exit the text editor.
3. When you are clean booting, you do not want anything to load automatically from your Startup group. To temporarily disable this functionality, hold down the SHIFT key as soon as you see the Windows logo screen. Keep the SHIFT key depressed until Windows is completely loaded.
4. Make sure that you are using a Windows video driver. To do this, exit Windows. At the C:\WINDOWS> prompt, type:

```
setup
```

Windows displays a System Information screen that indicates the type of display that you are using. If the Display field does not show EGA or VGA, follow the instructions on your screen to change the display to VGA.

NOTE: If you receive a Welcome To Windows Setup screen, you were not at the Windows prompt when you typed SETUP.

## Clean Booting Your System

---

A clean boot eliminates many variables that may be related to your Setup or installation problem. Follow these steps to clean boot your system:

1. Create a bootable floppy disk by putting a blank floppy disk in your A drive, and then typing the following at the MS-DOS command prompt:

```
format a: /s
```

2. Use a text editor, such as MS-DOS Editor or Windows Notepad, to

create a CONFIG.SYS file on the bootable floppy disk that you created in step 1.

A simple CONFIG.SYS contains the following:

```
files=50
buffers=20
device=c:\windows\himem.sys
<third-party disk partitioner>
<third-party disk compression driver>
<Other third-party driver if necessary to boot your computer>
shell=c:\<valid path>\command.com /p
```

NOTE: If you have the following line in your current CONFIG.SYS file, you should include it in your clean boot CONFIG.SYS file:

```
device=c:\windows\smartdrv.sys /double_buffer
```

For more on SMARTDRV.SYS, query on the following words here in the Microsoft Knowledge Base:

SMARTDrive and 4.0 and design and overview

NOTE: If there is hard disk partitioning, disk compression, or other third-party software required to boot your computer, these files must be included in your CONFIG.SYS file.

3. Using a text editor, create an AUTOEXEC.BAT file on the bootable floppy disk that you created in step 1. A simple AUTOEXEC.BAT contains the following:

```
prompt pg
path=c:\windows
set temp=c:\<valid path>
```

4. Exit from the text editor.
5. With the clean, bootable floppy disk in drive A, reboot your system.
6. If you still experience Setup problems, try starting Windows in standard mode by typing

```
win /s
```

at the A prompt. Windows does not try to create any virtual memory in standard mode. If running in standard mode resolves your problem, contact Windows Product Support for further assistance.

For more information on the clean booting your system, query on the following words in the Microsoft Knowledge Base:

```
clean and boot and WINDOWS
kbsetup
Additional reference words: 1.00 1.10 2.00
KBCategory: kbsetup kbtshoot
KBSubcategory: StpOthr
```



## **INF: Troubleshooting Tips for Error Values**

**Article ID: Q112103**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SYMPTOMS

=====

You see any of the following error values in a field on a table, query, form, or report:

#Error, #Num!, #Name?, #Div/0!, #Deleted, or #Locked

### CAUSE

=====

Microsoft Access places an error value in a field when it cannot find necessary information, execute an expression, or store a value within the limits for the field.

### RESOLUTION

=====

The sections below explain what the error values mean, and list some reasons why Microsoft Access might display them:

#### #Error

-----

This value means that Microsoft Access cannot evaluate the expression. For example, you may have supplied incorrect or too few arguments for an aggregate function, used a parameter query as the domain for an aggregate function, or the expression may contain a circular reference.

The following three examples demonstrate possible causes for the #Error value:

Aggregate function:  
=DAvg("[Sales]","Employe")

In the above example, the table name should be "Employee," not "Employe."

Circular reference: (Query)  
FirstName: [FirstName] & " " & [LastName]

The alias FirstName is also part of the expression in the example above.

Circular reference: (Forms and Reports)  
Name: FirstName  
ControlSource: =[FirstName] & " " & [LastName]

The text box name is also part of the expression in this example.

In a table or form, this error can also occur when the DefaultValue property setting for a field or control is not appropriate for the DataType or FieldSize property setting, or when an expression includes a bound control defined using the Trim() function and the underlying field contains no data. The following example demonstrates a field whose DefaultValue property setting is not appropriate for its FieldSize setting:

```
Field: State (Text, FieldSize = 2)
DefaultValue: ="Cal" (3 characters)
```

In a query, this error can occur when the value of a calculated field is greater than the value allowed by the field's FieldSize property setting. For example, if you add or multiply two Integer values and the resulting Integer is greater than the value permitted in an Integer field, Microsoft Access displays the #Error value in the field. The following example demonstrates a value that is larger than is permitted for the field:

```
Field: Age (Integer, current record value = 50)
Expr1: [Age] * 1000
```

```
#Num!

```

This error value means that the value in the field is too large (either positively or negatively) to be stored in the field, based on the field's DataType or FieldSize property setting.

```
#Name?

```

This error means that the name entered as the source of the value in the field is invalid. The name may be misspelled, you may have omitted the equal sign (=) before the expression, or the source itself may have been deleted.

The following example demonstrates a missing equal sign (=) in an expression:

```
ControlSource: [FirstName] & " " & [LastName]
Should be: =[FirstName] & " " & [LastName]
```

The next example demonstrates an invalid ControlSource property name:

```
ControlSource: =[FirstNam] & " " & [LastName]
Should be =[FirstName] & " " & [LastName]
```

When you are referring to a control on a subform or subreport, you must refer to it via the main form or report using the following syntax:

```
=Forms![<MainFormName>]![<SubFormName>].Form![<ControlName>]
```

-or-

```
=Reports![<MainReportName>]![<SubReportName>].Report![<ControlName>]
```

<MainFormName> or <MainReportName> is the name of the form or report that contains the subform or subreport.

<SubFormName> or <SubReportName> is the name of the subform or subreport. Note that this name does not have to be the same name as the name of the subform or subreport itself. To verify this name, check the Name property of the subform or subreport.

<ControlName> is the name of the control on the subform or subreport. To verify this name, check the Name property of the control.

Some other causes for the #Name? error value include:

- A field name on the form or report does not match the name of the field in the underlying table.
- A control name is the same as one of the fields on the underlying table.
- An expression designed to calculate a sum for a control might include a Sum() function. (The Sum() function can be used to calculate sums only for fields, not for controls.)

#Div/0!

-----

This value means that you are trying to divide a number by zero, either directly in an expression (for example, 8/0), or by using a value from a field whose value is zero.

#Deleted

-----

This value means that the record being referred to has been deleted.

#Locked

-----

The value can be caused by any of the following situations:

- The record has been locked by another user and Microsoft Access cannot read the data.
- There are two or more instances of Microsoft Access running on the same computer. Microsoft Access treats each open copy of the database as a separate user.
- Access Basic code has opened a recordset and has a lock on a record.

It is important to note that Microsoft Access uses "page locking" instead of "record locking." Microsoft Access stores records in groups of 2048 bytes, called pages. A single page may hold only one record, or it may hold many records. If each record only requires 200 bytes, then up to 10 records might be stored on a page. When Microsoft Access places a lock on a record, it locks that record's page. If there are 10 records on the page, then all 10 records on that page will be locked.

Additional reference words: 2.00 pounderror poundname

KBCategory: kbusage kbtshoot

KBSubcategory: ExrOthr





## INF: Two Functions to Calculate Age in Months and Years

Article ID: Q100136

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article explains how to create a function to calculate the age of a person or thing based on a given date.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

### MORE INFORMATION

=====

Enter the following code in a module:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'=====
' General Declaration
'=====
Option Explicit

'*****
' FUNCTION NAME: Age()
'
' PURPOSE:
' Calculates age in years from a given date to today's date.
'
' INPUT PARAMETERS:
' StartDate: The beginning date (for example, a birth date).
'
' RETURN
' Age in years.
'*****
Function Age (varBirthDate As Variant)
 Dim varAge As Variant

 If IsNull(varBirthdate) then Age = 0: Exit Function

 varAge = DateDiff("yyyy", varBirthDate, Now)
 If Date < DateSerial(Year(Now), Month(varBirthDate), _
 Day(varBirthDate)) Then
 varAge = varAge - 1
```

```

 End If
 Age = varAge
End Function
'*****
' FUNCTION NAME: AgeMonths()
'
' PURPOSE:
' Compliments the Age() function by calculating the number of months
' that have expired since the last month supplied by the given date.
' If the given date is a birthday, the function returns the number of
' months since the last birthday.
'
' INPUT PARAMETERS:
' StartDate: The beginning date (for example, a birthday).
'
' RETURN
' Months since the last birthday.
'*****
Function AgeMonths(ByVal StartDate As String)
 Dim tAge As Double
 tAge = (DateDiff("m", StartDate, Now))
 If (DatePart("d", StartDate) > DatePart("d", Now)) Then
 tAge = tAge - 1
 End If

 If tAge < 0 Then
 tAge = tAge + 1
 End If

 AgeMonths = tAge Mod 12

End Function

```

#### Testing the Age() and AgeMonths() Functions

---

The following steps describe how to test the Age() and AgeMonths() functions:

1. Open a module or create a new one.
2. From the View menu, choose Immediate Window.
3. Assume your friend's birth date was November 15, 1967 and today is June 3, 1993. Type the following in the Immediate window and press ENTER:

```
? Age("11/15/67")
```

Microsoft Access responds with the value 25 (years).

4. Type the following and press ENTER:

```
? AgeMonths("11/15/67")
```

Microsoft Access responds with the value 6, indicating that six months have passed since this person's last birthday. Your friend

is 25 years and six months old.

#### Using the Age() and AgeMonths() Functions

---

The following procedure explains how to mark old orders by placing the age value in a new control:

1. In the sample database NWIND.MDB, enter the Age() and AgeMonth() functions in a new module.
2. Open the Orders form in Design view and add an unbound text box.
3. Type the following in the ControlSource property of the new control:

```
=Age([Order Date]) & " ys " & AgeMonths([Order Date]) & " mo"
```

4. Switch to Form view. The age of the order is displayed in the new control.

Additional reference words: 1.00 1.10 2.00 birth birthday birthdate age abc

KBCategory: kbprg

KBSubcategory: PgmOthr

**INF: Two Methods to Refer to a Field in a Previous Record**  
**Article ID: Q101081**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Sometimes a field in a previous record must be referred to in a later record, such as in a checkbook program or a mileage log. This article explains how to refer to such a record.

NOTE: The procedures provided require a unique, sequential, numeric field, such as a Counter field or a transaction number.

MORE INFORMATION

=====

This section provides two sets of examples for database objects (forms, reports, and queries): one using the DLookup() function, the other using the sample database NWIND.MDB.

DLookup() Function Examples

-----

Use the DLookup() function for the following database objects, as in the following examples:

In a Form

-----

Type the following in the ControlSource of the text box:

```
=DLookup("[Field]","Table","[ID]=Forms![Form1]![ID]-1")
```

In a Report

-----

Type the following in the ControlSource of the text box:

```
=DLookup("[Field]","Table","[ID]=Reports![Report1]![ID]-1")
```

In a Query

-----

Type the following in the Field row of the query grid:

```
Expr1: DLookup("[Field1]","Table","[ID]=|[ID]|"&-1)
```

The "-1" indicates the previous record. This number can be changed to reference a different record.

Northwind Traders Examples

-----  
The following examples use the sample database NWIND.MDB:

In a Form  
-----

1. Create a blank form based on the Suppliers table.
2. Add the Supplier ID and Company Name fields.
3. Create an unbound text box.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Form: Form1

RecordSource: Suppliers

Form1 Controls

Text Box:
 ControlName: Supplier ID
 ControlSource: Supplier ID
Text Box:
 ControlName: Company Name
 ControlSource: Company Name
Text Box:
 ControlName: Field2
 ControlSource:
 =DLookUp("[Company Name]","Suppliers","[Supplier ID] = _
 Forms![Form1]![Supplier ID]-1")
```

In a Report  
-----

1. Create a blank report based on the Suppliers table.
2. Add the Supplier ID and Company Name fields.
3. Create an unbound text box.

```
Report: Report1

RecordSource: Suppliers

Report1 Controls

Text Box:
 ControlName: Supplier ID
 ControlSource: Supplier ID
Text Box:
 ControlName: Company Name
 ControlSource: Company Name
Text Box:
```

ControlName: Field2  
ControlSource:  
=DLookup("[Company Name]","Suppliers","[Supplier ID] = \_  
Reports![Report1]![Supplier ID]-1")

In a Query  
-----

1. Create a new query.
2. Add the Suppliers table.

Query: Query1  
-----

Type: Select Query  
Field: Supplier ID  
Field: Company Name  
Field: Expr1: DLookup("[Company Name]","Suppliers","[Supplier ID] = \_  
|[Supplier ID]|"&-1)

Additional reference words: 1.00 1.10 2.00 next  
KBCategory: kbusage  
KBSubcategory: ExrOthr

## **INF: Understanding Event Order in MS Access 2.0**

**Article ID: Q118762**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

In order to aid you in automating tasks in Microsoft Access successfully, it is helpful to have an understanding of the Microsoft Windows event model. Objects in Microsoft Access respond to many types of events, such as mouse events and keyboard events. Events are also triggered in response to record movements. There are several ways to determine the order of events for forms, reports, subforms, and subreports in Microsoft Access.

Below is a description of a form in the Northwind Traders sample database (NWIND.MDB) that allows you to experiment with and view the order of events.

### MORE INFORMATION

=====

There is a special version of the Orders form in the sample database NWIND.MDB called the Show Events form that records each event as it occurs. When you open the Show Events form, an accompanying form called Event History opens as well. The Event History form lists the events that occur in the Show Event form. As you use the mouse or keyboard to select or change controls in the Show Events form, the list in the Event History form changes to show all the events that have occurred. Note the events that occur as you use the mouse and keyboard to manipulate objects in the Show Events form. The Show Events and Event History forms can help you to understand the concept of events, and the order in which they occur.

The recording of events in the Show Events form is accomplished by a function called ShowEvent(). The ShowEvent() function is called in the event procedures of all the event properties of all the controls on the Show Events form. The ShowEvent() function sets the value of the text control to include the event that just occurred.

### REFERENCES

=====

For more information about event order, search for "order of events" then "Order of Events" using the Microsoft Access Help menu. The Order of Events topic includes the following topics:

- Events for Controls on Forms
- Events for Records on Forms
- Events for Forms
- Events for Keyboard and Mouse
- Events for Reports and Report Sections
- Events and Saving Data



Microsoft Access "Building Applications," version 2.0, Chapter 8,  
"Events," pages 184-201

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmrHowto

## INF: Updating One Table with Data from a Second Table

Article ID: Q103269

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes how to create a query that will update a record in one table with data from another table.

### MORE INFORMATION

=====

The sample query below adds data from the Current Week table to the records in the Yearly Rainfall table.

NOTE: This type of query will not work if the data being added is the result of a totaling or grouping query. For more information on how to update a record with the result of a totaling or grouping query, query on the following words here in the Microsoft Knowledge Base:

access and update and dlookup

1. Create a new table with the following fields and save it as Yearly Rainfall:

| City     | Inches | Last Updated |
|----------|--------|--------------|
| Baroda   | 0.5    | 4/2/93       |
| Basildon | 22.0   | 4/2/93       |
| Beaver   | 18.0   | 4/2/93       |

2. Create a new table with the following fields and save it as Current Week:

| City     | Inches | Week Of |
|----------|--------|---------|
| Baroda   | 0.1    | 4/9/93  |
| Basildon | 2.0    | 4/9/93  |
| Beaver   | 2.5    | 4/9/93  |

3. Create a new query based on the tables Yearly Rainfall and Current Week. Join the tables on the City field.
4. From the Query menu, choose Update.
5. Drag the Last Updated and Inches fields from the Yearly Rainfall table to the query grid.
6. Create the following entries in the Update To row of the query grid:

Field: Last Updated

Update To: [Current Week].[Week Of]

Field: Inches

Update To: [Yearly Rainfall].[Inches]+[Current Week].[Inches]

7. Save and then run the query.

The data in the table Yearly Rainfall will now be:

| City     | Inches | Last Updated |
|----------|--------|--------------|
| Baroda   | 0.6    | 4/9/93       |
| Basildon | 24.0   | 4/9/93       |
| Beaver   | 20.5   | 4/9/93       |

Additional reference words: 1.00 1.10 2.00 Total Archive History

KBCategory: kbusage

KBSubcategory: QryMktbl

**INF: Updating, Troubleshooting Novell NetWare VLM Installation**  
**Article ID: Q117918**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

This article describes how to update specific Novell NetWare drivers on your system to the latest Novell virtual loadable module (VLM) drivers, so that Microsoft Access will run correctly on your Novell network.

MORE INFORMATION

=====

Steps to Update Novell Network Drivers

-----

Microsoft Access requires specific versions (or later) of certain Novell network files to run correctly. The following steps describe how to be sure that the correct versions of these files are installed on your system:

1. Obtain the following files:

- WINUP9.EXE and DOSUP9.EXE (from Novell). These files are available on Novell's CompuServe NOVLIB forum. To obtain them, log on to CompuServe and type GO NOVFILES, or contact Novell technical support at (800) 638-9273. These files are also available through NetWare and authorized Novell resellers.
- WW0863.EXE (from Microsoft). This file is available on the Microsoft Software Library (MSL). You can download it from the MSL on the following services:
  - Microsoft Download Service (MSDL)  
Dial (206) 936-6735 to connect to MSDL  
Download WW0863.EXE from WINDOWS\31\APPNOTES
  - Internet (anonymous FTP)  
ftp ftp.microsoft.com  
Change to the SOFTLIB/MSLFILES subdirectory  
Get WW0863.EXE
  - CompuServe  
GO MSL  
Search for WW0863.EXE  
Display results and download

2. Run WW0863.EXE, which is a self-extracting file. When you run this file, it will expand into the following three files: VIPX.386, VTDA.386, and README.TXT. Follow these steps to install the new files:

- a. Rename or back up your original VIPX.386 file. To rename the file,

type the following line at the command prompt:

```
ren vipx.386 vipx.bak
```

- b. Copy the VTDA.386 file to your WINDOWS\SYSTEM subdirectory. To copy the file, type the following line at the command prompt:

```
copy <drive>:\<sourcedir>\vtda.386 c:\windows\system
```

- c. Using any text editor (such as Windows Notepad), open the SYSTEM.INI file in your Windows directory. Add the following lines to the [386Enh] section of the file, or change them to match these lines if they already exist:

```
InDOPolling=FALSE
TimerCriticalSection=10000
;device=*vtd
device=vtda.386
```

- d. Save and then close the SYSTEM.INI file.

3. DOSUP9.EXE and WINUP9.EXE are self-extracting files. Run both of these files to produce the following files:

|             |       |          |        |
|-------------|-------|----------|--------|
| LSL.COM     | v2.05 | 9-10-93  | 17805  |
| IPXODI.COM  | v2.12 | 10-7-93  | 30247  |
| VLM.EXE     | v1.10 | 12-9-93  | 36525  |
| VIPX.386    | v1.17 | 10-11-93 | 23855  |
| NETWARE.DRV | v3.02 | 11-24-93 | 146736 |

Copy the first three files, LSL.COM, IPXODI.COM, and VLM.EXE, to your NWCLIENT directory. (NOTE: C:\NWCLIENT is the default directory. Your directory location may differ, depending on your installation.) Copy the other two files, VIPX.386 and NETWARE.DRV, to your WINDOWS\SYSTEM subdirectory.

NOTE: There is one version of the NETWARE.DRV file for VLM, and another version for NETX. You must be sure to use the version for VLM. If you use the wrong version, network problems can occur.

These files add to or replace your existing network files. To see which version of each .EXE or .COM file you are currently running, type the name of the file and press ENTER. If the file is already loaded, you will get a message that states that the file is already loaded, and displays the version number of the currently loaded file.

4. Modify the NET.CFG file (or the SHELL.CFG file, depending on your network setup) by adding the following entry:

```
READ ONLY COMPATIBILITY = OFF
```

If your .CFG file has the heading "Netware DOS Requestor," then the preceding line should follow this heading. Otherwise you can place this line anywhere in the .CFG file.

5. Quit and then restart Microsoft Windows.

6. Make sure that every workstation on the network has the correct versions of the files listed in this article and has the .INI file entries listed in this article.

#### Troubleshooting Tips

-----

If, after following the above steps, you still encounter network problems, try the following tips:

- Make sure that everyone on the network who uses Microsoft Access has the necessary network rights. All Microsoft Access users must have File Scan and Read rights for the WINDOWS and ACCESS directories on the network. Also, users must have File Scan, Read, Write, Create, Erase, and Modify rights for the ACCESS\WORKDIR subdirectory.
- Use the network Supervisor or Admin account to run the Microsoft Access network Setup (SETUP /A) and the first workstation Setup.
- If the ACCESS directory on the server is marked as Read-Only, make sure that the ACCESS\WORKDIR subdirectory is marked as Create/Erase/FileScan/Modify/Read/Write for all users.
- Load NETX.EXE instead of VLM.EXE when you start the network. If you do this, make sure to change the NETWARE.DRV driver from the VLM version to the NETX version before you start Windows. To do this, make sure the following files are present in the WINDOWS\SYSTEM subdirectory:

|             |       |          |        |
|-------------|-------|----------|--------|
| NETX.EXE    | v3.32 | 11-17-93 | 78654  |
| NETWARE.DRV | v2.02 | 10-27-92 | 126144 |

#### Multiuser Database Corruption or Locking Issues

-----

If you experience multiuser database corruption or locking issues, perform the following steps:

1. On the NetWare server, create the following setting:

```
SET MAXIMUM RECORD LOCKS PER CONNECTION = 10000
```

2. On each workstation, add the following line to the [ISAM] section of the MSACC20.INI file in the Windows directory:

```
PageTimeout=300
```

3. If you are running Novell NetWare version 3.11, obtain the TTSFIX patch file described in the following article (and in the ACREADME.HLP file):

```
ARTICLE-ID: Q102522
TITLE : PRB: "Record Lock Threshold Exceeded" with Large
 Action Query
```

NetWare is manufactured by Novell Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 2.00 networking  
KBCategory: kbnetwork  
KBSubcategory: EvnNtw

**INF: Upper Limit for QueryTimeout and LoginTimeout Parameters**  
**Article ID: Q100176**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes the maximum limit allowed for the QueryTimeout and LoginTimeout parameters.

MORE INFORMATION

=====

As the README.TXT file for Microsoft Access 1.1 states, the two settings in the [odbc] section of the MSACCESS.INI file are as follows:

| Entry        | Value | Effect                                                                                                                                       |
|--------------|-------|----------------------------------------------------------------------------------------------------------------------------------------------|
| QueryTimeout | S     | Wait S seconds for queries sent to ODBC, then stop trying to process the query results (for asynchronous queries only). Default: 60 seconds. |
| LoginTimeout | S     | Wait S seconds for ODBC login response, and then stop trying to connect to a server. Default: 20 seconds.                                    |

The README.TXT file does not state the limits for these parameters.

Since the values for QueryTimeout and LoginTimeout are unsigned 4-byte values (long integers), the maximum value is +/- 2,147,483,648. If the value is set to zero, the timeout is disabled and the query or login never times out.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr



## **INF: Use a Query to Parse a Full Name into Two Fields**

**Article ID: Q95608**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0  
-----

The following query parses a field containing a full name in which the last and first names are separated by commas. For example a name could be entered as: Smith, John. The query would return two columns named Last: and First:. The query uses the Left\$ and Right\$ functions and searches for the comma using the InStr function.

Query: QueryTest

-----  
Field: Last: Left\$([Name],InStr(1,[Name],",")-1)

Show: True

Field: First: Right\$([Name],Len([name])-InStr(1,[Name],",")-1)

Show: True

You can modify the query to account for spaces in the name field. For example if the name field contained Smith,John with no spaces, you would remove the -1 for the first name field.

Additional reference words: 1.00 1.10 2.00 expressions string

KBCategory: kbusage

KBSubcategory: QryMktbl

**INF: Use Append Query to Set Initial Value of Counter Field**  
**Article ID: Q94821**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can change the starting value of a Counter field to a number other than 1 by using an append query.

MORE INFORMATION

=====

By design, Microsoft Access always numbers Counter fields beginning with the number 1. You cannot edit a Counter field or change its starting value.

However, you can force Microsoft Access to number a Counter field with a number you choose by doing the following:

1. Create a new table containing a single Number field. Set FieldSize to Long Integer.
2. Give the new Number field a name identical to the name of the Counter field in your original table.
3. Use an append query to join this new record to your original table. This action forces Microsoft Access to number any new Counter fields with your number + 1.

For the expanded procedure, see the "How to Renumber the Counter Field" section below.

Your original table has the following properties:

Table: Receiving

-----

FieldName: ID Number

Data Type: Counter

FieldName: Description

Date Type: Text

How to Renumber the Counter Field

-----

1. Create a new table with the following structure and save it as Sending:

Table: Sending

-----

FieldName: ID Number

Data Type: Number

FieldSize: Long Integer

2. Open the Sending table in Datasheet view. Add a new record by inserting a value in the Number field. The value you enter must be 1 less than the starting value you want for the Counter field in your original table. For example, if you want the Counter value in your original table to start at 100, enter 99 in the Number field.
3. Create a new append query to append the single record from the new table (Sending) to your original table (Receiving). To do this, choose Query, then choose New and select the Sending table. Save the query as AppendCounter.
4. From the Query menu, choose Append. Select Receiving from the Append To Table Name combo box, then choose OK.

At this point, you have instructed Microsoft Access to append (or join) a field (or fields) from the Sending table to the Receiving table. Now you need to tell Microsoft Access which field or fields to append.

5. Append the ID Number field from the Sending table to the Receiving table either by selecting ID Number from the Field combo box in the query grid or by dragging the ID Number field from the Sending table to the query grid.

Because Microsoft Access recognizes that the two fields in Sending and Receiving have identical names, it automatically places ID Number in the Append To field of the query grid, as follows:

```
Query: AppendCounter

FieldName: ID Number
AppendTo: ID Number
```

6. Run the query by choosing the exclamation point (!) button on the toolbar. The following message is displayed:

```
1 Row(s) will be appended.
```

Choose OK to append the record to the Receiving table.

7. Delete the Sending table, then delete the newly appended row from the Receiving table.

You are now ready to add a new record to your original table. The Counter will start at the number you have chosen.

NOTE: Do not compact the database before adding a new record to the original table. If you do, Microsoft Access will reset the Counter value to the number 1.

#### REFERENCES

=====

For more information on append action queries and Counter fields, search on the following words here in the Microsoft Knowledge Base:

counter and append

For more information on customizing Counter fields, search on the following words here in the Microsoft Knowledge Base:

counter and custom

Microsoft Access "User's Guide," version 1.0, page 174

Additional reference words: 1.00 1.10 2.00 appending starting initial

KBCategory: kbusage

KBSubcategory: QryMktbl

**INF: Use AutoLookup Technique to Look Up Info. Automatically**  
**Article ID: Q95048**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can create a query or form that automatically looks up information in the one-side table of a one-to-many relationship based on an entry in a many-side field. This process is sometimes called AutoLookup.

You can use AutoLookup to have a query automatically look up information in other tables and display it on a form or report.

You can look up information by using any one of the following three techniques:

1. Use AutoLookup. For more information on AutoLookup, see the "More Information" section below.
2. Use multiple DLookup() functions in forms and reports. For more information on the DLookup() function, query on the following words here in the Microsoft Knowledge Base:

DLookup and forms and reports

3. Use the Column property of a multiple-column combo box to update a text box control with new information as focus moves from row to row in the combo box. For more information on this, query on the following words here in the Microsoft Knowledge Base:

Column and property and combo and Update

MORE INFORMATION

=====

Use AutoLookup when you want Microsoft Access to automatically look up information in the one-side table based on an entry in a many-side field. The AutoLookup technique works in a query or in a form. AutoLookup also works with tables that have a one-to-one relationship if you use a left outer join. For more information on what data is updatable, search for "Underlying table or query," then "When is a query updatable?" using the Microsoft Access Help menu.

For example, the Categories and Products tables in the sample database NWIND.MDB have a one-to-many relationship. Each category may appear several times in the Products table. The Products table contains the Category ID field, which is the foreign key that identifies the category for a product.

When AutoLookup updates records, Microsoft Access automatically recalculates any totals or expressions that are dependent on the

updated information.

Microsoft Access version 1.1 has additional online Help topics for AutoLookup issues. In this context, the term "AutoLookup" is synonymous with the term "dynamic lookup." For more information, search for "dynamic lookup", then "Displaying Fields from Another Table or Query (Common Question)" using the Help menu. In Microsoft Access version 2.0, search for "AutoLookup."

The following two examples demonstrate how AutoLookup works:

Example 1: Create a Query That Uses AutoLookup

---

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new query. Add the Products and Categories tables. Products is the many-side table and Categories is the one-side table. The two tables have a many-to-one relationship based on the Category ID field (meaning there are many identical Category ID values in Products for each unique Category ID value in Categories).
3. Add the following fields from the Products table:  
  
    Product ID  
    Product Name  
    Category ID
4. Add the following fields from the Categories table:  
  
    Category Name  
    Description
5. Save the query as AutoLookup Example.
6. View the results of the query. To verify that the query performs AutoLookup, move to the end of the query and create a new record. Enter a valid Category ID value in the Category ID field. Microsoft Access will automatically look up the values associated with that Category ID value in the Categories table.

This process works because the Category ID field located in the query comes from the many-side table (Products). Microsoft Access performs AutoLookup on one-side tables when you enter identifying data in the many-side table. When you enter a valid value in the Products table's Category ID field, Microsoft Access knows that Category ID is the key field for the one-side table (Categories), so it looks up the information from Categories based on the Category ID value and automatically displays it in the query.

Example 2: Create a Form That Uses AutoLookup

---

1. Create a new form based on the query that you created in Example 1.
2. Locate the fields from the Products table on "one" side of the form

and those from the Categories table on the "many" side of the form.

3. Create a combo box bound to the Category ID field. Set the RowSource property to the Categories table. Display a single column, [Category ID], in the combo box.
4. Switch to Form view. When you enter new records and select a category from the Category ID combo box, all related information about that Category is displayed.

Note that if you had included the primary key value from the one-side table (Categories), you would not have been allowed to add new records to the query or to change the Category ID value. For this reason, you must drag the Category ID field from the Products table, rather than from the Categories table.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 10, "Creating Forms Based on More than One Table," page 282

Microsoft Access "User's Guide", version 2.0, Chapter 11, "Designing Select Queries"

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryJoin

**INF: Using "<", ">", and "=" Operators in Query Parameters**  
**Article ID: Q96463**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes a simple, non-programmatic way to allow a user to type the comparison operators "<", ">", and "=" in query parameters. Due to the nature of the method, query criteria should be kept as simple as possible. For more complex criteria, you will have to use Access Basic code.

MORE INFORMATION

=====

1. Create a table called Names with the following fields:

Table: Names

-----

Field: First Name

DataType: Text

Field: Last Name

DataType: Text

Field: Age

DataType: Number

Format: Single

2. Enter the following data:

| First Name | Last Name | Age |
|------------|-----------|-----|
| Tom        | Smith     | 35  |
| Anne       | Howard    | 7   |
| Jim        | Bowie     | 20  |
| Sue        | Thomas    | 44  |

3. Create the following query based on the Names table:

Query: List Names

-----

Field: First Name

Field: Last Name

Field: Age

First Criteria Line: =Mid([Enter Age],2)

Second Criteria Line: <Mid([Enter Age],2)

Third Criteria Line: >Mid([Enter Age],2)

Fourth Criteria Line: <leave blank>



Show: True

Field: Expr1:Left([Enter Age],1)  
First Criteria Line: "="  
Second Criteria Line: "<"  
Third Criteria Line: ">"  
Fourth Criteria Line: Is Null  
Show: False

The criteria are entered on successive lines (that is, "Criteria 1" is entered on the "Criteria" line, "Criteria 2" is entered on the "Or" line, and "Criteria 3" and "Criteria 4" are entered on the blank lines below that).

Save the query and run it. Sample output is given for each of the four types of input:

| [Enter Age]   | Output |        |    |
|---------------|--------|--------|----|
| <leave blank> | Tom    | Smith  | 35 |
|               | Anne   | Howard | 7  |
|               | Jim    | Bowie  | 20 |
|               | Sue    | Thomas | 44 |
| <21           | Anne   | Howard | 7  |
|               | Jim    | Bowie  | 20 |
| >21           | Tom    | Smith  | 35 |
|               | Sue    | Thomas | 44 |
| =7            | Anne   | Howard | 7  |

NOTE: The query is not designed to allow combinations of "<", ">", and "=", such as ">=20". You must always use one of the three operators, no more, no less, or you must leave the whole parameter blank.

The SQL for the query is given below:

```
SELECT DISTINCTROW
 [First Name], [Last Name], Age
FROM
 Names
WHERE
 (Age=Mid([Enter Age],2) AND Left([Enter Age],1)="=")
OR
 (Age<Mid([Enter Age],2) AND Left([Enter Age],1)"<")
OR
 (Age>Mid([Enter Age],2) AND Left([Enter Age],1)">")
OR
 (Left([Enter Age],1) Is Null)
WITH OWNERACCESS OPTION;
```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryParm

**INF: Using a Conditional Macro to Confirm Changes to a Field**  
**Article ID: Q99400**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how to prompt the user to confirm changes to a field using a conditional macro.

MORE INFORMATION

=====

This following procedure uses the Employees form in the sample database NWIND.MDB:

1. Create the following new macro by choosing the Macro button in the Database window and then choosing New:

| Macro Name | Condition                     | Action      | Defined Below |
|------------|-------------------------------|-------------|---------------|
| AreYouSure | MsgBox("Commit changes?",1)=2 | CancelEvent | 1.            |
|            | ...                           | SendKeys    | 2.            |

AreYouSure Actions

-----

1. CancelEvent
2. SendKeys  
    Keystrokes: {esc}  
    Wait: NO

NOTE: The ellipsis in the Condition column forces Microsoft Access to perform the action on that line if the condition on the preceding line is true. Microsoft Access evaluates macro conditions as true or false. If the expression is true, Microsoft Access performs the action; if it is false, Microsoft Access ignores the action.

2. Save the macro as AreYouSure.
3. Open the Employees Form in Design view by choosing the Form button in the Database window, selecting the Employees form, and choosing the Design button.
4. From the View menu, choose Property to display the property sheet.
5. Select the [Hire Date] text box control to display its properties.
6. Set the properties for the Employees form and [Hire Date] text box control as follows:

Form: Employees

Caption: Employees Form  
ControlSource: Employees  
Text box: Hire Date  
ControlName: Hire Date  
ControlSource: Hire Date  
Before Update: AreYouSure

7. From the View menu, choose Form. When the date in the [Hire Date] field is edited, a confirmation message appears. Choose OK to confirm your change, or Cancel to cancel your change.

NOTE: This example uses the MsgBox statement instead of the MsgBox action. The MsgBox action has only an OK Button, whereas the MsgBox statement can also include a Cancel button.

#### REFERENCES

=====

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

Microsoft Access "User's Guide," version 1.0, pages 528-554

Additional reference words: 1.00 1.10 2.00 forms macros data validation

KBCategory: kbusage

KBSubcategory: McrCond

**INF: Using Access Basic to Derive a Statistical Median**  
**Article ID: Q95918**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY  
=====

This article describes an Access Basic function that opens a table, reads the data, and finds the statistical median. The median is a measure of central tendency, another "middle" of a data set. The data set consisting of the numbers 1, 2, 3, 6, and 100 has a median of 3, the middle of the set. The data set consisting of the numbers 1, 2, 6, and 10 has a median of 4, the middle of the set [(2+6)/2=4].

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION  
=====

To create an Access Basic function that determines the statistical median, open a new or previously created module and enter the following code:

Note: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

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

Option Explicit

'=====
```

```
' Create the following Median() function in the module
' The following function will initialize:
' - A variable for the database object.
' - A variable for the snapshot.
' - Sets the database object to the opened database.
' - Creates a snapshot based on the database object.
' - A variable for the record count, the for statement,
' two field values and an offset. This function requires the
' table name and field to be analyzed to be include as argument
' strings.
'=====
```

```
Function Median (tName$, fldName$) As Single
 Dim MedianDB As Database
 Dim ssMedian As Snapshot
```

```

Dim RCount%, i%, x%, y%, OffSet%

Set MedianDB = CurrentDB()
Set ssMedian = MedianDB.CreateSnapshot("SELECT [" & fldName$ & _
 "]" FROM [" & tName$ & "]" ORDER BY [" & fldName$ _
 & "];")

ssMedian.MoveLast
RCount% = ssMedian.RecordCount
x% = RCount% Mod 2

If x% <> 0 Then
 OffSet% = ((RCount% + 1) / 2) - 2
 For i% = 0 To OffSet%
 ssMedian.MovePrevious
 Next i
 Median = ssMedian(fldName$)
Else
 OffSet% = (RCount% / 2) - 2
 For i% = 0 To OffSet%
 ssMedian.MovePrevious
 Next i
 x% = ssMedian(fldName$)
 ssMedian.MovePrevious
 y% = ssMedian(fldName$)
 Median = (x% + y%) / 2
End If
ssMedian.Close
MedianDB.Close
End Function

```

#### How to Use the Median() Function

---

Create a form with text box controls that will reflect all measures of central tendency of a data set. In the ControlSource property for the text box control, enter

```
=Median("<TableName>", "<FieldName>")
```

The value of this control will be the median of the data set. Another way to use this function is to call it from within another function that compares the median from different data sets. For example,

```

Function CompareMedians()
 Dim MyDB as Database
 .
 .
 .
 X = Median("<TableName>", "<FieldName>")
 Y = Median("<TableName>", "<FieldName>")
 If X > Y Then Debug.Print "The median for X is greatest."
End Function

```

Additional reference words: 1.00 1.10 2.00 statistics

KBCategory: kbprg

KBSubcategory: PgmOthr



**INF: Using Column Property of Combo Box to Update Text Box**  
**Article ID: Q93138**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can display multiple fields in a combo or list box on a form or report even when those fields come from a table that is not bound to the form or report. You can also update controls such as text boxes with new information based on what a user selects from a combo or list box.

MORE INFORMATION

=====

To accomplish these objectives, use one of these techniques:

- Use Row Fix-up in forms. For more information on Row Fix-up, query on the following words here in the Microsoft Knowledge Base:

Row and Fix-up

- Use multiple DLookup() functions in forms and reports. For more information on the DLookup() function, query on the following words here in the Microsoft Knowledge Base:

DLookup and forms and reports

- Use the Column property of a multiple-column combo box to update a text box control with new information as focus moves from row to row in the combo box. The next section describes this technique in detail.

Using the Column Property of a Combo Box

-----

By assigning the Column property of a multiple-column combo or list box to a text box, you can display one column from the current combo box selection in the text box. Microsoft Access automatically updates the text box when a selection is made from the combo box. The following example demonstrates how to do this:

1. Create your form or report based on the appropriate table or query.
2. Create a combo or list box that retrieves information from more than one field. For example, you might use the following multiple-column Select statement as the RowSource for a combo or list box to display information from several columns in the Categories table:

```
Select [Category Id], [Category Name], [Description] from
Categories Order by [Category Name];
```

For this example, set the ColumnCount property to 3, and set the ColumnWidth property to an appropriate size for the combo or list box. Once you have the combo or list box sized correctly and defined to return multiple fields, you can use the Column property to display the current selection in a text box control. Choose one entry in the list box first or it will return a null.

The Column property uses a reference argument to refer to a specific column in the multiple-column combo or list box. Use Column(0) to refer to the first column, Column(1) to refer to the second column, and so on.

This example uses Column(1) to refer to [Category Name]--the second column in the combo box.

3. To display the [Category Name] of the current combo box selection, create a text box control. Make the text box a calculated control by defining the following expression as the ControlSource for the text box:

```
=[<ComboControlName>].Column(1)
```

<ComboControlName> is the name of the combo box. The Column property will make the text box (calculated control) read-only.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 9, "Designing Forms," pages 233-245

Microsoft Access "User's Guide," version 2.0, Chapter 16, "Customizing Forms," "Creating a List Box or Combo Box with a Wizard"

For more information on the Column property, search for "combo box: properties" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsCmbo



**INF: Using Data from MS Access in a WordPerfect Mail Merge**  
**Article ID: Q104916**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - WordPerfect for DOS, versions 5.1 and 6.0
  - WordPerfect for Windows, versions 5.1 and 5.2
- 

SUMMARY

=====

The following information describes how to use data contained in a Microsoft Access database to perform a mail merge in WordPerfect for DOS and WordPerfect for Windows.

MORE INFORMATION

=====

WordPerfect for Windows and WordPerfect for DOS cannot read data from Microsoft Access in its native format. To use data from a Microsoft Access database in a WordPerfect mail merge, the data must first be exported from Microsoft Access in a format that WordPerfect recognizes as a valid mail merge data file (also called a "secondary file").

The following information discusses how to export data from Microsoft Access for use in a WordPerfect mail merge. Further, information is provided to demonstrate how to perform a simple mail merge with the exported data in WordPerfect for DOS and WordPerfect for Windows.

Exporting Data from Microsoft Access for Use in WordPerfect

---

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. From the File menu, choose Export.
3. From the Data Destination list, select Text (Delimited), dBASE III or dBASE IV and then choose OK.
4. Select Customers and then choose OK.

TIP: If you are not using Microsoft Access version 2.0, you can export data from a query to a delimited text file directly. For more information about this method, query on the following words here in the Microsoft Knowledge Base:

export and query and text and file

5. In the File Name box, type the name that you want for the exported file. In the Directories box, select the directory that you want the file to be placed in. You may want to place the file in your WordPerfect directory.
6. If you selected the Text (Delimited) data destination, the Export Text

Options will appear. Choose Options to see an expanded list of export options.

7. In the Text Delimiter box, select {none}.
8. In the Field Separator box, select {tab}.

TIP: If you plan to regularly export data to WordPerfect, consider saving these Export Text Options with a specification name to prevent having to manually specify these options each time. A specification can be used with the TransferText macro action to automate the export process further. See the Microsoft Access "User's Guide" for more information on using export specifications and the TransferText macro action.

9. Choose the OK button. Microsoft Access begins the export process, placing the data in the file specified in step 5.

#### Performing a WordPerfect Mail Merge with Microsoft Access Data

---

A WordPerfect merge is the process of combining information from two documents: a "primary file" (or form file) and a "secondary file" (or data file) to create a single document.

The primary file, which is typically a form letter, is used to direct the merging process. The secondary file contains records of information, typically names and addresses, used to fill in places in the primary file. The primary file contains placeholders (fields) that indicate where specific information, such as a first name or an address, from the secondary file should be merged in.

The placeholders appear in the primary file starting with {FIELD} followed by a number indicating which field in the secondary file to use. The placeholder ends with a tilde character (~). For example, {FIELD}2~ would be the placeholder for the second field in the secondary data file. In the CUSTOMER.TXT file exported earlier, this is the Company Name field.

NOTE: In WordPerfect version 6.0 for DOS the field looks like FIELD(2).

The fields appear in the secondary file in the same order as they appear in Microsoft Access. The fields in the Customers table in the NWIND database are ordered as follows:

1. Customer ID
2. Company Name
3. Contact Name
4. Contact Title
5. Address
6. City
7. Region
8. Postal Code
9. Country
10. Phone
11. Fax

## Creating the WordPerfect Primary Form File

---

### WordPerfect 5.1 for DOS:

1. Press F7, No, and then No again to create a new WordPerfect document.
2. Press SHIFT+F9, then press 1 (or F) for Field.
3. At the Enter Field: prompt, type the number indicating the field you want to reference from the secondary data file, then press ENTER.
4. Repeat steps 2 and 3 to create the following primary file:

```
{FIELD}2~
{FIELD}3~
{FIELD}5~
{FIELD}6~, {FIELD}7~ {FIELD}8~
```

Dear {FIELD}3~,

Thank you for your inquiry regarding Northwind Trader products...

Sincerely,

Andrew Fuller  
Vice President, Sales

5. Press F10. At the Document To Be Saved: prompt, type "PRIMARY.DOC" (without the quotation marks) and press ENTER.

### WordPerfect 6.0 for DOS:

1. From the File menu, choose New to create a new WordPerfect document.
2. From the Tools menu, choose Merge, then choose Define.
3. Press 1 (or F) to choose Form and press OK.
4. The Merge Codes (Form File) dialog box is now displayed. Choose OK.
5. At the Field: prompt, type the number indicating the field you want to reference from the secondary data file, then press ENTER.
6. Repeat steps 2 and 3 to create the following primary form file:

```
FIELD(2)
FIELD(3)
FIELD(5)
FIELD(6), FIELD(7) FIELD(8)
```

Dear FIELD(3),

Thank you for your inquiry regarding Northwind Trader products...

Sincerely,

Andrew Fuller  
Vice President, Sales

7. From the File menu, choose Save As, type "PRIMARY.DOC" (without the quotation marks) in the Filename: box and then press ENTER.

WordPerfect 5.x for Windows:

1. From the File menu, choose New to create a new WordPerfect document.
2. From the Tools menu, choose Merge (CTRL+F12), then choose Field.
3. In the Insert Merge Code dialog box, type the number indicating the field you want to reference in the secondary file, then choose the OK button.
4. Repeat steps 2 and 3 to create the following primary file:

```
{FIELD}2~
{FIELD}3~
{FIELD}5~
{FIELD}6~, {FIELD}7~ {FIELD}8~
```

Dear {FIELD}3~,

Thank you for your inquiry regarding Northwind Trader products...

Sincerely,

Andrew Fuller  
Vice President, Sales

5. From the File menu, choose Save As, type "PRIMARY.DOC" (without the quotation marks) in the Save As box, and choose the Save button.

Merging the Primary and Secondary Files in WordPerfect

---

WordPerfect 5.1 for DOS:

1. Press F7, No, then No again.
2. Press CTRL+F9, then press 1 (or M) for Merge.
3. Type "PRIMARY.DOC" (without the quotation marks) at the Primary file: prompt and press ENTER.
4. When the prompt for the secondary file appears, press CTRL+F5 to activate the DOS Text Delimited File: prompt.
5. Type "CUSTOMER.TXT" (without the quotation marks) and press ENTER. The Merge: DOS Text File menu screen is activated.
6. Choose 1 (or F) for Field delimiters.

7. At the Begin prompt, delete any contents that appear then press the DOWN ARROW key.
8. At the End prompt, press the TAB key so that [Tab] appears, then press the DOWN ARROW key.
9. Choose 2 (or R) for Record delimiters.
10. At the Begin prompt, delete any contents that appear, then press the DOWN ARROW key.
11. At the End prompt, press CTRL+M so that [CR] appears, then press the DOWN ARROW key.
12. Press ENTER to begin the merge.

WordPerfect 6.0 for DOS:

1. From the File menu, choose New.
2. From the Tools menu, choose Merge, then choose Run. The Run Merge dialog box appears.
3. Type "PRIMARY.DOC" (without the quotation marks) in the Form File: box.
4. Type "CUSTOMER.TXT" (without the quotation marks) in the Data File: box.

NOTE: You may want to include the full drive and path with the file names in the Primary and Secondary File boxes if these files are not located in the WordPerfect directory.

5. Select DOS Delimited Text from the File Format list and choose the Select button. The Delimited Text Options dialog box appears.
6. Choose 1 (or F) for Field Delimiter:, press F5, select Tab, then press the DOWN ARROW key. ([Tab] should now appear in the box.)
7. In the Record Delimiter: field, make sure the selected value is [CR][LF]. Press F5 and select these items from the list if needed. Press the DOWN ARROW key.
8. In the Field Encapsulate Character: box, delete any entry and press the DOWN ARROW key until the OK button is selected.

NOTE: You may want to select the Save Setup Options check box if you intend to merge on a regular basis to avoid having to reset these settings in the future.

9. Press the OK button, then press the Merge button to begin the merge.

WordPerfect 5.x for Windows:

1. From the File menu, choose New.

2. From the Tools menu, choose Merge, then choose Merge again. The Merge dialog box appears.
3. Type "PRIMARY.DOC" (without the quotation marks) in the Primary File: box.
4. Type "CUSTOMER.TXT" (without the quotation marks) in the Secondary File: box.  
  
NOTE: You may want to include the full drive and path with the file names in the Primary and Secondary File boxes if these files are not located in the WordPerfect directory.
5. Check the ASCII Delimited Text (DOS) check box, or press ALT+A.
6. Choose the OK button. The Text File Delimiters dialog box appears.
7. Under Field Delimiters, leave the Begin: box empty. In the End: box, delete the current contents and choose [TAB] from the list that appears pressing the "<" button located immediately to the right of the End: box.
8. Under Record Delimiters, leave the Begin: box empty. In the End: box, delete the current contents and choose [CR] from the list that appears pressing the "<" button located immediately to the right of the End: box.
9. Choose the OK button to begin the merge.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, chapter 9

Microsoft Access "User's Guide," version 1.x, chapter 4

"WordPerfect for DOS Reference," version 5.1, pages 420-431

"WordPerfect for DOS Workbook," version 5.1, pages 237-246

"WordPerfect for DOS Reference," version 6.0, pages 412-432

"WordPerfect for DOS Workbook," version 6.0, pages 195-204

"WordPerfect for Windows Reference," version 5.1, pages 300-308

"WordPerfect for Windows Workbook," version 5.1, pages 203-210

Additional reference words: 1.00 1.10 2.00 WP51 WPWIN print merge

KBCategory: kbinterop

KBSubcategory: IntpOthr

**INF: Using Date/Time Data Type and Calculating Elapsed Time**  
**Article ID: Q88657**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article describes how Microsoft Access stores data with a Date/Time data type, and demonstrates how to calculate elapsed time.

MORE INFORMATION

=====

Microsoft Access uses double-precision, floating-point numbers (up to 15 decimal places) to store date/time values. Using this numbering system, you can add, subtract, and compare date/time values as you would other numeric values. The integer portion of the double-precision number represents the date; the decimal portion represents the time.

The acceptable range for date information is January 1, 100 A.D. (-657,434) through December 31, 9999 A.D. (2,958,465), where January 1, 1900 is 2. Negative numbers represent dates prior to December 30, 1899.

The acceptable range for time information is between 0:00:00 and 23:59:59, or 12:00:00 A.M. and 11:59:59 P.M., inclusive. Time values representing 24 hours are stored internally as double-precision fractional numbers between 0.0 and 0.99999. Below are some examples of these fractional numbers:

2.0 represents 1/1/1900 12:00:00 A.M.

0.25 represents 6:00:00 A.M. (0.25 \* 24 hours = 6 hours)

33914.125 represents 11/6/1992 3:00:00 A.M. (0.125 \* 24 hours = 3 hours)

24794.84375 represents 11/18/67 8:15:00 P.M. (0.84375 \* 24 hours = 20.25 or 20 hours and 15 minutes (.25 \* 60 minutes = 15 minutes)

Using Null Values

-----

Microsoft Access recognizes when a portion of the data in a date/time field is missing. For example, if you enter only a time value, Microsoft Access does not display the false (0) date value of 12/30/1899. Calculations that include a null date/time field return a null value.

Formatting and Storing Date/Time Values

-----

Date/time fields can show the date, the time, or both, depending on how you format and display the date/time field. Microsoft Access always stores both the date and the time.

For more information on formatting, search for "Format" then "Format Property" using the Microsoft Access Help menu.

#### Comparing Dates Using Functions

---

Microsoft Access also uses double-precision, floating-point numbers to calculate functions that contain date or time arguments. This method can occasionally yield unexpected results.

For example, if you test the following expression in the Immediate window of a module, the result is 0 (false), even if today's date is 1/31/92:

```
?Now()=DateValue("1/31/92")
```

This function returns -1 (true) only when the current date and time are 1/31/92 12:00:00 A.M. This occurs because the Now() function returns the double-precision number of the current date and time. This number may not be equal to the integer number of the date only, which is returned by the DateValue() function.

To avoid this problem, use either of the following statements to compare today's date with another date:

```
?Date()=DateValue("1/31/92")
?Int(Now())=DateValue("1/31/92")
```

The Int() function strips the fractional portion of the double-precision number from the Now() function, returning only the number representing the date.

#### Computing and Displaying Elapsed Time

---

Below are two sample functions, HoursTest() and ElapsedTest(). They return the amount of elapsed time between two given date/time values, using different display formats. Both functions require you to specify time1 and time2 as parameters.

The first example, HoursTest(), displays the elapsed time in hours, minutes, and seconds:

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when recreating this code in Access Basic.

```
Function HoursTest()
' This function displays the total hours elapsed in this format:
' 48:57:04
 Time1 = CVDDate("1/3/92 4:05:06")
 Time2 = CVDDate("2/2/92 5:02:10")
 HoursTest = Format(Int(Abs(Time1 - Time2) * 24), "###:") + _
```



```
 Format(Abs(Time1 - Time2), "nn:ss")
End Function
```

The HoursTest() function uses the CVDate() function to convert the string expression to a date value. The HoursTest() function then takes the absolute value of the difference between the two date/time values and multiplies this new value by 24:

```
Abs(Time1 - Time2) * 24
```

HoursTest() uses the Int() function to return only the integer portion of the date/time value. The resulting integer is the total number of hours elapsed:

```
Int(Abs(Time1 - Time2) * 24)
```

The first Format() function returns the elapsed hours. The second Format() function takes the absolute value of the difference between the two time values and uses the "nn:ss" argument to capture only the minutes and seconds. HoursTest() concatenates the results of two Format() functions, as follows:

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

The second example, ElapsedTest(), adds the number of elapsed days to the resulting value:

```
Function ElapsedTest()
' This function displays the elapsed time as the total number of
' days, hours, minutes, and seconds. For example: 4 Days 01:04:05 Hours.
 Time1 = CVDate("1/31/92 4:01:05")
 Time2 = CVDate("2/4/92 5:05:10")
 ElapsedTime = Abs(Time1 - Time2)
 ElapsedDays = Int(ElapsedTime)
 ElapsedPartialDay = (ElapsedTime - ElapsedDays)
 ElapsedTest = ElapsedDays & " Days " & Format(ElapsedPartialDay, _
 "HH:NN:SS") & " Hours"
End Function
```

The ElapsedTest() function uses a process similar to that used by HoursTest(), but displays its result in a different format.

#### Microsoft Excel Dates vs. Microsoft Access Dates

-----

Microsoft Excel also uses double-precision numbers to store date/time values, but Microsoft Excel values are limited to the range 1/1/1900 through 12/31/2078. The Microsoft Excel date and time system begins with the double-precision number 1.0, which represents 1/1/1900 00:00:00 (12:00:00 A.M.).

Originally, Microsoft Access supported the same date range as Microsoft Excel (1900-2078). But when Microsoft extended the date range in Microsoft Access to 100-9999, the 1900 leap year anomaly was corrected. This means that for dates between 1/1/1900 and 2/29/1900, the internal integer portion of the date values for Microsoft Access

and Microsoft Excel differ by one.

Additional reference words: 1.00 1.10 2.00 date/time fractional

KBCategory: kbusage

KBSubcategory: TblDatyp

**INF: Using DLookup() to Look Up Values in Forms and Reports**  
**Article ID: Q93025**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can display multiple fields in a combo or list box on a form or report even when those fields come from a table that is not bound to the form or report. You can also update controls such as text boxes with new information based on what a user selects from a combo or list box.

To accomplish these objectives, use one of these three techniques:

1. Use row fix-up in forms. For more information on row fix-up, query on the following words here in the Microsoft Knowledge Base:

"row fix-up" and "multiple" and "one to many"

2. Use multiple DLookup() functions in forms and reports. See the "More Information" section below.

3. Use the Column property of a multiple-column combo box to update a text box control with new information as focus moves from row to row in the combo box. For more information on this, query on the following words here in the Microsoft Knowledge Base:

"column property" and "combo box" and "update"

MORE INFORMATION

=====

DLookup() Function

-----

Here is the syntax for DLookup():

```
DLookup(expr, domain, [criteria])
```

The DLookup() function returns a value from a field in a specified set of records called the domain. The DLookup() function specifies the criteria for the domain. To make the domain dependent on one or more values listed in controls on a form or report, refer to the controls in the DLookup() criteria argument.

The following example looks in the Employees table (the domain) and returns the Last Name of the Employee ID specified in the [ControlName] control on the Form.

NOTE: In the following sample code, an underscore ( \_ ) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
=DLookup("[Last Name]", "Employees", "[Employee ID] =_
Forms!FormName![ControlName]")
```

For reports, use Reports!ReportName![ControlName].

DLookup() returns one value from a single field even if more than one record satisfies the criteria. If no record satisfies the criteria or if the domain contains no records, DLookup() returns Null.

#### REFERENCES

=====

For more information on the DLookup() function, search for "DLookup" then "DLookup Function" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 functions

KBCategory: kbusage

KBSubcategory: FmsHowto

**INF: Using Find Method to Find a Quotation Mark or Apostrophe**  
**Article ID: Q104823**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article demonstrates how to use the Find method to search for text strings containing quotation marks (") or apostrophes ('). Searching for text strings containing these characters requires special syntax because the quotation mark and the apostrophe are used as delimiters in Access Basic.

MORE INFORMATION

=====

Notes

-----

- This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.
- In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

How to Find an Apostrophe

-----

The following examples use the Customers table in the sample database NWIND.MDB. Several of the company names in the Customers table contain apostrophes. If you are recreating these examples in Microsoft Access version 2.0, replace the company name "Babu Ji's Exports" in the examples with the company name "B's Beverages."

To find a specific company, you would normally use the following syntax:

```
Function FindaPost1 ()
 Dim MyDB As Database, MyDynaset As Dynaset
 Set MyDB = CurrentDB()
 Set MyDynaset = MyDB.CreateDynaset("Customers")
 'Find this company.
 MyDynaset.FindFirst "[company name]='Around the Horn'"
 If Not MyDynaset.nomatch Then
 MsgBox MyDynaset.[company name]
 Else
 MsgBox "No Match"
 End If
```

```
MyDynaset.Close
End Function
```

However, the following syntax will generate a syntax error because of the apostrophe in the company name:

```
Function FindaPost2 ()
 Dim MyDB As Database, MyDynaset As Dynaset
 Set MyDB = CurrentDB()
 Set MyDynaset = MyDB.CreateDynaset("Customers")
 'The following line will generate an error.
 MyDynaset.FindFirst "[company name]='Babu Ji's Exports'"
 If Not MyDynaset.nomatch Then
 MsgBox MyDynaset.[company name]
 Else
 MsgBox "No Match"
 End If
 MyDynaset.Close
End Function
```

To search for a company name that includes an apostrophe, replace the single quotation marks around the company name with two sets of double quotation marks, as in the following example:

```
Function FindaPost3 ()
 Dim MyDB As Database, MyDynaset As Dynaset
 Set MyDB = CurrentDB()
 Set MyDynaset = MyDB.CreateDynaset("Customers")
 'Find this company (with an apostrophe in the name).
 MyDynaset.FindFirst "[company name]=""Babu Ji's Exports""
 If Not MyDynaset.nomatch Then
 MsgBox MyDynaset.[company name]
 Else
 MsgBox "No Match"
 End If
 MyDynaset.Close
End Function
```

How to Find a Quotation Mark

-----

The following example uses the Employees table in the sample database NWIND.MDB. The Notes field for some employees contains quotation marks.

If you use the following syntax to find a string containing quotation marks, a compile error will be generated:

```
Function FindQuotel ()
 Dim MyDB As Database, MyDynaset As Dynaset
 Set MyDB = CurrentDB()
 Set MyDynaset = MyDB.CreateDynaset("Employees")
 'the following line generates a compile error.
 MyDynaset.FindFirst "[notes] like '*The art of the cold_
 call.*'"
 If Not MyDynaset.nomatch Then
 MsgBox MyDynaset.[Last Name]
 Else
```

```
 MsgBox "No Match"
 End If
 MyDynaset.Close
End Function
```

To search for an item containing quotation marks, concatenate Chr(34) with the quotation marks, as in the following example:

```
Function FindQuote2 ()
 Dim MyDB As Database, MyDynaset As Dynaset
 Set MyDB = CurrentDB()
 Set MyDynaset = MyDB.CreateDynaset("Employees")
 'find this note that contains a quote.
 MyDynaset.FindFirst "[notes] like '*' & Chr(34) & "The art of_
 the cold call." & Chr(34) & '*'"
 If Not MyDynaset.nomatch Then
 MsgBox MyDynaset.[Last Name]
 Else
 MsgBox "No Match"
 End If
 MyDynaset.Close
End Function
```

#### REFERENCES

=====

For more information about the Find method, search for "find method" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmObj

**INF: Using IIf() in Crosstab Query to Limit Column Headings**  
**Article ID: Q120883**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can use the IIf() function to group values under a small number of headings in a crosstab query. This article contains an example demonstrating how to group records by country and count the number of orders placed for each customer.

MORE INFORMATION

=====

The following example uses the IIf() function to list companies whose name begins with the letter "A" individually, and group all other companies under the "OTHERS" column heading:

1. Open the sample database NWIND.MDB.
2. Create the following new crosstab query based on the Orders and Customers tables:

Query: MyQuery

-----  
Type: crosstab query  
Join: Orders.[Customer ID]<->Customers.[Customer ID]  
Field Name: Country  
    Table Name: Customers  
    Total: Group By  
    Crosstab: Row Heading  
Field Name: IIf([Company Name] Like "A\*", [Company Name], "OTHERS")  
    Total: Group By  
    Crosstab: Column Heading  
Field Name: Order ID  
    Table Name: Orders  
    Total: Count  
    Crosstab: Value

3. Run the query.

Note that you could use this technique to list data for the current month in one column, and the sum of prior months' data in another.

REFERENCES

=====

For more information about the IIf() function, search for "IIf" then "IIf Function" using the Microsoft Access Help menu.

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



Queries," pages 150-155

Microsoft Access "User's Guide," version 1.1, Chapter 6, "Designing Select Queries," pages 154-159

Microsoft Access "User's Guide," version 2.0, Chapter 12, "Advanced Queries," pages 279-285

Microsoft Access "Language Reference," version 1.0, pages 248-249

Additional reference words: 1.00 1.10 2.00 iif fixed

KBCategory: kbusage

KBSubcategory: QryCross

**INF: Using Microsoft Access 2.0 Databases with Visual Basic**  
**Article ID: Q112104**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Visual Basic version 3.0
- 

SUMMARY

=====

This article describes how to use Microsoft Access version 2.0 databases in Visual Basic version 3.0.

MORE INFORMATION

=====

Microsoft Visual Basic version 3.0 includes dynamic-link libraries (DLLs) that you can use to communicate with the Microsoft Access Jet database engine version 1.1 (Jet 1.1) that is used by Microsoft Access version 1.1. Jet 1.1 is also included with Visual Basic version 3.0 and can be freely distributed with applications created with it; there is no need to purchase Microsoft Access to obtain Jet 1.1.

You can use the DLLs in Visual Basic 3.0 to create Jet 1.1 applications. However, these DLLs will not communicate with the Microsoft Access Jet database engine version 2.0 (Jet 2.0) that is used in Microsoft Access version 2.0. Visual Basic version 3.0 includes Jet 1.1 only; it does not include Jet 2.0. Jet 2.0 consists of the MSAJT200.DLL file, plus the various ISAM DLLs (PDX200.DLL, XBS200.DLL, and so on).

So that you can use Microsoft Access version 2.0 databases in Visual Basic version 3.0, a "compatibility layer," or series of files, was created to replace these DLLs. For example, the compatibility layer directly replaces the VBDB300.DLL file with a new version with the same name. When you use the compatibility layer in conjunction with Jet 2.0, you can use Microsoft Access version 2.0-created databases.

Note that it is not necessary to use the compatibility layer and Jet 2.0 combination to create database applications with Visual Basic 3.0; you can continue to use the DLLs included with Visual Basic 3.0 as before with no problem, as well as Jet 1.1 databases. Also, you can continue to use your current Jet 1.1 applications, even if you convert to Microsoft Access version 2.0, since it can still read Jet 1.1 databases. Just do not convert your Jet 1.1 databases to Jet 2.0 if you do not want to use the compatibility layer.

The compatibility layer is required only if you want to create applications that use Jet 2.0 and the new features it contains, such as:

- Data-definition queries
- Engine-level validation
- Programmatic access to security
- DAO (data access objects)
- Cascading deletes

- Temporary queries

The compatibility layer and Jet 2.0 will be available in the Microsoft Access Developer's Toolkit (ADT) version 2.0. The compatibility layer only (without the Jet files) will be available in the Microsoft Office Developer's Kit (ODK).

Additional reference words: 2.00 mapping vb interoperability

KBCategory: kbinterop

KBSubcategory: IntpOthr

## INF: Using Microsoft Access Macro to Quit Windows

Article ID: Q89596

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

It is possible to quit Microsoft Windows from within a Microsoft Access application by using the Windows application programming interface (API) function `ExitWindows()`. To do this, you must create a module that declares the `ExitWindows()` function and then can call the Microsoft Access function in a macro using the `RunCode` action.

### MORE INFORMATION

=====

The call to the Windows dynamic-link library (DLL) behaves in the same way as does the `Exit` command from the File menu in Program Manager. Each application must agree to be closed; for example, if you choose `Cancel` when you are prompted to save a file, your exit request is also canceled.

To create a Microsoft Access function to quit Windows, use the following steps:

1. From the File menu, choose `New` and then `Module`.
2. In the `Declarations` section, type the following statements:

NOTE: In the following sample code, an underscore (`_`) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

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

3. Create a Microsoft Access function that calls the `ExitWindows()` function, as follows:

```
Function ExitNow ()
 DimRetVal as Integer
 RetVal = ExitWindows(0, 0)
End Function
```

4. Save the module.
5. From the File menu, choose `New` and then `Macro`. Add the following action and its properties:

| Action | Function Name |
|--------|---------------|
| -----  | -----         |

```
RunCode ExitNow()
```

6. Save the macro.

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

Microsoft does not recommend exiting from Microsoft Access using Windows API calls, though exiting can be safely accomplished. In some cases, temporary files can be left in the Windows Temp directory; however, you can safely delete the Temp files. To automatically remove temp files each time the computer is started, add the following line to the AUTOEXEC.BAT file:

```
del c:\windows\temp\~*.tmp
```

NOTE: This line should be placed before the line that begins "Windows...." This line assumes that c:\windows\temp is the location of the Windows Temp directory specified by the SET TEMP statement in the AUTOEXEC.BAT file.

#### REFERENCES

=====

"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

Additional reference words: 1.00 1.10 2.00 SDK

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: Using Microsoft Word to Edit and Spell Check Memo Field**  
**Article ID: Q96544**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You can use a word processor to edit memo fields stored in Microsoft Access tables. Then you can check the spelling and grammar in your data. This article describes how to use Microsoft Word for Windows to edit memo fields.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access.

MORE INFORMATION

=====

Note the following information before trying the example:

- Microsoft Word for Windows must have the DOS TEXT Import/Export filter installed.
- Only the most basic formatting will be preserved because memo fields do not support tabs, margins, fonts, or other fancy formatting.
- Memo fields are limited to 32,000 characters (about 8-12 pages in Microsoft Word for Windows). Excess characters will be truncated.
- The subdirectory containing WINWORD.EXE should be included in the path statement in your AUTOEXEC.BAT file or you can include the path in the following line of code in the example that follows:

```
X% = Shell("c:\word6\WINWORD.EXE TEMP.TXT", 3)
```

- This example is presented in the simplest terms. If you open the Microsoft Word document and return to Microsoft Access before you save the file, you will receive an "Access denied" error message when the Access Basic code tries to open the document that is already currently open in Microsoft Word.

The following example uses the Employees table in the NWIND.MDB database.

1. Create the following function in a new module.

```
'*****
'Declarations section of the module
'*****
```

```
Option Explicit
```

```

'=====
'The following function EditMemo() calls Microsoft
'Word for Windows
'=====
Function EditMemo (MyEditBox As Control)
Dim X%, LineInFile$, NewLine$, FileNum%, CarriageReturn_LineFeed$

CarriageReturn_LineFeed$ = Chr$(13) & Chr$(10)
FileNum% = FreeFile

Open "TEMP.TXT" For Output As #FileNum%

'save the text to temporary file for Microsoft Word for WINDOWS

If Not IsNull(MyEditBox) Then
 Print #FileNum%, MyEditBox
Else
 MsgBox "There is no text to edit. Abort."
 Exit Function
End If

Close #FileNum%

X% = Shell("WINWORD.EXE TEMP.TXT", 3)

MsgBox "Press Any Key to Return" ' required to pause the function

'Assume completion of the edits in Microsoft Word, now read the text
'and place it back into the edit control.

FileNum% = FreeFile
Open "TEMP.TXT" For Input As #FileNum%

NewLine$ = ""
Do While Not EOF(FileNum)
 Line Input #FileNum%, LineInFile$
 If NewLine$ <> "" Then
 NewLine$ = NewLine$ & CarriageReturn_LineFeed$
 End If
 NewLine$ = NewLine$ & LineInFile$
Loop
MyEditBox = NewLine$

Close #FileNum%

End Function

```

2. Create a new blank form and base it on the Employees table.
3. Place two text box controls (Last Name and Notes) on the form and assign the following properties:

```

Object: Text box

ControlName: Last Name
ControlSource: Last Name

```

Object: Text box

-----  
ControlName: Notes  
ControlSource: Notes  
OnDblClick: =EditMemo([Notes])

4. Save the form and switch to Form view.
5. Double-click the memo field to launch Microsoft Word for Windows. You will be presented with the Convert File dialog box. Select Text Only and then choose the OK button.
6. Perform the operations you want.
7. Exit from Microsoft Word for Windows and save the document.
8. Microsoft Access loads the changes back into the memo field.

For a more elegant solution, one that replaces the MsgBox routine, query on the following words here in the Microsoft Knowledge Base:

b\_waccess and waitshell

Additional reference words: 1.00 1.10 2.00 spell check  
KBCategory: kbprg kberrmsg  
KBSubcategory: PgmOthr



## **INF: Using OLE Automation to Change a Graph's Type**

Article ID: Q120418

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SUMMARY

=====

This article describes how to use OLE Automation to dynamically change a chart's type. For example, you could use this method to change a bar chart to a column chart as you print a report or view records in a form.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

### MORE INFORMATION

=====

The following example demonstrates how to use OLE Automation to change a chart's type at run time:

1. Open the sample database NWIND.MDB.
2. Create a new form based on the Sales Totals query. Choose the Form Wizards button.
3. In the "Which Wizard do you want?" screen, select Graph, and then choose OK.
4. In the Available Fields box, select Name and then choose the ">" button, then select SumOfSubtotal and choose the ">" button.
5. Choose the Next button twice, then choose the Finish button.
6. View the form in Design view.
7. Set the embedded graph's Name property to MyGraph.
8. Add an option group to the form. Set the option group's Name property to Graph\_Type.
9. Add the following six toggle buttons to the option group:

```
Name: Area
 OptionValue: 1
Name: Bar
 OptionValue: 2
Name: Column
 OptionValue: 3
Name: 3D_Area
 OptionValue: -4098
```

Name: 3D\_Bar  
OptionValue: -4099  
Name: 3D\_Column  
OptionValue: -4100

10. Set the option group's AfterUpdate property to the following event procedure:

```
Sub Graph_Type_AfterUpdate ()
 Me![MyGraph].Object.Application.Chart.Type = Me![Graph_Type]
End Sub
```

11. View the form in Form view.

12. Click the various toggle buttons. The chart will change to match your selection.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 8, "Events," pages 183-201, and Chapter 13, "Communicating with Other Applications," pages 281-297

Additional reference words: 2.00 Graph5  
KBCategory: kbusage  
KBSubcategory: FmsOthr

**INF: Using SetSysModalWindow API Call in Access Basic**  
**Article ID: Q88171**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

In Microsoft Access Basic, you can make an application modal so that users cannot switch to other Windows-based programs while the Microsoft Access application is active. This functionality is achieved with the following Windows application program interface (API) functions:

SetSysModalWindow

-and-

FindWindow

This article discusses the use of the SetSysModalWindow API function and provides an example of its use in Microsoft Access Basic.

MORE INFORMATION

=====

Microsoft Windows is designed so that the user can switch between applications without terminating one program to run another. There may be times, however, when a Microsoft Access application needs to restrict the user from switching to another application. Examples of this are a security system or time-critical application that needs to run uninterrupted.

SetSysModalWindow Usage Notes

-----

Consider all of the following usage notes before making a window system-modal.

1. If an application passes the handle of the window as an argument to the SetSysModalWindow, the user is limited to that window. This prevents the user from moving to other applications or bringing up the Task List (either with the mouse or by pressing CTRL+ESC) .
2. If an application uses the SetSysModalWindow function in conjunction with the GetSystemMenu function, the Control (system) menu is disabled and the user is prevented from quitting Windows by pressing ALT+F4.
3. A window that is system-modal cannot be minimized, maximized, moved, or sized. Microsoft Access help is created with a separate Windows Help file; therefore, help windows created with the Windows Help Compiler are not available to the user.

4. When an application uses the SetSysModalWindow API function, the system-modal window must be closed before other applications or windows can be accessed. Therefore, when using this function, be sure that the application has a means to exit Microsoft Access.
5. Any child window that is created by a system-modal window becomes a system-modal window. For example, if Microsoft Access is system-modal and a form is opened, that form becomes system-modal until it is closed.

#### How to Use the SetSysModalWindow Function

-----

To use the SetSysModalWindow API function within Microsoft Access Basic, create the sample code below.

NOTE: In this sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

1. Within the Declarations Section, enter the following:

```
Option Explicit

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

Declare Function FindWindow% Lib "User" (
 ByVal lpzClassName As Any,
 ByVal lpzWindow As Any)
```

2. Create a procedure by entering the following:

```
Function System_Modal_Test ()

 Const lpClassName = "OMain"
 ' Declares the class name constant used in
 ' the FindWindow API function.

 Dim AccessHandle
 Dim Success As Integer

 ' FindWindow returns the Handle for Microsoft Access.
 AccessHandle = FindWindow(lpClassName, 0&) 'Note this is a zero

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

End Function
```

3. From the File menu, choose Save and save the module as "Modal Access" (without the quotation marks). Choose the OK button.
4. From the Run menu, choose Compile All.

5. From the View menu, choose Immediate Window. In the Immediate window, type the following, then press ENTER:

```
?System_Modal_Test()
```

After running the program, test the modal functionality by trying to select other windows or resize or move the system-modal window.

#### REFERENCES

=====

For more information on the Modal property in Microsoft Access, search for "Modal" then "Modal Property" using the Microsoft Access Help menu.

For more information on using the FindWindow API function, query on the following words here in the Microsoft Knowledge Base:

Access and FindWindow

For more information on API functions, refer to:

- "Microsoft Windows Software Development Kit," Microsoft Press, 1992
- "Programming Windows: The Microsoft Guide to Writing Applications for Windows 3," Charles Petzold, Microsoft Press, 1990
- "Programmer's Reference Library: Microsoft Windows 3.1 Guide to Programming Reference," Volumes 1-6, Microsoft Press, 1992

For more information on Windows API calls, query on the following words here in the Microsoft Knowledge Base:

handle and API and DLL and kernel and user and GDI

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: Using SHARE.EXE and VSHARE.386 with Microsoft Access 2.0**  
**Article ID: Q112125**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SUMMARY

=====

Microsoft Access version 2.0 includes a version of VSHARE.386 that is compatible with Microsoft Windows 3.1, Windows for Workgroups 3.1, and Windows for Workgroups 3.11.

If you are using applications that support object linking and embedding (OLE) 2.0, you must run either SHARE.EXE or VSHARE.386. VSHARE.386 eliminates the need for SHARE.EXE when you run Windows 3.1 or Windows for Workgroups in 386 enhanced mode. If you run Windows 3.1 in standard mode, you still need to run SHARE.EXE.

MORE INFORMATION

=====

Microsoft Access Setup will automatically install VSHARE.386 by copying VSHARE.386 into the WINDOWS\SYSTEM directory and adding this line to the [386Enh] section of the SYSTEM.INI file:

```
device=vshare.386
```

Setup will not automatically remove SHARE.EXE from the AUTOEXEC.BAT file, since it may still be required by MS-DOS applications running outside of Windows.

If SHARE.EXE is not required for any applications running in MS-DOS outside of Windows, it can be removed from the AUTOEXEC.BAT file to free up approximate 5K of conventional memory. Leaving SHARE.EXE in the AUTOEXEC.BAT file will not cause any harm.

If Windows is run in standard mode then SHARE.EXE must be in the AUTOEXEC.BAT file. Add the following line to the AUTOEXEC.BAT file to install SHARE.EXE:

```
<path>\SHARE.EXE /L:500 /F:5100
```

In the example above, substitute the drive and directory where the SHARE.EXE file is located for <path>. For example, if the SHARE.EXE file is located in the DOS directory on drive C, add the following line to your AUTOEXEC.BAT file:

```
C:\DOS\SHARE.EXE /L:500 /F:5100
```

After adding SHARE.EXE to the AUTOEXEC.BAT file, restart the computer.

Additional reference words: 2.00 WinIss  
KBCategory: kbenv

KBSubcategory: EvnOs

## **INF: Using SQL Server ODBC Driver with Sybase Servers**

**Article ID: Q101071**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

The SQL Server ODBC driver was designed to work with both Microsoft SQL Server and Sybase SQL Server computers. The ODBC organization at Microsoft has completed some limited interoperability testing with Sybase version 4.2 on the SPARC platform, using the Microsoft TCP/IP network protocol and the Sybase Net-Library for this stack, and has not uncovered any significant problems. Below are two issues to consider if you use the SQL Server ODBC driver with Sybase servers.

### MORE INFORMATION

=====

#### Issue 1

-----

The ODBC driver requires the catalog procedures that ship with Sybase SQL Server 4.2a. Until the Sybase 4.9.1 release, these catalog procedures did not ship with Sybase servers. For the ODBC driver to work with Sybase servers, the INSTCAT.SQL script must be run against the Sybase server to install the catalog procedures.

Sybase changed the system catalogs in its version 4.8 release, requiring a modified script. This script is called INSTCAT.48 and is available through Microsoft Product Support Services and on CompuServe. This script is required with Microsoft Access version 1.0 and Microsoft Visual Basic version 2.0 applications.

With Microsoft Access 1.1 and Visual Basic 3.0, the INSTCAT.SQL file applies to all versions of Microsoft SQL Server, as well as to all versions of Sybase SQL Server.

#### Issue 2

-----

The SQL Server ODBC driver that ships with Microsoft Access 1.0 and Visual Basic 2.0 comes with the Sybase Net-Library. To connect to Sybase, you needed to use Microsoft SQL Bridge or install Net-Library on each client computer.

Both Microsoft Access and Visual Basic 2.0 use the network in a way that is not typical of Windows-based applications (for instance, these applications use multiple, simultaneous connections and asynchronous calls). It is probable that there are network configurations in which Net-Library will fail with these products.

Microsoft and Sybase are now discussing how best to perform



interoperability testing with the various network configurations supported by Sybase (FTP Software, Wollongong, Novell, TCP/IP, DECnet, and so forth).

The following products are manufactured by vendors independent of Microsoft: DECnet is manufactured by Digital Equipment Corporation; FTP Software is manufactured by FTP Software, Inc.; Novell networking products are manufactured by Novell, Inc.; SPARC is manufactured by Sun Microsystems, Incorporated; Net-Library and Sybase SQL Server are manufactured by Sybase, Inc. We make no warranty, implied or otherwise, regarding these products' performance or reliability.

The CompuServe file name containing the INSTCAT.SQL and the INSTCAT.48 is either SQL100.EXE or SQL110.EXE and is located in the ODBC Connectivity library in the MSACCESS forum.

#### REFERENCES

=====

For further information about running Microsoft Access with Sybase SQL Server, search for the following word here in the Microsoft Knowledge Base:

q96468

Additional reference words: 1.00 1.10 2.00 SQL Server Oracle Net-Library  
KBCategory: kbusage  
KBSubcategory: ObcOthr

**INF: Using the DDE Item SQLText to Request the Text of a Query**  
**Article ID: Q102519**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SUMMARY

=====

Microsoft Access version 1.1 introduces the new DDE item, SQLText, which can be used to retrieve the SQL statement representing the table or query topic on which the DDE link has been established.

The SQLText item allows the DDE client to request that the SQLText be either returned as a whole or divided into substrings. The length of the substring is specified by the client. The SQLText is returned as an array of substrings.

This article describes the use of this DDE item to request the text of a query and place it in a Microsoft Excel worksheet.

MORE INFORMATION

=====

The syntax for the SQLText item name is as follows:

SQLText;<number>

<number> is the maximum number of characters for each substring. If a value for <number> is not provided, the entire text is sent as one string.

The last substring of the array may be shorter than the previous substrings if there are not enough characters.

Creating the Sample Macro

-----

NOTE: The following example assumes that you have a working knowledge of Microsoft Excel macros. For information on using macros, please refer to the Microsoft Excel documentation.

The following sample macro should be pasted or entered in a Microsoft Excel macro sheet, starting with cell A1. (Some formulas are dependent on starting at that location in the macro sheet.) Also, formulas designated in opening and closing braces ({} ) must be entered as array formulas, using the CTRL+SHIFT+ENTER key combination. For example, use the three steps below to yield the following array formula:

```
{=FORMULA.ARRAY (TRANSPOSE (SQL)) }
```

1. Go to the cell that contains the formula.
2. Remove the braces ({} ).

3. Press CTRL+SHIFT+ENTER.

To verify that the formula is entered as an array, the braces will reappear around the formula in the Microsoft Excel formula bar. For more information on using array formulas, please refer to the Microsoft Excel documentation.

```
SQLText
StringLength=50
chan=INITIATE("MSACCESS","NWIND.MDB;Query Sales for 1991")
{=FORMULA.ARRAY(REQUEST(chan,"SqlText;255"),B1:U1)}
=IF(LEN(B1)<255)
= SET.NAME("SQLLen",LEN(B1))
=ELSE()
{= SET.NAME("SQLLen",SUM(IF(ISNA(B1:U1),0,LEN(B1:U1))))}
=END.IF()
=SET.NAME("SQLPieces",INT(SQLLen/StringLength)+1)
=SET.NAME("SQL",REQUEST(chan,"SqlText;"&StringLength))
=TERMINATE(chan)
=NEW(1)
=SELECT("r1c1:r"&SQLPieces&"c1")
{=FORMULA.ARRAY(TRANSPOSE(SQL))}
=COPY()
=PASTE.SPECIAL(3)
=CANCEL.COPY()
=FORMULA.REPLACE(CHAR(13)&CHAR(10)," ",2,2,FALSE)
=COLUMN.WIDTH(,,3)
=RETURN()
```

How the Sample Macro Works

-----  
The length of the result string is set to 50, a channel is initiated to the Microsoft Access query, the SQL statement is requested in 255-character chunks, and the total length of the query is calculated.

Using SQLLen, the number of rows needed to display the query is calculated. The SQLText is requested in chunks of 50, based on the variable StringLength, and then the DDE channel is terminated.

Finally, a new worksheet is opened and the appropriate number of cells are selected. The Formula.Array() function places the SQLText in the currently selected cells. The result originally appears as a formula, but the Copy and Paste.Special() commands change it to text. The carriage returns and line feeds are changed to spaces and the ColumnWidth property is set to Best Fit.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpDde

## INF: Using the Generic Print Driver with Forms/Reports

Article ID: Q92815

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

One way to get Microsoft Access output from forms or reports into a text file is to use the Windows Generic print driver and print to a file. However, using the Generic print driver (TTY.DRV) can be a little tricky. Listed below are some hints on how to set up your forms and reports for the Generic print driver.

### MORE INFORMATION

=====

The Windows Generic print driver simulates a line-oriented printer with a fixed pitch font, like a TTY device or a line printer. Because of this, it can only print text, and it can only position the text on a grid that is 10 characters per inch in the horizontal direction and 6 lines per inch in the vertical direction. (By contrast, most Windows print drivers can position text on any pixel position.)

This means that any form or report using the Generic print driver to create a text file should be layed out so that everything lines up with this character grid.

Here are some considerations for the form/report layout:

- Ensure that each section of your report has a height that is an integral multiple of 1/6".
- Lay out the text items in your report such that each falls on 1/6" boundaries.
- Since the default hardware font for most printers is a Courier 10cpi font, you should set the font for all of your text items to be Courier 10.
- Do not use Bold fonts, as they will appear three times in your output. (The Generic driver simulates bold text by overstriking.)

Here are some problems that frequently occur when using the Generic print driver to do text output:

- Blank lines appear in the output--text items or sections are slightly farther apart than 1/6".
- Overlapped lines appear in the output--text items or sections are slightly closer together than 1/6"
- Text appears three times in the output--text is formatted as

Bold.

Additional reference words: 1.00 1.10 2.00 outputto

KBCategory: kbenv

KBSubcategory: EvnOdr

**INF: Using the INSTCAT.SQL Script with an ODBC Application**  
**Article ID: Q104601**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SUMMARY

=====

The INSTCAT.SQL script should be run on SQL Server for products that use the SQL Server driver. Because different versions of the INSTCAT.SQL script ship with different versions of the products and of the SQL Server driver, this article discusses compatibility issues.

MORE INFORMATION

=====

The INSTCAT.SQL script, when run on SQL Server, installs several stored procedures that are later used by the ODBC SQL Server driver to query the system catalog. Products such as Microsoft Visual Basic and Microsoft Access, which use the SQL Server driver for ODBC access to SQL Server, cannot be fully functional unless this script is run on the server.

The INSTCAT.SQL script may have been run on SQL Server during the installation of SQL Server itself. As a rule of thumb, we recommend that the INSTCAT.SQL script that ships with Microsoft Access or Visual Basic be used instead. (Note that this is an uncompressed file on the driver disk.) To verify which version of the INSTCAT.SQL script is running on the server, the following command can be issued from any client utility (ISQL, SAF, and so forth):

```
sp_server_info 500
```

This command returns a version string, such as 01.02.3015, in the attribute\_value column.

The table below lists the sizes, dates, and versions of the two INSTCAT.SQL scripts that are shipping with Microsoft Access version 1.1, version 2.0 and Visual Basic version 3.0:

NOTE: These INSTCAT.SQL scripts must also be run on Sybase SQL Server machines across all platforms and versions when using Microsoft Access 1.1 and greater or Visual Basic 3.0.

| Product              | Size   | Version    |
|----------------------|--------|------------|
| -----                | ----   | -----      |
| Visual Basic 3.0     | 93834  | 01.01.2806 |
| Microsoft Access 1.1 | 93834  | 01.01.2807 |
| Microsoft Access 2.0 | 100363 | 01.02.3015 |

However, the above versions of the INSTCAT.SQL script are incompatible with Microsoft Access version 1.0. As documented in the README.TXT included with the Visual Basic 3.0 and Microsoft Access 1.1 products,

if you are also using Microsoft Access 1.0 as a client for SQL Server, you should use the INSTCAT.SQL script that was included with Microsoft Access 1.0.

NOTE: Visual Basic 3.0 and Microsoft Access 1.1 will work correctly with the INSTCAT.SQL script from Microsoft Access 1.0, except that Visual Basic 3.0 will be unable to delete indexes from tables.

Sybase SQL Server is manufactured by Sybase, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.10 2.00 ADK

KBCategory: kbusage

KBSubcategory: ObcOthr

**INF: Using the Scroll Bar Control to Move Through Records**  
**Article ID: Q114552**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Developer's Toolkit version 2.0
- 

SUMMARY

=====

Included in the Microsoft Access Developer's Toolkit (ADT) are three custom OLE controls, similar to those available in Microsoft Visual Basic. Among them is the scroll bar control, which you can use to move through records in a form. You must have the ADT installed to use this control.

MORE INFORMATION

=====

This article assumes that you have the ADT installed, and that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

The following example demonstrates how to use the scroll bar control:

1. Open the sample database NWIND.MDB.
2. Open the Customers form in Design view. From the Edit menu, choose Insert Object. In the dialog box, select the Insert Control option button. A list of controls will be displayed. If the scroll bar control is not listed, choose Add Control, and then select the file MSASB20.OCX from your ADT directory. Select Scroll Bar Control and then choose OK.
3. Note the scroll bar in the top left corner of the form. Click the right mouse button on the scroll bar, and then choose Scroll Bar Control Object, then Properties. Change the orientation from horizontal to vertical, and then choose OK. Size the scroll bar appropriately.
4. Click the secondary mouse button on the scroll bar, and then choose Properties. Change the Name property to "Scrollbar1" (without quotation marks).
5. From the View menu, choose Code. Add this code to the general Declarations:

```
Option Explicit
Dim rec_count As Integer
Dim rs As Recordset
```

6. In the Object combo box, select Form. In the Procedure combo box, select Load. Add the following code to the Form Load procedure:



```

Sub Form_Load ()
' Find out how many records are in the form.
Set rs = Me.recordsetclone
rec_count = rs.recordcount

' Initialize the scroll bar properties:
Me!scrollbar1.object.Max = rec_count
Me!scrollbar1.object.Min = 0
Me!scrollbar1.object.SmallChange = 1
Me!scrollbar1.object.LargeChange = 5
End Sub

```

7. Click the secondary mouse button on the scroll bar, and then choose Build Event. Add the following code to the scroll bar's Change event:

```

Sub scrollbar1_Change ()
Dim x As Integer

' Moves to the first record in the form's dynaset, then moves
' .. forward x records, x being the value of the scroll bar.
rs.MoveFirst
x = Me!scrollbar1.object.value
rs.Move x

' Check to see if at the beginning or end of the
' .. recordset, and if so, take appropriate action.
If rs.eof Then
rs.MoveLast
End If

If rs.bof Then
rs.MoveFirst
End If

' Coordinate the form bookmark and the dynaset's bookmark.
Me.bookmark = rs.bookmark

End Sub

```

8. Add this code to the form's Current event:

```

Sub Form_Current ()
' Reinitialize the rec_count variable in case records are added.
rec_count = rs.recordcount
Me!scrollbar1.object.Max = rec_count
End Sub

```

When you use the scroll bar, the appropriate record will be displayed. However, if you move through the form using other methods, the scroll bar will not reflect your movements.

#### REFERENCES

=====

Microsoft Access "Building Applications," Chapter 11

For more information about the scroll bar control, search for "scroll bar

control" then "Scroll Bar Control" using the Scroll Bar Control Help system in the Microsoft ADT program group.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsEvt

**INF: Using the Windows 3.1 API to Connect to Network Resources**  
**Article ID: Q90862**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

The Microsoft Windows version 3.1 application programming interface (API) can be used to connect to and disconnect from network drives and printers in an Access Basic module.

MORE INFORMATION

=====

To use the API functions, follow the steps below. You will probably want to use variables for the parameters.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

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

```
'-----
Option Explicit
Declare Function WNetAddConnection% Lib "User" (ByVal lpszNetPath$, _
 ByVal lpszPassword$, _
 ByVal lpszLocalName$)

Declare Function WNetCancelConnection% Lib "User" (ByVal lpszName$, _
 ByVal fForce%)

Const WN_SUCCESS=0 ' The function was successful.
Const WN_NET_ERROR=2 ' An error occurred on the network.
Const WN_BAD_PASSWORD=6 ' The password was invalid.

Dim Results%, Force%
'-----
```

2. Create a function that makes the connection:

```
Function AddConnection ()
 ' lpszLocalName$ can be in the form D: or E: and LPT1 or LPT2.
 Results% = WNetAddConnection("\server\share", "password", "y:")
End Function
```

Some of the possible return values for Results% are WN\_SUCCESS, WN\_NET\_ERROR, and WN\_BAD\_PASSWORD.

For more information on how to obtain an unused drive resource

programmatically, search the Knowledge Base for the following keywords:

network and getdrivetype

3. Create a function that cancels the connection. The parameter `fForce%` specifies whether any open files or open print jobs on the device should be closed before the connection is canceled. If this parameter is `FALSE` and there are open files or jobs, the connection will not be canceled.

```
Function CancelConnection ()
 Force%=1
 ' NOTE: If Force%=0 and files were open,
 ' the connection will not be 'canceled.

 Results% = WNetCancelConnection("y:", 1)
End Function
```

Two of the most common return values for `Results%` are `WN_SUCCESS` and `WN_NET_ERROR`.

#### REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmApi

**INF: Using Undocumented CreateForm(), CreateReport() Functions**  
**Article ID: Q93096**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article discusses the Microsoft Access undocumented functions CreateForm() and CreateReport(). If you intend to write your own FormWizard or ReportWizard, you can use these functions to create and customize a blank form or report to which you can add controls.

This article assumes that you are familiar with Access Basic and designing forms and reports.

MORE INFORMATION

=====

CreateForm()/CreateReport() is roughly equivalent to choosing "New... Form/New... Report" from the File menu at the top of the screen. When the function is executed, a new empty form/report appears in an iconized state.

Both functions return an object value that you can use for further manipulation, and neither function requires parameters.

To use these functions, first define a form or report object variable, then assign the variable to the function name. An example of how to do this is:

```
Dim MyForm As Form
Set MyForm = CreateForm()
```

After opening the form/report in design mode after executing the commands above, you can bind the form to a table or query by modifying the form's/report's RecordSource property:

```
MyForm.RecordSource = "Categories"
```

With the form/report in design mode, you can access and change any of the other design-time properties of the form/report. You can also access and change properties of each of the form's/report's sections via the Section property. The Section property is actually an array with each array value being a reference to a form's/report's section. For forms, the Section property array is broken down as:

```
Section(0) - Detail Section
Section(1) - Form Header
Section(2) - Form Footer
Section(3) - Page Header
Section(4) - Page Footer
```

For reports, the section property array is broken down as:

- Section(0) - Detail Section
- Section(1) - Report Header
- Section(2) - Report Footer
- Section(3) - Page Header
- Section(4) - Page Footer
- Section(5) - Group Level 1 Header
- Section(6) - Group Level 1 Footer
- Section(7) - Group Level 2 Header
- ...etc.

With this information, you can customize the design of a form/report section programmatically. The following example creates a new report, hides the page footer by setting its Visible property to false, sets the Height property of the detail section, and enables its KeepTogether property:

```
Dim MyReport As Report
Set MyReport = CreateReport()
MyReport.Section(4).Visible = False
MyReport.Section(0).Height = 1760
MyReport.Section(0).KeepTogether = True
```

#### REFERENCES

=====

For more information on Microsoft Access libraries, query on the following words here in the Microsoft Knowledge Base:

libraries and debugging and creating

For more information on other ReportWizard and FormWizard functions, query on the following words here in the Microsoft Knowledge Base:

CreateControl and CreateReportControl and CreateForm and  
CreateReport and "Section Property" and DeleteControl and  
DeleteReportControl and CreateGroupLevel

For more information on the CreateForm(), CreateReport(), CreateControl(), and CreateReportControl() functions, search for each function name and then view its topic using the Microsoft Access version 2.0 Help menu.

Additional reference words: 1.00 1.10 2.00 wizards

KBCategory: kbprg

KBSubcategory: MdlLib

**INF: Using Variables in the Criteria Row of the QBE Grid**  
**Article ID: Q114511**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

You cannot use a variable in the criteria row of the Query by Example (QBE) grid, because Microsoft Access assumes the variable name is a literal text string. However, you can use a function name in the QBE grid, and have the function return the value of the variable.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

The following example demonstrates how to create a form based on a query that uses the value of a global variable as a criterion:

1. Open the sample database NWIND.MDB.
2. Create a new query based on the Employees table. Drag the First Name and Last Name fields from the field list to the QBE grid.
3. In the criteria cell of the Last Name column, type the following function name:

```
Return_Variable()
```

4. Save the query as Variable Criteria Query and then close it.
5. Create a new module with the following in the Declarations section:

```
Option Explicit
Global MyVar
```

6. Enter the following functions in the module:

```
Function Return_Variable()
 Return_Variable = MyVar
End Function

Function Set_MyVar(Name_from_Button)
 MyVar = Name_from_Button
 DoCmd Requery
End Function
```

7. Save the module as Variable Criteria Module and then close it.
8. Create a new, blank form based on the Variable Criteria Query query.
9. Place text box controls for the First Name and Last Name fields on the form.
10. Create the following two command buttons on the form:

Object: command button

-----  
Name: Fuller Button  
Caption: Fuller  
On Click: =Set\_MyVar("Fuller")

Object: command button

-----  
Name: Peacock Button  
Caption: Peacock  
On Click: =Set\_MyVar("Peacock")

NOTE: In Microsoft Access version 1.x, the Name property is called the ControlName property, and the OnClick property is called the OnPush property.

11. Set the form's DefaultView property to Single Form and then switch the form to Form view.
12. Choose the Fuller button on the form. Note that the record for Andrew Fuller appears.
13. Choose the Peacock button on the form. Note that the record for Margaret Peacock appears.

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.1, Chapter 1, "Writing a New Function," pages 7-19

Microsoft Access "Building Applications," version 2.0, Chapter 3, "Introducing Access Basic," pages 66-73

Additional reference words: 1.00 1.10 2.00 parameter

KBCategory: kbusage

KBSubcategory: QryParm



## **INF: Using Version 1.1 Databases in Version 2.0**

**Article ID: Q112122**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SUMMARY

=====

This article describes several issues involved in using Microsoft Access version 1.x databases in Microsoft Access version 2.0.

### MORE INFORMATION

=====

The following issues should be kept in mind when you are using version 1.x databases in Microsoft Access version 2.0:

- A. When you open an unconverted Microsoft Access 1.x database in Microsoft Access 2.0, you will receive the message:

Database <database name> was created by a previous version of Microsoft Access. You won't be able to save changes made to object definitions in this database.

You cannot make any design changes to database objects created in version 1.x until you convert the database to version 2.0. Note that database objects may behave differently in version 2.0 because of new and changed features.

You can open and use unconverted databases in version 2.0. However, if you want to make design changes to them, use Microsoft Access version 1.x.

- B. You cannot change permissions in version 1.x databases using Microsoft Access version 2.0. However, if you use a version 1.x SYSTEM.MDA file with Microsoft Access 2.0, the existing security scheme will operate properly.

You can use Microsoft Access version 1.x to make security changes to the version 1.x database, but make sure to use the version 1.x SYSTEM.MDA file.

- C. You can update fields on the one side of a one-to-many join.

- D. A combo or list box displays data as it is formatted in the control's underlying table or query. However, if the row source is a SQL SELECT statement, no formatting is applied.

If you want the data in a combo or list box to be formatted, apply a format expression to the field in the query.

- E. If you attempt to open an object in a version 1.x database whose name contains a backquote character (`) in version 2.0, you will receive

the error message:

```
Couldn't find object <objectname>
```

While the backquote character (`) is permitted in object names in version 1.x, it violates standard naming conventions in version 2.0. In addition to the backquote (`), you cannot use a period (.), an exclamation point (!), or brackets ([]) in object names in version 2.0.

Use Microsoft Access 1.x to rename objects in the version 1.x database to remove these characters.

- F. The CTRL+F4 key combination closes only MDI windows (such as the Database window) in version 2.0. Use ALT+F4 to close pop-up windows. In Microsoft Access 1.x, CTRL+F4 closes both pop-up windows and MDI windows. This behavior is changed in version 2.0 to be consistent with Microsoft Windows behavior.

This means that you should open your version 1.x databases in Microsoft Access version 1.x and change all occurrences of {^F4} to {%F4} in all SendKeys actions in macros and Access Basic code.

- G. SendKeys actions to change menus, dialog boxes and property sheets do not necessarily produce the same results in version 2.0 as they do in version 1.x, since the File, Edit, Layout, and Format menus are changed in version 2.0. The Paste Special, Options and Security dialog boxes are changed as well. Several changes and additions to object property sheets also affect how SendKeys actions set properties.

The following items describe how to accommodate these changes:

- Open your version 1.x databases in Microsoft Access version 1.x and replace all SendKeys actions in macros and code that reference changed menus with DoMenuItem actions.
- Set and change security using data access objects (DAO) instead of SendKeys. This requires that you first convert the database to version 2.0 database format.
- Set properties at run time instead of using code that opens objects in Design view, changes properties, then opens the same objects in Form view. This requires that you first convert the database to version 2.0 database format.
- Use the GetOption and SetOptions methods instead of SendKeys to change options. This requires that you first convert the database to version 2.0 database format.

Since changes like these can occur with each new version of Microsoft Access, it is best to avoid using the SendKeys action whenever possible.

- H. Tables are the only objects that you can export to a Microsoft Access 1.x database from a Microsoft Access 2.0 database. You cannot export any other object from a version 2.0 database to a version 1.x database.

Objects in a version 1.x database opened in version 2.0 can be exported

to a version 2.0 database. However, this process converts those objects to version 2.0 objects.

You can attach a version 1.x table using version 2.0. You cannot attach a version 2.0 table using version 1.x.

- I. In version 1.x, validation rules for bound controls override validation rules for bound fields in the underlying table. In version 2.0, the validation rules for the field and bound control are combined.

For example, in version 1.x, if the field validation rule requires the value of a numeric field to be greater than 100, and the validation rule for a bound control on a form requires the value to be less than 75, an entry of 50 in the bound control is allowed.

The same entry would not be allowed in version 2.0, since the field and bound control validation rules are combined, and there is no number that is both less than 75 and greater than 100.

It is recommended that you review all of your form and field validation rules and change them where conflicts occur.

#### REFERENCES

=====

For more information about converting version 1.x databases to version 2.0 format, search for "Convert Database" then "Convert Database Command" using the Microsoft Access Help menu.

For more information about DAO, search for "DAO" then "Objects and Collections Reference" using the Microsoft Access Help menu.

Additional reference words: 2.00 Compatibility Converting

KBCategory: kbinterop

KBSubcategory: IntpOthr

**INF: Using WIN.INI & ODBC.INI When Attaching SQL Server Table**  
**Article ID: Q96110**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article discusses how the ODBC Driver manager and the SQL Server driver use the WIN.INI and ODBC.INI files during connection time.

MORE INFORMATION

=====

For the purposes of this article, the following is assumed:

- You set up a data source for the SQL Server driver called mysqlsvr.
- The data source is located on a server called Unixmc.
- You are connecting to a Sybase SQL Server table on Unix whose address is 11.1.4.40,3180.
- The net-library you are using is wdbftptc.dll.

Given these assumption, if the setup was done correctly, the ODBC.INI file will have the following entry in the [ODBC Data Sources] section:

```
mysqlsvr=SQL Server
```

There will be a section called [mysqlsvr] in the ODBC.INI and it will have entries of the following form:

```
DRIVER=c:\windows\system\sqlsrvr.dll
Description=SQL\Server on server Unixmc
OemToAnsi=No
Network=wdbftptc
Address=11.1.4.40,3180
```

The DRIVER entry points to the location of the SQL Server driver. It is located on the System subdirectory of the Windows directory.

The Description entry describes what kind of a data source this is. This is something that you can change when setting up the data source by using ODBC Administrator.

The OemToAnsi entry indicates whether conversion needs to be done from the OEM code page to the ANSI code page. Note that this is not the same as the translation DLL.

The Network entry indicates the name of the net-library.

The Address entry indicates the network address.

The WIN.INI file will have the following entry in its [SQLSERVER] section:

```
mysqlsrvr=wdbftptc,11.1.4.40,3180
```

When you run a SQLConnect or SQLDriverConnect command and choose to connect to the mysqlsvr data source, the Driver manager looks in the ODBC.INI file to find the name and location of the SQL Server driver. It looks for the DRIVER entry described above and loads it into memory.

Next, the SQL Server driver looks in the WIN.INI file's [SQLSERVER] section for mysqlsvr entry where it picks up the name of the net-library. Then it loads the net-library name (wdbftptc in this case) into the memory. Next, it looks at the entry again to get the address of the server which it passes on to the net-library. If there is no address following the name of the net-library, it passes the name of the data source to the net-library, as the network address.

Db-library users should note that unlike W3DBLIB.DLL, the SQL Server driver does not look at the dsquery entry unless dsquery is a data source in the ODBC.DLL file.

If you are using Microsoft Access, this sequence of operations take place when you attach a SQL Server table. If you are using Microsoft Visual Basic version 2.0, it takes place in response to the OpenDataBase call.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr

**INF: Validation Rule to Accept Only Alphabetic Characters**  
**Article ID: Q114554**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

This article demonstrates a sample user-defined Access Basic function called IsAlpha() that you can use to validate a string to make sure that every character in the string is a letter in the English alphabet.

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

MORE INFORMATION

=====

The IsAlpha() function accepts a string argument and checks each character in the string to make sure it is a character in the English alphabet. The function does this by checking the ASCII value of each character in the string to see if it falls in the range from 65 to 90 (for uppercase characters), or in the range from 97 to 122 (for lowercase characters), of the ANSI character set. The function returns True if all the characters in the string fall within one of these two ranges, or False if one or more characters in the string does not fall within one of these two ranges. You can use this function as a validation rule for a field in a table in Microsoft Access version 1.x, or as a validation rule for a control on a form in Microsoft Access versions 1.x and 2.0.

The following example shows how to use the function as a validation rule for a control on a form.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the sample database NWIND.MDB.
2. Create a new module and enter the following code:

```
'*****
'Declarations section of the module
'*****
Option Compare Database
Option Explicit

'=====
```

' The following function is named IsAlpha(). It accepts a string  
' argument and returns either True (-1) or False (0).

```

'=====

Function IsAlpha (MyString As String) As Integer
Dim LoopVar As Integer
Dim SingleChar As String

LoopVar = 1

If IsNull(MyString) Then
 IsAlpha = False
 Exit Function
End If

For LoopVar = 1 To Len(MyString)
 SingleChar = UCase(Mid$(MyString, LoopVar, 1))
 If SingleChar < "A" Or SingleChar > "Z" Then
 IsAlpha = False
 Exit Function
 End If
Next LoopVar

IsAlpha = True

End Function

```

3. Open the Categories form in Design view.
4. Set the Category Name field's ValidationRule property to:
 

```
IsAlpha([Category Name]) = True
```
5. Save the Categories form, and then view the form in Form view.
6. From the Records menu, choose Data Entry.
7. Type "ABCD" (without quotation marks) in the Category Name field. Note that the validation rule allows you to create this entry and exit the field.
8. In another new record, type "AB,D" (without quotation marks) in the Category Name field. Note that the validation rule will not accept the entry.

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.0, Chapter 3, "Access Basic Essentials," pages 24-26

Microsoft Access "Language Reference," version 1.0, "Asc Function," pages 40-41, and "Mid\$ Function," pages 302-303

Microsoft Access "User's Guide," version 2.0, Chapter 8, "Entering a Validation Expression," pages 147-148

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: GnlValid



## **INF: Why OLE Objects Cause Databases to Grow**

**Article ID: Q123151**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

A Microsoft Windows OLE object is stored as an OLE stream that consists of two portions:

- Information to render the object.
- Native data that can be modified by the OLE server that created the object.

Microsoft Access stores the entire OLE stream in an OLE wrapper in a type of Long Binary field.

The amount of information used to render the object is often greater than the size of the object itself. Adding such OLE objects can increase the size of a database significantly. To minimize size increases caused by adding OLE objects to a database, use either of these methods:

- Make the object easier to render (by lowering the resolution of the object, or decreasing its physical size).
- Display the object as an icon.

### MORE INFORMATION

=====

An object's rendering information can be in any of the following formats:

- Windows Bitmap format (BMP)
- Windows Metafile format (WMF)
- Windows Device-Independent Bitmap format (DIB)

An OLE server offers formats in order from the server's most to least preferred format. An OLE client enumerates the available formats and uses the first compatible format. Generally, you can override the OLE client's choice of formats with the Paste Special command.

Normally, either WMF or DIB is chosen, since OLE servers render these formats most easily. For example, Microsoft Word for Windows normally renders using WMF, while Windows Paintbrush renders using DIB. An OLE client can always choose the format it prefers first. Since DIB is a form of BMP, most servers that render in DIB also render in BMP, although DIB is often offered first. Microsoft Access always chooses DIB over BMP.

OLE servers that can display pictures stored in other compressed formats, such as the JPEG format (JPG) or the CompuServe GIF format (GIF), must still send the two portions of information described earlier in this

article. The server must send information to render the object using BMP, WMF, or DIB, as well as native data that can be used to edit the object. When this occurs, the compressed format is converted to a format to be rendered. The size of the OLE stream is the sum of the converted rendering object, the native data, and some OLE overhead information. Because the OLE stream contains all this information, small objects in compressed formats can become quite large when stored in a Microsoft Access database.

One way to work around this behavior is to store the object as an icon. If you do not need to display the object in a Microsoft Access form or report, you can store the object as an icon that represents the object type. Storing an object as an icon causes the OLE server to send the object with rendering information consisting of only the icon rather than the complete object, using significantly less storage space in the database. When you double-click the icon the OLE server is launched with the native data that the icon represents.

Another example of a small file growing to become a large OLE object is a Microsoft Word file that contains complex text formatting or an image on the first page of the document. The small amount of native data requires so much information for rendering that the OLE object is quite large. To work around this behavior, place a page with simple formatting at the beginning of the document, or store the document as an icon.

#### Additional Considerations

-----

OLE has a preferred order for using the standard presentation formats. Regardless of where WMF appears in the format order, it is used if it is available (metafiles provide better scaling than bitmaps). If WMF is not available, DIB is preferred over BMP. OLE currently supports only WMF, DIB, and BMP.

One of the options in the OLE Specification calls for OLE servers to add more rendering formats to the three default formats that Windows supports. Microsoft Access does not currently support this OLE feature.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 19, "Using Pictures, Graphs, and Other Objects," pages 461-482

"Microsoft Object Linking and Embedding Programmer's Reference," Volume 1, pages 418-420

Additional reference words: 1.00 1.10 2.00 big

KBCategory: kbole

KBSubcategory: IntpOle

**INF: Word for Windows Print Merge Using Microsoft Access Data**  
**Article ID: Q103629**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Word for Windows, versions 1.1, 1.1a, 2.0, 2.0a, 2.0b, and 2.0c
- 

SUMMARY

=====

There is no Microsoft Word for Windows file converter capable of reading Microsoft Access database files. In order to use data from a Microsoft Access database in a Microsoft Word for Windows print merge, the data must first be exported from Microsoft Access in a format that Microsoft Word for Windows recognizes.

In Microsoft Access version 1.0, you can export the data in dBASE IV format. In Microsoft Access versions 1.1 and 2.0, you can export the data in Word For Windows Merge format. We do not recommend that you export Microsoft Access data to Word for Windows using the Text (Delimited format).

MORE INFORMATION

=====

Exporting Data from Microsoft Access for Use in a Print Merge

-----

1. Open the Microsoft Access database table that contains the data you want to use in a Microsoft Word for Windows print merge.
2. From the File menu, choose Export. In Microsoft Access version 1.1 or 2.0, select Word For Windows Merge in the Data Destination box. In Microsoft Access version 1.0 select dBASE IV as the data destination. Choose OK.
3. From the Table list, select the table to be exported, and choose OK.

NOTE: Microsoft Access only shows you a list of available tables to export. If you want to export a subset of a table, you can create a query to result in the subset you want, and export the results to a print merge data file. For information on how to export the results of a query to a file, query on the following words here in the Microsoft Knowledge Base:

access and export and query and transfer and results

4. Enter a unique filename in the File Name box. Select the appropriate destination directory in the Directory box and then choose OK.

Notes

-----

- Microsoft Access 1.1 will default the export filename to the same name

as your table, with a .TXT extension, and will place the file in the Microsoft Access directory. Microsoft Access 1.0 will default the export filename to the same name as your table, with a .DBF extension. In either case, you may want to change the directory to your Microsoft Word for Windows directory and give the file a .DOC extension to make it easier to find in Word for Windows.

- In Microsoft Access versions 1.1 and 2.0, you can select custom formats in the Export Options dialog box. These options determine how dates, times, and numbers are formatted in the data file. Although you can select custom formats, you do not need to make any changes in this dialog box.
- In Microsoft Access 2.0, you can export the data with the Output To command from the File menu. You can output the table to Rich Text format, which Word for Windows can read.

This format will maintain the table structure, if the table does not exceed Microsoft Word for Windows internal limits. This method does not correct the field names, which can cause problems with a Microsoft Word for Windows print merge. Problems with Microsoft Access field names in a Word for Windows print merge are discussed later in this article.

#### Using the Exported Data in Microsoft Word for Windows

---

1. Open or create the main document in Microsoft Word for Windows.
2. From the File menu, choose Print Merge.
3. Choose Attach Data File.
4. From the file list, select the exported file, and then choose OK.

NOTE: You may have to type "\*.TXT" or "\*.DBF" (without the quotation marks) in the File Name box to see text or dBASE files in the filename list.

5. Use the Insert Merge Fields tool on the print merge toolbar to insert fields in your main document. Consult the Microsoft Word for Windows documentation for information on how to use the merge features.

#### Problems Exporting Data Using the Text (Delimited) Format

---

One method of exporting data to Microsoft Word for Windows is to use the Text (Delimited) format. However, exporting the data in this format may require some changes to the exported data file before it can be used for a print merge.

The problem is that a number of restrictions are placed on field names used in a Microsoft Word for Windows print merge, including the following:

- Field names cannot contain spaces.
- Field names cannot begin with a number.
- Field names cannot be longer than 20 characters in length.
- Field names cannot contain special characters.

Microsoft Access field names do not have these restrictions. Consequently, you can export data from Microsoft Access that contains field names incompatible with Microsoft Word for Windows. If you use the Text (Delimited) format for exporting data, you may have to change the field names in the exported data file so that they meet the restrictions listed above.

If you use the dBASE IV or Word For Windows Merge formats, Microsoft Access will perform field name conversions to ensure that the data files will work correctly with Microsoft Word for Windows.

#### Microsoft Access Field Name Conversion

-----

As discussed previously, Microsoft Word for Windows places a number of restrictions on the field names used in a print merge. Since Microsoft Access is not limited by these restrictions, it must perform field name conversions when exporting data to a print merge data file.

Microsoft Access performs field name conversions using the following rules:

- Any spaces or illegal characters are converted to underscores.
- If the first character in a field name is a number or an underscore, Microsoft Access adds the prefix "m\_". For example, 1Field becomes m\_1Field, and \_Field2 becomes m\_\_Field2.
- Microsoft Access truncates field names that are longer than 20 characters.
- If a conversion results in a duplicate field name, Microsoft Access adds a number to the end of each duplicate name (replacing the final character, if the field name contains twenty characters). For example, the field names Order#, Order\$, and Order% become Order\_, Order\_1, and Order\_2.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr

**INF: Writing Functions Called from Events or Expressions**  
**Article ID: Q97514**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

Functions can be used in a variety of places in Microsoft Access. How you write your functions depends on where the functions are going to be called from.

This article assumes that you are familiar with Access Basic and with creating applications for Microsoft Access using the programming tools provided with Microsoft Access.

MORE INFORMATION

=====

There are two main styles for writing Access Basic functions:

- Functions can be called from event properties, such as the AfterUpdate property for a control on a form. Typically, you would call the function with a parameter. The function then acts on or modifies the parameter.
- Functions can be used in expressions, such as calculated controls. The difference is in how the result is returned when the function exits.

NOTE: This article does not address event procedures in Microsoft Access version 2.0, since these are Sub procedures and not functions.

The examples below use the Proper() function to illustrate the differences between the two function styles. Proper() converts the first letter of a word to uppercase and the other letters to lowercase.

Calling a Function from an Event Property

-----

The Proper() function can be written so it can be called from an event, such as the AfterUpdate property of a control on a form. In this example we will call it ProperAU() as a reminder that it should be called from the AfterUpdate property.

Enter the following function in a module:

```
Function ProperAU(Field As Control)
 Field=UCCase(Left(Field,1)) & LCase(Mid(Field,2))
End Function
```

NOTE: The result of the calculation updates the field that was

passed as a parameter.

#### Example

-----

1. Open the Customers form in Design view.
2. View the Property sheet by choosing Properties from the View menu.
3. Add the following statement to the AfterUpdate property of the First Name field:

```
Object: Text Box

ControlName: First Name
AfterUpdate: =ProperAU([First Name])
```

Now, whenever the employees name is typed into the Employee form, it will be converted to the correct format when the user presses TAB or ENTER.

#### Calling a Function from an Expression

-----

The Proper() function can be written so it can be called from an expression, or calculated control. In this example we will call it ProperCC() as a reminder that it should be used in calculations.

Enter the following function in a module:

```
Function ProperCC(Field)
 ProperCC=UCase(Left(Field,1)) & LCase(Mid(Field,2))
End Function
```

NOTE: The result of the calculation is assigned to the function. This way, it can be used in an expression or calculated control.

#### Example

-----

1. Open the Customers form in Design view.
2. Add the following calculated control to the form:

```
Object: Text Box

ControlName: Proper Last Name
ControlSource: =ProperCC([Last Name])
```

Now when you type in the Last Name field, you will see the correct capitalization in the Proper Last Name field.

NOTE: You will not be able to type in the Proper Last Name field. ProperCC() does not change underlying data like ProperAU() does. For this reason, ProperCC() is useful in reports and expressions and can be used more places than ProperAU().

You can use ProperCC in the same manner as any of the built-in functions listed in the "Language Reference," such as UCase, LCase, and so on.

#### Determining the Type of Function You Need

---

| Where used                               | Function style |
|------------------------------------------|----------------|
| AfterUpdate, BeforeUpdate, and so on     | Event          |
| RunCode macro action                     | Event          |
| Calculated controls on forms and reports | Expression     |
| Calculated fields in a query             | Expression     |
| SetValue macro action expression         | Expression     |
| Default values in a table or form        | Expression     |
| Called from another function or sub      | Expression     |

Additional reference words: 1.00 1.10 2.00 code

KBCategory: kbprg

KBSubcategory: PgmOthr



## **INF: Xtrieve Data Dictionary (.DDF) File Information**

**Article ID: Q109449**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SUMMARY

=====

This article describes Xtrieve data dictionary (.DDF) files.

### MORE INFORMATION

=====

Each Xtrieve data dictionary consists of three files: FILE.DDF, FIELD.DDF, and INDEX.DDF.

The first three records in the FILE.DDF file contain information about the .DDF files for the data dictionary. The first record contains the name and path to itself (FILE.DDF), the second record contains the name and path to FIELD.DDF, and the third record contains the name and path to INDEX.DDF. The remaining records contain the path and filename of each Btrieve file available for use by Microsoft Access as a table.

Microsoft Access reads the FILE.DDF file to locate the FIELD.DDF and INDEX.DDF files. If you move these files, make sure to update the FILE.DDF file to reflect their current location. Or, if you change the names of these files, make sure to update the FILE.DDF file to reflect the new names.

To modify the FILE.DDF file you must use Xtrieve, or any hex editor. If you use a hex editor to modify the file, make sure not to change the record or field lengths in the file.

Xtrieve and Btrieve are manufactured by Btrieve Technologies, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00 Attaching

KBCategory: kb3rdparty

KBSubcategory: IsmBtrv

**PRA: "Can't Place Item at This Position" Error Message**  
**Article ID: Q121359**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you use a Report Wizard to create a new report, and select only a Date/Time field for the report, you receive the error message "Can't place item at this position."

RESOLUTION

=====

Select at least one other field in addition to the Date/Time field when you create a new report.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new report based on the Orders table. Choose the Report Wizards button.
3. In the "Which Wizard do you want?" box, select Tabular, and then choose the OK button.
4. In the Available Fields box, select Order Date, and then choose the ">" button. Choose the Finish button. You will receive the error message stated above. Choose OK.

Note that the mouse pointer remains an hourglass, instead of returning to its normal arrow shape. To restore the pointer, follow these steps:

1. Open any module.
2. From the View menu, choose Immediate Window.
3. In the Immediate window, type the following line and then press ENTER:

```
DoCmd hourglass false
```

Additional reference words: 2.00  
KBCategory: kbusage kberrmsg  
KSubcategory: FmsWiza

**PRA: "Couldn't Find Object '<Name>'" with SQL Server Table**  
**Article ID: Q124239**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you try to attach a SQL Server table, you receive the error message:

    Couldn't find object '<tablename>'

CAUSE

=====

This problem occurs only if you have installed the Microsoft Access version 2.0 Service Pack. This problem is caused by the Microsoft SQL Server ODBC driver, but occurs only if you are using the Microsoft Jet database engine version 2.5, which is installed with the Service Pack. The problem with the SQL Server driver causes the Microsoft Jet database engine version 2.5 to be unable to attach any table with a name with the following pattern:

    (number of non-underscore characters in table name) + ((number of  
underscores in table name) x 3) >= 33

Thus, you can attach a table with the name

    a\_bceghi\_jklopq\_stuv\_x24

but you cannot attach a table with the name:

    a\_bcdeghi\_jklopq\_stuv\_x25

RESOLUTION

=====

To work around this problem, rename the table so that it does not have a name matching the pattern described above. Or, create a SQL Server view based on the table and then attach the view. For example, you could use the following sample SQL statement to create a view:

```
create view AbcDefGhiJklMnoPqr
as
select * from abc_def_ghi_jkl_mno_pqr;
```

You could then attach the view instead of attaching the table. In order to update the view, you would need to create an index specification on the view, as in the following example:

```
create unique index Index1 on <owner>_AbcDefGhiJklMnoPqr (field1)
```

For more information about this type of index, please see the Microsoft Access "User's Guide," version 2.0, Chapter 9, "Importing, Exporting, and

Attaching," page 169.

STATUS

=====

Microsoft has confirmed this to be a problem in the Microsoft SQL Server ODBC driver. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

REFERENCES

=====

For more information about the Microsoft Access version 2.0 Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kbusage kberrmsg

KBSubcategory: ObcSqlms

**PRA: "Invalid Reference" Saving Record on Form with Button**  
**Article ID: Q123858**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

When you try to save a record by choosing a command button on a form that runs a DoMenuItem action, you receive the error message:

Invalid reference to field 'Form\_BeforeUpdate'

CAUSE

=====

The Cancel argument is being set in the form's BeforeUpdate event procedure.

RESOLUTION

=====

There are two ways to work around this problem. The first method traps and handles the error, and the second cancels the event in a different manner.

Method 1

-----

You can work around the problem by trapping and handling the error message. When you execute the DoMenuItem action to save the record, trap for error 2465 and replace it with a more descriptive error message. For example, you could attach the following sample Access Basic code to the command button:

```
Sub Button1_Click ()
 On Error GoTo Err_Button1_Click
 DoCmd DoMenuItem A_FORMBAR, A_FILE, A_SAVERECORD, , A_MENU_VER20
Exit_Button1_Click:
 Exit Sub
Err_Button1_Click:
 Select Case Err
 Case 2465
 MsgBox "Descriptive error message of your choice here"
 Case Else
 MsgBox Error$
 End Select
 Resume Exit_Button1_Click
End Sub
```

Method 2

-----

In step 5 of the example in the "Steps to Reproduce Problem" section below, use

DoCmd CancelEvent

instead of:

```
Cancel = True
```

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

The problem described above usually results in the error message "Invalid reference to 'Form\_BeforeUpdate'," but may also result in the error message "Invalid reference to field 'Button0\_Click'."

This problem can also occur if you call a macro instead of code in the BeforeUpdate property. The only method of saving a record that works as you expect is choosing Save Record from the File menu.

Steps to Reproduce Problem

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new form based on the Employees table. Set the form's DefaultView property to Single Form.
3. Place a text box control on the form. Set the text box's ControlSource property to First Name.
4. Create a Save Record button using the Command Button Wizard. To create this button, follow these steps:
  - a. Choose the Control Wizards button in the toolbox (the button should appear sunken).
  - b. Place a command button on the form.
  - c. In the Categories box, select Record Operations. In the When Button Is Pressed box, select Save Record. Choose the Next button.
  - d. Choose the Next button, then choose the Finish button.
5. Set the form's BeforeUpdate property to the following event procedure:

```
Sub Form_BeforeUpdate (Cancel As Integer)
 Cancel = True
End Sub
```

6. View the form in Form view and enter any text in the text box.

7. Choose the command button to save the record. You receive the error message stated above.

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: FmsProb



## **PRA: "No Permission for DBSTRUCT.MDB" Error Message**

**Article ID: Q120580**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Developer's Toolkit version 2.0
- 

### SYMPTOMS

=====

After you log on to Microsoft Access as any user other than the Admin user, you receive the following error message when you try to open the sample database DBSTRUCT.MDB:

No permission for '<path>\DBSTRUCT.MDB'

### CAUSE

=====

Only the Admin user has Open/Run permission on the DBSTRUCT database. Neither the Admins group nor the Users group has Open/Run permission on the database.

### RESOLUTION

=====

You can use the DBSTRUCT database by logging on to Microsoft Access using a new, unsecured system database (or SYSTEM.MDA file), or you can modify the DBSTRUCT database's permissions so that other users can open it. Both of these methods are demonstrated below.

#### How to Use a New, Unsecured System Database

-----

1. Quit Microsoft Access if it is running, and then make a backup copy of the SYSTEM.MDA file. Name the backup copy of the file SYSTEM.BAK.
2. Double-click the MS Access Workgroup Administrator icon in your Microsoft Access program group. (NOTE: This icon may be in another group, such as the Microsoft Office program group.)
3. In the Workgroup Administrator dialog box, choose the Create button to create a new system database.
4. In the Workgroup Owner Information dialog box, type appropriate information in the Name, Organization, and Workgroup ID boxes, and then choose the OK button.
5. In the Workgroup System Database dialog box, enter a path and unique name for the new system database file in the Database box, and then choose OK. For example, enter:

C:\ACCESS\UNSECURE.MDA

NOTE: Do not use the same name as your secure system database file because Microsoft Access will attempt to overwrite the file when it creates the new file. Also, the path in the example above is for illustrative purposes only; you should use the correct path for your Microsoft Access installation.

6. In the Confirm Workgroup Information dialog box, choose OK. When you receive a message stating that the new system database was created, choose OK.
7. Choose the Exit button to quit the Workgroup Administrator.
8. Start Microsoft Access. You should not receive a Logon dialog box, which means that you are logging on to Microsoft Access as the default Admin user. Since the Admin user has Open/Run permission on the DBSTRUCT database, you should now be able to open the database.

#### How to Modify Permissions on the DBSTRUCT Database

---

1. Start Microsoft Access and log on as the Admin user.
2. Open the sample database DBSTRUCT.MDB.
3. From the Security menu, choose Permissions.
4. In the Object Type box, select Database.
5. Select the Groups option button, and then select the Users group in the User/Group Name box.
6. In the Permissions group, select the Open/Run check box, and then choose the Assign button.
7. In the User/Group Name box, select the Admins group.
8. In the Permissions group, select the Open/Run check box, and then choose the Assign button.
9. Choose the Close button, and then quit Microsoft Access.
10. Restart Microsoft Access and log on as you normally do. You should now be able to open the DBSTRUCT database.

#### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

#### MORE INFORMATION

=====

#### Steps to Reproduce Problem

---

1. In an unsecured Microsoft Access system, create a new user named USER1 and add the user to the Admins group.
2. Assign a password to the default Admin user so that you will receive a Logon dialog box when you restart Microsoft Access.
3. Quit and then restart Microsoft Access. Log on to Microsoft Access as USER1.
4. Try to open the DBSTRUCT database. Since USER1 does not have Open/Run permission on the database, you will receive the error message stated above.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 14,  
"Securing Your Application," pages 313-344

Additional reference words: 2.00 security login ADT  
KBCategory: kbusage kberrmsg  
KBSubcategory: ScrtPerm

**PRA: "Unexpected Error from External Database Driver [7]"**  
**Article ID: Q116383**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you are working with an attached Btrieve table, you receive the error message:

Unexpected error from external database driver [7]

Or, a query based on an attached Btrieve table does not return any records in Microsoft Access version 2.0. This same query does return records correctly in Microsoft Access version 1.x.

CAUSE

=====

The problems stated above are caused by an error in the BTRV200.DLL Btrieve ISAM driver that is installed by Microsoft Access 2.0.

RESOLUTION

=====

An updated Btrieve ISAM driver is available in the Microsoft Software Library (MSL). This driver is for use by registered owners of Microsoft Access version 2.0 and Microsoft Visual Basic version 3.0 for Windows - Professional Edition only. By downloading this driver, you are indicating that you are a registered owner of one of these products.

You can download BTR200.EXE, a self-extracting file, from the MSL on the following services:

- CompuServe
  - GO MSL
  - Search for BTR200.EXE
  - Display results and download
- Microsoft Download Service (MSDL)
  - Dial (206) 936-6735 to connect to MSDL
  - Download BTR200.EXE
- Internet (anonymous FTP)
  - ftp ftp.microsoft.com
  - Change to the softlib/mslfiles directory
  - Get BTR200.EXE

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version

2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

#### MORE INFORMATION

=====

You can differentiate between the original and updated drivers by checking the file sizes. The original BTRV200.DLL file has a file size of 110,592 bytes. The updated file has a file size of 110,624 bytes. For a network installation of Microsoft Access, install the updated file in the ACCESS\SYSTEM directory. For a stand-alone installation of Microsoft Access, install the updated file in the WINDOWS\SYSTEM directory.

NOTE: Do not install the updated file before installing Microsoft Access. You should install Microsoft Access first, then install the updated driver.

#### How to Install the Updated Driver

-----

##### Stand-Alone Installation:

1. Make a backup copy of the original BTRV200.DLL file by renaming it, or by copying it to another location. The BTRV200.DLL file is in the WINDOWS\SYSTEM directory.
2. Copy the updated BTRV200.DLL file to the WINDOWS\SYSTEM directory.

##### Network Installation:

1. The network administrator should make a backup copy of the original BTRV200.DLL file by renaming it, or by copying it to another location. The BTRV200.DLL file is in the ACCESS\SYSTEM directory on the network server.
2. The network administrator should copy the updated BTRV200.DLL file to the ACCESS\SYSTEM directory on the network server.
3. The network administrator should make a backup copy of the SETUP.INF file by renaming it, or by copying it to another location. The SETUP.INF file is located in the ACCESS installation directory on the network server.
4. The network administrator should open the SETUP.INF file in any text editor (such as Windows Notepad), and change the line that reads

```
"DLL"= 1,SYSTEM\BTRV200.DLL,,,,,,1033,,,,,,SHARED,110592,,,,2.0.0.13,
```

to read:

```
"DLL"= 1,SYSTEM\BTRV200.DLL,,,,,,1033,,,,,,SHARED,110592,,,,2.0.0.15,
```

5. After making this change, the network administrator should save and then close the SETUP.INF file.

This modification changes the internal version from 2.0.0.13 to 2.0.0.15. Changing the internal version number in the SETUP.INF file will cause future workstation installations of Microsoft Access to

install the updated Btrieve driver on workstations that have a copy of the original Btrieve driver.

6. The network administrator should make the updated BTRV200.DLL file available to those who have installed Microsoft Access to their local workstation hard disks. These users should copy the updated file to their WINDOWS\SYSTEM directories.

Additional reference words: 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmBtrv

**PRA: 1.x Table/Query Datasheet Properties Not Converted to 2.0**  
**Article ID: Q117538**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

The row heights and fonts of tables and queries open in Datasheet view that were converted from Microsoft Access version 1.x format to version 2.0 format are reset to their default values. The row heights and fonts are not set to what they were in Microsoft Access version 1.x.

CAUSE

=====

When you convert a database from Microsoft Access version 1.x to version 2.0 format, the following property settings for tables and queries are reset to their default values:

- Description
- FrozenColumns
- RowHeight
- DatasheetFontHeight
- DatasheetFontItalic
- DatasheetFontName
- DatasheetFontUnderline
- DatasheetFontWeight

RESOLUTION

=====

There are two ways to work around this problem:

- After converting the database, manually set each of the affected properties back to the correct setting by using the commands on the Format menu.
- Write a custom function to automatically set the datasheet properties for each table. The following example demonstrates how to set the row height of the Customers table in the sample database NWIND.MDB to display two lines of data. The property setting will take effect the next time you open the Customers table in Datasheet view. This example uses the SetTableProperty procedure, which is in the NWIND database's global module:

```
Dim MyDB As Database, MyTableDef As TableDef
Set MyDB = DBEngine.Workspaces(0).Databases(0)
Set MyTableDef = MyDB![Customers]
SetTableProperty MyTableDef, "RowHeight", DB_LONG, 450 (450 twips)
```

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Start Microsoft Access version 1.x and open the sample database NWIND.MDB. Open the Customers table in Datasheet view.
2. From the Layout menu, choose Row Height. Change the value from 10.5 to 20.
3. Save the changes and quit Microsoft Access.
4. Start Microsoft Access version 2.0.
5. From the File menu, choose Convert. Convert the Microsoft Access version 1.x sample database NWIND.MDB to version 2.0 format. Name the converted database NWIND20.MDB
6. Open the NWIND20 database and then open the Customers table in Datasheet view.
7. Check the row height setting that you set in step 2. Note that the value has been reset to the default value of 10.5.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: TblOthr



**PRA: ADT Application Setup Always Defaults to Drive C**  
**Article ID: Q121959**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

When you are installing an application created with the Microsoft Access Developer's Toolkit (ADT), the default drive for installation is always drive C, even if you designated a different drive as the default in the ADT Setup Wizard.

CAUSE

=====

The Setup program automatically chooses the first available hard disk for the default installation drive. If no local hard disk is available, Setup chooses the first available network drive.

STATUS

=====

Microsoft has confirmed this to be a problem in the Microsoft Access Developer's Toolkit version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Run the Setup Wizard using the sample database NWIND.MDB as your custom application.
2. In the Default Installation Directory box, enter the following line:  
  
R:\TEST
3. When the Setup Wizard is finished, run the custom Setup program. Note that Setup uses drive C as the default installation drive instead of drive R.

REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0, Chapter 2, "Creating a Custom Setup Program," pages 19-37

Additional reference words: 2.00  
KBCategory: kbsetup

KBSubcategory: StpOthr

## **PRA: ADT Application Setup Fails with Error -1809**

**Article ID: Q115880**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

### SYMPTOMS

=====

You receive the following error message when you install a custom application created with the Microsoft Access Developer's Toolkit (ADT):

Setup couldn't create a SYSTEM.MDA file. After Setup is completed, reboot your computer, then run the Workgroup Administrator before you run this application. (-1809)

### CAUSE

=====

This problem occurs if no form of file sharing (SHARE.EXE, VSHARE.386, or network locking) is running when the Workgroup Administrator (WRKGADM.EXE) tries to create a SYSTEM.MDA file. In order for the Workgroup Administrator to be able to create the SYSTEM.MDA file, some form of file sharing must be running.

### RESOLUTION

=====

There are two workarounds for this problem:

- Ship a SYSTEM.MDA file with your custom application. To do this, add the SYSTEM.MDA file in the Files To Include box in the ADT Setup Wizard. This way, the Workgroup Administrator does not need to create a SYSTEM.MDA file. Note that the file must be named SYSTEM.MDA in order for this workaround to work. No other name (such as MYSYS.MDA) will work.
- Test to see if some form of file sharing is running before starting the custom Setup program. You can test to see if file sharing is running in a number of ways. To test for file sharing in Microsoft Windows, you can start the MS-DOS prompt and then type "share" (without quotation marks). If file sharing is running, you will receive a message stating "SHARE already installed." You can also test for file sharing with a small Microsoft Visual Basic or Microsoft Visual C application (these applications are not discussed in this article).

### STATUS

=====

Microsoft has confirmed this to be a problem in the Microsoft Access Developer's Toolkit version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Create application disks using the ADT Setup Wizard. Do not include a SYSTEM.MDA file with your application.
2. Install your application on a computer that is not running some form of file sharing (either loading SHARE.EXE in the AUTOEXEC.BAT file or loading VSHARE.386 in the [386Enh] section of the SYSTEM.INI file).

You will receive the error message stated above near the end of the Setup process.

Additional reference words: 2.00

KBCategory: kbsetup kberrmsg

KBSubcategory: StpFail

## **PRA: ADT Splash Screen Does Not Get Repainted**

**Article ID: Q117916**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0  
-----

### SYMPTOMS

=====

If you specify a splash, or startup, screen in your custom application's .INI file, and you have security enabled in your application, the application's Logon dialog box will overlay the splash screen. If you move the Logon dialog box, the splash screen does not get repainted.

### RESOLUTION

=====

Do not move the Logon dialog box when you are logging into a Microsoft Access Developer's Toolkit-developed run-time application.

### STATUS

=====

Microsoft has confirmed this to be a problem in the Microsoft Access Developer's Toolkit (ADT) version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

=====

#### Steps to Reproduce Problem

-----

1. Create a Microsoft Access application.
2. Use the ADT Setup Wizard to create Setup disks for your application. Make sure to include a secured SYSTEM.MDA file and a .BMP file with your application.
3. Add the following lines to your application's .INI file:  

```
[Run-Time Options]
StartupScreen=<yourapp>.bmp
```
4. Use the Setup disks to install your application.
5. Double-click the icon that your application's Setup creates. When the Logon dialog box appears, move the dialog box. Note that the bitmap image does not get repainted.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: GnlAcini



## **PRA: Attached Paradox 4.0 Table Displays "#Deleted"**

**Article ID: Q123598**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you attach a Paradox version 4.0 table with a single primary key that is a one-character alphanumeric field, "#Deleted" is displayed in all the records in the table.

### RESOLUTION

=====

Increase the FieldSize property setting of the primary key field, add another field to the primary key, or delete the current primary key.

### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

=====

#### Steps to Reproduce Problem

-----

1. Create a Paradox 4.0 table with a one-field primary key that is a one-character alphanumeric field.
2. Add a record to the table.
3. Start Microsoft Access and open any database.
4. Attach the table you created in step 1.
5. View the table in Datasheet view. "#Deleted" is displayed in the record.

### REFERENCES

=====

For more information about attaching Paradox tables, search for "Paradox files" then "Importing and Attaching Paradox Tables" using the Microsoft Access Help menu.

Additional reference words: 2.00 pounddeleted pound deleted

KBCategory: kb3rdparty kberrmsg

KBSubcategory: IsmPdox





**PRA: AutoDialer Saves Only Five Characters of Prefix**  
**Article ID: Q121861**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

The Prefix box in the AutoDial dialog box saves only the first five characters you type. If you type more than five characters in the Prefix box and close the AutoDial dialog box, there will be only five characters in the Prefix box the next time you open the AutoDial dialog box.

RESOLUTION

=====

Even though you can store only five characters or less in the Prefix box, you can enter a prefix of more than five characters and use it. The AutoDialer will correctly dial a prefix of more than five characters, but it will not save more than five characters. If you need to use a prefix of more than five characters for a phone number, type the extra characters before calling that number.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open the Customers form in Form view.
3. From the View menu, choose Toolbars. In the Toolbars dialog box, choose the Customize button.
4. In the Categories box, select Record.
5. Drag the Autodialer button (the button with a picture of a telephone on it) from the Buttons box to the toolbar.
6. In the Customize Toolbars box, choose the Close button.
7. Place the insertion point in the Phone field, and then choose the Autodialer button.

8. In the Prefix box, enter the prefix 9-34567, and then select the Use Prefix check box. Choose OK. When you receive the message "Pick up the phone!" choose OK.
9. Choose the Autodialer button again. Note that the prefix has been truncated at five characters.

#### REFERENCES

=====

For more information about the AutoDialer, search for "AutoDialer" then "AutoDialer" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsProb

**PRA: Calendar Control Font and Color Properties Unavailable**  
**Article ID: Q115125**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

The arrow to the right of the Calendar Control Properties drop-down list box is unavailable. You cannot access the Font and Color properties for the Calendar control.

CAUSE

=====

The file OC1016.DLL did not get registered properly in the REG.DAT file when you installed the Microsoft Access Developer's Toolkit.

RESOLUTION

=====

NOTE: This resolution assumes that the Microsoft Access Developer's Toolkit is installed in the C:\ACCESS\ADT directory, and that Microsoft Windows is installed in the C:\WINDOWS directory. If you are using different directories, modify the line in step 2 appropriately.

1. In Windows Program Manager, choose Run from the File menu.
2. In the Command Line box, type the following line, and then choose OK:

C:\ACCESS\ADT\MSAREG C:\WINDOWS\SYSTEM\OC1016.DLL

The registration process will take from 1 to 5 seconds to complete. You will not receive any notification that the process has been completed.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. In any database, create a blank, new form.
2. From the Edit menu, choose Insert Object.

3. Select the Insert Control option button.
4. In the Control Type box, select Calendar Control, and then choose OK.
5. From the Edit menu, choose Calendar Control Object, then choose Properties. Note that the arrow to the right of the Calendar Control Properties drop-down list box is dimmed.

Additional reference words: 2.00 gray grayed custom ocx

KBCategory: kbusage

KBSubcategory: GnlApp

**PRA: Can Print Macro Definition Without Read Design Permission**  
**Article ID: Q120144**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When a user has Open/Run permissions on a macro, but not Read Design permissions, the user can view the macro's design using the Print Definitions command.

CAUSE

=====

This problem only occurs if the user is granted Open/Run permissions on a macro as an individual user. The problem does not occur if the user is granted Open/Run permissions on the macro as a member of a group.

RESOLUTION

=====

To prevent a user from being able to read a macro's design, give the user Open/Run permissions on the macro as a member of a group, instead of as an individual user.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

NOTE: This example assumes that there is no password on your Admin user account.

1. Create a new database called TEST.MDB.
2. Create the following new macro and then save the macro as Test Security and close it:

Test Security Actions

-----

MsgBox

Message: This is a test

3. From the Security menu, choose Users, and then in the Users dialog

box choose the New button. In the Name box, enter "Joe" (without quotation marks), and in the Personal ID box enter "abcd" (without quotation marks). Choose OK. Make sure that Joe is a member of only the Users group and then choose the Close button.

4. From the Security menu, choose Permissions. In the Object Name box, select the Test Security macro, and then assign the following permissions on the macro:

Admins group: all permissions  
Guests group: no permissions  
Users group: no permissions  
Joe: Open/Run permissions

When you are done, close the Permissions dialog box.

5. From the Security menu, choose Change Password. Change the Admin user's password by typing "1234" (without quotation marks) in the New Password and Verify boxes, and then choose OK.
6. Quit Microsoft Access and then start it again. In the Logon dialog box, type "Joe" (without quotation marks) in the Name box, and then choose OK.
7. Open the TEST database, and then select the Test Security macro in the Database window. Choose the Design button to open the macro in Design view. You will receive a message stating that you do not have permissions to read the macro. This is correct behavior, as the user Joe has only Open/Run permissions on this macro.
8. From the File menu, choose Print Definition. In the Print Macro Definition dialog box, choose OK. An Object Definition report will be generated, in which you can view the macro's names, conditions, actions, arguments, and permissions, even though you do not have Read Design permissions on the macro.

NOTE: To avoid having to log in to Microsoft Access as the Admin user after following the above steps, clear the password for the Admin user by following these steps:

1. Quit Microsoft Access and then start it again.
2. In the Logon dialog box, type "Admin" (without quotation marks) in the Name box, type "1234" (without quotation marks) in the Password box, and then choose OK.
3. Open any database.
4. From the Security menu, choose Users.
5. In the Name box, select Admin, then choose the Clear Password button. Choose the Close button.
6. Quit Microsoft Access.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 14,  
"Securing Your Application," pages 313-344

For more information about database security, search for "security: basics"  
then "Securing a Database" using the Microsoft Access Help menu.

For more information about the Print Definition command, search for "Print  
Definition" then "Print Definition Command" using the Microsoft Access Help  
menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: ScrtPerm

**PRA: Cannot Add Record After Changing DefaultEditing Property**  
**Article ID: Q122673**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

If you open a form with the form's DefaultEditing property set to Can't Add Records, using Access Basic code from a global module to set the DefaultEditing property to Allow Edits will not cause the form to allow new records to be added.

RESOLUTION

=====

One way to work around this problem is to first set the DefaultEditing property to Data Entry, and then to Allow Edits. The following example demonstrates how to add a command button to a form that will set the DefaultEditing property so that you can add new records.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open the Customers form in Design view.
3. Set the form's DefaultEditing property to Can't Add Records.
4. Add a command button named Add to the form.
5. Set the new command button's OnClick property to the following event procedure:

```
Sub Add_Click ()
 Me.DefaultEditing = 1 'Sets DefaultEditing to Data Entry
 Me.DefaultEditing = 2 'Sets DefaultEditing to Allow Edits
End Sub
```

6. Save the form, and then close it.
7. Open the form in Form view. Note that you cannot add new records.
8. Choose the Add button. Note that you can add new records.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.



## MORE INFORMATION

=====

### Steps to Reproduce Problem

-----

1. Follow steps 1-3 in the "Resolution" section above.
2. Save the form, close it, and then open it in Form view. Note that you cannot add new records.
3. Create a new module and then choose Immediate Window from the File menu.
4. Type the following line in the Immediate window and then press ENTER:  

```
Forms![Customers].DefaultEditing = 2
```
5. Select the Customers form. Note that you still cannot add new records.

## REFERENCES

=====

For more information about the DefaultEditing property, search for "DefaultEditing" then "AllowEditing, DefaultEditing Properties" using the Microsoft Access Help menu.

Additional reference words: 2.00 insert

KBCategory: kbusage

KBSubcategory: Gn1Othr

## **PRA: Cannot Close Form from Control's OnExit Property**

**Article ID: Q115574**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you try to close a form from the OnExit (or OnEnter) property of a control on the form, you receive the error message:

Can't run this action while processing a form or report event.

### CAUSE

=====

A macro or procedure that closes the active form is called from the form's OnExit or OnLostFocus property.

### RESOLUTION

=====

To work around this problem, create a new control containing a flag on the form. Then, refer to that flag's value before closing the form, as demonstrated in the following example:

1. On the form in question, place a new, unbound text box control.
2. Set the new control's Name property to FlagControl. Set its DefaultValue property to 0.
3. Set the OnExit property of the control from which you want to close the form to the following event procedure:

```
Me![FlagControl]=-1
```

To create this event procedure, click in the OnExit property field, choose the Build button at the right side of the field, and then select Code Builder. After you enter the code, close the module.

4. From the Edit menu, choose Select Form.
5. Set the form's TimerInterval property to 500.
6. Set the form's OnTimer property to the following event procedure:

```
If Me![FlagControl] Then
 DoCmd Close
End If
```

7. View the form in Form view, and then exit the control (either by pressing TAB or by clicking another control).

## STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

## MORE INFORMATION

=====

### Steps to Reproduce Problem

-----

1. Set the OnExit property of any control on a form to the event procedure:

    DoCmd Close

2. View the form in Form view, and then exit the control (either by pressing TAB or by clicking another control).

Additional reference words: 2.00

KBCategory: kbprg kberrmsg

KBSubcategory: PgmOthr

**PRA: Cannot Connect to SQL Server w/ NetWare, Workgroups**  
**Article ID: Q90151**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Windows for Workgroups version 3.11
- 

SYMPTOMS

=====

You are unable to communicate with SQL Server in Microsoft Access when you are running Microsoft Access under Microsoft Windows for Workgroups on a Novell NetWare network.

CAUSE

=====

The Windows for Workgroups IPX/SPX driver (MSIPX.COM) does not work correctly with Novell's named pipe requester (DOSNP.EXE) to connect to or communicate with SQL through named pipes.

RESOLUTION

=====

Use the NetWare Integration Kit (NIK), which provides native SPX/IPX connectivity to SQL Server. This connectivity eliminates the need for DOSNP.EXE. The NIK enables Microsoft Access and Windows for Workgroups to communicate correctly with SQL Server on a Novell network.

You can obtain the NIK by purchasing SQL Server for OS/2 version 4.2B or the upgrade kit from the usual retail channels.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Windows for Workgroups version 3.11. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Novell's implementation of IPX is a file called IPX.COM. Microsoft's equivalent NDIS-compliant implementation of IPX is a file called MSIPX.COM, which is used by Microsoft Windows for Workgroups. MSIPX.COM is functionally equivalent to Novell's IPX.COM, but has the additional capability of using NDIS-compliant network adapter drivers.

When you install the NetWare Named Pipe Requester, a terminate-and-stay-resident program (TSR) called DOSNP.EXE installs into memory. This TSR works with IPX.COM to provide named-pipe functionality (which is the communication protocol used by SQL Server) to applications. However, DOSNP.EXE does not work correctly with MSIPX.COM at this time. Since Windows for Workgroups uses MSIPX.COM, the NetWare Named Pipe Requester

does not work with Windows for Workgroups. Therefore, Microsoft Access is not able to properly communicate with SQL Server.

Additional reference words: 1.00 1.10 2.00 importing exporting attaching  
KBCategory: kbusage  
KBSubcategory: ObcOthr

**PRA: Cannot Delete Templates Created with the Setup Wizard**  
**Article ID: Q123011**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

There is no documented way to delete a saved template in the Microsoft Access Developer's Toolkit (ADT) version 2.0.

RESOLUTION

=====

To delete saved templates, reinstall the ADT. Reinstalling the ADT overwrites the SETUPWIZ.MDB file and any saved templates. Another method for deleting templates, although undocumented and therefore unsupported, is to delete the template records in the zws\_TemplateFiles and zws\_SetupTemplates tables in SETUPWIZ.MDB.

STATUS

=====

Microsoft has confirmed this to be a problem in the Microsoft Access Developer's Toolkit version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0, Chapter 2, "Creating a Custom Setup Program," pages 20-21

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: WzProb

## **PRA: Cannot Edit Record from Two Forms in One Session**

**Article ID: Q122744**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

You cannot edit the same record from two different forms or datasheets in a single session, or instance, of Microsoft Access, even though you have set the Microsoft Access locking scheme to No Locks, or optimistic record locking.

### CAUSE

=====

The record locking scheme in Microsoft Access defaults to Edited Record, or pessimistic record locking, on a per-record basis within a single instance of Microsoft Access.

### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0, 1.1, and 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

=====

Microsoft Access has three record locking schemes. The No Locks, or optimistic, record locking scheme permits multiple users to edit the same record simultaneously. When one user saves a record, other users editing the same record receive notification from Microsoft Access, and are allowed to overwrite the record, abandon it, or copy changes to the Clipboard.

The Edited Record, or pessimistic, record locking scheme permits only one user to edit a record at a time. When a record is being edited, other users who attempt to edit the record are locked out.

The All Records locking scheme instructs Microsoft Access to lock all the records in a form or datasheet (and underlying tables) for the entire time you have the form or datasheet open. This is an extremely restrictive record locking scheme.

To set the record locking scheme, choose Options from the View menu, select the Multiuser/ODBC category, and then select the scheme you want for the Default Record Locking item. You can also set the record locking scheme by setting the RecordLocks property for a form. In Microsoft Access version 2.0, you can also set the RecordLocks property for queries and reports. The Default Record Locking setting applies to table datasheets and other objects that do not have an explicit RecordLocks property.

No matter which record locking scheme you set, however, Microsoft Access ignores the setting and uses pessimistic record locking within a single instance of Microsoft Access.

#### Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. From the View menu, choose Options. In the Category box, select Multiuser/ODBC.
3. Set the Default Record Locking item to No Locks, and then choose OK.
4. Open the Categories table in Datasheet view. Modify the first record in the table, but do not save the changed record.
5. Open the Categories form in Form view and try to modify the first record in the form. Note that you cannot edit the record.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.1, Chapter 16, "Using Microsoft Access in a Multiuser Environment," pages 388-389

Microsoft Access "User's Guide," version 2.0, Chapter 6, "Sharing Data in a Multiuser Environment," pages 106-108

Additional reference words: 1.00 1.10 2.00 page

KBCategory: kbusage

KBSubcategory: GnlMu



**PRA: Cannot Embed OLE Objects Using Access Basic Code**  
**Article ID: Q120262**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

You can use Access Basic code to embed one OLE object in a bound control on a form successfully. However, subsequent OLE objects that you try to embed with Access Basic code are visible in the form, but are not saved when you save the record.

CAUSE

=====

You are not moving the focus in your Access Basic code to the OLE controls in which you are trying to embed objects.

RESOLUTION

=====

Set the focus in your Access Basic code to subsequent OLE controls before saving the record.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Create a new table with two OLE object fields called OLE1 and OLE2. Set the Required property for both fields to Yes.
2. Create a new form based on the table you created in step 1. Add the two OLE fields to the form.
3. Set the form's OnLoad property to the following event procedure:

```
Sub Form1_Load
 Me![OLE1].SourceDoc = "c:\windows\arcade.bmp"
 Me![OLE1].Action = OLE_CREATE_EMBED

 Me![OLE2].SourceDoc = "c:\windows\arcade.bmp"
 Me![OLE2].Action = OLE_CREATE_EMBED
End Sub
```

4. View the form in Form view. Note that both the OLE controls contain the Arcade bitmap.
5. From the File menu, choose Save Record to save the record. You will receive the error message:

Field '<tablename>.OLE2' can't contain a null value

NOTE: If you did not set the Required property for both fields to Yes, you will not receive an error message, and the second OLE object field will contain a null value.

To correct the problem, change the event procedure for the OnLoad property to be:

```
Sub Form1_Load
 Me![OLE1].SourceDoc = "c:\windows\arcade.bmp"
 Me![OLE1].Action = OLE_CREATE_EMBED

 Me![OLE2].SourceDoc = "c:\windows\arcade.bmp"
 Me![OLE2].Action = OLE_CREATE_EMBED
 Me![OLE2].Setfocus
End Sub
```

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 19, "Using Pictures, Graphs, and Other Objects," pages 470-481

Microsoft Access "Building Applications," version 2.0, Chapter 4, "Adding and Editing Data," pages 64-65

Additional reference words: 2.00 ab link

KBCategory: kbprg kberrmsg

KBSubcategory: PgmPrcls

**PRA: Changes to Print Setup Not Saved with Form or Report**  
**Article ID: Q114514**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

After changing your form or report to print in a different paper orientation (such as landscape), it does not print in the orientation you chose. After checking the Print Setup dialog box, you find that none of the changes you made were saved.

CAUSE

=====

Changes made in the Print Setup dialog box are not saved under the following conditions:

- The report contains functions that change one or more run-time settings and you choose Print Setup from the File menu while previewing the report.
- When you open a form into Form view, Datasheet view, or Print Preview, switch to Form Design view, then choose Print Setup from the File menu.

RESOLUTION

=====

For reports, make changes in the Print Setup dialog box from the Database window, rather than while previewing the report.

For forms, make changes in the Print Setup dialog box while the form is open in Design view, rather than while previewing the form.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Open any form in Datasheet view.
2. Preview the form.
3. From the File menu, choose Print Setup. Change the orientation from

Portrait to Landscape.

4. Switch the form back to Design view, and then save and close the form.
5. Open the form in Design view and choose Print Setup from the File menu.  
Note that the orientation is still set to Portrait.

#### REFERENCES

=====

For more information about changing form or report print settings, search for "Print Setup" then "Print Setup Command (File Menu)" using the Microsoft Access Help menu.

Additional reference words: 2.00 Papersize Margins Saving

KBCategory: kbprint

KBSubcategory: PtrOthr

**PRA: Compile Errors Not Reported When Form Opened with Code**  
**Article ID: Q121013**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

In Microsoft Access version 1.x, compile errors in procedures triggered by the Open event are not reported if you use Access Basic code to open a form in Dialog window mode.

In Microsoft Access version 2.0, compile errors in procedures triggered by the Load event are not reported if you use Access Basic code to open a form in any window mode.

RESOLUTION

=====

Compile all your Access Basic code before you open the form.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0, 1.1, and 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Start Microsoft Access and open any database.
2. Create a new module.
3. From the Edit menu, choose New Procedure.
4. In the New Procedure dialog box, select the Function option button, and then type "Force\_Error" (without quotations marks) in the Name box. Choose OK.
5. Enter the following code in the module:

```
Function Force_Error()
 MsgBox 1
End Function
```

6. Save the module as Test and then close it.
7. Create a blank new form. In Microsoft Access version 1.x, set the

form's OnOpen property to:

```
=Force_Error()
```

In Microsoft Access version 2.0, set the form's OnLoad property to:

```
=Force_Error()
```

8. Save the form as MyForm and then close it.
9. Open the MyForm form in Form view. You will receive a compile error message.
10. Open the Test module, and then choose Immediate Window from the View menu.
11. Type the following line in the Immediate window and then press ENTER:

```
DoCmd OpenForm "MyForm",,,,A_DIALOG
```

Note that you do not receive any error message.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage kbusage kberrmsg

KBSubcategory: FmsEvt

**PRA: Database Documentor Generates Incorrect Form Information**  
**Article ID: Q120483**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you use the Database Documentor or Print Definition command to document a form's properties, the report that is generated contains incorrect information about the report's AllowEditing and AllowUpdating properties.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

The following table details the actual and reported settings for a form's AllowEditing and AllowUpdating properties:

| Property       | Actual Setting | Reported Setting |
|----------------|----------------|------------------|
| AllowEditing   | Unavailable    | Available        |
|                | Available      | Unavailable      |
| Allow Updating | Default Tables | No               |
|                | Any Tables     | 1                |
|                | No Tables      | 2                |

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. In the Database window, select the Categories form.
3. From the File menu, choose Print Definition. In the Print Form Definition dialog box, choose OK.
4. Note that in the resultant Object Definition report, the AllowEditing property setting is reported as Unavailable and the AllowUpdating property setting is reported as No. These are not the correct settings for these properties. The AllowEditing property is actually set to Available, and the AllowUpdating property is actually set to Default Tables.

REFERENCES

=====

For more information about the Database Documentor, search for "Database Documentor" then "Database Documentor Command" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: OtpProb



**PRA: Database Documentor Incorrectly Reports Permissions**  
**Article ID: Q119654**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

The Database Documentor in Microsoft Access version 2.0 incorrectly reports permissions for the Users group and any new groups that you create. This problem also occurs with the Print Definitions command.

RESOLUTION

=====

There is no known resolution to this problem at this time.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. From the Security menu, choose Permissions.
3. Select the Groups option button.
4. In the User/Group Name box, select Users.
5. In the Object Name box, select the Categories table.
6. Assign full permissions to the Users group on the Categories table. (Select the Read Design, Modify Design, Administer, Read Data, Update Data, Insert Data, and Delete Data check boxes.)
7. Choose the Assign button and then the Close button.
8. From the File menu, choose Add-ins then choose Database Documentor.
9. In the Print Definitions Of Objects box, select the Categories table, and then choose the Select button.
10. Choose the Options button and then make sure that the Permissions By User And Group check box is selected. Choose OK.

11. Choose OK to generate the Object Definition report. Note that for permissions for the Users group, the Database Documentor reports "Read Permissions," which is incomplete for this table.

This problem also occurs if you create a new group and assign full permissions on a table to the group.

#### REFERENCES

=====

For more information about the Database Documentor, search for "Database Documentor" then "Database Documentor Command" using the Microsoft Access Help menu.

Additional reference words: 2.00 secure

KBCategory: kbusage

KBSubcategory: RptsOthr

**PRA: Empty Recordset Report Hangs with Group KeepTogether**  
**Article ID: Q114549**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

Microsoft Access stops while you are previewing a report, and may mark the database as corrupt. Running Repair Database on the database eliminates the error message that states that the database is corrupt.

CAUSE

=====

Microsoft Access can stop unexpectedly while you are previewing a report if the report is based on an empty recordset, and the group KeepTogether property is set to With First Detail or Whole Group.

This is a known problem with "retreating" in Microsoft Access version 2.0 reports. Retreating refers to returning to a earlier report section in order to determine where certain controls and sections are on a report, and whether they will fit in a given space. Examples of when retreating is used include:

- Where group levels (except for page headers and footers) are used, and the KeepTogether property is set to either Whole Group or With First Detail.
- In subreports or subforms where the CanGrow or CanShrink property is set to Yes.

In these situations, the Format event occurs as Microsoft Access determines how the sections will fit on the printed page. If a section cannot be printed on the current page, Microsoft Access retreats so that the section can be printed on the following page.

RESOLUTION

=====

To avoid this problem, test the report's underlying recordset to see if data is returned. If no data is returned, use the CancelEvent action to stop the report from processing. Two methods for avoiding the problem are described below.

Method 1

-----

Use this method when the report is based directly on a table or on a query that does not have parameters.

Call the following Sub procedure from the report's OnOpen property:

```

Sub Report_Open (cancel As Integer)
 If IsNull(DLookup("<AnyFieldInQuery>", "<QueryName>")) Then
 DoCmd CancelEvent
 End If
End Sub

```

## Method 2

-----

Use this method if the report is based on a parameter query. The query must be open, with the parameter values filled in, before the number of records in the recordset can be referenced.

Use the following sample code to cancel the report if the recordset is empty. This sample code references two date parameters. The key is to have the parameters filled in with values.

```

Sub Report_Open (Cancel As Integer)
 Dim MyDb As Database, MyQuery As QueryDef, DataRecs As Recordset
 Set MyDb = DBEngine.Workspaces(0).Databases(0)
 Set MyQuery = MyDb.QueryDefs("QueryName")
 MyQuery("Beginning Date:") = #1/1/96# ' or Forms!FormName![Control]
 MyQuery("Ending Date:") = #1/3/96# ' or Forms!FormName![Control]
 Set DataRecs = MyQuery.OpenRecordset()
 If DataRecs.EOF Then DoCmd CancelEvent
 DataRecs.Close
 MyQuery.Close
End Sub

```

## STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

## MORE INFORMATION

=====

### Steps to Reproduce Problem

-----

**WARNING:** Following these steps causes Microsoft Access to stop. Be sure to save any work in progress before following these steps.

1. Open the sample database NWIND.MDB.
2. In the Database window, select the Orders table, and then choose the AutoReport button on the toolbar.
3. View the new report in Design view.
4. Choose the Sorting And Grouping button on the toolbar.
5. Set the following group properties:

Customer ID

```
GroupHeader: Yes
KeepTogether: With First Detail
Order ID
GroupHeader: Yes
KeepTogether: With First Detail
Order Date
Ascending sort only
```

6. Set the height for the Customer ID header section and the Order ID header section to zero.

7. Save the report as A Test Report.

8. Create a new macro with the OpenReport action and the following arguments:

```
Report Name: A Test Report
View: Print Preview
Where: [Order Id] = 222
```

9. Save the macro as A Test Macro and then run it. The report will begin formatting, but will never finish. Microsoft Access will stop.

10. Close Microsoft Access and restart Microsoft Windows.

11. Start Microsoft Access and open the NWIND database. If you receive the prompt

```
NWIND.MDB is corrupted or is not a database file. Attempt to
repair?
```

choose Yes.

12. To avoid the problem in this example, add the following code to the report's OnOpen property:

```
Sub Report_Open (cancel As Integer)
 If IsNull(DLookup("[Order Id]","Orders", "[Order Id] = 222")) Then
 DoCmd CancelEvent
 End If
End Sub
```

#### REFERENCES

=====

For more information about the Retreat event, search for "Retreat" then "Retreat Event" using the Microsoft Access Help menu.

Additional reference words: 2.00 hang

KBCategory: kbusage

KBSubcategory: RptSort

**PRA: Error "Setup Cannot Create SYSTEM.MDA File (-1809)"**  
**Article ID: Q115066**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

When you are running Microsoft Access Setup, after the last disk has loaded, Setup has updated your system, and the Program Manager icons have been created, you get the error message:

Setup couldn't create a SYSTEM.MDA file. After Setup is completed, reboot your computer, then run the Workgroup Administrator before you run this application. (-1809)

CAUSE

=====

This error occurs when a file-sharing utility has not been loaded prior to running Setup.

Setup uses a program called the Workgroup Administrator (WRKGADM.EXE) to create the SYSTEM.MDA file. The Workgroup Administrator requires a file-sharing utility such as SHARE.EXE or VSHARE.386 (the virtual device driver file-sharing utility used in Windows 386 enhanced mode) to work correctly.

RESOLUTION

=====

To solve this problem, follow these steps:

1. After Setup completes, quit Microsoft Windows.
2. If you run Windows version 3.1 in standard mode, make sure that SHARE.EXE is loaded in your AUTOEXEC.BAT or CONFIG.SYS file, or load it at the MS-DOS command prompt.
3. If you run Windows version 3.1 or Windows for Workgroups version 3.1x in 386 enhanced mode, make sure that VSHARE.386 is loaded in the [386Enh] section of your SYSTEM.INI file.
4. Start Microsoft Windows.
5. Start the Workgroup Administrator by double-clicking the MS Access Workgroup Administrator icon, and then create a new SYSTEM.MDA file.
6. Start Microsoft Access.

Microsoft Access must have a valid SYSTEM.MDA file to work correctly. If you try to start Microsoft Access before you create the SYSTEM.MDA file, you will receive the error message:

Couldn't find file 'C:\ACCESS\system.mda'.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

Additional reference words: 2.00 setup installing admin

KBCategory: kbsetup

KBSubcategory: StpOthr

## **PRA: Error (-1310) Running Crosstab or Totalling Query**

Article ID: Q119077

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you run a query that contains aggregate functions, you receive the error message:

Reserved error (-1310); there is no message for this error.

This problem only occurs with queries for which all of the following conditions are true:

- The query does not contain remote (ODBC) tables.
- The query contains three or more tables, and can be divided into two or more queries to obtain the same results.
- The query does not have any criteria on the aggregate fields.

### CAUSE

=====

This error is caused by the query optimizer. Any change made to the query or its underlying tables (such as adding criteria to the query, adding indexes to or removing them from underlying tables, or adding rows to underlying tables, and then resaving the query) that causes Microsoft Access to change its query optimization strategy can either result in or resolve this error.

### RESOLUTION

=====

#### Method 1

-----

The best way to prevent this error is to divide the query into two or more separate queries that produce the same result. For example, the following sample query returns the average unit price for each customer and the quantity of items sold, using the sample database NWIND.MDB. This query will result in the error message stated above:

```
Tables: Customers, Orders, Order Details
Group By: [Customers].[Company Name]
Group By: [Order Details].[Quantity]
Avg: [Order Details].[Unit Price]
```

You can divide this query into two queries by following these steps:

1. Remove the Customers table from the query.



2. Drag the Customer ID field from the Orders table to the query grid.
3. In the Total row of the Customer ID column, select Group By.
4. Verify that the query runs correctly, then save it as TempTotals.
5. Create a new query based on the TempTotals query and the Customers table. Note that the table and query automatically join on the Customer ID field.
6. Drag the Company Name field from the Customers table, and the Quantity and AvgOfUnit Price fields from the TempTotals query, to the query grid.

#### Method 2

-----

Adding criteria to your query can prevent the error from occurring. Add criteria that do not affect the outcome of the query, such as comparing a field to itself. For example, you could use the following criterion for a query that includes the Customers, Orders, and Order Details tables:

```
Customers.[Customer ID]=Customers.[Customer ID]
```

#### STATUS

=====

This problem no longer occurs with the Microsoft Jet database engine version 2.5, which is available with the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

```
ARTICLE-ID: Q122927
```

```
TITLE : WX1124: Microsoft Access Version 2.0 Service Pack
```

#### MORE INFORMATION

=====

#### Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Create a new query based on the Customers, Orders, and Order Details tables. Note that the tables join automatically.
3. Drag the Company Name field from the Customers table to the query grid. Drag the Quantity and Unit Price fields from the Order Details table to the query grid.
4. From the View menu, choose Totals. In the Total row for the Company Name and Quantity columns, select Group By. In the Total row for the Unit Price column, select Avg.

Note that the SQL statement for this query is:

```
SELECT DISTINCTROW Customers.[Company Name],
```

```
[Order Details].Quantity, Avg([Order Details].[Unit Price])
AS [AvgOfUnit Price]
FROM Customers INNER JOIN (Orders INNER JOIN
 [Order Details] ON Orders.[Order ID] = [Order Details].[Order ID])
 ON Customers.[Customer ID] = Orders.[Customer ID]
GROUP BY Customers.[Company Name], [Order Details].Quantity;
```

5. Run the query. You will receive the error message stated above.

Additional reference words: 2.00 queries querying jet25 jet 2.5  
KBCategory: kbusage kberrmsg  
KBSubcategory: QryTotal

**PRA: Error Deleting Value from Attached SQL Server Table**  
**Article ID: Q117616**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you attach an SQL Server table that has a column with a NOT NULL restriction to Microsoft Access, add a value to the restricted column in the attached table, then delete the value and attempt to commit the record, you receive the error message:

[Microsoft][ODBC SQL Server Driver][SQL Server] The column <column name> in table <table name> may not be null. (#233)

CAUSE

=====

When the value is deleted, Microsoft Access first attempts to write a null to the restricted column in the attached table. This results in an internal error from SQL Server, so Microsoft Access then attempts to write a zero-length string. Since SQL Server does not recognize zero-length strings, it treats the string as a null, resulting in the error message.

RESOLUTION

=====

Do not enter zero-length strings in SQL columns with the NOT NULL restriction.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Create a table on SQL Server with a column with the NOT NULL restriction.
2. Start Microsoft Access and open any database.
3. Attach the table that you created in step 1.
4. Enter a value in the column with the NOT NULL restriction.

5. Move to the next column.
6. Move back to the previous column, select the value, then delete it.
7. Commit the record by moving to another record. You will receive the error message stated above.

#### REFERENCES

=====

For more information about using zero-length strings, search for "zero-length strings" then "Working with Nulls and Zero-Length Strings" using the Microsoft Access Help menu.

For additional information about zero-length strings and SQL Server, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q117614

TITLE : Error Adding Zero-Length String to SQL NOT NULL Column

Additional reference words: 2.00

KBCategory: kbinterop kberrmsg

KBSubcategory: ObcSqlms

**PRA: Error In SQL Pass-Through Action Query Not Detected**  
**Article ID: Q115897**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When an error occurs while you are performing a SQL pass-through action query on a remote SQL server, the error is not reported to Microsoft Access.

CAUSE

=====

Microsoft Access does not detect errors that occur while performing data-definition queries.

RESOLUTION

=====

Avoid using data-definition queries with remote SQL servers. Instead, perform these functions (DROP TABLE, CREATE TABLE, and so on) from the Admin utility included with SQL Server.

If you use DB\_FAILONERROR as an option for action queries, errors will be detected.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

Steps to Reproduce Problem

-----

This example attempts to delete a nonexistent table from a SQL Server database. Use your own database information in the "Set db = DBEngine(0)" line in this example.

1. Start Microsoft Access and open any database.
2. Create a new module.
3. Enter the following function in the module.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Function test ()
 On Error GoTo Handler
 Dim db As Database, qd As QueryDef

 Set db = DBEngine(0).OpenDatabase("", False, False, _
 "ODBC;DSN=dabusql;UID=sa;PWD=;DATABASE=mult;")

 Set qd = db.CreateQueryDef("")
 qd.SQL = "DROP TABLE tsint2222" ' Nonexistent table
 qd.Execute ' Error expected here
 MsgBox "DONE", 0
 Exit Function

Handler:
 MsgBox Error$
 Exit Function
End Function
```

4. From the View menu, choose Immediate Window.

5. Type the following in the Immediate window, and then press ENTER:

```
? Test()
```

Note that you do not receive any error message.

Additional reference words: 2.00 dao programming queries  
KBCategory: kbinterop  
KBSubcategory: ObcSqlms

**PRA: Error Message "Couldn't Open SYSTEM.MDA"**

**Article ID: Q123589**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you start Microsoft Access, you receive either of the following error messages:

- Couldn't open SYSTEM.MDA
- or-
- <Database> is corrupted or is not a database file. Attempt to repair?

CAUSE

=====

The Microsoft Jet database engine version 2.0 may erroneously leave an internal database flag set that in some cases results in one of the error messages stated above.

RESOLUTION

=====

To reset the internal database flag, have all users log out of the database and then run the repair utility. (To run the repair utility, choose Repair from the File menu.) If you receive the "Couldn't open SYSTEM.MDA" error message, all users should quit Microsoft Access. When the first user logs back in, the SYSTEM.MDA file is repaired automatically. No data will be lost as a result of the internal database flag being set erroneously.

STATUS

=====

This problem no longer occurs with the Microsoft Jet database engine version 2.5, which is available with the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kbusage

KBSubcategory: GnlBaddb

**PRA: Error Message "Internal Database Error -1310"**

**Article ID: Q109346**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you run a large query (about 25K or more in size), you receive the error message

Internal database error -1310 has occurred

even if the SQL statement is correct.

CAUSE

=====

This error is caused by an out of memory condition in an internal memory segment.

RESOLUTION

=====

Break up large queries that generate this error into several smaller queries.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0, 1.1, and 2.0. We are researching this problem, and will post new information here in the Microsoft Knowledge Base as it becomes available.

Additional reference words: 1.00 1.10 2.00 Queries OOM

KBCategory: kbusage

KBSubcategory: QryParm



**PRA: Error Message "Undefined Function in Expression"**

**Article ID: Q123344**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you call a user-defined function from within an aggregate function (such as the Sum() function) in a report, you receive the following error message:

Undefined function '<function name>' in expression.

CAUSE

=====

The user-defined function is located in the report module.

RESOLUTION

=====

Transfer the user-defined function from the report module to a global module.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Open the Alphabetical List Of Products report in Design view.
3. Add a text box with the following property to both the detail section and the Product Name footer section of the report:

ControlSource: =Sum(Notwork())

4. From the View menu, choose Code.
5. Enter the following user-defined function in the report module:

```
Function Notwork()
 Notwork=8
```

End function

6. Close the report module.

7. Preview the report.

You receive the error message stated above.

#### REFERENCES

=====

For more information about creating modules, search for "modules: opening" then "Creating and Opening a Module" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: ExprFnc

**PRA: Error Previewing Subreports that Call Filter Macro**  
**Article ID: Q103131**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you attempt to print a form or report containing a subreport/subform that calls a macro that in turn executes an ApplyFilter action, then you call that macro from the OnOpen property of the subreport/subform, the following error message is displayed:

Action isn't valid because the form or report isn't bound to a table or query.

Even if you do not receive this error message, note that the ApplyFilter action has not been executed. Currently, this functionality is not available.

CAUSE

=====

When printing, Microsoft Access erroneously attempts to apply the filter that is run from the macro to the outermost (main) report/form, rather than to the subreport/subform.

The error occurs because the main report/form does not contain a RecordSource to perform the ApplyFilter action against.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0, 1.1 and 2.0. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new macro that executes an ApplyFilter action. Enter criteria such as the following for the Where condition:

[Last Name]="Fuller"

3. Create a new, blank, unbound report.
4. Drag the Employee Sales Subreport from the Database window to

create a subreport on the new report.

5. Double-click the subreport to edit it. For the OnOpen property of the subreport, select the macro you created in step 2 from the combo box.
6. Preview the report. The following error message appears:

Action isn't valid because the form or report isn't bound to a table or query.
7. Change the RecordSource property of the main report to point to the Employees table.
8. Preview the report. Note that the error message no longer appears, however, neither the subreport nor the main report have been filtered.

Additional reference words: 1.00 1.10 2.00 Subreports Reports Subforms  
KBCategory: kbusage  
KBSubcategory: RptSub

**PRA: Error When Import dBASE, FoxPro, or Paradox Table**  
**Article ID: Q95327**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you import a dBASE, FoxPro, or Paradox table in Microsoft Access version 1.x, or a Paradox table in Microsoft Access version 2.0, without including the file extension, you receive the following error message:

    Couldn't find object '<filename>'

RESOLUTION

=====

Rename the file so that it has an extension. It does not have to have the correct extension, such as .DBF for a dBASE table, but it must have some sort of extension.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0 and 1.1. This problem no longer occurs in Microsoft Access version 2.0 for dBase and FoxPro, but does still occur for Paradox. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Rename a dBASE, Paradox, or FoxPro file so that it has no extension. For example, rename CUSTOMER.DBF to CUSTOMER.
2. Open a Microsoft Access 1.x database.
3. From the File menu, choose Import.
4. In the Import dialog box, select the appropriate data source. For example, select dBASE III or IV to import a dBASE file.
5. Under List Files of Type, select All Files (\*.\*).
6. Locate and select the import file created in step 1 without an extension.
7. Choose the Import button.

You will see the following error message:

Couldn't find object '<filename>'.

Additional reference words: 1.00 1.10 2.00

KBCategory: Kbusage kberrmsg

KBSubcategory: ObcOthr

**PRA: Errors After Reinstalling MS Access to Workstation**  
**Article ID: Q120058**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you reinstall Microsoft Access to a workstation and then start Microsoft Access, you receive an error message similar to:

```
 Couldn't find file 'G:\ACCESS\SYSTEM.MDA'
```

If you modify the MSACC20.INI file so that the SystemDB line in the [Options] section of the file points to the correct path for the SYSTEM.MDA file and then start Microsoft Access, you receive the following error message repeatedly while the Add-ins are being loaded:

```
 Tried to load module with duplicate procedure definition:
 doc_CleanObjTables
```

CAUSE

=====

When you reinstall Microsoft Access to the workstation, incorrect settings are written to the MSACC20.INI file. For example, the line

```
 SystemDB=G:\ACCESS\WORKDIR\SYSTEM.MDA
```

is changed to:

```
 SystemDB=G:\ACCESS\SYSTEM.MDA
```

Note that the \WORKDIR subdirectory is missing from the path for the SYSTEM.MDA file after the reinstallation.

The [Libraries] section of the file might look like the following example before the reinstallation:

```
[Libraries]
G:\ACCESS\WORKDIR\wzLib.mda=rw
G:\ACCESS\WORKDIR\wzTable.mda=rw
G:\ACCESS\WORKDIR\wzQuery.mda=rw
G:\ACCESS\WORKDIR\wzFrmRpt.mda=rw
G:\ACCESS\WORKDIR\wzBldr.mda=rw
```

After the reinstallation, the [Libraries] section might look like:

```
[Libraries]
G:\ACCESS\WORKDIR\wzLib.mda=rw
G:\ACCESS\WORKDIR\wzTable.mda=rw
G:\ACCESS\WORKDIR\wzQuery.mda=rw
G:\ACCESS\WORKDIR\wzFrmRpt.mda=rw
```

G:\ACCESS\WORKDIR\wzBldr.mda=rw  
wzLib.mda=rw  
wzTable.mda=rw  
wzQuery.mda=rw  
wzFrmFpt.mda=rw  
wzBldr.mda=rw

Note that duplicate entries that do not have a path are listed in the [Libraries] section after the reinstallation.

#### RESOLUTION

=====

Make the following changes to the MSACC20.INI file:

1. Modify the SystemDB line in the [Options] section to point to the correct path for the SYSTEM.MDA file.
2. Delete the entries in the [Libraries] section that do not specify a full path.

#### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

#### MORE INFORMATION

=====

#### Steps to Reproduce Problem

-----

1. On a workstation running Microsoft Access, quit Microsoft Access and then make a backup copy of your MSACC20.INI file (for example, copy the file to MSACC20.SAV).
2. Double-click the MS Access Setup icon in your Microsoft Access program group, and then choose the Reinstall button.
3. After Setup is finished, open the MSACC20.INI file and note the incorrect settings in the [Options] and [Libraries] sections.
4. Either correct the MSACC20.INI file as described in the "Resolution" section above, or replace it with your backup copy.

#### REFERENCES

=====

For more information about settings in the MSACC20.INI file, search for "MSACC20.INI" then "Customizing MSACC20.INI Settings" using the Microsoft Access Help menu.

Additional reference words: 2.00 net network  
KBCategory: kbsetup kberrmsg  
KBSubcategory: StpNet





**PRA: Export to MS Excel Uses Version 4.0 File Format**  
**Article ID: Q120490**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

If you have Microsoft Excel version 5.0 installed on your computer, and you export a table to Microsoft Excel using the Analyze It With MS Excel button, the table is exported to Microsoft Excel 5.0 as a Microsoft Excel version 4.0 spreadsheet. If you make any changes to the sheet in Microsoft Excel and then try to save it, you receive the message:

'<Spreadsheet name>' was created in a previous version of Microsoft Excel. Do you want to update it to Microsoft Excel 5.0 format?

CAUSE

=====

The Analyze It With MS Excel button exports only in Microsoft Excel version 4.0 file format.

RESOLUTION

=====

To export a table in Microsoft Excel version 5.0 file format, choose Export from the File menu, then select Microsoft Excel 5.0 in the Data Destination box.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

NOTE: This example assumes that you have Microsoft Excel version 5.0 installed on your computer.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. In the Database window, select the Categories table.
3. Choose the Analyze It With MS Excel button on the toolbar.
4. Once the table is exported to Microsoft Excel, change any cell in

the spreadsheet.

5. From the File menu, choose Save. You will receive the message:

'CATEGORI.XLS' was created in a previous version of Microsoft Excel. Do you want to update it to Microsoft Excel 5.0 format?

#### REFERENCES

=====

For more information about using the Analyze It With MS Excel button to export to Microsoft Excel, search for "Analyze It button" then "Output To MS Excel button" using the Microsoft Access Help menu.

Additional reference words: 2.00 exporting

KBCategory: kbinterop kberrmsg

KBSubcategory: IsmExl5

## **PRA: Exporting Data to MS Excel 5.0 Twice Causes Error**

Article ID: Q119806

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Excel for Windows, version 5.0
- 

### SYMPTOMS

=====

If you export data from Microsoft Access to a Microsoft Excel version 5.0 workbook twice, either manually or using the TransferSpreadsheet macro action, and then try to delete extra worksheets from the workbook, you will receive the error message:

Not enough memory

### RESOLUTION

=====

Do not export Microsoft Access data directly to a Microsoft Excel 5.0 workbook. Instead, use one of the following methods:

- Export the data, either manually or using the TransferSpreadsheet macro action, to a Microsoft Excel version 4.0 workbook.
- Export the data, either manually or using the TransferSpreadsheet macro action, to a temporary Microsoft Excel 5.0 workbook, and then use a Microsoft Excel macro to transfer the data from one workbook to another.

If you accidentally export data to a Microsoft Excel 5.0 workbook, follow these steps to correct the problem:

1. In Microsoft Excel, open the workbook to which you exported the data, and then choose Save As from the File menu.
2. In the Save File As Type box, select Microsoft Excel 4.0 Workbook, and then choose OK. Note that the file's extension changes to ".XLW".
3. Close the workbook.
4. Open the .XLW file you created in step 2.
5. From the File menu, choose Save As.
6. In the Save File As Type box, select Microsoft Excel Workbook, and then choose OK.
7. When you are prompted, choose Yes to replace the existing file.

### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Create a new Microsoft Excel version 5.0 workbook. Type "1" (without quotation marks) in cell A1, and type "2" (without quotation marks) in cell B1.
2. Delete worksheets 5-16 by selecting their tabs and then choosing Delete Sheet from the Edit menu.
3. Save the workbook as BOOK1.XLS. Make sure to save the file in Microsoft Excel Workbook format (the default format).
4. Close the workbook and then minimize Microsoft Excel.
5. Start Microsoft Access version 2.0 and open the sample database NWIND.MDB.
6. Create the following new macro, and then save it as TestMacro:

| Macro Name | Macro Actions       |
|------------|---------------------|
| TestMacro  | TransferSpreadsheet |

| TestMacro Actions                     |
|---------------------------------------|
| TransferSpreadsheet                   |
| Transfer Type: Export                 |
| Spreadsheet Type: Microsoft Excel 5.0 |
| Table Name: Customers                 |
| File Name: <drive>:\<path>\BOOK1.XLS  |

7. Create a blank, new form.
8. Add a command button to the form. Set the command button's OnClick property to the TestMacro macro.
9. Save the form, and then view it in Form view.
10. Choose the command button twice.
11. Switch to Microsoft Excel.
12. Open the BOOK1.XLS workbook.
13. Try to delete the new worksheets. You will receive the error message stated earlier in this article.

NOTE: This problem occurs with the second export to the same Microsoft Excel 5.0 workbook. If you export the data only once, the problem does not

occur. The problem will also occur if you export the data, delete the new worksheet, export the data again, and then try to delete the new worksheet.

#### REFERENCES

=====

For more information about the TransferSpreadsheet macro action, search for "TransferSpreadsheet" then "TransferSpreadsheet Action" using the Microsoft Access Help menu.

Microsoft Access "User's Guide," version 2.0, Chapter 26, "Using Macros to Print Reports or Transfer Data," page 700

Additional reference words: 2.00 interoperability

KBCategory: kbinterop kberrmsg

KBSubcategory: IsmEx15

**PRA: Expression Builder Creates Incorrect Section Syntax**  
**Article ID: Q119805**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you use the Expression Builder to refer to a form or report's Section property, the syntax created by the Expression Builder is incorrect.

CAUSE

=====

The Expression Builder incorrectly references sections as if they were controls on the form or report.

RESOLUTION

=====

To reference a form or report's Section property, create the expression manually instead of using the Expression Builder.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Create a command button in the form header section of any form.
2. In the command button's OnClick property field, choose the Build button.
3. In the Choose Builder box, select Code Builder, then choose OK.
4. Choose the Build button on the form module's toolbar.
5. In the Expression Builder, select Detail in the second column, and Visible in the third column, and then choose the Paste button.
6. Choose the "=" button, then type "0" (without quotation marks). The expressions should read as follows:

Me![Detail].Visible = 0

7. Choose OK.
8. Close the form module and then view the form in Form view.
9. Choose the new command button. You will receive the error message:

Invalid reference to field 'Detail'

The correct expression should read as follows:

Me.Section(0).Visible = 0

#### REFERENCES

=====

For more information about the Expression Builder, search for "Expression Builder" then "Expression Builder" using the Microsoft Access Help menu.

Additional reference words: 2.00 eb wizard  
KBCategory: kbusage kberrmsg  
KBSubcategory: ExrBldr



**PRA: Extra White Space Between Last Section and Report Footer**  
**Article ID: Q116061**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

There is extra white space on your report after the last detail record or section of the report, before the report footer section prints. The extra white space is as tall as the detail section height. Note that this problem occurs only with reports that have a report footer section.

This problem can occur with reports created in Microsoft Access version 2.0, and with reports created in Microsoft Access version 1.x that you are using in version 2.0.

RESOLUTION

=====

Change the report to have a vertical layout by following these steps:

1. Open the report in Design view. From the File menu, choose Print Setup. Choose the More button.
2. In the Items Across box, enter "2" (without quotation marks).
3. In the Item Layout area, select the Vertical option button.
4. In the Items Across box, enter "1" (without quotation marks).
5. Choose OK.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Open the Alphabetical List Of Products report in Design view.
3. Increase the height of the report footer section to 1 inch.
4. Set the report footer section's BackColor property to 16711680 (blue).

5. Set the Product Name footer section's BackColor property to 255 (red).  
(NOTE: Changing the sections' colors helps to illustrate the problem.)
6. Preview the report. Note the extra white space between the Product Name footer (red) and the report footer (blue).
7. Close the report without saving the changes.

Additional reference words: 2.00 line spacing whitespace canshrink blank  
KBCategory: kbusage  
KBSubcategory: RptLayout

**PRA: GP Fault Creating Report Based on Query with Subquery**  
**Article ID: Q121898**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you create a report based on a parameter query that contains a subquery, you receive the following general protection (GP) fault error message:

MSACCESS caused a General Protection Fault in module MSAJT200.DLL  
at <address>

This error occurs regardless of whether the parameter is declared, and regardless of whether the report is created manually or with a Report Wizard.

RESOLUTION

=====

Convert your parameter query to a make-table query and then run the query to create a new table. Use the new table as the record source for the report, rather than the parameter query.

STATUS

=====

This problem no longer occurs with the Microsoft Jet database engine version 2.5, which is available with the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927  
TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

WARNING: Following these steps will cause a GP fault on your computer. Make sure to save and close any open work on your computer before following these steps.

1. Start Microsoft Access and create a new database called SUBQRY.MDB.
2. Import the Categories and Products tables from the sample database NWIND.MDB into the SUBQRY database.
3. Create the following new query based on the Categories and Products

tables, and then save the query as Query1.

NOTE: In the following sample query, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this query.

Query: Query1

```

Field: Category ID
 Table: Categories
Field: Supplier ID
 Table: Products
 Criteria: [Enter a Supplier ID]
Field: Cat: (SELECT [Category Name] FROM [Categories] _
 WHERE [Products].[Category ID]=_
 [Categories].[Category ID])
```

4. Close the query.
5. Select the Query1 query in the Database window.
6. Click the AutoReport button on the toolbar.

Additional reference words: 2.00 gpf jet25 jet 2.5  
KBCategory: kbusage kberrmsg  
KBSubcategory: RptProb

**PRA: GP Fault in STORAGE.DLL with Forms Switchboard Form**  
**Article ID: Q121694**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

If you try to change the color of the background bitmap for the Forms Switchboard form in the version 1.x sample database NWIND.MDB in Microsoft Access version 2.0, you will receive a general protection (GP) fault in STORAGE.DLL.

RESOLUTION

=====

This problem does not occur with any other forms, including those that use the same bitmap. Creating a new form with the same bitmap may be a workaround.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

STORAGE.DLL is an OLE dynamic-link library (DLL) file that is used by applications to provide a more abstract view of OLE files. It allows a document file to be viewed as if it were an entire file system, complete with subdirectories and files that can be moved, copied, and deleted. This structure is necessary for providing a location for OLE objects (each object requires its own subdirectory inside a file), and for providing compatibility among applications.

Steps to Reproduce Problem

-----

Notes:

- Following these steps will cause a GP fault on your computer. Make sure to save and close any open work on your computer before following these steps.
  - This example assumes that you have a copy of the sample database NWIND.MDB from Microsoft Access version 1.x on your computer.
1. Start Microsoft Access version 2.0 and create a new database.
  2. From the File menu, choose Import. In the Data Source box, select

Microsoft Access, and then choose OK.

3. In the File Name box, select the version 1.x NWIND.MDB file, and then choose OK.
4. In the Object Type box, select Forms. In the Objects In NWIND.MDB box, select Forms Switchboard, and then choose the Import button.
5. When you receive the message "Successfully imported 'Forms Switchboard'," choose OK, and then choose the Close button.
6. Open the Forms Switchboard form in Design view.
7. Double-click the background bitmap to start Windows Paintbrush.
8. Select the Paint Roller tool and change the bitmap's color.
9. Quit Windows Paintbrush and update the object.
10. Save and then close the Forms Switchboard form.
11. Open the Forms Switchboard form in Form view. You receive a GP fault in STORAGE.DLL.

#### REFERENCES

=====

For additional information about the STORAGE.DLL file, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q108392  
TITLE : Definitions of the OLE 2.0 DLLs

Additional reference words: 2.00 gpf  
KBCategory: kbusage kberrmsg  
KBSubcategory: FmsProb

**PRA: GP Fault Using Query Wizards and Mono VGA Video Driver**  
**Article ID: Q121468**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

You receive a general protection (GP) fault error message when you use the Find Unmatched Query Wizard or the Crosstab Query Wizard.

CAUSE

=====

You are using the Monochrome VGA video driver for your Microsoft Windows video driver.

RESOLUTION

=====

To work around this problem, use the standard VGA video driver, and select the Monochrome color scheme in Windows Control Panel.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

The Monochrome VGA video driver is a popular driver for use on laptop computers. However, using this video driver will cause a GP fault when you use the Microsoft Access Wizards mentioned above.

Steps to Reproduce Problem

-----

NOTE: This example assumes that you have the Monochrome VGA video driver installed.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. In the Database window, choose the Query button, and then choose the New button. In the New Query box, choose the Query Wizards button.
3. In the "Which Wizard do you want?" box, select Find Unmatched Query. Choose OK.
4. In the "Select the table or query that contains the records you want

to see in the query's results..." box, select Categories, and then choose the Next button.

5. In the "Select the table or query that contains the related records." box, select Customers, and then choose the Next button.
6. In the Fields In Categories box, select Category Name. In the Fields In Customers box, select Customer ID. Choose the "<=>" button, and then choose the Next button.
7. Choose the ">>" button, and then choose the Next button. You will receive a GP fault error message.

Additional reference words: 2.00 gpf

KBCategory: kbusage kberrmsg

KBSubcategory: WzProb



**PRA: GPF Closing Data Outline Control Properties Box**  
**Article ID: Q117610**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you add a data outline control to a form or report, then edit the control's properties, the Properties dialog box is normally modal, meaning that you cannot select anything outside the dialog box.

If, however, you click the property sheet for the form or report containing the data outline control, the property sheet gets the focus, and the modality of the data outline control's Properties dialog box is lost. You can then select anything in the database.

When the modality of the control's Properties dialog box is broken as outlined above, your mouse pointer may disappear, and you may receive a general protection (GP) fault.

CAUSE

=====

The GP fault and disappearing mouse pointer problem occurs only if you close a form while "orphan" windows are still open. These orphan windows are data outline control windows such as the Properties dialog box, or the Edit Control window.

This problem occurs because there is no way for a non-in-place active OLE server application (such as the data outline control) to instruct its container (in this case, Microsoft Access) to disable modeless dialog boxes when opening a modal dialog box on the container's behalf.

RESOLUTION

=====

Avoid closing the form or report's Properties dialog box while you are editing the data outline control.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

WARNING: Following these steps will cause a GP fault. You should save and close any open work on your computer before following these steps.

1. Start Microsoft Access and open any database.
2. Create a new, blank form.
3. Open the form's property sheet.
4. Place an unbound OLE control on the form.
5. In the Insert Object dialog box, select the Insert Control option button.
6. In the Control Type box, select Data Outline Control, then choose OK.
7. Click the secondary mouse button in the data outline control, choose Data Outline Control Object, then choose Properties.
8. Click anywhere outside the Properties dialog box (except in the form's property sheet). Note that since the Properties dialog box is modal, you cannot select anything outside it.
9. Click the form's property sheet to select it, then select Event Properties in the property sheet.
10. Select the OnEnter property, then choose the Build button at the right side of the property field. Select Code Builder, then choose OK. The form module will open behind the Properties dialog box.
11. From the File menu, choose Exit. If you are prompted to save your changes, choose No. A GP fault will result.

#### REFERENCES

=====

For more information about OLE custom controls, search for "OLE custom controls" then "Insert Object Command (Edit Menu)" using the Microsoft Access Help menu.

Additional reference words: 2.00 gpf  
KBCategory: kbusage kberrmsg  
KBSubcategory: WzOthr

**PRA: GPF Using List Box Fill Function with Requery Method**  
**Article ID: Q117615**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you fill a list box using an Access Basic function, and the function returns a static variable upon receiving the close code (8), and you then use the Requery method to requery the list box, you receive a general protection (GP) fault.

CAUSE

=====

The temporary string created by the list box fill function is not freed, and conflicts with another variable used internally by Microsoft Access.

RESOLUTION

=====

Use one of the following techniques to work around this problem:

- Use a Requery action instead of the Requery method for requerying the list box.
- Do not return a static string when your function exits through the close code (8). That is, dimension the variable to make it dynamic, rather than declaring it static.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

**WARNING:** Following these steps will result in a GP fault. Save and then close any open work on your computer before following these steps.

**NOTE:** In the following example, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this example.

1. Start Microsoft Access and open the sample database NWIND.MDB.

2. Create a new, blank form.
3. Click the Code button on the toolbar, or choose Code from the View menu, and then enter the following code in the form module:

```

Option Compare Database 'Use database order for string _
 comparisons.

Dim iTotTables As Integer
Dim My_CurrDB As Database
Dim TableList2() As String
Dim iTotFlds As Integer
Dim FieldList() As wlib_FldInfo

Function FillTables (fld As Control, handle As Long, lRow As _
Long, lCol As Long, iMsg As Integer) As Variant

 Static returnval As Variant
 ' If you change the above line to the following the error
 ' will not occur:
 ' Dim returnval As Variant

 Select Case iMsg
 Case LB_INITIALIZE
 returnval = -1&
 Case LB_OPEN
 returnval = Timer
 Case LB_GETROWCOUNT
 returnval = iTotTables
 Case LB_GETCOLUMNCOUNT
 returnval = 1
 Case LB_GETCOLUMNWIDTH
 returnval = -1&
 Case LB_GETVALUE
 returnval = TableList2(lRow + 1)
 Case LB_GETFORMAT
 Case LB_CLOSE
 Case LB_END
 End Select
 FillTables = returnval
End Function

```

4. Add the following command button and list box to the form:

```

Command button:
Name: Button0
Caption: Requery Table List

```

```

List box:
Name: Field1
RowSourceType: FillTables

```

5. Set the command button's OnClick property to the following event procedure:

```

Sub Button0_Click ()
 Set My_CurrDB = DBEngine.Workspaces(0).Databases(0)

```

```
 iTotTables = wlib_GetObjList(My_CurrDB, A_TABLE, TableList2(), _
 False, &HFFFFFFF)
 Me!Field1.Requery
 End Sub
```

6. View the form in Form view.

7. Click the Requery Table List button 8 or more times. You will receive the error message:

General Protection Fault in MSABC200.DLL at 000B:07E1

#### REFERENCES

=====

For more information about using Access Basic code to fill a list box, search for "filling list boxes/combo boxes" then "Filling a List Box or Combo Box Using an Access Basic Function" using the Microsoft Access Help menu.

Additional reference words: 2.00 programming gpf

KBCategory: kbprg kberrmsg

KBSubcategory: PgmOthr

**PRA: Graph 5.0 Legend Moves Each Time Graph Is Refreshed**  
**Article ID: Q117613**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Graph version 5.0
- 

SYMPTOMS

=====

Each time a Microsoft Graph 5.0 object is refreshed, the legend moves to a new location.

RESOLUTION

=====

Although you cannot prevent the legend from moving when the graph is refreshed, you can use OLE Automation to reset the legend's position. See the "Workaround" section later in this article for an example of this technique.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Graph version 5.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

A graph in a Microsoft Access form or report is refreshed when any of the following occurs:

- If the graph is linked to the data in the form or report, the graph is refreshed each time a new record is displayed.
- When you press F9.
- When the form containing the graph is refreshed by Access Basic code or the Refresh macro action.

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Open the form Sales By Product in Design view.
3. Double-click the graph object to open it in Microsoft Graph 5.0.
4. From the Insert menu, choose Legend.
5. With the legend selected, choose Selected Legend from the Format menu.

Click the Placement tab, then choose the Bottom option button. Choose OK.

6. Quit Graph to return to Microsoft Access.
7. View the form in Form view.
8. Click a record selector button at the bottom of the form to move to another record. Note that the legend moves up the graph when the graph is refreshed. As you move through the records, the legend moves further up the graph.

#### Workaround

-----

This example continues the example started in the "Steps to Reproduce Problem" section of this article. After completing the steps in that section, follow these steps to use OLE Automation to reset the legend to the bottom of the graph each time the graph is refreshed:

9. View the form in Design view.
10. Select the graph object and display the property sheet. Select the OnUpdated property and then click the Build button to the right of the property. Select Code Builder and then choose OK.
11. Enter the following line in the module:

```
Me!Embedded13.Object.Legend.Top = 171
```

NOTE: The value 171 was obtained by opening the Sales By Product form in Design view, opening the form module, and then entering the following line in the Immediate window:

```
?Forms![Sales By Product]!Embedded13.Object.Legend.Top
```

12. From the Run menu, choose Compile Loaded Modules, then close the module.
13. Save and then close the form.
14. Open the form in Form view and move through the records. Note that the legend remains at the bottom of the graph.

To use this workaround in a report, specify the code in step 11 in the OnFormat property setting of the report section that contains the graph.

#### REFERENCES

=====

For more information about using OLE Automation, see Chapter 13 of the Microsoft Access version 2.0 "Building Applications" manual.

The properties, methods, and objects supported by Microsoft Graph version 5.0 are documented in a Help file called VBA\_GRP.HLP. This file is available in the ADT/ADK section of the MSACCESS forum on CompuServe.

Additional reference words: 2.00 Chart Example  
KBCategory: kbtool  
KBSubcategory: IntpGrph



**PRA: Help Topic "Converting Code" Sample Code Incorrect**  
**Article ID: Q114732**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

The sample code for running a pass-through query in the Microsoft Access version 2.0 Help system topic "Examples of Converting Code to Version 2.0" is not correct. If you run this code, you will receive one of the following error messages:

Object variable not set or Name not found in this collection

-or-

Invalid SQL statement; expected 'DELETE', 'INSERT', 'PROCEDURE', 'SELECT', or 'UPDATE'

RESOLUTION

=====

The incorrect sample code is as follows:

```
' Given open Database object MyDB.
Dim Q As QueryDef
Set Q = MyDB.CreateQueryDef("MYODBCQuery")
Q.SQL = "Exec My_Stored_Procedure"
Q.Connect = "ODBC; DSN=MyServer; UID=sa; PWD=hithere; DATABASE=pubs"
Q.ReturnsRows = True

Dim R As Recordset
Set R = Q.OpenRecordset()
```

To correct the code, make the following two changes:

1. Change the line that reads

```
Q.ReturnsRows = True
```

to be:

```
Q.ReturnsRecords = True
```

The correct name for the property is ReturnsRecords.

2. Move the line that reads

```
Q.SQL = "Exec My_Stored_Procedure"
```

so that it is directly beneath the line that reads:

```
Q.Connect = "ODBC; DSN=MyServer;..."
```

The syntax "Exec My\_Stored\_Procedure" is only valid in a pass-through query. Setting the connect property changes the new query from a select query to a pass-through query.

The corrected code should be as follows:

```
' Given open Database object MyDB.
Dim Q As QueryDef
Set Q = MyDB.CreateQueryDef("MYODBCQuery")
Q.Connect = "ODBC; DSN=MyServer; UID=sa; PWD=hithere; DATABASE=pubs"
Q.SQL = "Exec My_Stored_Procedure"
Q.ReturnsRecords = True

Dim R As Recordset
Set R = Q.OpenRecordset()
```

#### REFERENCES

=====

For more examples of converting code to version 2.0, search for "converting code" then "Examples of Converting Code to Version 2.0" using the Microsoft Access Help menu.

Additional reference words: 2.00 documentation

KBCategory: kbusage kbdocerr

KBSubcategory: DcmOthr

**PRA: Horizontal/Vertical Lines Missing from 600 dpi Printout**  
**Article ID: Q120762**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Windows NT workstation version 3.5
  - Microsoft Windows NT Server version 3.5
- 

SYMPTOMS

=====

Horizontal or vertical lines are missing from a document printed on a printer that prints 600 dots per inch (dpi).

CAUSE

=====

This problem occurs when you print from Microsoft Access to a 600 dpi printer using a printer driver that does not support the DRAWPATTERNRECT escape. When this occurs, Microsoft Access may miscalculate the number of pixels needed to print a solid line.

RESOLUTION

=====

Change the printer resolution from 600 dpi to 300 dpi.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

NOTE: This example assumes that you are printing to a 600 dpi printer.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Preview the Catalog report.
3. Print the first page. Note that several vertical lines are missing from the printed page.

Additional reference words: 2.00

KBCategory: kbui

KBSubcategory: UiToolbr

**PRA: Imported Microsoft Excel Data Contains Wrong Columns**  
**Article ID: Q120131**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you are importing data from a Microsoft Excel spreadsheet, Microsoft Access may import the wrong data if the specified range extends beyond column Z (the 26th column), or if the specified range begins after column Z.

CAUSE

=====

Microsoft Access miscounts columns by one after the 26th column. For example, if you specify an import range of A1:AD10 (30 columns and 10 rows), the actual range that will be imported is A1:AC10 (29 columns and 10 rows).

If the import range begins with a column beyond column Z, the first column imported will be one column to the left of the specified column. For example, if you specify an import range of AF1:BB4 (23 columns and 4 rows), the actual range that will be imported is AE1:BA4 (23 columns and 4 rows).

RESOLUTION

=====

There are three workarounds for this behavior:

- Create a named range in Microsoft Excel that includes all the cells you want to import, and then use that named range to import the data into Microsoft Access.
- Add an extra column to the range you specify when you are importing the data. For example, if you want to import the 27 columns in the range A1:AA1, specify the range A1:AB1.
- If the range you want to import begins after column Z, offset the column range by one. For example, if you want to import the range AF1:BB1, specify the range AG1:BC1.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

## Steps to Reproduce Problem

---

1. Create a new spreadsheet in Microsoft Excel version 3.x or later, and type any data in cells A1 through AA1.
2. Save the file as EXCELTST.XLS and then quit Microsoft Excel.
3. Start Microsoft Access and open the sample database NWIND.MDB.
4. From the File menu, choose Import.
5. In the Data Source box, select the appropriate Microsoft Excel source, and then choose OK.
6. In the File Name box, select the EXCELTST.XLS file you created in step 2, and then choose the Import button.
7. In the Import Spreadsheet Options dialog box, select the Create New Table option button, and then enter "A1:AA1" (without quotation marks) in the Range box. Choose OK.
8. Open the new EXCELTST table in Datasheet view. Note that only 26 columns (instead of the 27 you specified) were imported.

## REFERENCES

---

Microsoft Access "User's Guide," version 2.0, Chapter 9, "Importing, Exporting, and Attaching," pages 175-179

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: IeaEx1

**PRA: Input Mask Wizard Not Available with All DataTypes**  
**Article ID: Q113882**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you attempt to invoke the Input Mask Wizard you get the message:

The Input Mask Wizard only works with Text or Date/Time fields.

CAUSE

=====

The Microsoft Access Help topics "InputMask Property" and "Adding an Input Mask" do not identify which data types the Input Mask Wizard works with.

The Input Mask Wizard can be used with Text and Date/Time data types, but cannot be used with Memo, Number, Currency, Counter, Yes/No, or OLE Object data types.

RESOLUTION

=====

If you cannot use the Input Mask Wizard, enter the input mask manually in the InputMask property.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. Microsoft is researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

The Input Mask Wizard is invoked by choosing the Build button to the right of the InputMask property. The InputMask property applies to table fields, query fields, and text box controls on forms and reports. The Input Mask Wizard can be used with Text and Date/Time data types, and, in forms and reports, it can be used with unbound text box controls. Trying to use the Input Mask Wizard with all other data types will produce the error message stated earlier in this article.

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Select the Employees table in the Database window, and then choose the

AutoForm button on the toolbar.

3. Switch the form to Design view.
4. Select the Employee ID text box control in the detail section.
5. If the property sheet is not open, open it by choosing the Properties button on the toolbar.
6. Click in the InputMask property field.
7. Choose the Build button to invoke the Input Mask Wizard.

Additional reference words: 2.00 builders wizards

KBCategory: kbusage

KBSubcategory: WzOthr

## **PRA: Mailing Label Wizard Creates One-Page Report**

**Article ID: Q115124**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

A mailing label report created with the Mailing Label Wizard contains only one page, no matter how many records are in the underlying table or query. This problem occurs if your default printer is a laser printer, and you are using any of the following Avery mailing labels:

4145, 4162, 4163, 4249, 4250, 4251, 4253, 4254, or 5615

If your default printer is a dot-matrix printer, most mailing label reports using one-up continuous labels will contain only one page.

### CAUSE

=====

This problem occurs if you use the Row Spacing setting in the Print Setup dialog box to reserve the space between each record or label.

This problem also occurs with reports converted from Microsoft Access version 1.x, if the Row Spacing setting is used.

### RESOLUTION

=====

If you set the Row Spacing property to 0 the report can preview/print all data. From the File menu choose Print Setup, then press the More button and set the Row Spacing property to 0, then Press OK.

### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

=====

#### Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Create a new report based on the Customers table. Choose the Report Wizards button.
3. Select the Mailing Label Wizard, then choose OK.



4. In the Available Fields box, select Contact Name, then choose the ">" button. Choose Next.
5. Repeat step 4.
6. In the "What label size do you want?" box, select any of the following labels:  
  
4145, 4162, 4163, 4249, 4250, 4251, 4253, or 4254
7. Under Label Type, select the Continuous option button, and then choose Next.
8. Choose Next, and then choose Finish.

Note that the mailing label report that is created contains only one page.

#### REFERENCES

=====

For more information about creating mailing label reports, search for "creating mailing labels" then select a topic using the Microsoft Access Help menu.

Additional reference words: 2.00 mailing label

KBCategory: kbusage

KBSubcategory: RptLbl

**PRA: Navigation Buttons Not Enabled, Disabled Correctly**  
**Article ID: Q119473**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

The navigation buttons on a form's toolbar are not enabled and disabled correctly when you move between a main form and a subform.

CAUSE

=====

The screen is not being refreshed correctly.

RESOLUTION

=====

Press the F9 key or requery the form to refresh the screen.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the sample database NWIND.MDB
2. Open the Orders form in Form view.
3. Make sure the Orders form toolbar is visible. If it is not, choose Toolbars from the View menu, select Form View, choose the Show button, then choose Close.
4. From the View menu, choose Toolbars. Choose the Customize button, then select Records in the Categories box. Drag the Previous Record and Next Record buttons from the Buttons box to the form's toolbar. Choose Close.
5. Note that the Previous Record button is disabled, since the first record is the current record (there is no previous record).

6. Click the Next Record button. Note that the second record becomes the current record, and that the Previous Record button is enabled.
7. Select the first record in the subform. Note that the Previous Record button is still enabled, even though there is no previous record in the subform.
8. Select the second record in the subform, and then select the first record in the subform again. Note that the Previous Record button is now disabled.
9. Click anywhere in the main form. Although the second record is the current record, the Previous Record button still appears disabled. Note that you can use the Previous Record button even though it appears disabled.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Appendix B, "Working with Toolbars," pages 711-727

Additional reference words: 2.00 navigate tool bar

KBCategory: kbui

KBSubcategory: UiToolbr

**PRA: No Serial Number In the About Microsoft Access Dialog Box**  
**Article ID: Q114319**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

There is no serial number listed in the About Microsoft Access dialog box.

CAUSE

=====

A limited number of Microsoft Access 2.0 disks were inadvertently manufactured without a serial number listed in the About Microsoft Access dialog box.

RESOLUTION

=====

The serial number (also known as the Product ID number, or PID) is contained in the product, but is not visible in the dialog box. The serial number is only used internally by Microsoft Access, and it is not necessary that you see or know it.

If, however, you want to find the serial number of your copy of Microsoft Access version 2.0, follow these steps:

1. Rename the SETUP.STF file in the \SETUP subdirectory of your Microsoft Access program directory to SETUP.BAK.
2. Run the Setup program from your original Microsoft Access disk number 1.
3. Choose OK until you reach the screen that displays the Product ID number at the top. Write this number down, and then choose OK.
4. Choose the Exit Setup button to cancel the Setup program.
5. Rename the SETUP.BAK file back to SETUP.STF.

STATUS

=====

Microsoft has confirmed this to be a problem with a limited number of Microsoft Access version 2.0 installation disks. This problem has been corrected.

Additional reference words: 2.00

KBCategory: kbsetup

KBSubcategory: StpOthr

**PRA: No Visible Property for Form Page Header/Footer Sections**  
**Article ID: Q120524**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

In Microsoft Access version 2.0, form page header and page footer sections do not have a Visible property as they do in Microsoft Access version 1.x.

If you convert a form that was created in version 1.x to version 2.0, and that form has a page header or page footer section whose Visible property is set to No, there is no way in version 2.0 to set the Visible property for that section back to Yes.

CAUSE

=====

In Microsoft Access version 2.0, the Visible property can be set at run time for the form header, form footer, and detail sections. As a side effect, the Visible property is disabled for the page header and page footer sections.

RESOLUTION

=====

One workaround for this problem is to create a new form in Microsoft Access version 2.0 and then copy all of the controls from the version 1.x form into the new form. When you do this, the page header and page footer sections in the new form will be visible.

Another workaround is to use a report to print the data instead of a form. (Since the page header and page footer sections are used only when you print the form, you could easily substitute a report instead.) Page header and page footer sections in reports in Microsoft Access version 2.0 do have a Visible property. To save a form as a report, open the form in Design view, and then choose Save As Report from the File menu.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

REFERENCES

=====

For more information about the Visible property, search for "Visible" then "Visible Property" using the Microsoft Access Help menu.

Additional reference words: 2.00 forms invisible hide

KBCategory: kbusage  
KBSubcategory: FmsProp

**PRA: ODBCTimeout Ignores MSACC20.INI ODBC Setting**  
**Article ID: Q113883**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

The ODBCTimeout property in new queries defaults to 60 seconds, regardless of the value set in the [ODBC] section of your MSACC20.INI file.

In databases converted from Microsoft Access version 1.x, the ODBCTimeout property defaults to 0, regardless of the value set in the [ODBC] section of your MSACC20.INI file.

RESOLUTION

=====

You can use the sample subprocedure listed below to set the ODBCTimeout property for all the queries in your database. Note that the ODBCTimeout property is ignored in queries that do not use attached tables.

Sample Subprocedure

-----

```
Sub ChangeAllQueryTimeOuts (timeinterval As Integer)
 Dim db As Database, qd As QueryDef, i As Integer
 Set db = dbengine.workspaces(0).databases(0) 'Use current database.
 BeginTrans
 'Start a transaction.
 'Loop through Querydefs and change the
 'ODBC timeout.
 For i = 0 To db.querydefs.count - 1
 Set qd = db.querydefs(i)
 qd.odbc.timeout.value = timeinterval 'Set ODBC timeout.
 Next i
 CommitTrans
 'Finished: commit the transaction.
 db.querydefs.Refresh 'Update the Querydefs collection.
End Sub
```

How to Run the Subprocedure

-----

To run the ChangeAllQueryTimeOuts subprocedure, type the following in the module's Immediate window, and then press ENTER:

```
ChangeAllQueryTimeOuts <time value>
```

Substitute a value, in seconds, for <time value> in the sample line above.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version

2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

#### MORE INFORMATION

=====

The QueryTimeout setting in the [ODBC] section of the MSACC20.INI file applies only to queries that do not have a setting for the ODBCTimeout property. If a query's ODBCTimeout property is blank, then the QueryTimeout setting is used.

An ODBCTimeout value of 0 means that there is no timeout. If the ODBC server does not respond, the query will never timeout and return control.

In Microsoft Access version 1.x, The QueryTimeout and ODBCTimeout values are set by the QueryTimeout setting in the [ODBC] section of the MSACCESS.INI file. This value applies to all the ODBC queries run in the current Microsoft Access session.

In Microsoft Access version 2.0, these values can differ. Each query can have its own ODBCTimeout setting.

#### Steps to Reproduce Problem

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open any query in Design view.
3. Set the query's ODBCTimeout property to null (that is, delete any existing value for the property).
4. Save and then close the query.
5. Reopen the query in Design view. Note that the ODBCTimeout property is set to 0.

#### REFERENCES

=====

For more information about the ODBCTimeout and QueryTimeout settings, search for "ODBCTimeout" and "QueryTimeout" using the Microsoft Access Help menu.

Additional reference words: 2.00 ODBC Attaching  
KBCategory: kbusage  
KBSubcategory: ObcOthr



**PRA: ODBCTimeout Set to Zero for New Pass-Through Queries**  
**Article ID: Q117537**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

When you create a SQL pass-through query using Access Basic, the ODBCTimeout property setting for the query is set to zero, despite the timeout setting you specify.

RESOLUTION

=====

When you use Access Basic to create a SQL pass-through query, set the ODBCTimeout property after you set the Connect property for the query. If you do not do this, the ODBCTimeout property is set to zero.

MORE INFORMATION

=====

The ODBCTimeout property setting specifies the number of seconds Microsoft Access waits before a timeout error occurs on a query involving an ODBC database. When this property is set to zero, no timeout occurs.

Steps to Reproduce Problem

-----

NOTE: This example assumes that you have an ODBC data source (such as Microsoft SQL Server) available.

1. Open the sample database NWIND.MDB.
2. Create a new module and enter the following sample procedure:

```
Sub CreateSPTQuery()
 Dim db As Database, q As QueryDef
 Set db = dbengine(0)(0)
 'Create a new QueryDef.
 Set q = db.CreateQueryDef()
 q.Name = "MySPTQuery"
 'Modify the line below to select from an existing ODBC table.
 q.SQL = "Select * From authors;"
 q.ODBCTimeout = 60
 q.Connect = "ODBC;"
 q.ReturnsRecords = True
 'Add the QueryDef to the QueryDefs collection.
 db.QueryDefs.Append q
 db.QueryDefs.Refresh
 'Print the ODBCTimeout setting to the Immediate window.
 Debug.Print db.QueryDefs("MySPTQuery").ODBCTimeout
End Sub
```

3. In the Immediate window, test this procedure by typing:

```
CreateSPTQuery
```

Microsoft Access will print a zero (0) to indicate the ODBCTimeout setting for the query.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, pages 291-294

Additional reference words: 2.00 spt passthrough

KBCategory: kbusage

KBSubcategory: QryPass

**PRA: OLE Objects Clipped Using Fahrenheit Video Driver**  
**Article ID: Q109344**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

OLE objects inserted into a Microsoft Access form will be clipped on the bottom and on the right side if you are using an Orchid Fahrenheit video driver in Microsoft Windows.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0, 1.1, and 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

The following Orchid Fahrenheit drivers are known to cause this problem:

- All 800 x 600 resolution drivers
- The 1024 x 768 16 colors large font driver

Steps to Reproduce Problem

-----

1. Install any of the above listed video drivers in Microsoft Windows.
2. Start Microsoft Access. Open any database.
3. Open a new form.
4. From the Edit menu, choose Insert Object.
5. Select Microsoft Graph, and then choose OK.
6. Close Microsoft Graph. Choose Yes in the Update Changes box.

Note that the chart is clipped on the bottom and on the right side.

Additional reference words: 1.00 1.10 2.00 Interoperability

KBCategory: kbole

KBSubcategory: IntpOle

**PRA: OnError Property Does Not Trap Locking Error in Form**  
**Article ID: Q116065**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

If you have a form's RecordLocks property set to No Locks and a multiuser record locking error occurs, the form's Error event is not triggered, and the normal "This record has been changed by another user since you started editing it..." dialog box appears.

CAUSE

=====

The Error event is triggered when a run-time error is produced in Microsoft Access. The multiuser record locking error mentioned above is not a trappable run-time error.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

The Error event is triggered when a run-time error is produced in Microsoft Access. If you set the form's OnError property to a macro or event procedure, Microsoft Access will run the macro or event procedure when a run-time error occurs.

If you set a form's RecordLocks property to No Locks, no error will be trapped when two users modify the same record at the same time. Instead, Microsoft Access will display a dialog box to the user who attempted to save the record last, prompting the user to save the changes, discard the changes, or copy the changes to the Clipboard.

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB. Make sure to clear the Exclusive check box in the Open Database dialog box.
2. Open the Employees form in Design view.
3. Set the form's OnError property to:

`=MsgBox("An error has occurred.")`

4. From the View menu, choose Options.
5. In the Category box, select Multiuser/ODBC. In the Items box, set Default Record Locking to No Locks.
6. Save and close the Employees form.
7. Start another instance of Microsoft Access, or start Microsoft Access on another computer.
8. In the second instance of Microsoft Access, or on the other computer, open the same NWIND.MDB file. Make sure to clear the Exclusive check box in the Open Database dialog box.
9. In both copies of Microsoft Access, open the Employees form in Form view.
10. In both copies of Microsoft Access, edit the Title field of the first record.
11. In the second copy of Microsoft Access, move to the next record.
12. In the first copy of Microsoft Access, move to the next record. Note that the Error event is not triggered.

#### REFERENCES

=====

For more information about Error events, search for "error events" then "Error and Timing Events" using the Microsoft Access Help menu.

Microsoft Access "User's Guide," version 2.0, pages 107 and 224

Additional reference words: 2.00 optimistic

KBCategory: kbusage

KBSubcategory: FmsProp

**PRA: Only One Page Created for Legal Size Form or Report**  
**Article ID: Q120146**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you specify a legal-size page (8.5 by 14 inches) for your form or report, only one page is created when you print or preview the report.

RESOLUTION

=====

To work around this problem, follow these steps to force the printer setup for the form or report to a vertical layout:

1. In Microsoft Access, choose Print Setup from the File menu.
2. In the Print Setup dialog box, choose the More button. Note that the Item Layout option buttons are dimmed.
3. Change the Items Across setting to 2 and then press the TAB key. Note that the Item Layout option buttons are no longer dimmed.
4. Select the Vertical option button in the Item Layout group.
5. Set the Items Across setting back to 1.
6. Choose OK.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Create a blank, new report based on the Customers table. Drag the Company Name field from the field list to the report's detail section.
3. Set the Height property for each of the following sections as indicated:

| Section | Height |
|---------|--------|
|---------|--------|

-----  
page header 1.75 inches  
detail 10.75 inches  
page footer 0 inches

4. From the File menu, choose Print Setup.
  - a. In the Paper group, select "Legal 8 1/2 x 14 in" in the Size box.
  - b. In the Orientation group, select the Portrait option button.
  - c. In the Margins group, set the Top setting to 0.25 inches.
  - d. In the Margins group, set the Bottom setting to 1.25 inches.
  - e. Choose OK.
  
5. Preview the report. Note that only one page is created for the report.

Additional reference words: 2.00  
KBCategory: kbusage  
KBSubcategory: RptProb

**PRA: Options in Form and Report Wizards Are Not Displayed**  
**Article ID: Q103173**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you select the Form Wizard button or the Report Wizard button, an empty list box is displayed.

CAUSE

=====

The [Form Wizards] and [Report Wizards] sections of the MSACCESS.INI file (in Microsoft Access version 1.x) or the MSACC20.INI file (in Microsoft Access version 2.0) either do not exist or are empty.

RESOLUTION

=====

In version 2.0, using Notepad or another text editor, add the following sections in your MSACC20.INI file. The MSACC20.INI file is located in your Windows directory.

```
[Form Wizards]
Single-Column=zwform,1
Tabular=zwform,2
Graph=zwgraph,,
Main/Subform=zwmainsub,,
Autoform=zwautoform,,

[Report Wizards]
Single-Column=zwreport,3
Groups/Totals=zwreport,4
Mailing Label=zwmailinglabel,,
Summary=zwreport,12
Tabular=zwreport,11
Autoreport=zwautoreport,,
MSWordMailmerge=PM_Entry,,

[Libraries]
wzlib.mda=rw
```

In version 1.x, using Notepad or another text editor, add the following sections in your MSACCESS.INI file. The MSACCESS.INI file is located in your Windows directory.

```
[Form Wizards]
Single-Column=zwInitWizard, 1
Tabular=zwInitWizard, 2
Graph=zwInitWizard, 6
Main/Subform=zwInitWizard, 7
```



[Report Wizards]  
Single-Column=zwInitWizard, 3  
Groups/Totals=zwInitWizard, 4  
Mailing Label=zwInitWizard, 5

[Libraries]  
wizard.mda=ro

Additional reference words: 1.00 1.10 2.00 option wizard blank disappear  
ini

KBCategory: kbusage

KBSubcategory: FrmrptWzd

## **PRA: Oracle Driver Errors on Delimited Column Names**

**Article ID: Q114840**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0 and 1.1 and 2.0
- 

### SYMPTOMS

=====

This information is extracted from an article in the LANPROD knowledge base. To find the most current information, see the "References" section later in this article.

The Oracle ODBC Driver gives an error on any SELECT statement that contains an outer join (in the ODBC escape-clause syntax) on a delimited column name that is greater than 17 characters. As a result, when you are using Microsoft Access or Microsoft Visual Basic to connect to an Oracle server, you cannot perform outer joins on columns whose names are longer than 17 characters.

### RESOLUTION

=====

Rename the joined columns on the server so that each column has less than 17 characters, then re-attach the tables.

### REFERENCES

=====

For more information, please see the following article(s) in the Microsoft Knowledge Base:

ARTICLE-ID: Q104958

TITLE : BUG: Oracle Driver Errors on Delimited Column Names

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr

## **PRA: Ordinal Number in SQL ORDER BY Clause Ignored**

**Article ID: Q123709**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you order the data in a query using an ordinal number in the query's SQL ORDER BY clause, the ORDER BY clause is ignored. The data is not ordered, and no error message is generated.

### CAUSE

=====

The Microsoft Jet database engine version 2.0 does not support the ordering of data by ordinal numbers in a SQL ORDER BY clause.

### RESOLUTION

=====

The Microsoft Jet database engine version 2.5 does support the ordering of data by ordinal numbers in a SQL ORDER BY clause.

### STATUS

=====

This problem no longer occurs with the Microsoft Jet database engine version 2.5, which is available with the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

### MORE INFORMATION

=====

Ordering data by ordinal numbers refers to ordering data using a numeric reference in the SQL ORDER BY clause, rather than by a field reference. For example, in the sample query

```
SELECT DISTINCTROW Table1.Names, Table1.Numbers
FROM Table1
ORDER BY Table1.Names DESC;
```

replace the field reference in the ORDER BY clause with:

```
ORDER BY 1 DESC;
```

The number in the ORDER BY clause refers to the order of the fields in the SELECT statement. In the example above, 1 refers to the first field, Names.

## Steps to Reproduce Problem

---

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new query based on the Employees table.
3. Drag the First Name and Last Name fields from the field list to the query grid.
4. From the View menu, choose SQL.
5. Change the SQL statement so that it reads as follows:

```
SELECT DISTINCTROW Employees.[First Name], Employees.[Last Name]
FROM Employees
ORDER BY 1 DESC;
```

6. Run the query. Note that the data is not sorted on the First Name field as it should be.

Additional reference words: 2.00 jet25 jet 2.5 ordering

KBCategory: kbusage

KBSubcategory: QrySqlvw

## **PRA: Print Definition Report Truncated**

Article ID: Q120876

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you use the Database Documentor or the Print Definition command to print or preview a report containing design information for a database object, the report is truncated.

### CAUSE

=====

The default printer's orientation is set to landscape, and the paper size is set to legal (8.5 by 11 inches).

### RESOLUTION

=====

Change the printer settings so that the paper size is not set to legal or so that the orientation is not set to landscape.

### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

=====

#### Steps to Reproduce Problem

-----

1. In Windows Control Panel, double-click the Printers icon, and then change the settings for the default printer so that its orientation is set to landscape and the paper size is set to legal (8.5 by 11 inches).
2. Start Microsoft Access.
3. Open the sample database NWIND.MDB.
4. In the Database window, select the Employees table.
5. From the File menu, choose Print Definition. In the Print Table Definition dialog box, choose OK.
6. Note that the report that is created is truncated after two pages.

7. Return the default printer settings to their original values.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: WzOthr

**PRA: Printer Fonts Cause Text to Shift Down and Right**  
**Article ID: Q103431**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The first page of a report prints fine, but the text is shifted down and to the right on the other pages. This shift occurs when you use printer fonts.

CAUSE

=====

The problem is related to the GENDRV.DLL file. Any printer driver that uses this .DLL file creates the same distortion when using any printer fonts.

The following printers use the GENDRV.DLL file:

Generic Printer Driver-GENDRV.DLL (GENDRV.EXE) 06/24/93 version (1.68)

-----

|                            |              |
|----------------------------|--------------|
| Canon LBP 8-III            | (CANON.EXE)  |
| Digital DECLaser 1100      | (DEC1.EXE)   |
| Digital DECLaser 2100/Plus | (DEC1.EXE)   |
| Digital DECLaser 2200/Plus | (DEC1.EXE)   |
| Digital DECLaser 3200      | (DEC1.EXE)   |
| Digital LA70               | (DEC1.EXE)   |
| Digital LA75               | (DEC1.EXE)   |
| Digital LA75 Plus          | (DEC1.EXE)   |
| Digital LA324              | (DEC1.EXE)   |
| Digital LJ250/252          | (DEC1.EXE)   |
| Digital LN03/Plus          | (DEC1.EXE)   |
| IBM Laser Printer 4029     | (IB4029.EXE) |
| Olivetti DM 124 C          | (OLIVE.EXE)  |
| Royal CJP 450              | (OLIVE.EXE)  |
| Seiko CH 4104              | (SEIKO.EXE)  |
| Seiko CH 5504              | (SEIKO.EXE)  |
| Seiko CH 5514              | (SEIKO.EXE)  |

NOTE: The IBM4029 is the only printer driver certified by Microsoft Compatibility Labs (MCL). All other printers using the GENDRV.DLL file have been neither tested nor certified by MCL.

RESOLUTION

=====

Use TrueType fonts or, for the IBM4029 printer, use the the most current Microsoft Windows drivers for this printer, which are available by calling Lexmark at (606) 232-3000. These drivers are also available from the Lexmark bulletin board at (606) 232-5238.

The printer driver for Windows version 3.0 is called 29W30\_PS.EXE; the printer driver for both Windows 3.0 and 3.1 is called 29W3X\_PP.EXE. A README.DOC file with installation instructions is included with each driver.

#### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0, 1.1, and 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

#### Steps to Reproduce Problem

-----

To duplicate this problem using the sample database NWIND.MDB, do the following:

1. Open the List of Products by Category report in Design view.
2. From the File menu, choose Print Setup. Change the print driver to Canon LBP 8-III or to one of the print drivers listed above.
3. Select all controls in the Page Header section.
4. From the Edit menu, choose either Dutch Scalable or any printer font.
5. Print two pages of the report.

On the first page, the page header controls will be within the lines; on the second page, the controls will be below and to the right of the lines.

The IBM4029 printer driver is manufactured by Lexmark and the other printer drivers by other vendors independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprint

KBSubcategory: PtrOthr



**PRA: Report Based on Fast Query Seems to Run Slowly**  
**Article ID: Q123713**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

A report that is based on a query that runs quickly seems to run slowly, taking several minutes to finish.

CAUSE

=====

In certain queries containing many-to-many join fields, Microsoft Access can choose an incorrect index when it interprets the query. Using the wrong index results in poor performance.

STATUS

=====

This problem no longer occurs with the Microsoft Jet database engine version 2.5, which is available with the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kbusage

KBSubcategory: RptOthr

## **PRA: Report Output to RTF Format Contains Blank Lines**

**Article ID: Q120274**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

A report that you output to a Rich Text Format (RTF) file contains two extra blank lines for each record. These blank lines do not appear when you preview the report.

### CAUSE

=====

This problem can occur if a report section's CanGrow property is set to Yes when you output the report to an RTF file.

### RESOLUTION

=====

To prevent the blank lines from appearing in the RTF file, set the section's CanGrow property to No. Note that if you are using the CanGrow property to make sure that all the data in a field is printed, setting the CanGrow property to No can truncate the data.

### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

=====

When you output a report to text format, the CanGrow property is ignored and the extra blank lines are not created. When you output a report to Microsoft Excel format, the CanGrow property is used but the extra blank lines are not created.

### Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB and then create a blank, new report based on the Categories table.
2. Create the following text boxes in the report's detail section:

Text box:

Name: Category ID  
ControlSource: Category ID  
Top: 0

Left: 0  
Width: 1  
Height: 0.1667  
Text box:  
Name: Category Name  
ControlSource: Category Name  
Top: 0  
Left: 1.1  
Width: 1  
Height: 0.1667  
Text box:  
Name: Description  
ControlSource: Description  
CanGrow: Yes  
Top: 0  
Left: 2.2  
Width: 1  
Height: 0.1667

3. Set the detail section's Height property to 0.1667 inches.
4. Preview the report. Note that there are no blank lines between records.
5. From the File menu, choose Output To.
6. In the Select Format box, select Rich Text Format, and then choose OK.
7. In the File Name box, type a name for the RTF file and then choose OK.
8. Open the RTF file in any word processor that can open RTF files (such as Microsoft Word for Windows). Note the extra blank lines between the records.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Appendix D, "Using Other Windows-Based Applications with Microsoft Access," pages 737-738

Additional reference words: 2.00 can grow  
KBCategory: kbusage  
KBSubcategory: OtpRtf

**PRA: Report Section with Subreport Not Included In Output File**  
**Article ID: Q113562**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you use the Output To command to output a report to a file, any section of a report including a subreport is not included in the output file.

CAUSE

=====

It is by design that when you use the Output To command to output a report to a file, subreports are not included in the output file. However, the presence of a subreport in any section of a report prevents the entire section from being included in the output file.

RESOLUTION

=====

Remove the subreport from the report section and output the report again.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem, and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Preview the report Sales By Year.
3. In the Sales By Year Dialog form, enter "6/1/91" (without quotation marks) for the beginning date, and enter "6/30/91" (without quotation marks) for the ending date. Choose OK.
4. Note that one page is displayed, with a page header section, group header section, detail section, and page footer section.
5. From the File menu, choose Output To.
6. In the Select Format box, select Rich Text Format (\*.rtf), and then choose OK.

7. Choose OK to accept the default filename (sales\_by.rtf) for the output file.
8. When you are alerted that the subreport will not be included, choose OK.
9. Open the SALES\_BY.RTF file in any editor (such as Microsoft Word for Windows, or Microsoft Write) that can open a Rich Text format file. Note that the entire group header section is missing from the file, because the group header section of the Sales By Year report contains a subreport.

NOTE: The height of the detail section in the Sales By Year report is set to 0.1667 inches to add space between detail records. Setting the height of the detail section to 0.1563 inches avoids double spacing in the output file.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: RptSub

**PRA: Reserved Error -1001 with Linked Crosstab Subreport**  
**Article ID: Q104758**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SYMPTOMS

=====

You receive the following error message when you are printing or previewing a report:

The crosstab query underlying a subform or subreport must have fixed column headings (Microsoft Access version 2.0)

-or-

Reserved error (-1001); there is no message for this error (Microsoft Access version 1.1)

CAUSE

=====

A main report with a bound subreport that is based on a crosstab query cannot be previewed or printed unless the crosstab query uses fixed column headings.

RESOLUTION

=====

Add fixed column headings to the crosstab query's properties:

In Microsoft Access version 2.0, with the crosstab query in Design view, choose Query Properties from the View menu. In the ColumnHeadings property, type the column headings for your query.

In Microsoft Access version 1.1, with the crosstab query in Design view, choose Query Properties from the View menu. In the FixedColumnHeadings property, type the column headings for your query.

Another workaround is to delete the LinkChildFields and LinkMasterFields properties of the subreport. You will then be able to run your report, but the report will show every record in the subreport.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Create a crosstab query based on the Employees table. Save the query as Employee Crosstab.

Query: Employee Crosstab

-----  
Type: Crosstab Query

Field: Last Name

Total: Group By

Crosstab: Row Heading

Field: Birth Date

Total: Group By

Crosstab: Column Heading

Field: Reports To

Total: Sum

Crosstab: Value

3. Using the Single-Column ReportWizard, create a new report based on the Employee Crosstab query. Include all the query fields in your report, and sort by the Last Name field. Save the report as Employee Crosstab SubReport.
4. Using the Single-Column ReportWizard, create another new report based on the Employee Crosstab query. Include only the Last Name field in this report, and sort by the Last Name field. Save this report as Employee Main Report.
5. Embed the Employee Crosstab SubReport in the detail section of the Employee Main Report. You can do this by dragging the subreport from the Database window to the detail section of the main report (which must be in Design view).
6. In Design view of the main report, view the Properties sheet for the Employee Crosstab SubReport. Verify that the LinkChildFields and LinkMasterFields properties are set to Last Name.
7. Try to preview or print the Employee main report. You will receive the error message stated above.

Additional reference words: 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptSub

## **PRA: Resources Lost Printing Reports with OLE Objects**

**Article ID: Q123591**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you print a form or report containing several OLE objects (such as charts created in Microsoft Graph), Microsoft Access appears to reduce the available Microsoft Windows system resources until the available system resources reach zero.

When the Windows system resources reach zero, different results can occur, depending on the operating system and version of Windows running on your computer. Windows may seem to run slowly, parts of the Windows environment may seem to disappear, or you may receive a general protection (GP) fault error message.

### STATUS

=====

This problem no longer occurs with OLE 2.02, which is available with the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

### MORE INFORMATION

=====

The new OLE 2.02 dynamic-link library (DLL) files have been created to coincide with the release of Microsoft Windows NT Workstation 3.5. These new DLLs provide hooks for 16/32 interoperability for 32-bit applications.

OLE 2.01 contains limitations that can cause problems in large networks and with screen palettes. OLE 2.02 corrects the following problems:

- A limit of 20 users who can access a read-only file.
- A 4 GB high lock on a network read (such as on DEC Pathworks version 4.x).
- A GP fault during multiple palette use.

Additional reference words: 2.00 jet25 jet 2.5 gpf

KBCategory: kbole

KBSubcategory: IntpOle



**PRA: RunningSum Text Box in Report Not Accumulating Values**  
**Article ID: Q115877**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

When a text box in a report that uses the RunningSum property is based on another control in the report, it does not accumulate the running sum. Instead, it reflects only the value of the control it references.

RESOLUTION

=====

Instead of referring to another control in the text box's ControlSource property, refer to a field from the underlying table or query, or to an explicit calculation.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

This problem is usually encountered in databases that have been converted from version 1.x to 2.0.

Steps to Reproduce Problem

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new report based on the Products table.
3. Place two new text boxes side by side in the report's detail section. Note that the Name property of the left text box is Field0, and the right text box is Field2.
4. Set the following properties for the text boxes:

    Name: Field0  
    ControlSource: =[Unit Price]

    Name: Field2  
    ControlSource: =[Field0]  
    RunningSum: Over All

5. Preview the report and scroll to the last page.

Note that the Field2 text box reflects only what is contained in the Field0 text box, even though its RunningSum property is set to Over All. It does not accumulate the values contained in the two text boxes as it should.

To correct this problem for fields in the underlying record set, set the Field2 text box's ControlSource property to:

```
=[Unit Price]
```

Note that this does not correct the problem for expression fields such as DLookup and Sum. You can also try placing the expression in the underlying query or incrementing a global variable using Access Basic.

#### REFERENCES

=====

For more information about the RunningSum property, search for "RunningSum" then "RunningSum Property" using the Microsoft Access Help menu.

Additional reference words: 2.00 Reports

KBCategory: kbusage

KBSubcategory: RptTotal

## **PRA: SETUP /Q Always Installs Complete Version**

Article ID: Q119026

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

If you follow the directions in the "Network Installations" section of the ACREADME.HLP file to modify the SETUP.STF file for a quiet Setup (SETUP /Q), Setup will set up a complete copy of Microsoft Access, whether you choose the Typical, Custom, Laptop, or Workstation Setup option.

### CAUSE

=====

There is an additional line in the SETUP.STF file that needs to be modified. The ACREADME.HLP file does not mention this line.

### RESOLUTION

=====

Follow these steps to modify the line that contains the Batch Mode Root Object ID variable:

1. Make a backup copy of the SETUP.STF file.
2. Open the SETUP.STF file in a text editor or a spreadsheet. Opening the file in a spreadsheet is the preferred method.
3. Search for the string "Batch Mode Root Object ID" (without quotation marks). If you open the file in Microsoft Excel, the string will be in cell A19.
4. In the cell to the right of the string (cell B19), change the value

9:01

to:

7:01

5. Save and then close the file.

### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

=====

## Steps to Reproduce Problem

---

1. Install Microsoft Access on your server using the SETUP /A command. When you are prompted, choose to keep all the Microsoft Access files on the server.
2. Install Microsoft Access on a workstation using the SETUP command (do not use the /Q switch). Setup will create an icon for Microsoft Access that points to the server, as it should.
3. Modify the SETUP.STF file as outlined in the ACREADME.HLP file to automatically set up Microsoft Access using the workstation settings.
4. Run the SETUP /Q command to install Microsoft Access on another workstation. Note that Setup copies the MSACCESS.EXE file to the workstation's local hard disk.

Additional reference words: 2.00

KBCategory: kbsetup

KBSubcategory: StpNetinst

**PRA: Subreport Placed Low on Main Report Causes Blank Pages**  
**Article ID: Q117917**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you print or preview a report containing a subreport placed low in the main report's detail section, blank pages are generated after the subreport.

CAUSE

=====

The extra pages are the result of the main report incorrectly determining where the subreport begins and ends.

RESOLUTION

=====

Move the subreport higher in the main report's detail section, or create a static grouping section and place the subreport in that section. To create a static group footer, follow these steps:

1. Click the Sorting And Grouping button on the toolbar and type the following in the Field/Expression:

="A"

Then set the GroupFooter property to Yes.

2. Move the subreport from the detail section into the new group footer section.
3. Preview the report.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0, 1.1, and 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Create a new, unbound report.

3. In the report's detail section, create three label controls with the following properties:

Label1

Name: Label1

Caption: This is the top of the detail section.

Top: 0

Label2

Name: Label2

Caption: This is the middle of the detail section.

Top: 10

Label3

Name: Label3

Caption: This will print just above the subreport control.

Top: 19.5

4. Create a subreport control in the detail section. Drag the Alphabetical List Of Products report to the subreport control.

5. Set the subreport control's properties as follows:

Subreport Control

Name: Alphabetical List of Products

SourceObject: Report.Alphabetical List of Products

Left: 0

Top: 21

CanGrow: Yes

6. Preview the report. Note that pages 6-11 are blank.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptSub

**PRA: Text Moves When Locked and Enabled Properties Set to No**  
**Article ID: Q119286**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When a text box's Locked and Enabled properties are both set to No, the text in the text box moves to a different place on the form.

CAUSE

=====

This problem can occur if you select Ocean as your Microsoft Windows color scheme, and the text in the text box is centered or right-aligned. Note that the disabled text color in the Ocean color scheme is black; the disabled text color is gray in all other color schemes.

RESOLUTION

=====

There are three resolutions to this problem:

- Left-align the text in the text box by setting the text box's TextAlign property to Left.
- Adjust the size of the text box to match the width of the text being displayed in the box. The larger the text box is relative to the text being displayed, the more marked the problem will be.
- Change the color for disabled text to some color other than black using the Color utility in Windows Control Panel.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0, 1.1, and 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Start Windows Control Panel.
2. Double-click the Color icon.
3. Choose the Color Palette button.

4. In the Color Schemes box, select Ocean, then choose OK.
5. Start Microsoft Access.
6. Open the sample database NWIND.MDB.
7. Create a new, blank form based on the Customers table.
8. Place a text box on the form. Set the text box's ControlSource property to:  
  
Customer ID
9. View the form in Form view. Note that the text in the text boxes is displayed correctly.
10. View the form in Design view. Set the text box's Enabled and Locked properties to No, and its TextAlign property to Right.
11. View the form in Form view. Note that the text in the text boxes moves to the right.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 16, "Customizing Forms," pages 415-418

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsProb



## **PRA: Toolbar Not Updated When Switching Window Types**

Article ID: Q121897

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you switch from one window to another (for example, from the Database window to a Form Design window), occasionally the toolbar does not get updated correctly, or multiple buttons on the toolbar appear sunken.

### CAUSE

=====

This problem is caused by the way Microsoft Windows version 3.1 and Microsoft Windows for Workgroups version 3.11 handle right mouse button click events. This behavior occurs only when you release the left and right mouse buttons at exactly the same time.

### RESOLUTION

=====

To work around this problem, click the toolbar once with the right mouse button, and then once with the left mouse button. The toolbar will be updated correctly.

### STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Windows version 3.1 and Microsoft Windows for Workgroups version 3.11. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

### MORE INFORMATION

=====

#### Steps to Reproduce Problem

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Select the Employees form in the Database window.
3. Position the mouse pointer on the Design button, and then click both the left and right mouse buttons and hold them down.
4. Release both buttons simultaneously. The Employees form will open in Design view, but the toolbar from the Database window will remain.

Additional reference words: 2.00 ui

KBCategory: kbui  
KBSubcategory: UiToolbr

**PRA: TransferDatabase Fails in Code, Not in Immediate Window**  
**Article ID: Q117612**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you use the TransferDatabase action within a transaction in an Access Basic function, you receive the error message:

    Couldn't update, locked by another user on this system.

However, if you set a breakpoint in the function and single-step through it, you do not receive the error message.

CAUSE

=====

This error occurs only when you run a TransferDatabase action nested in a transaction on a table that is already attached to Microsoft Access.

The error occurs because a transaction updates the MySysObjects table, placing a write lock on the table. The write lock is not released while the transaction is still active. The DoCmd TransferDatabase statement causes Microsoft Access to start a new session and try to create a new table. However, the Microsoft Jet database engine needs to update the MySysObjects table because it is adding a record, but the MySysObjects table is still locked.

RESOLUTION

=====

Do not use nested TransferDatabase actions in transactions against attached tables. Instead, use the following techniques:

- Move the TransferDatabase action above or below the transaction.
- Move the data from the attached table into a local table and perform the TransferDatabase action on it instead.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Start Microsoft Access and create a new database.
2. Import the Employees table from the sample database NWIND.MDB into the new database.
3. Attach the Order Details table from NWIND.MDB.
4. Create a new module and enter the following function:

```
Sub TestIt ()
 Dim ws as Workspace
 Dim db As Database, rs As Recordset
 BeginTrans
 set ws=dbengine.workspaces(0)
 Set db = ws.databases(0)
 Set rs = db.OpenRecordset("Order Details", DB_OPEN_DYNASET)
 rs.FindFirst "[Order Id]=10010"
 rs.Edit
 rs![Order Id] = 10001
 rs.Update
 DoCmd TransferDatabase A_EXPORT, "Microsoft Access",_
 "C:\ACCESS\SAMPAPPS\NWIND.MDB", A_TABLE, "Employees",_
 "Employees2", False
 CommitTrans
End Sub
```

5. From the View menu, choose Immediate Window.
6. In the Immediate window, type the following line and then press ENTER:

```
TestIt
```

#### REFERENCES

=====

For more information about the TransferDatabase action, search for "TransferDatabase" then "TransferDatabase Action" using the Microsoft Access Help menu.

Additional reference words: 2.00 programming attaching  
KBCategory: kbusage kberrmsg  
KSubcategory: McrProb

**PRA: Update or Delete Query Fails Without Generating Error**  
**Article ID: Q117163**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you run an update or delete query in Access Basic using the Execute method, some records are not modified or deleted, and no error message is generated.

CAUSE

=====

Those records that were not modified or deleted were locked when the action query was run. A record is locked whenever it is being edited, whether that editing is done through the user interface, from Access Basic code, or by an action query.

RESOLUTION

=====

In Microsoft Access version 2.0, use the DB\_FAILONERROR option when you use the Execute method to run an action query from Access Basic code. This switch issues a rollback and generates an error when locking conflicts occur, resulting in the canceling of all updates.

In Microsoft Access version 1.x, create a dynaset in Access Basic that includes all the records that should be modified. Within transactions (BeginTrans and CommitTrans), modify each record in the dynaset until all the records are updated. This technique will generate an error that you can trap if a locking conflict occurs. You can then issue a rollback on the transaction and try to update the record again until the update succeeds.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 1.x. The problem was corrected in Microsoft Access version 2.0 by including the DB\_FAILONERROR option with the Execute method.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

Steps to Reproduce Problem

-----

1. Start Microsoft Access and open any database.
2. Create a new table called Table1. Add a text field called Info to the table. View the table in Datasheet view and add the following records:

```

Info

ddd
aaa
ccc
ddd
ddd
eee

```

3. Create a new query based on Table1. Drag the Info field to the query grid. Type "ddd" (without quotation marks) in the Criteria field. Save the query as Query1. Run the query and note that three records are displayed. Close the query.
4. Create an update query based on Table1. Drag the Info field to the query grid. Type "zzz" (without quotation marks) in the Update To row, and type "ddd" (without quotation marks) in the Criteria row. Save the query as UDQuery.
5. Open a new module and enter the following sample code:

```
Option Explicit
```

```

Function RunUDQuery ()
 Dim db As Database, qdef As QueryDef
 'Use only one of the following lines--remove the other line!
 Set db = CurrentDB() 'Use this line in version 1.x
 Set db = DBEngine(0)(0) 'Use this line in version 2.0
 Set qdef = db.OpenQueryDef("UDQuery")
 qdef.Execute
End Function

```

6. Create a new form based on Table1. Add a text box based on the Info field to the form. Add a command button to the form, and set the button's OnClick property to "=RunUDQuery()" (without quotation marks). (Note that in Microsoft Access version 1.x, the OnClick property is called the OnPush property.) Save the form as Form1.
7. View Form1 in Form view and type "xxx" (without quotation marks) in the Info text box. Note that the record selector changes from an arrow to a pencil, indicating that the record is being edited, and is locked. Click the command button on the form to run the update query. Note that no error message is generated.
8. Press ESC to undo your typing in the form. Open Query1 and note that one record still matches the "ddd" criteria. This record was not updated by the update query because the record was being edited when the update query was run.

To correct this problem in Microsoft Access version 2.0, use the following sample code in place of the code in step 5:

Option Explicit

```
Function RunUDQuery ()
 Dim db As Database, qdef As QueryDef
 Set db = DBEngine(0)(0)
 Set qdef = db.OpenQueryDef("UDQuery")
 On Error GoTo ErrorHandler
 qdef.Execute DB_FAILONERROR
 Exit Function
```

```
ErrorHandler:
 MsgBox "Update Failed " & Err & " " & Error
 Exit Function
End Function
```

#### REFERENCES

=====

Microsoft Access "Introduction to Programming," version 1.1, Chapter 8,  
page 108

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbusage  
KBSubcategory: QryMktbl

**PRA: Words Split When CanGrow=Yes and KeepTogether=No**

Article ID: Q113301

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The Report Wizard splits a text box in the middle of a word instead of between lines.

CAUSE

=====

This problem can occur if all of the following conditions are met:

- The first line of a multiple-line text box is the last line that will fit on the page.
- The text box's CanGrow property is set to Yes.
- The section's CanGrow property is set to Yes.
- The section's KeepTogether property is set to No.

RESOLUTION

=====

Set the text box's height to two lines and its CanShrink property to Yes. The Report Wizard will not split a control into a size smaller than its size in Design view.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access versions 1.0, 1.1, and 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

Additional reference words: 1.00 1.10 2.00 truncation

KBCategory: kbusage

KBSubcategory: RptProp



**PRA: WRKGADM.EXE File Installed Without Being Specified**  
**Article ID: Q122241**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

When you install a custom application that you created with the Microsoft Access Developer's Toolkit (ADT) version 2.0, a file named WRKGADM.EXE is installed in the application directory even though you did not specify this file for installation when you created the application's custom Setup.

CAUSE

=====

The WRKGADM.EXE file is the Workgroup Administrator utility, and is required to create a system database if you do not include one with your run-time application. Setup copies the WRKGADM.EXE file to the application directory and then runs it to create the system database. After running the file, Setup is unable to delete it.

RESOLUTION

=====

If you include a system database file called SYSTEM.MDA with your custom application, Setup will not copy the WRKGADM.EXE file to the application directory. If you want to include a system database with some name other than SYSTEM.MDA, include the custom system database and any file called SYSTEM.MDA (for instance, a text file) with your application.

STATUS

=====

Microsoft has confirmed this to be a problem in the Microsoft Access Developer's Toolkit version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

Additional reference words: 2.00

KBCategory: kbsetup

KBSubcategory: StpOthr

**PRB: GoToRecord Macro Does Not Work on Subform**

**Article ID: Q92685**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

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

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.

The GoToRecord action would look like this:

```
GoToRecord
 Object Type:
 Object Name:
 Record : Next
```

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB and then create the following new macro:

| Macro Name | Condition | Action     |
|------------|-----------|------------|
| Macro1     |           | GoToRecord |

Macro1 Actions

-----

```
GoToRecord
 Object Type: Form
 Object Name: Categories Subform
 Record: Next
```

2. Open the Categories form and from the File menu, choose Run Macro, then

select Macro1. You will get the following error message:

Object 'Categories Subform' isn't open

3. Now change Macro1 to the following:

| Macro Name                       | Condition | Action                    |
|----------------------------------|-----------|---------------------------|
| Macro1                           |           | GoToControl<br>GoToRecord |
| Macro1 Actions                   |           |                           |
| -----                            |           |                           |
| GoToControl                      |           |                           |
| Control Name: Categories Subform |           |                           |
| GoToRecord                       |           |                           |
| Record: Next                     |           |                           |

Run the macro in the same way as in step 2. The macro will go to the second record in the Categories Subform.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsEvnt

## **PRB: "#Error" Message Referencing Subreport Values (2.0)**

**Article ID: Q114513**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SYMPTOMS

=====

Referencing a subreport control from a main report when the subreport does not return any records results in a "#Error" error message.

### CAUSE

=====

This error occurs because the subreport does not return any values.

### RESOLUTION

=====

Make sure that a subreport referenced in a main report returns values. Alternatively, you can use an Access Basic function to retrieve values from the subreport and trap for the error should it occur. See the "Steps to Solve Problem" section later in this article for an example of this technique.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

#### Steps to Reproduce Problem

-----

1. Open the sample database NWIND.MDB.
2. Create a new, blank query based on the Customers table.
3. Drag the Customer ID and Company Name fields from the field list to the query grid.
4. In the criteria cell for the Company Name column, enter the following expression:  
  
    Like "P\*"
5. Save the query as Sorter.
6. Create a new report based on the Orders table. This report will be the subreport.

7. Drag the Freight field from the field list to the report's Detail section.
8. In the report's footer section, place a text box with the following properties:
 

```
ControlSource: =Sum([Freight])
Name: Sum Freight
```
9. Save the report as SubRpt.
10. Create a new report based on the Sorter query. This report will be the main report.
11. From the Format menu, turn off the Page Header/Footer command. From the View menu, choose Sorting And Grouping. In the Field/Expression box, select Customer ID, and then set the Group Header property to Yes.
12. Drag the Customer ID field from the field list to the main report's Customer ID header section.
13. Drag the SubRpt report from the Database window to the detail section of the main report. Below the new subreport, place a label control with the caption "Total Order Amount Per Customer:" (without quotation marks) and a text box containing the following formula:
 

```
=([SubRpt].Report![Sum Freight])
```
14. Set the subreport's LinkMasterFields and LinkChildFields properties to [Customer ID].
15. Save the main report as MainRpt.
16. Preview the main report. Note that in the record for company Paris Specialties, the error message "#Error" is displayed for the subreport reference ([SubRpt].Report![Sum Freight]).

#### Steps to Solve Problem

---

1. Create the a new Access Basic function called ErrAvoid():
 

```
Function ErrAvoid (n As Variant)
 On Error GoTo Trap
 ErrAvoid = n
 Exit Function
Trap:
 ErrAvoid = 0
 Resume Next
End Function
```
2. Replace "`=([SubRpt].Report![Sum Freight])`" in step 13 of the previous procedure with:
 

```
=ErrAvoid([SubRpt].Report![Sum Freight])
```
3. Preview the main report.

Note that in the record for company Paris Specialties, \$0.00 is now displayed instead of the "#Error" message. The ErrAvoid() function traps for errors caused by the subreport containing no values. When an error is encountered, a value of zero is assigned.

Additional reference words: 2.00 pounderror

KBCategory: kbusage kberrmsg

KBSubcategory: RptSub

**PRB: "#ERROR" Message When Referencing Subreport Controls**  
**Article ID: Q102521**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Referencing a subreport control from a main report when the subreport does not return any matching records results in a "#ERROR" message for the referenced subreport control.

RESOLUTION

=====

Make sure that any subreport fields referenced in main reports have values associated with them by adding a function to test for this condition. For more information, see the "Steps to Solve Problem" section below.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Create a new query based on the Customers table. Drag the Company Name field from the field list to the query grid. Type the following in the Criteria field for the column:

Like "P\*"

Drag the Customer ID field from the field list to the next column on the query grid. Save the query as Sorter. Note that this query will limit the recordset to only customer records beginning with the letter "P", and includes a customer with no orders.

3. Create a new subreport based on the Orders table; call it SubRpt.
4. Drag the Freight field to the detail section of the subreport. In the report footer section, place a text box with the following formula as the ControlSource property:

=Sum([Freight])

Set the ControlName property of the text box to ToTFreight. The subreport will print the detailed freight items with a grand total for

all freight charges.

5. Create a new report called Main Report based on the Sorter query.
6. From the View menu, choose Sorting And Grouping. In the dialog box, select Company Name from the Field/Expression combo box and set the Group Header property to Yes.
7. Drag the Company Name field to the new Company Name header section.
8. Drag the SubRpt subreport from the Database window to the detail section of Main Report. Below the subreport, place a label control with the caption "Total Order Amount Per Customer:" and a text box control with the ControlSource property set to the following formula:

```
=([SubRpt].Report![ToTFreight])
```

Set both the LinkMasterFields and the LinkChildFields properties for the subreport control to [Customer ID]. Setting these properties filters the subreport to display the Freight charges per customer.

9. Preview the report.

Note that Parisian Specialties (in version 1.x), or Paris Specialites (in version 2.0), which has no orders, displays "#ERROR" for the referenced subreport control.

#### Steps to Solve Problem

1. Create the following new Access Basic function called ErrAvoid():

```
Function ErrAvoid (n As Variant)
 On Error GoTo Trap
 ErrAvoid = n
 Exit Function
Trap:
 ErrAvoid = 0
 Exit Function
End Function
```

2. Replace "=[SubRpt].Report![ToTFreight]" in step 8 of the previous procedure with the following:

```
=ErrAvoid([SubRpt].Report![ToTFreight])
```

3. Preview the report.

Parisian Specialties, or Paris Specialites, now shows \$0.00 dollars in sales, instead of "#ERROR." The ErrAvoid() function traps for errors that are caused by the subreport control containing no values. When an error is encountered, "Object Has No Value," then a value of zero, is assigned to the main report control.

Additional reference words: 1.00 1.10 2.00 pounderror pounderr link master  
KBCategory: kbusage



KBSubcategory: RptSub

**PRB: "Can't Bind Name '[XXX]'" Error with Crosstab Query**  
**Article ID: Q91710**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

If you create a crosstab query using an implicit parameter [XXX] in a WHERE clause (criteria), Microsoft Access returns the following error when you run the query:

Can't bind name '[XXX]'

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

CAUSE

=====

A crosstab query dynamically generates column names. Therefore, Microsoft Access cannot tell whether [XXX] is referring to a parameter or a column name until after the query is bound.

When you build a graph, Microsoft Access uses a crosstab to build the data to graph. If the query is a parameter query, you have to define the parameters explicitly as you would for a normal crosstab query.

RESOLUTION

=====

To avoid this error, define [XXX] as an explicit parameter by adding it to the Query Parameters dialog box.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create a new crosstab query on the Order Details table.
3. Drag the fields Unit Price, Discount, and Order ID to the QBE grid, and set the values in the grid as follows:

Crosstab Query: Query1

-----

Field: Unit Price  
Total: Group By  
Crosstab: Row Heading  
Criteria: > [Enter an Amount:]  
Field: Discount  
Total: Group By  
Crosstab: Column Heading  
Field: Order ID  
Total: Count  
Crosstab: Value

4. Run the query. The following error message is displayed:

Can't bind name '[Enter an Amount:]'.

5. Choose OK to return to query design, and then select Parameters from the Query menu.

6. Enter the following under Parameters:

[Enter an Amount:]

7. Select Currency as the Data Type, and choose OK.

8. Run the query and type any amount (for example, type "2").  
The query executes.

#### REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00 can't

KBCategory: kbusage

KBSubcategory: QryCross

**PRB: "Can't Have More Than 10 Fields In an Index" Error Msg**  
**Article ID: Q115902**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

You receive the following error message when using a domain function or running a query:

Can't have more than 10 fields in an index

or you receive the error message:

Field won't fit

CAUSE

=====

This error can occur if the following conditions are met:

- The underlying table contains a multiple-field index.
- More than one field in the index is filtered by the query's criteria.
- One of the criteria contains the = operator (for example, "='Joe']").
- One of the criteria includes the Like operator with a wildcard (for example, "Like 'Smith\*']").

The "Field won't fit" error message occurs because of small numeric indexes.

RESOLUTION

=====

There are three workarounds for this behavior:

- Remove the multiple-field index on the underlying table and replace it with multiple single-field indexes. Note that you cannot use this method if the multiple-field index is the table's primary key. For example, change the multiple-field index

| Index Name | Field Name |
|------------|------------|
| NameIndex  | Last Name  |
|            | First Name |

to:

| Index Name | Field Name |
|------------|------------|
|------------|------------|

-----

LNameIndex    Last Name  
FNameIndex    First Name

- Change the order of the fields in the two-field index on the underlying table. For example, change the two-field index

| Index Name  | Field Name          |
|-------------|---------------------|
| Primary Key | Invoice #<br>Part # |

to:

| Index Name  | Field Name          |
|-------------|---------------------|
| Primary Key | Part #<br>Invoice # |

- Replace the = operator with a Like operator and a wildcard. For example, change the query

```
? DLookup("[Birth Date]", "Employees", "[First Name] like 'L*' and _
[Last Name]='Callahan'")
```

to:

```
? DLookup("[Birth Date]", "Employees", "[First Name] like 'L*' and _
[Last Name] like 'Callahan*')"
```

NOTE: In the above sample queries, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscores from the ends of the lines when re-creating these queries.

STATUS  
=====

This problem no longer occurs with the Microsoft Jet database engine version 2.5, which is available with the Microsoft Access version 2.0 Service Pack. For information about how to obtain the Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927  
TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

MORE INFORMATION  
=====

Steps to Reproduce Behavior  
-----

1. Open the sample database NWIND.MDB.
2. Create a copy of the Employees table. Save the copy table as Test.
3. Open the Test table in Design view.
4. From the View menu, choose Indexes.

5. Delete the existing index on the Last Name field.
6. Create a new index called NameKey. In the Field Name column, select Last Name.
7. Select the next cell in the Index Name column. Leave the cell blank. In the Field Name column, select First Name. Close the Index dialog box.
8. Open any module and then choose Immediate Window from the View menu.
9. Type the following line in the Immediate window and then press ENTER.

NOTE: In the following line, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this line.

```
? DLookup("[Birth Date]", "Test", "[First Name] like 'L*' and _
[Last Name]='Callahan'")
```

#### REFERENCES

=====

For more information about the error message "Can't have more than 10 fields in an index" search for "error messages: reference" using the Microsoft Access Help menu.

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kbusage kberrmsg

KBSubcategory: QryProb

**PRB: "Can't Restore Field's Previous Value..." Error Message**  
**Article ID: Q120971**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

You receive the following error message:

Can't restore field's previous value; choose Undo Current Record or Undo Current Field from the Edit menu.

CAUSE

=====

Access Basic code canceled the BeforeUpdate event for a control in a record and Microsoft Access attempted to restore the field's original value. However, this value either violates a table-level validation rule or is a null value in a field whose Required property is set to Yes.

RESOLUTION

=====

There are two ways to work around this behavior:

- Remove the table-level validation rule and set the field's Required property to No. Validate the data using validation rules at the form level, rather than at the table level.
- Create a DefaultValue property setting for the field, so that the field always contains a valid value in a new record.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Open the Orders table in Design view and set the following property for the Ship Via field:  
  
ValidationRule: >0
3. Save the table. When you are prompted "Data integrity rules have been changed..." choose Yes. Close the table.
4. Open the Orders form in Design view.
5. Select the Ship Via option group and then set its BeforeUpdate property to the following event procedure:

```
Sub Ship_Via_BeforeUpdate (Cancel As Integer)
 Cancel = True
End Sub
```

6. View the Orders form in Form view.
7. From the Records menu, choose Data Entry.
8. Select any of the check boxes in the Ship Via option group. You will receive the error message stated above.

This error occurs because the option group's BeforeUpdate event was canceled. In canceling the event, Microsoft Access attempted to restore the original value of the field. Since there is no default value for the field, Microsoft Access tried to insert a null value in the field, but the null value violated the table-level validation rule (>0).

#### REFERENCES

=====

For more information about table-level validation, search for "validation: rules and text" then "Validating Data in a Table" using the Microsoft Access Help menu.

For more information about the Required property, search for "requiring data entry" then "Required Property" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: FmsEvt



**PRB: "Cannot Change...SETUP.INI" Installing on Novell Server**  
**Article ID: Q120145**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you install Microsoft Access version 2.0 on a Novell NetWare server using the SETUP /A command, you receive an error message similar to:

Cannot change the file named SETUP.INI

-or-

Cannot change the properties of the file named SETUP.INI

This message varies somewhat depending on the version of NetWare you are using, but the common element is a failure in writing to either the SETUP.INI file or the \_MSSETUP.EXE file.

Note that this error can also occur when installing the Microsoft Access version 2.0 Service Pack. The solution is the same.

CAUSE

=====

The workstation's read-only compatibility setting is turned on in the SHELL.CFG or NET.CFG file. The file contains a line that reads:

READ ONLY COMPATIBILITY=ON

RESOLUTION

=====

1. Turn the workstations's read-only compatibility setting off. Change the line to read:

READ ONLY COMPATIBILITY=OFF

2. Restart the workstation and then run the Setup command (SETUP /A) again.

MORE INFORMATION

=====

NetWare is manufactured by Novell, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

REFERENCES

=====

Please see the following topics in the ACREADME.HLP file:

- Using Microsoft Access Version 2.0 with Other Software
- Setting up Microsoft Access on a Novell Network
- Using Microsoft Access on Novell NetWare Servers and Banyan VINES Servers

Additional reference words: 2.00 jet25 jet 2.5 office

KBCategory: kbsetup kberrmsg

KBSubcategory: StpNetinst

## **PRB: "Couldn't Find File WIZARD.MDA" Error Message**

**Article ID: Q116150**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you start Microsoft Access, you receive the error messages:

    Couldn't find file 'WIZARD.MDA'

    -and-

    Some libraries or modules could not load. Click OK or press ENTER to continue.

### CAUSE

=====

Your MSACC20.INI file has been deleted from the Windows directory, or it has been renamed or moved. Also, the Microsoft Access version 1.x MSACCESS.INI file is still in the Windows directory. If Microsoft Access version 2.0 cannot find the MSACC20.INI file, it will use the version 1.x MSACCESS.INI file.

### RESOLUTION

=====

1. Delete or rename the MSACCESS.INI file.
2. Locate your MSACC20.INI file and place it in your Windows directory. If it has been deleted, reinstall Microsoft Access to rebuild the MSACC20.INI file.

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

NOTE: These steps assume that the Microsoft Access version 1.x MSACCESS.INI file exists in your Windows directory.

1. In Windows File Manager, select the MSACC20.INI file in your Windows directory and rename it to some other name.
2. Start Microsoft Access version 2.0. The error messages stated above will be generated.
3. Quit Microsoft Access.
4. In File Manager, rename the file that you changed in step 1 back to

MSACC20.INI.

REFERENCES

=====

For more information about the MSACC20.INI file, search for "INI files" then "Customizing MSACC20.INI Settings" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbsetup kberrmsg

KBSubcategory: StpErr

**PRB: "Couldn't Find Installable ISAM" Error Message**  
**Article ID: Q90111**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Importing, exporting, or attaching a file using an indexed sequential access method (ISAM) driver returns an error message if Microsoft Access cannot find the ISAM driver.

CAUSE

=====

ISAM drivers are used by Microsoft Access to update file formats other than its own. If the path to the ISAM driver in your MSACCESS.INI (Microsoft Access version 1.x) or MSACC20.INI (version 2.0) file is invalid, or the ISAM driver does not exist, you will receive the following error message:

    Couldn't find installable ISAM.

This error can also be caused by entering an invalid database name for a SQL database when you are using the TransferDatabase action in a macro. When you use an invalid connect string, Microsoft Access looks for an installable ISAM for the SQL database. If an installable ISAM cannot be located, the error message stated above will occur.

RESOLUTION

=====

Edit the .INI file to specify the correct path in the [Installable ISAMs] section, or reinstall Microsoft Access using the Custom option.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Start Microsoft Access and open any database.
2. Create a new macro.
3. Select TransferDatabase as the action. Set the following arguments:

    Transfer Type: Import, Export, or Attach  
    Database Type: <SQL Database>

Database Name: pubs  
Object Type: Table  
Source: db.authors  
Destination: Test  
Structure Only: No

4. Save and run the macro.

You get the error message "Couldn't find installable ISAM."

The invalid database name argument for a SQL database causes the error.

To enter a valid database name, you must include the entire ODBC connection string for the attached SQL table.

#### REFERENCES

=====

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

Microsoft Access "User's Guide," version 2.0, Chapter 9, "Importing, Exporting, and Attaching," pages 159-174

For more information about the TransferDatabase action, search for "TransferDatabase" then "TransferDatabase Action" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: ObcOthr

**PRB: "Couldn't Find Object <Tablename>" Error with Btrieve**  
**Article ID: Q93685**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you import or attach Btrieve files, you receive the following error message:

    Couldn't find object <tablename>

CAUSE

=====

1. Microsoft Access requires all three Xtrieve data dictionary files (FILE.DDF, INDEX.DDF, and FIELDS.DDF) in order to import or attach Btrieve files. The path statements in FILE.DDF may not be correct for the current environment.

-or-

2. The descriptive Btrieve table name begins with a SQL keyword that causes Microsoft Access to return the above error message.

RESOLUTION

=====

1. Inspect the path statements in FILE.DDF. Either move the Btrieve data files to the locations specified in FILE.DDF, or use Btrieve or a third-party tool to modify the FILE.DDF file so that the path statements to the data files are correct.
2. Check the descriptive Btrieve table name to see if it begins with any of the following SQL keywords: Select, Insert, In, Update, Alter, Transform, Delete, Create, Drop, Procedures, or Parameters.

MORE INFORMATION

=====

Cause 1

-----

Microsoft Access requires all three Xtrieve data dictionary files (FILE.DDF, INDEX.DDF, and FIELDS.DDF) to be able to import or attach Btrieve files.

FILE.DDF contains the MS-DOS filename of each data table in the dictionary. These filenames can be in any of the following valid MS-DOS formats:

- File Only (for example, MYDATA.DAT)

- Relative Path (for example, ..\DEV\TEST\MYDATA.DAT)
- Fully qualified (for example, C:\TOOLS\DEV\TEST\MYDATA.DAT)
- Universal Naming Convention (UNC) (for example, (\\MYSERVER\USERS\JOE\TOOLS\DEV\TEST\MYDATA.DAT)

When you are importing or attaching a Btrieve file, Microsoft Access retrieves the MS-DOS filename from FILE.DDF and attempts to open the file. If the file does not exist in the location specified, you will see this error message:

Couldn't find object

In the case of the File Only and Relative Path formats, the ability to open the file depends on the accuracy of the current directory. For example, in the File Only format, MYDATA.DAT must exist in the current directory. In the Relative Path format, MYDATA.DAT must exist in the ..\DEV\TEST directory, which in turn must exist relative to the current directory.

Exactly which directory is the current directory is complicated by the fact that Windows-based applications all share the same environment and the same current directory. For example, if you change the current directory in Microsoft Excel by choosing Open from the File menu and changing directories to open a file, you also change the current directory for Microsoft Access.

Therefore, to fix the problem, you need to find out what path statements are included in FILE.DDF. You can do this by opening FILE.DDF in Microsoft Write, the word processor provided with Windows. When Microsoft Write asks if you want to convert this file to Microsoft Write format, choose No Conversion.

**WARNING:** Do not save the file after you finish viewing it with Microsoft Write. If you do, you will corrupt FILE.DDF beyond repair.

After opening FILE.DDF in Microsoft Write, you will see several characters that look like a box. Scroll to the end of the file. There you will see the MS-DOS file names of the Btrieve data files. The extension is usually .DAT, but it does not have to be. Once you have noted the names and paths, close FILE.DDF taking care NOT to save the changes.

Now that you have the file names, here are two things you can do to fix the problem:

1. Move the files into the correct location(s). This is the easiest method and the one Microsoft recommends.
2. Use Btrieve or a third-party tool to modify the FILE.DDF file so that the paths are correct. This method is much more complicated. Microsoft recommends you do not attempt them unless you are a Btrieve expert.

Cause 2



-----

When you first attach a Btrieve table, Microsoft Access displays the descriptive table name, which is different than the DOS filename for the Btrieve table. For example, "IN Product File" is the descriptive name while INPRODUC.DAT is the DOS filename.

Microsoft Access reads in the descriptive name during the initial attachment and checks the name to see if it is a SQL keyword. If it is, Microsoft Access will return the above error message.

Therefore, to fix the problem, you will need to modify the descriptive name to change it to another name that does not conflict with any of the SQL keywords listed above. Using the above example, change the descriptive name from "IN Product File" to "IN\_Product File."

You will need to use Btrieve or a third-party tool to modify the FILE.DDF file to change the descriptive name. This method is complicated and Microsoft recommends you do not attempt this unless you are a Btrieve expert.

Additional reference words: 1.00 1.10 2.00 novell ddf 3rdparty  
KBCategory: kb3rdparty kberrmsg  
KBSubcategory: IsmBtrv

**PRB: "Couldn't Insert or Paste; Data Too Long for Field" Msg**  
**Article ID: Q121512**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you enter data in a new record on a form or in a table, you receive the following error message:

    Couldn't insert or paste; data too long for field

CAUSE

=====

This error occurs if one of the fields in the record has a default value that exceeds that field's FieldSize property setting.

RESOLUTION

=====

Increase the field's FieldSize property setting to be equal to or larger than the field's default value, or change the field's default value so that it does not exceed the field's FieldSize property setting.

MORE INFORMATION

=====

The "Couldn't insert or paste; data too long for field" error message is not displayed until you enter data in a record because the record is not marked for saving to the table until you make some change to the record.

In Microsoft Access version 2.0, this error will also cause a "#Error" message to be displayed in the field whose default value exceeds its FieldSize property setting.

Steps to Reproduce Behavior

-----

1. Start Microsoft Access and open any database.
2. Create and then save the following new table:

    Table: Too Long

    -----

    Field Name: One

        Data Type: Text

        FieldSize: 5

        DefaultValue: Too Long

    Field Name: Two

        Data Type: Text

3. Create a form based on the Too Long table.
4. View the form in Form view, and type any data in the Two field. You will receive the error message stated above.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.1, Chapter 3, "Changing and Customizing Tables," page 38

Microsoft Access "User's Guide," version 2.0, Chapter 8, "Changing and Customizing Tables," page 134

Additional reference words: 1.00 1.10 2.00 data entry

KBCategory: kbusage kberrmsg

KBSubcategory: GnrlDatntry

**PRB: "Couldn't Lock File; SHARE.EXE Hasn't Been Loaded" Error**  
**Article ID: Q116387**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

When you try to open a Microsoft Access database, you receive the following error message even though you are running SHARE.EXE or VSHARE.386:

    Couldn't lock file; SHARE.EXE hasn't been loaded.

CAUSE

=====

This error can occur if Microsoft Access cannot perform record locking on the database's corresponding .LDB file. If, for example, the .LDB file is set to read-only, either on your local hard disk or on a network server, this problem will occur.

With an administrative network setup (setup /a), the .LDB files under the \WORKDIR directory are not created until the first user opens Microsoft Access. This error can also occur if the first user does not have Create rights.

RESOLUTION

=====

Set the .LDB file to read-write, not read-only.

MORE INFORMATION

=====

Microsoft Access may automatically create a locking information (.LDB) file for a database. This file will have the same name as the database itself, but will have an .LDB extension, and will be created in the same directory as the database.

Even if an .LDB file on a network server is set to read-write, network security may cause Microsoft Access to behave as if the file were read-only.

Follow these steps to check whether an .LDB file is read-write or read-only:

1. Start File Manager.
2. Locate and then select the .LDB file.
3. From the File menu, choose Properties. If the Read Only check box is selected, clear the check box to set the file to read-write.

## REFERENCES

=====

For more information about .LDB files, search for "LDB files" then "Locking Information Files (Common Questions)" using the Microsoft Access Help menu.

Additional reference words: 2.00 kberrmsg

KBCategory: kbenv

KBSubcategory: EvnOthr

**PRB: "Couldn't Open File" Error Message Using Paradox Table**  
**Article ID: Q91711**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You cannot open an attached Paradox table from within Microsoft Access, or you experience an "Unexpected Condition" error message in Paradox when you attempt to view a table.

Paradox and Microsoft Access should be able to simultaneously access the same Paradox tables if there is only one PARADOX.NET file in use by both programs.

CAUSE

=====

Microsoft Access

-----

If you receive the error message "Couldn't open file 'filename'" when you try to open an attached Paradox table while you are in Microsoft Access, you may have two PARADOX.NET files. In this case, Microsoft Access may be using one PARADOX.NET file while Paradox is using the other.

There are other conditions that can cause this error message as well. For more information, search for "Error Messages Reference" using the Microsoft Access Help menu.

Paradox

-----

If you receive an error message similar to

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

when you attempt to open a table while you are in Paradox, the cause is the same as above: there are two PARADOX.NET files, with Microsoft Access using one and Paradox using the other.

RESOLUTION

=====

When you attach 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 stand-alone configurations.

Microsoft Access creates a PARADOX.NET file in the Microsoft Access

program directory the first time you try to access Paradox data. 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 will be similar to:

```
[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, run the NUPDATE.EXE utility. Refer to the Paradox "Network Administrator's Guide" for more information on this topic.

Additional reference words: 1.00 1.10 2.00 ISAM PDOX  
KBCategory: kb3rdparty  
KBSubcategory: IsmPdox

**PRB: "Couldn't Update; Currently Locked by User" Error Message**  
**Article ID: Q94080**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The following message appears either when two users simultaneously try to update a similar object in a shared database, or on a single-user system if the system runs out of locks:

    Couldn't update; currently locked by user '<user name>' on machine  
    '<machine name>'

CAUSE

=====

On a multiuser system, a conflict exists with the SYSTEM.MDA file. By design, Microsoft Access cannot simultaneously update the MsysObjects table in the SYSTEM.MDA file for multiple users.

On a single-user system, there are not enough locks specified in the AUTOEXEC.BAT file.

RESOLUTION

=====

Multiuser System

-----

On a multiuser system, try saving the object again. If no one is updating a similar object at the same time, the Save command should work. This error can occur in Access Basic code also, so you will need to handle error trapping for this.

Single-User System

-----

1. Exit Windows.
2. Delete all the .LDB files from the Microsoft Access directory.
3. Delete all the .TMP files from the Windows directory and from the WINDOWS\TEMP directory.
4. Edit the AUTOEXEC.BAT file. Look for a line similar to:

    C:\DOS\SHARE.EXE

    Add "/L:500" (without quotation marks) to the end of the line, so it reads:



C:\DOS\SHARE.EXE /L:500

Note that this line will not be present if you run VSHARE.EXE, OS/2, or you are running on a network.

5. After saving the AUTOEXEC.BAT file, restart your computer.

MORE INFORMATION

=====

The SYSTEM.MDA file conflict is likely to occur only on systems with many (100+) users.

Steps to Reproduce Behavior

-----

1. User A and user B open the same database on a shared drive.
2. Both users create a new object of similar type (such as a table, form, report, query, or macro).
3. Both users choose Save from the File menu and enter a unique filename.
4. Simultaneously, user A and user B press ENTER to update the object.
5. Microsoft Access presents the error message listed above. Choose OK to dismiss the message and try again.

Additional reference words: 1.00 1.10 2.00 network

KBCategory: kbusage kberrmsg

KBSubcategory: GnlMu

**PRB: "Disk or Network Error" Err. Msg. Attaching Btrieve Data**  
**Article ID: Q123787**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SUMMARY

=====

When you try to attach a Btrieve (.DDF) file, you receive the error message "Disk or network error."

CAUSE

=====

This error occurs when you try to attach a Btrieve file created with Btrieve version 6.x using the Btrieve client software for version 5.x

RESOLUTION

=====

Use version 6.x of the Btrieve Requester to attach data created with Btrieve version 6.x

MORE INFORMATION

=====

There is currently no client-only version of Btrieve 6.x. To attach Btrieve 6.x data, you must use the Btrieve client-server software that requests data from the Btrieve NLM. If you try to attach Btrieve 6.x data with the client-only version of Btrieve 5.x, you will receive the error message stated above. Note that Btrieve 6.x can read the 5.x format, but you must have version 6.x drivers to read 6.x data files.

Microsoft Access can use data from version 5.x or 6.x of Btrieve, but the client software must be appropriate for the version of Btrieve that created the data.

The latest client software for Btrieve is available on CompuServe in the NOVLIB forum, in the Btrieve/NW API section, in a file called BTRREQ.EXE. The name of this file may change, so search for the latest version of the client software.

Btrieve is manufactured by Novell, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

REFERENCES

=====

For more information about using Btrieve data with Microsoft Access, view the "Using Microsoft Access with Btrieve" topic in the Microsoft Access Readme Help file (ACREADME.HLP).

Additional reference words: 1.00 1.10 2.00  
KBCategory: kb3rdparty kberrmsg  
KBSubcategory: IsmBtrv

**PRB: "Error Reading Table or Query '<Name>'" Error Message**  
**Article ID: Q117540**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

In Microsoft Access version 1.x, the error message "0 parameters supplied, <number> expected" occurs when you select a Form or Report Wizard.

In Microsoft Access version 2.0, the error message "Error reading table or query '<name>'" occurs when you select a Form or Report Wizard.

CAUSE

=====

The query that is selected for the Microsoft Access Wizard is a parameterized crosstab query. The Wizards require fixed column headings so that you can select the fields to display on the form or report.

RESOLUTION

=====

There are three workarounds for this behavior:

- Define column headings for the crosstab query. In Microsoft Access 2.0, use the ColumnHeadings property. In Microsoft Access 1.x, choose Query Properties from the View menu in the query's Design view to define fixed column headings.
- Instead of using parameters, enter value ranges in the crosstab query. For example, in the criteria row of the parameterized field, replace

between [start date] and [end date]

with

between #5/15/91# and #5/25/91#

- In Microsoft Access 2.0, output the results of the query to a Microsoft Excel spreadsheet file, and then import that file into Microsoft Access. For more information on exporting a query and importing a Microsoft Excel file, search for "exporting data: basics" and "importing data: basics" using the Microsoft Access Help menu.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB. Create a new query based on the

Orders table. Then, from the Query menu, choose Crosstab. Set up the following query:

Query: Test Wizard

-----  
Type: crosstab query

Field: Employee Id

Table: Orders

Total: Group By

Crosstab: Row Heading

Field: Order Date

Table: Orders

Total: Group By

Crosstab: Column Heading

Criteria: Between [start date] and [end date]

Field: Freight

Table: Orders

Total: Sum

Crosstab: Value

2. From the Query menu, choose Parameters. Define Start Date and End Date as Date/Time data types.
3. Save the query and then run it. Type "5/10/91" (without quotation marks) as the start date and "5/15/91" (without quotation marks) as the end date. Close the query.
4. Select the Database window. Click the Form button, then click New. In the Select A Table/Query box, select the Test Wizard query. Click the Form Wizards button. Select any of the Form Wizards and then choose OK.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.x, Chapter 6, "Designing Select Queries," pages 154-158

Microsoft Access "User's Guide," version 2.0, Chapter 12, "Advanced Queries," pages 279-284

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: WzOthr

**PRB: "Help Topic Does Not Exist" Invoking Custom Help File**  
**Article ID: Q103179**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

Pressing F1 to invoke a custom Help file topic for a form control produces the message "Help topic does not exist" twice.

CAUSE

=====

The number entered in the HelpContextID property could not be mapped to a Help file context string in the custom Help file.

RESOLUTION

=====

Make sure a [MAP] section exists in the Help project file that correctly maps the HelpContextID context number to the context string in the custom Help file.

MORE INFORMATION

=====

The number entered in the HelpContextID property for a Microsoft Access form control can be mistaken for the context string used to identify a topic in the help system. The HelpContextID is actually known as a context number. A context number is an optional number that you can associate with context strings by using the [MAP] section in the help project file. A context number can be assigned to each object in an application to enable context sensitivity.

A typical [MAP] section in the Help project file will look like the following:

```
[MAP]
TitlePageContextString 3
Topic1ContextString 1
Topic2ContextString 2
```

The context string appears first, followed by the HelpContextID context number. The context string and context number are separated by white space, using either space characters or tabs.

In the example above, the # footnote for the first topic would be set to the string TitlePageContextString. Within Microsoft Access, you would set HelpContextID to the number 3.

REFERENCES

=====

Microsoft Access Distribution Kit "Help Compiler Guide," version 1.1,  
Chapter 6

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: PgmOthr

**PRB: "Internal Database Error (-5001)" Msg Using Btrieve NLM**  
**Article ID: Q97531**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You receive the following error message when you are using the Btrieve Network Loadable Module (NLM) with Microsoft Access version 1.0:

Internal Database Error (-5001)

NOTE: You might also encounter an "Unexpected Error from External Database Driver [1017]" error message if you are missing the WBTRVRES.DLL file. This message occurs in Microsoft Access versions 1.0, 1.1, and 2.0.

CAUSE

=====

You do not have correct versions of the BREQUEST.EXE, WBTRCALL.DLL, and WBTRVRES.DLL files. In order for the Btrieve NLM to function properly with Microsoft Access, Microsoft recommends that the NLM requester (client) use BREQUEST.EXE, WBTRCALL.DLL, WBTRVRES.DLL version 6.0, and BTRIEVE.NLM version 5.15.

RESOLUTION

=====

Use the following procedure to obtain the correct versions of these Btrieve files:

1. Sign on to CompuServe.
2. At the exclamation point (!) prompt, type "go novlib" (without the quotation marks).
3. Choose the BTRIEVE/XQL library section.
4. Download the BT515.EXE file and place it in an empty directory. Then unzip the file by running BT515.EXE. BT515.EXE is a self-extracting zip file and will extract several files and place them in directories.
5. After the files are extracted, follow the instructions in the BT515.TXT file to properly install these files and for information on how to apply patches to the BTRIEVE.NLM file on your server.
6. Add the following lines to your WIN.INI file:

```
[BREQUESTDPMI]
datalength=4096
tasks=10
```



local=no  
chkparms=no

MORE INFORMATION

=====

Btrieve and the Btrieve NLM are manufactured by Novell, Inc., a company independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 1.00 1.10 2.00

KBCategory: kb3rdparty kberrmsg

KBSubcategory: IsmBtrv

**PRB: "Invalid Entries in [Btrieve] Section..." Error Message**  
**Article ID: Q121650**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

When you try to attach a Btrieve table in Microsoft Access, you receive the following error message:

Invalid entries in [Btrieve] section in WIN.INI

CAUSE

=====

There are multiple Options lines in the [Btrieve] section of your WIN.INI file. The error message occurs even if one of the Options lines is correct.

RESOLUTION

=====

Remove the duplicate Options lines from the [Btrieve] section of the WIN.INI file. The [Btrieve] section of the WIN.INI file should look like the following example:

```
[Btrieve]
Options=/m:64 /p:4096 /b:16 /f:20 /l:40 /n:12 /t:C:\ACCESS\BTRIEVE.TRN
```

MORE INFORMATION

=====

The [Btrieve] section of the WIN.INI file (for Btrieve version 5.1x) or the NOVDB.INI file (for Btrieve version 6.x) includes settings used by all Btrieve for Windows applications. The settings must be compatible with all Btrieve for Windows applications.

If the WIN.INI file does not contain a [Btrieve] section, the section will be added by the Setup program when you install Microsoft Access. If the WIN.INI file does already contain a [Btrieve] section with an Options line, Microsoft Access Setup will add a second options line beginning with "access options." This line contains default settings and is harmless.

Third-party software products that use Btrieve files can also alter the [Btrieve] section of the WIN.INI file. These products can add duplicate Options lines to the [Btrieve] section of the WIN.INI file.

Steps to Reproduce Behavior

-----

1. Make a backup copy of your WIN.INI file.
2. Open the WIN.INI file in Windows Notepad.

3. Locate the [Btrieve] section of the file, and then modify it to match the following example.

NOTE: In the following example, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this example.

```
[Btrieve]
Options=/m:64 /p:1024 /f:16 /l:20
Options=/m:64 /p:4096 /b:16 /f:20 /l:40 /n:12 /t:C:\ACCESS_
BTRIEVE.TRN
```

4. Save and then close the WIN.INI file.
5. Start Microsoft Access and open any database.
6. From the File menu, choose Attach Table.
7. In the Data Source box, select Btrieve, and then choose the OK button.
8. In the File Name box, select the Btrieve file you want to attach, and then choose the Attach button. You will receive the error message stated above.

Btrieve is manufactured by Novell, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

#### REFERENCES

=====

For more information about using Btrieve with Microsoft Access, view the following topics in the ACREADME.HLP file:

- "Using Microsoft Access with Btrieve"
- "Setting Microsoft Access Options for Btrieve"
- "Setting Btrieve Options"

Additional reference words: 2.00 switches

KBCategory: kb3rdparty kberrmsg

KBSubcategory: IsmBtrv

**PRB: "Invalid Operation" Error Msg. Running QueryDef Example**  
**Article ID: Q124341**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you run the sample QueryDef code on page 251 of the Microsoft Access version 2.0 "Building Applications" manual, you receive the error message "Invalid operation."

CAUSE

=====

This error occurs because the Append method is used incorrectly with the QueryDefs collection. Although the Append method is a valid method for the QueryDefs collection, it is used incorrectly in the sample code.

The CreateQueryDef method creates a new QueryDef object and appends it to the QueryDefs collection if it is supplied with a valid name. If you try to append the QueryDef object again using the Append method, you receive an error message because the QueryDef object has already been appended to the collection.

RESOLUTION

=====

The sample code on page 251 should read as follows.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Dim MyDB As Database, MyQuery as QueryDef
Set MyDB = DBEngine.Workspaces(0).Databases(0)

' Create QueryDef.
Set MyQuery = MyDB.CreateQueryDef("All Cust", "SELECT * FROM _
Customers;")

' Set SQL property.
MyQuery.SQL = "UPDATE DISTINCTROW Products SET _
Products![Supplier ID] = 2 WHERE Products![Supplier ID] = 1;"

MyQuery.Execute ' Invoke query.
MyQuery.Close ' Close query.
```

MORE INFORMATION

=====

You can use the Append method with the QueryDefs collection when the CreateQueryDef method does not assign a name to the QueryDef object, as

in the following example:

```
Dim MyDB As Database, MyQuery As QueryDef
Set MyDB = DBEngine.Workspaces(0).Databases(0)

' Create QueryDef object without a name.
Set MyQuery = MyDB.CreateQueryDef()
MyQuery.Name = "All Cust"
MyQuery.SQL = "SELECT * FROM Customers;"
MyDB.QueryDefs.Append MyQuery
```

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 11,  
"Working with Sets of Records," page 251

Additional reference words: 2.00  
KBCategory: kbusage kbdocerr kberrmsg  
KBSubcategory: DcmMnl

**PRB: "Not Enough Memory to Run SETUP.EXE" Error Message**  
**Article ID: Q94928**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

An attempt to run the Microsoft Access Setup program fails and Setup displays one of the following out-of-memory error messages:

Insufficient Memory

-or-

Not enough memory to run SETUP.EXE

When this error occurs, sufficient space is available on the disk you are installing Microsoft Access to and your AUTOEXEC.BAT file contains a valid SET TEMP statement that specifies a directory that has free disk space.

CAUSE

=====

One cause of this error relates to an incorrect CMOS setting for the floppy disk drive from which Microsoft Access is installed.

RESOLUTION

=====

Modify the CMOS setting to correctly configure the disk drive. If you need assistance to change the settings appropriately, contact your dealer or hardware manufacturer.

MORE INFORMATION

=====

The error messages above can occur when you install Microsoft Access from a 1.2 MB disk drive and the CMOS setting for that drive indicates that it is a 360 KB disk drive.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbsetup kberrmsg

KBSubcategory: StpOthr

**PRB: "ODBC Call Failed" When Exporting Table to SQL Server**  
**Article ID: Q109341**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You receive the error message "ODBC - Call failed" when you export a Microsoft Access table to a SQL Server database using ODBC if the table field names are SQL reserved words. This error message is followed by another which reads:

[Microsoft][ODBC SQL Server Driver][SQL Server] Incorrect syntax near the keyword '<keyword>'. (#156)

CAUSE

=====

This problem will occur if the table you are exporting contains a field name that is also an SQL keyword. The Microsoft ODBC driver for SQL Server translates the export command into an SQL statement that can be interpreted by the SQL Server. If a field name in the Microsoft Access table is an SQL keyword, the SQL Server will interpret the field name as an SQL command keyword.

RESOLUTION

=====

Rename the fields in the Microsoft Access table to non-SQL reserved words. Field names such as Order, In, From, or Where should not be used when you create Microsoft Access tables.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Create a new table with a field named Order.
2. Save the table with the name Test Table.
3. From the File menu, choose Export.
4. In the Data Destination box select <SQL Database>, and then choose OK.
5. In the Select Microsoft Access Object box, choose Test Table. Choose OK.
6. In the Export dialog box, choose OK.
7. In the SQL Data Sources dialog box, select a SQL Server. Choose OK.

8. If necessary, log on to the SQL Server using the SQL Server Login dialog box.

The error messages stated above will be displayed.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr



**PRB: "ORA-00904: Invalid Column Name" Error Message in Query**  
**Article ID: Q108439**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SYMPTOMS

=====

If you have a query based on an attached Oracle synonym that has a sort on a column name, and the owner of the synonym has read-only permissions on the table on which the synonym is based, you will receive the error messages:

ODBC - call failed

-and-

[PageAhead][ODBC Oracle Driver][Oracle OCI]ORA-00904: invalid column name. [#904]

CAUSE

=====

This error occurs because the owner of the synonym has read-only permissions on the Oracle table on which the synonym is based.

RESOLUTION

=====

There are four ways to correct this problem:

- Create a Public synonym in Oracle.
- In Microsoft Access, attach directly to the table on which the synonym is based.
- In Oracle, assign the owner of the synonym permissions greater than read-only.
- In Microsoft Access, remove the "order by" clause from the query.

MORE INFORMATION

=====

The following is an example of an SQL statement that will cause this error:

```
SELECT * FROM <owner>.<synonym_name>
ORDER BY <owner>.<synonym_name>.<column_name>
```

Note that the same errors will occur if this command is issued in SQL\*DBA or SQL\*PLUS, which are Oracle's interface tools. If you remove the ORDER BY

clause, the statement will run. The statement will run with the ORDER BY clause if you remove the <owner>. clause instead. The statement will run with all the clauses included if you increase the owner's permissions on the table on which the query is based.

#### Steps to Reproduce Behavior

-----

1. In Oracle, create a table. Assign all permissions on the table to user A, and read-only permissions to user B.
2. Log into Oracle as user A. Create synonym A based on the new table.
3. Log into Oracle as user B. Create synonym B based on the new table.
4. Start Microsoft Access. Attach synonym A with user A's login account. Create a query based on synonym A with a sort on one of the columns.
5. Run the query. Note that the query runs correctly.
6. Attach synonym B with user B's login account. Create a query based on synonym B with a sort on one of the columns.
7. Run the query. The error messages stated above will occur.

NOTE: You can reproduce this behavior in Oracle's SQL\*DBA as well. To do so, connect as user A and issue the following command:

```
SELECT * FROM userA.synonymA ORDER BY userA.synonymA.column1;
```

Note that the command runs correctly. Next, connect as user B and issue the following command:

```
SELECT * FROM userB.synonymB ORDER BY userB.synonymB.column1;
```

An error message will occur.

Additional reference words: 1.10 2.00 adk

KBCategory: kbinterop kberrmsg

KBSubcategory: IntpOthr

**PRB: "ORA-9241" Error Deleting Then Attaching ORACLE Table**  
**Article ID: Q119470**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

If you attach several ORACLE tables, delete a table, then immediately attach another ORACLE table, you receive the following error message when you try to log in to the table:

```
[Microsoft][ODBC DLL] Driver's SQLSetConnectOption failed (#0)
[PageAhead][ODBC Oracle Driver]Driver not capable. (#0)[PageAhead]
[ODBC Oracle Driver][Oracle OCI] ORA-09241: Message not found;
product=RDMS; facility=ORA; language=NULL. (#9241)
```

CAUSE

=====

Preliminary tests indicate that this behavior is due to the ORACLE 6.0 ODBC driver. The ORACLE 7.0 ODBC driver does not exhibit this behavior.

According to ORACLE's error documentation, the ORA-9241 error message means "ss\_tln: error translating logical name," which indicates that an error occurred in the low-level network communication process, such as an environment configuration problem or a lack of file handles.

RESOLUTION

=====

If you are running an ORACLE 7.0 server, you can work around this behavior by obtaining the ORACLE 7.0 ODBC driver. You can download this driver from the Oracle forum on CompuServe, or you can purchase it from Intersol (formerly Q+E).

If you are running an ORACLE 6.0 server, there is currently no workaround other than upgrading the server to ORACLE version 7.0 and using the ORACLE 7.0 ODBC driver.

MORE INFORMATION

=====

ORACLE is manufactured by Oracle Corporation, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: ObcOracl

**PRB: "Out of Memory" Error Message Loading Large Module**  
**Article ID: Q114515**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

You get an "Out of Memory" error message when you attempt to load a large module when Microsoft Access Wizards (such as the Form Wizard or Report Wizard) are loaded.

RESOLUTION

=====

Break up large modules into multiple smaller modules. Also, load as few Wizards as possible when dealing with large modules.

STATUS

=====

Microsoft has confirmed this to be a problem in Microsoft Access version 2.0. We are researching this problem and will post new information here in the Microsoft Knowledge Base as it becomes available.

Additional reference words: 2.00 oom

KBCategory: kbprg kberrmsg

KBSubcategory: PgmObj

**PRB: "Out of Memory" or "Query Too Complex" Running Report**  
**Article ID: Q103429**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You receive an "Out of Memory" or "Query too Complex" error message when you run your report.

CAUSE

=====

Reports create temporary queries for each section of the report, including the report header, page header, group header, detail section, group footer, page footer, and report footer.

All of the temporary queries for each report are combined into a segmented virtual table (SVT). The final output must be compiled within a 64K segment limit. When this limit is reached, you may see either the "Query too Complex," or the "Out of Memory" error message.

Similar 64k limits are used to compile and store all of the expressions from page to page when the report is being processed or to store the unbound controls or label information. If any of these segments exceed the limit, controls on the report may start displaying the "#Name?" error message.

RESOLUTION

=====

The error messages stated above are related to the complexity of the underlying query or the report itself. The following items suggest several ways to reduce the complexity of your queries or report:

- Shorten table names, column names, and control names. Reducing a 30-character name to the minimum length may help.
- Reduce the number of text fields on the report.
- Avoid extra overhead by removing any fields in the underlying query that are not used in the final output for the report.
- Reduce expressions in underlying queries. Reducing space used for expressions in the select list helps to avoid these errors. If possible, place the expressions directly in the report.
- Move complex expressions to a user-defined function that does all the evaluation.
- Avoid stacked query objects, such as situations in which Query1 is used to pull data from Table2, where Query2 filters the data.

Pulling information together in one query is preferable to having multiple queries each doing portions of the task.

- Avoid basing main reports and their subreports on the same queries. Look for stacked query objects and for tables that are unnecessary to the subset.
- Use subreports to break up a complex report into several less complicated reports.
- If the report is based on a query, build a temporary table from the query to base the report on instead of the query. Create a Make Table query that includes all the fields from the original query to build the temporary table.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptOthr

**PRB: "Out of Memory" or Low System Resources Printing Lines**  
**Article ID: Q121358**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you print a form or report containing graphic lines created with the line control, you either receive an "Out of memory" error message, or you notice that Microsoft Windows system resources decrease as you print multiple forms or reports.

CAUSE

=====

When you print forms or reports containing graphic lines, Microsoft Access stores the image created to print the lines. The image is not cleared until you quit Microsoft Access. As you print multiple forms or reports, the Windows system resources decrease. Quitting and then restarting Microsoft Access restores the lost system resources.

RESOLUTION

=====

Instead of using the line control to create a line on a form or report, use a rectangle control or the borders of a text box.

The following sample code demonstrates how to open all the reports in a database and then replace all the lines controls in the reports with rectangle controls.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Option Compare Database ' Use database order for string comparisons.
Option Explicit
```

```
Sub ChgLinToRect (Rptname As String)
 ' Opens the report and changes lines to rectangles.
 Dim i, numctrls
 Dim rpt As Report
 Dim newrect As Control
 DoCmd OpenReport Rptname, A_DESIGN
 Set rpt = Reports(Rptname)
 numctrls = rpt.count
 For i = 0 To numctrls - 1
 If IsLineControl(rpt(i)) Then
 ' If it is a line control then create a new rectangle
 ' with properties like the line.

 Set newrect = CreateReportControl(rpt.name, 101, _
```

```

 rpt(i).section, "", "", rpt(i).left, rpt(i).top, _
 rpt(i).width, rpt(i).height)

 newrect.BorderStyle = rpt(i).BorderStyle
 newrect.BorderColor = rpt(i).BorderColor
 newrect.BorderWidth = rpt(i).BorderWidth
 newrect.BorderLineStyle = rpt(i).BorderLineStyle
 newrect.SpecialEffect = 0
 ' Delete the line.
 DeleteReportControl Rptname, rpt(i).name
 ' Decrease counters by one.
 i = i - 1
 numctrls = numctrls - 1
End If
Next i
DoCmd DoMenuItem 7, 0, 2
DoCmd Close
End Sub

Function DoAllReps ()
' This function loops through all reports, changing lines to
' rectangles.
Dim db As Database
Dim i
Set db = DBEngine.Workspaces(0).Databases(0)
For i = 0 To db.Containers("Reports").Documents.Count - 1
 ChgLinToRect (db.Containers("Reports").Documents(i).Name)
Next i
End Function

Function IsLineControl (Ctl As Control)
' This function tests to see if a control is a line control.
Dim x
On Error Resume Next
x = Ctl.lineslant
IsLineControl = (Err = 0)
End Function

```

To run this code, call the DoAllReps() function in a module's Immediate window by typing the following line and then pressing ENTER:

```
?DoAllReps()
```

NOTE: There will be a great deal of screen activity caused by the property sheets and windows opening and closing.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprint kberrmsg

KBSubcategory: PtrGraph



**PRB: "Output To" of Print Definition Contains Only Headers**  
**Article ID: Q112794**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you use the Output To command to create an output file from a Print Definition report, the file contains pages that have only headers. The rest of each page is blank.

CAUSE

=====

All the information in Print Definition reports, except for the header, is contained in subreports. The Output To command does not support subforms or subreports. Print Definition reports cannot be saved because they are created temporarily in a library database.

RESOLUTION

=====

You can work around this design limitation by saving the Print Definition report as a table, or by printing it to a file. Both of these methods are described below.

Saving the Print Definition Report as a Table

-----

When you preview a Print Definition report, the Save As Table command is available in the File menu. Choosing this command saves the contents of the Print Definition report as a table named Object Definition. If a table already exists with this name, the new table is named Object Definition1. You can create your own queries and reports on the Object Definition table to analyze the structure of your database objects.

Printing the Print Definition Report to a File

-----

To print the Print Definition report to a file, open the report and use the Output To command to print it to a text file using the Windows Generic print driver attached to the FILE port.

For more information about how to print to a text file using the Windows Generic printer, query on the following words here in the Microsoft Knowledge Base:

print and file and generic

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. In the Database window, select the Shippers table.
3. From the File menu, choose Print Definition, then choose OK.
4. Choose either the Publish It With MS Word or the Analyze It With MS Excel button on the toolbar.

NOTE: If you do not have either Microsoft Word for Windows or Microsoft Excel installed on your computer, you can add a button to the toolbar to output text files. To do this, click the secondary mouse button on the toolbar, choose Customize, choose File in the Categories box, and then drag the Output To Notepad button to the toolbar.

The file that is output will contain text similar to the following:

|                              |                       |
|------------------------------|-----------------------|
| C:\ACCESS\SAMPAPPS\NWIND.MDB | Friday, April 1, 1994 |
| Table: Shippers              | Page: 1               |

Note that the page is blank except for the header.

REFERENCES

=====

For more information about the Output To command, search for "Output To" then "Output To Command" using the Microsoft Access Help menu.

Additional reference words: 2.00 printing outputto database documentor  
KBCategory: kbusage  
KBSubcategory: Gn1Othr

**PRB: "Record Lock Threshold Exceeded" with Large Action Query**  
**Article ID: Q102522**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

If you exceed the maximum number of locks set per connection for the Novell server, you will get the following error message on the server:

Record Lock threshold exceeded

This error can also cause the Novell server to crash.

RESOLUTION

=====

To solve this problem, use the following steps:

1. Obtain the TTSFIX.NLM dynamic patch, either directly from Novell or from CompuServe. On CompuServe, look for 311PTD.EXE in the NOVFILES forum.
2. Set the Record Lock threshold to some high value greater than the default of 500, preferably between 6,000 and 10,000, even though the server should no longer crash. However, Microsoft Access may still encounter problems when the server exceeds the lock count.

There are no reports of any problems with the Record Lock threshold set to a high value. Even with this maximum set, if the threshold is reached, Microsoft Access will stop; with the fix, the Novell network will not crash.

3. Load the TTSFIX.NLM dynamic patch on the server to ensure that it will not crash if a lock count is exceeded.

NOTE: The 311LPTD.ZIP file also contains PATCHMAN.NLM and TTSFIX.NLM. We recommend that these files be loaded in the AUTOEXEC.NCF of the server. This action requires a LOAD PATCHMAN.NLM, followed by a LOAD TTSFIX.NLM in the AUTOEXEC.NCF.

The following table details possible lock settings:

|                                             | Range       | Defaults       |
|---------------------------------------------|-------------|----------------|
| -----                                       |             |                |
| #1 SET Maximum record locks per connection: | 10-10,000   | default 500    |
| #2 SET Maximum file locks per connection:   | 10-1,000    | default 250    |
| #3 SET Maximum record locks:                | 100-200,000 | default 20,000 |
| #4 SET Maximum file locks:                  | 100-100,000 | default 10,000 |

#1 is the setting that should be increased from the default to some value within the 6,000 to 10,000 range.

NOTE: This problem is related to the record locks setting on the Novell server. This problem is not related to the MS-DOS SHARE.EXE locks setting.

Additional reference words: 1.00 1.10 2.00 novell server 3.11

out of memory

KBCategory: kbusage kberrmsg

KBSubcategory: GnlMu

**PRB: "Save Changes to Form" Prompt When No Changes Made**  
**Article ID: Q115876**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

Each time you open a form in Form view and then close it, you are prompted to save the changes to the form, even if you have not made any changes to the form.

CAUSE

=====

The form contains an OLE Object, and the object frame's Locked property is set to No.

RESOLUTION

=====

To keep from being prompted to save your changes each time you close the form, set the object frame's Enabled property to No and its Locked property to Yes. Note that if you set both the Enabled and the Locked properties to Yes, you will receive the following message when you close the form, and any changes to the form will be discarded:

The object is locked and any changes made will be discarded  
when the form is closed.

Workaround to Allow Editing

-----

The following example demonstrates how to keep from being prompted to save your changes each time you close the form, but still allow changes to be made to the form.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Open the Sales By Product form in Design view.
2. Add a new command button to the form.
3. Click the secondary mouse button on the new command button. Choose Build Event, then select Code Builder and choose OK. Enter the following code in the module:

```
Me!Embedded13.Enabled = -1
Me!Embedded13.Locked = 0
DoCmd GoToControl "Embedded13"
```

4. Close the module.
5. Set the graph object's Enabled property to No and its Locked property to Yes.
6. Set the graph object's OnExit property to the following event procedure:

```
DoCmd GoToControl "Product Name"
Me!Embedded13.Enabled = 0
Me!Embedded13.Locked = -1
SendKeys "%FS"
```

To create this event procedure, click in the OnExit property field, then choose the Build button at the right side of the field. Select Code Builder, and then choose OK. Enter the code in the module, and then close the module.

7. Save and then close the form.

When you open the form, the graph will not be enabled. If you close the form, you will not be prompted to save your changes. However, if you choose the new command button, the graph will be enabled. When you select a different control on the form, the graph will be locked and the form will be saved.

Note that if you choose the command button and then immediately close the form, you will be prompted to save your changes. This is due to the timing of the OnExit property.

#### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Open the Sales By Product form.
3. Close the form without making any changes. Note that you are prompted to save your changes even though you did not make any.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsOthr

**PRB: "Setup Parse Error..." Error Message in Run-Time Setup**  
**Article ID: Q116063**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

When you are running Setup from disks you created with the Microsoft Access Developer's Toolkit, you receive the following error message after the "Setup is searching for installed components" screen:

Setup Parse Error in Initialize Pass. Object ID 1: AppSearch Object:  
Bad Default Path Field.

After this error message, Setup quits without completing.

CAUSE

=====

When you ran the Setup Wizard to create the disks, you specified C:\ as the default installation directory, or you specified a directory name that is more than eight characters long. For example, specifying the directory C:\MYPROGRAM would cause the error message to occur.

Including special characters in the installation directory or application name can also generate this message. For example, C:\!MYAPP in the default installation directory or "My Excellent! App" in the Application Name would cause this error.

Regarding the Application Name, the problem is not with the Application Name itself but with the fact that it also creates a Program Manager Group Name, which cannot interpret the special character.

RESOLUTION

=====

Create a new set of installation disks. In the Setup Wizard's Default Installation Directory box, make sure to specify the full path name for the directory into which your application should be installed (for example, C:\MYAPP).

REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0, Chapter 2, "Creating a Custom Setup Program"

Additional reference words: 2.00

KBCategory: kbsetup kberrmsg

KBSubcategory: StpOthr

**PRB: "Unexpected Error from External Database Driver [20]"**  
**Article ID: Q100173**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SYMPTOMS

=====

When you are using the network loadable module (NLM) version of Btrieve and try to import, export, or attach a Btrieve table you receive the error message "Unexpected error from external database driver [20]."

CAUSE

=====

Btrieve returns this error message if you try to access Btrieve data with Microsoft Access before loading the BREQUEST.EXE program. The Xtrieve manual lists this error with the following description:

- 20 Record Manager inactive.  
You made a request before starting the Record Manager.  
Restart the Record Manager.

RESOLUTION

=====

You must start the BREQUEST.EXE program before starting Microsoft Windows. You may want to create a batch file to execute the BREQUEST.EXE program before starting Windows.

This error can also occur if the NLM version of the WBTRCALL.DLL file is used on a non-NLM version of Btrieve. The NLM version of WBTRCALL.DLL is approximately 13K, while the non-NLM version of WBTRCALL.DLL is approximately 51K. Verify that the NLM version of WBTRCALL.DLL is not in the path prior to the non-NLM version to ensure that the correct version is in use.

MORE INFORMATION

=====

Btrieve is manufactured by Novell, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

REFERENCES

=====

For a complete list of Xtrieve error messages, refer to the Novell Xtrieve manual, Appendix A.

Additional reference words: 1.10 2.00 brequest



KBCategory: kb3rdparty kberrmsg  
KBSubcategory: IsmBtrv

**PRB: "Unexpected Error from External Database Driver [22]"**  
**Article ID: Q101319**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SYMPTOMS

=====

When you try to open an attached Btrieve table, you receive the following error message:

Unexpected error from external database driver [22]

CAUSE

=====

Btrieve returns this error message when there is a mismatch between the FIELD.DDF file and the Btrieve data table. This problem usually occurs when the internal structure of the Btrieve table has been changed without corresponding changes having been made to the FIELD.DDF file. The Xtrieve manual lists this error with the following description:

22 Data buffer length too short

The data buffer parameter is not long enough to accommodate the length of the data record defined when the file was created. Verify that the length of the data buffer is at least as long as the file's defined record length.

RESOLUTION

=====

Make sure that the internal structural definition of the Btrieve table is the same as the FIELD.DDF file. This can be done using Novell's Xtrieve program or a third-party DDF builder to update the FIELD.DDF file.

MORE INFORMATION

=====

Btrieve is manufactured by Novell, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

REFERENCES

=====

For a complete list of Xtrieve error messages, refer to the Novell Xtrieve manual, Appendix A.

Additional reference words: 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmBtrv



**PRB: 'Unable to Load Communication Module'**  
**Article ID: Q95612**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The following error message is displayed:

```
Unable to load communication module.
Driver has not been properly installed.
Unable to connect to data source.
```

CAUSE

=====

The error occurs because the DBNMP3.DLL file you are using may be old, you may have duplicate DBNMP3.DLL files on your system and the wrong one is being used, or the DBNMP3.DLL file you are using may be damaged or corrupted.

This error can also be caused by an incorrect [SQLSERVER] entry in your WIN.INI file. This entry should look like:

```
[SQLSERVER]
MY_SERVER=DBNMP3,\\MY_SERVER\PIPE\SQL\QUERY
```

If there are any spaces in the above entry, or if DBNMP3 is misspelled, you will receive the error.

RESOLUTION

=====

In Microsoft Access version 1.0, the DBNMP3.DLL file installed by Microsoft Access version 1.0 is dated 9/15/92, has a size of 8241 bytes. In Microsoft Access version 1.1, the DBNMP3.DLL file is dated 5/26/93 with a file size of 9941 bytes. In both versions, this file should be installed in the \WINDOWS\SYSTEM directory.

MORE INFORMATION

=====

It is possible that there are other situations that may indirectly cause the above error message to display. To troubleshoot this problem, do the following:

- Use File Manager to search for multiple, corrupted, or outdated copies of these files:

| Filename | Date | Size |
|----------|------|------|
| -----    |      |      |

ODBC.DLL 10/16/92 or later (44736 bytes)  
SQLSRVR.DLL 10/16/92 or later (135792 bytes)  
NETAPI.DLL depends on the network being used  
COMMDLG.DLL 10/25/92 or later (89248 bytes)

Note: The sizes and dates referenced are correct if the files are installed by Access version 1.0.

- Make sure the files listed above are in their appropriate locations. All of the files except NETAPI.DLL are usually installed in the Windows \SYSTEM subdirectory. The NETAPI.DLL file is a network specific driver and is usually located in the network's program directory, which should be in the path.
- Use another application, such as Q+E, SAF, or PowerBuilder, on the same machine to attempt to attach to the same data source. If you cannot access the data source from another application, then it is probably a problem with the network, such as NETAPI.DLL, or a problem with the server, not with Microsoft Access or ODBC.
- Attempt to attach to another data source, if available, such as another SQL server. If you can attach to another server, then the original server may be down or you may need to reconfigure the entry for that server using the ODBC Administration Utility.
- Attempt to attach to the same data source from another machine. If you cannot access the SQL server from another machine, it may indicate a network-wide problem or that the server is down or not communicating.
- Try increasing your ODBC timeout settings in the ODBC section of the MSACCESS.INI file.
- Check with the system administrator to see if the SQL server is up, available for transactions, and operating properly.
- Reinstall ODBC after performing a clean boot and renaming the ODBC.DLL, DBNMP3.DLL, and SQLSRVR.DLL files.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, pages 660-662

Additional reference words: 1.00 1.10 2.00 kbtshoot

KBCategory: kbusage

KBSubcategory: ObcOthr

## **PRB: AccessWizards Button Unavailable**

**Article ID: Q90097**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

In Microsoft Access version 1.x, the FormWizards or ReportWizards button is disabled and not available.

In Microsoft Access version 2.0, you receive the following error message:

Cannot Find Wizard or Error in Wizard code. Check the [Libraries] section of your MSACC20.INI file.

### CAUSE

=====

One reason for this behavior is that no table or query object exists in the Database window. You can correct this by creating a table.

Another reason for this behavior is that the MSACCESS.INI file (or the MSACC20.INI file in Microsoft Access version 2.0) is missing information in its [Libraries] section.

### RESOLUTION

=====

To check the .INI file, do the following:

In Microsoft Access version 1.x:

1. Open the MSACCESS.INI file in a text editor.
2. Find the [Libraries] section.
3. Verify that the following entry appears:

```
[Libraries]
wizard.mda = ro
```

In Microsoft Access version 2.0:

1. Open the MSACC20.INI file in a text editor.
2. Find the [Libraries] section.
3. Verify that the following entry appears:

```
[Libraries]
wzfrmrpt.mda = ro
```

Additional reference words: 1.00 1.10 2.00 report wizard grey disable  
KBCategory: kbusage  
KBSubcategory: FrmrptWzd

## **PRB: Adding Records to AutoLookup Form Generates Error**

**Article ID: Q116062**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SYMPTOMS

=====

When you are adding a record to a form based on an AutoLookup query, either of the following messages is generated when you try to enter anything in the foreign key field:

To make changes to this field, first save the record

-or-

Current field must match join key "?" on 'one' side of outer join

### CAUSE

=====

You have one or more default values set on control(s) bound to the one-side table in your AutoLookup query.

### RESOLUTION

=====

Make sure that the control(s) on the "one" side have no default values set.

### MORE INFORMATION

=====

When you enter a value in the foreign key field in an AutoLookup query, fields from the one-side table should update automatically with related information. However, if you are adding a new record and one or more fields from the "one" side has a default value, Microsoft Access assumes you are trying to add a record to the "one" side as well as to the "many" side.

If there are no default values for any of the many-side fields, when you enter a value in the foreign key field after making an entry in another field from the many-side table, you receive the error message:

Current field must match join key "?" on 'one' side of outer join

If there are default values for any of the many-side fields, or if you have not entered any value in any field from the many-side table, you will receive the following warning message in the status bar after you enter a value in the foreign key field:

To make changes to this field, first save the record

Steps to Reproduce Behavior  
-----



1. Open the sample database NWIND.MDB.
2. Use a Form Wizard to create a new form based on the AutoLookup query. Include all the fields in the form.
3. View the form in Design view. Set the Region control's DefaultValue property to 0.
4. View the form in Form view. From the Records menu, choose Data Entry. Try to type any value in the Customer ID field. Note the message in the status bar.
5. Press the ESC key.
6. In the Order ID field, enter "1111" (without quotation marks). In the Customer ID field, enter any value. You will receive an error message when you try to exit the field.

#### REFERENCES

=====

For more information about the AutoLookup feature, search for "AutoLookup" then "Looking Up Information Automatically (AutoLookup)" using the Microsoft Access Help menu.

Additional reference words: 2.00 rowfixup defaultvalue

KBCategory: kbusage

KBSubcategory: FmsProp

**PRB: ADT Repair/Compact Documentation Correction**

**Article ID: Q119753**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

The Microsoft Access Developer's Toolkit "Advanced Topics" manual states the following on page 17:

Also be aware that if your application is secured, the user running the repair or compact operation must have Open Exclusive permission for the database.

This information is incorrect.

RESOLUTION

=====

The user running the repair operation does not need to have Open Exclusive permission for the database. Any user validated by the database's SYSTEM.MDA file can run the repair operation, even if that user has no explicit or implicit permissions for the database. However, in order to compact the database, the user must have Open/Run as well as Open Exclusive permissions for the database.

REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0, Chapter 1, "Developing Run-Time Applications: an Overview," page 17

Additional reference words: 2.00

KBCategory: kbusage kbdocerr

KBSubcategory: DcmMnl

**PRB: Append Query Results in Duplicate Counter Numbers**  
**Article ID: Q94105**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

A table that you append data to using an append query now has duplicated numbers in a field defined as a Counter data type. The expected result was appended data with incremented numbers in the Counter field.

CAUSE

=====

Your append query is explicitly referencing the field with the Counter data type. You are, in essence, overriding the automatic counter function that you specified in your table design. This may result in duplicate numbers.

RESOLUTION

=====

Do not include a field with a Counter data type in the Append To row of the QBE grid in your append query. By omitting this field in your append query, Microsoft Access will automatically generate incremental numbers in the appended table's Counter field.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

You will only get duplicates if there is no Primary Key or Index (No Duplicates) defined on the Counter field in the table that you are appending to. By definition, primary keys are unique.

Including fields with Counter data types in your append query might also result in non-contiguous numbers in the table that you are appending to. For example, if the table you are appending to has values of 1 to 20 in the Counter field, and the table you are appending from has values of 20, 34, and 55, you will end up with one duplicate value. The next time you add a new row to the appended to table, the Counter field will be assigned a value of 56.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryMktbl



**PRB: AutoLookup Query Not Performed with 1-to-1 Relationship**  
**Article ID: Q121332**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

Microsoft Access version 2.0 will not perform an AutoLookup query when the tables in the query have a one-to-one relationship.

CAUSE

=====

Microsoft Access determines the relationship between the tables in the query by examining the indexes on the join fields. Microsoft Access will ignore the relationship that you specify when it is determining whether to perform an AutoLookup query. If your tables have one of the following index combinations, Microsoft Access will not perform an AutoLookup query:

| Table 1                | Table 2                |
|------------------------|------------------------|
| -----                  | -----                  |
| Primary Key            | Primary Key            |
| Primary Key            | Indexed, No Duplicates |
| Indexed, No Duplicates | Primary Key            |

RESOLUTION

=====

To cause Microsoft Access to perform an AutoLookup query, make sure that the foreign key is not indexed as the primary key, or as Indexed, No Duplicates.

MORE INFORMATION

=====

It may be necessary to add a unique index to the foreign key on the "many" side of a relationship in order to maintain the integrity of the data. Consider the following tables:

| Employees   | Offices     |
|-------------|-------------|
| -----       | -----       |
| Employee ID | Office ID   |
| Last Name   | Assigned To |

These tables are related on the Employees.[Employee ID] and Offices.[Office ID] fields. To ensure that an employee is not assigned to more than one office, add a unique index to the Offices.[Assigned To] field. Regardless of whether there is no relationship, a one-to-many relationship, or a one-to-one relationship defined between these tables, Microsoft Access will determine that the relationship is one-to-one when determining whether to perform an AutoLookup query.

This behavior does not exist in Microsoft Access version 1.x. Version 1.x will perform an AutoLookup query when the tables have a one-to-one relationship.

#### Steps To Reproduce Behavior

-----

1. Open the sample database NWIND.MDB and create the following new table:

```
Table: Offices

Name: Office ID
 Data Type: Counter
Name: Building
 Data Type: Text
Name: Assigned To
 Data Type: Number
 FieldSize: Long Integer
 DefaultValue: <blank>
```

2. Select the Office ID field, and then choose Set Primary Key from the Edit menu.
3. View the table in Datasheet view, and then add the following data to the table:

```
Office ID Building Assigned To

 1 A
 2 A
 3 B
```

4. Create a new query based on the Employees and Offices tables.
5. Join the tables on the Employees.[Employee ID] and Offices.[Assigned To] fields. Double-click the join line and then select the "Include ALL records from 'Offices' and only those records from 'Employees' where the joined fields are equal" option button. Choose OK.
6. Drag all the fields from the Offices field list to the query grid, and then drag the First Name and Last Name fields from the Employees field list to the query grid.
7. Run the query. Type "7" (without quotation marks) in the first office's Assigned To field, and then press ENTER. Note that the AutoLookup query fills in the First Name and Last Name fields for the record.
8. Save the query as AssignOffices and then close it.
9. Open the Offices table in Design view.
10. Set the Assigned To field's Indexed property to Yes (No Duplicates).
11. Save the table and then close it.

12. Run the AssignOffices query. Type "3" (without quotation marks) in the second office's Assigned To field and then press ENTER. Note that Microsoft Access does not fill in the First Name and Last Name fields.

13. Press SHIFT+F9 to requery. Note that the fields are filled.

#### REFERENCES

=====

For more information about AutoLookup queries, search for "looking up information automatically" then "Looking Up Information Automatically (AutoLookup)" using the Microsoft Access Help menu.

Additional reference words: 2.00 row fix-up updateable  
KBCategory: kbusage  
KBSubcategory: QryFixup

**PRB: Avery 4xxx-Series Labels Do Not Work on Laser Printer**  
**Article ID: Q89621**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you are using Microsoft Access ReportWizards for mailing labels, note that the Avery labels 4xxx series are intended for dot-matrix printers, and the 5xxx series are intended for laser printers. The dot-matrix, or continuous, labels will print incorrectly when they are printed on a laser printer.

CAUSE

=====

These labels are meant for dot-matrix printers and their sizes do not match the standard page sizes available for laser printers.

STATUS

=====

This behavior is by design.

REFERENCES

=====

For more information on creating mailing labels for dot-matrix printers, query on the following words here in the Microsoft Knowledge Base:

creating and mailing and labels and "dot-matrix" and printers

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptLabel



## **PRB: BeforeUpdate Macro Running Twice**

**Article ID: Q92810**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

The BeforeUpdate macro appears to be running twice, rather than only once.

### CAUSE

=====

The BeforeUpdate macro will run whenever an edited record is saved. And if you add records using the Paste Append command, it will run when each of those records is added. If you find that the BeforeUpdate macro is running more than once, check to see exactly what actions you are taking.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. Create a macro with a MsgBox action and then a CancelEvent action.
2. Save this macro as Macrol1.
3. Create a bound form with text boxes.
4. Set the form's BeforeUpdate property to Macrol1.
5. Switch the form from Form Design view to Form view.
6. Enter data in a field, then select the data in that field.
7. From the Edit menu, choose Copy.
8. From the Edit menu, choose Paste Append.
9. Choose the OK button when the Message box (that you created in Macrol1) appears.
10. Close the form.
11. Choose the OK button when the Message box (that you created in

Macrol) appears.

12. Choose the OK button when the following error message appears:

The record being edited can't be saved. If you close the form, the changes you've made to the record will be lost. Close anyway?

Note that the message box you created in Macrol reappeared in step 11. The macro that you assigned to the BeforeUpdate property is running twice: when the record that you added using the Paste Append command is saved, and when the edited record is saved.

#### REFERENCES

=====

For more information, search for "BeforeUpdate, AfterUpdate Properties" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 before update

KBCategory: kbusage

KBSubcategory: McrName

**PRB: Blank Lines in Exported Report Formatted Incorrectly**  
**Article ID: Q122245**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you export a report from Microsoft Access to Microsoft Word version 6.0 for Windows by choosing the Output To command from the File menu and then selecting Rich Text Format, blank lines in the exported Word document are formatted with a different font than the rest of the report. The blank lines are formatted with the MS Sans Serif font, while the rest of the document is formatted with the same font used in the original report.

RESOLUTION

=====

Follow these steps to reformat the document in Word so that the entire document uses the same font:

1. Open the document in Word.
2. From the Edit menu, choose Select All.
3. From the Format menu, choose Font.
4. Select the font you want.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. In the Database window, select the Alphabetical List Of Products report.
3. From the File menu, choose Output To.
4. In the Select Format box, select Rich Text Format (\*.rtf), and then choose OK.
5. In the File Name box, type a name for the Word document, and then choose OK.
6. Start Word and open the document you created in step 5. Note that blank lines in the report are formatted with a different font than the text.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: OtpProb



**PRB: Btrieve ISAM Is Not Installed with Typical Setup**  
**Article ID: Q115570**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

If you install Microsoft Access version 2.0 using the Typical setting, you will be unable to import, export, or attach Btrieve tables. The Btrieve option will not appear in the Data Source or Data Destination box for these operations.

CAUSE

=====

Installing Microsoft Access using the Typical setting does not install the Btrieve indexed sequential access method (ISAM) driver that is necessary for accessing Btrieve data.

RESOLUTION

=====

There are two ways to add the Btrieve ISAM to your system:

- Install Microsoft Access using the Complete/Custom setting instead of the Typical setting.
- or-
- Add the Btrieve components by running Setup from the Microsoft Access Setup icon. To do this, follow these steps:
  1. Double-click the MS Access Setup icon (by default, this icon is in the Microsoft Office program group).
  2. Choose the Add/Remove button.
  3. In the Options box, select ISAM Drivers, and then choose the Change Option button.
  4. In the Options box, select the Btrieve ISAM check box, and then choose OK.
  5. Choose Continue.

REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 1, page 7

Additional reference words: 2.00  
KBCategory: kb3rdparty

KBSubcategory: IsmBtrv

## **PRB: Calendar Control Cannot Format Individual Dates**

**Article ID: Q114832**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Developer's Toolkit version 2.0
- 

### SYMPTOMS

=====

The Microsoft Access Developer's Toolkit (ADT) version 2.0 contains OLE custom controls similar to those in Microsoft Visual Basic. One of the custom controls is the Calendar control. You can alter the formatting of all the text on the Calendar control, but you cannot alter the formatting of individual dates on the control.

### CAUSE

=====

This is a design limitation of the Calendar custom control.

### RESOLUTION

=====

This article assumes that you have installed the ADT, and that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

A workaround for this behavior is to use a text box on your form that reflects the date selected in the Calendar control. The following example demonstrates how to create such a text box:

1. Open, in Design view, the form to which you want to add the text box. From the Edit menu, choose Insert Object. Select the Insert Control option button. In the Control Type box, select Calendar Control. If the Calendar control does not appear in the box, choose the Add Control button and select the MSASB20.OCX file from your ADT directory. After you have selected the Calendar control, choose OK.
2. Set the Calendar control's Name property to Calendar1.
3. Create a new text box on the form. Set the text box's Name property to CalendarDate.
4. To display the form's module, select the Calendar control, click in the control's OnClick property field, choose the Build button to the right of the field, select Code Builder, and then choose OK.
5. Enter the appropriate code so that the procedure looks like the following:

```
Sub calendar1_Click ()
 Me.calendarDate.value = Format(Me.calendar1.object.value, "dddddd")
End Sub
```

This procedure sets the value of the CalendarDate text box to the value of the date selected in the Calendar control, with a long date format. Note that you can use the format of your choice--you do not have to use the long date format.

6. Close the module.

7. View the form in Form view, and select different dates.

#### STATUS

=====

Changes to the Calendar custom control are being reviewed and will be considered for possible inclusion in a future release of the Microsoft Access Developer's Toolkit.

#### REFERENCES

=====

For more information about the Calendar custom control, double-click the Calendar Control Help icon in the ADT program group in Program Manager.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: IntpOle



**PRB: Cannot Add Security Items to Custom Menu**

Article ID: Q103992

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Access Distribution Kit version 1.1
  - Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

If you add a security command to a custom menu with a DoMenuItem action, the security command will be unavailable.

CAUSE

=====

You can only perform DoMenuItem actions with commands available from the current context. If you have a form open, you can only perform menu items available to a form. Since security items are available only from the Database window, they will not work when invoked from a form.

RESOLUTION

=====

You can add the security command Change Password to a custom menu using the following sample macro:

| Macro Name      | Action       | Defined Below |
|-----------------|--------------|---------------|
| Change Password | Echo         | 1             |
|                 | SelectObject | 2             |
|                 | DoMenuItem   | 3             |
|                 | DoMenuItem   | 4             |
|                 | SelectObject | 5             |

Change Password Actions

- 
1. Echo  
Echo On: No
  2. SelectObject  
Object Type: Form  
Object Name: <name of the form that the menu is on>  
In Database window: Yes
  3. DoMenuItem  
Menu Bar: Database  
Menu Name: Security  
Command: Change Password
  4. DoMenuItem  
Menu Bar: Database  
Menu Name: Window  
Command: Hide
  5. SelectObject  
Object Type: Form

Object Name: <name of the form that the menu is on>  
In Database window: No

NOTE: Actions 1 and 5 are used if you need to have the Database window hidden while using the retail version of Microsoft Access. The Echo action in step 1 ensures that the Database window is not visible when it is activated in step 2. Step 4 hides the Database window again. If you are using the Microsoft Access Distribution Kit (ADK), steps 1 and 5 can be omitted since the Database window is always hidden.

The only side effect to this workaround is that when the Change Password dialog box appears, you will see the Database window menu items in the background. However, you will not be able to choose them.

STATUS  
=====

This behavior is by design.

MORE INFORMATION  
=====

Steps to Reproduce Behavior  
-----

1. Create the following macro and save it as Security Menu:

| Action  | Defined Below |
|---------|---------------|
| -----   | -----         |
| AddMenu | 1             |

Security Menu Actions  
-----

1. AddMenu  
    Menu Name: Security Menu  
    Menu Macro Name: Security Menu Password  
    Status Bar Text: Change Password

2. Create the following macro and save it as Security Menu Password:

| Macro Name      | Action     | Defined Below |
|-----------------|------------|---------------|
| -----           | -----      | -----         |
| Change Password | DoMenuItem | 1             |

Security Menu Password Actions  
-----

1. DoMenuItem  
    Menu Bar: Database  
    Menu Name: Security  
    Command: Change Password

3. Create a form and set the OnMenu property to Security Menu.

4. View the form in Form view and note that Change Password is unavailable on the Security menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage  
KBSubcategory: McrOthr

## **PRB: Cannot Attach Paradox Table on Read-Only Drive**

**Article ID: Q92653**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

You cannot attach a Paradox table (\*.DB) located on a read-only drive. If you try to attach a Paradox table on a read-only drive, you receive the following error message:

    Couldn't open file '<FILENAME#DB>' (in Microsoft Access version 1.x)

-or-

    Couldn't lock table '<tablename.db>', currently in use by user '<username>' on machine '<machinename>' (in Microsoft Access version 2.0)

### CAUSE

=====

Neither Paradox nor the Paradox indexed sequential access method (ISAM) shipped with Microsoft Access supports databases located on read-only drives.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

Paradox is manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmPdcox

## **PRB: Cannot Choose Custom Bitmap for Toolbar Button**

**Article ID: Q112135**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

You cannot add user-designed bitmaps to toolbar buttons in Microsoft Access version 2.0. Instead, you can only use the 84 predefined bitmaps that are included with Microsoft Access.

### STATUS

=====

This behavior is by design. This feature is being reviewed and will be considered for revision in a future release of Microsoft Access.

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. From the View menu, choose Toolbars.
2. Choose the Customize button.
3. Click the secondary mouse button on the button in the toolbar whose bitmap you want to change.
4. Choose the Choose Button Face command.

You must select a bitmap from the 84 predefined bitmaps; you cannot choose your own custom bitmap.

Additional reference words: 2.00 UI Limitations

KBCategory: kbui

KBSubcategory: UifOthr

## **PRB: Cannot Clear Selection in Combo Box**

**Article ID: Q123737**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SYMPTOMS

=====

When you delete the selected entry in a combo box, you receive the following error message:

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

### CAUSE

=====

The combo box's LimitToList property is set to Yes. When a combo box's LimitToList property is set to Yes, an empty selection does not match any value in the list. This behavior makes clearing a selection in the combo box difficult.

### RESOLUTION

=====

You can work around this behavior using any of the following methods.

#### Method 1

-----

Provide a different mechanism for clearing the combo box selection. For example, you could specify code in the combo box's OnDblClick property, or in a command button's OnClick property, to set the combo box selection to Null. Given a combo box called MyCombo, you could specify the following event procedure in the combo box's OnDblClick property to set the combo box selection to Null:

```
Sub MyCombo_DblClick (Cancel As Integer)
 Me!MyCombo = Null
End Sub
```

#### Method 2

-----

Use the NotInList event to detect null selections. When a null selection is detected, suppress the error message and enter a null value in the combo box. For example, given a combo box called MyCombo, you could add the following event procedure to the combo box's OnNotInList property:

```
Sub MyCombo_NotInList (NewData As String, Response As Integer)
 Dim Msg As String
 ' Determine if selection is null.
 If NewData = "" Then
 Response = DATA_ERRCONTINUE
 End If
End Sub
```

```

 Msg = "Do you want to clear the combo box selection?"
 If MsgBox(Msg, 32 + 4) = 6 Then
 ' Set the combo box value to Null.
 Screen.ActiveControl = Null
 ' Close the dropped combo box list.
 SendKeys "{F4}"
 Else
 ' Undo the empty selection.
 SendKeys "{ESC}"
 End If
End If
End Sub

```

When you clear a selection in the combo box, the OnNotInList property event procedure will be triggered, prompting you to verify that you want to clear the selection. If you choose Yes, a null value will be entered in the combo box. If you choose No, the original selection will be restored.

Note that when you choose Yes to verify that you want to clear the selection, the combo box list will drop and then close, and the combo box will remain selected. If you pressed the TAB key to exit the combo box, you will need to press TAB again.

#### Method 3

-----

Do not set the combo box's LimitToList property to Yes. Instead, simulate the LimitToList functionality using the DLookup() function in the combo box's BeforeUpdate property to determine whether non-null selections are in the list.

#### Method 4

-----

Add a row to the combo box that can be selected to enter a null value. If the combo box is based on a table, add a null entry to the table. Or, you could use an Access Basic function to fill the combo box list, adding a null value to the top of the list. For more information about adding a value to a list with an Access Basic function, open the Solutions sample application (SOLUTION.MDB). In the Select A Category Of Examples box, select "Get more mileage from combo boxes, list boxes, subforms, and subreports." In the Select An Example box, select "Add '(all)' to a list."

#### STATUS

=====

This behavior is being reviewed and will be considered for change in a future release of Microsoft Access.

#### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB and create a new, blank form.

2. Create a combo box with the following properties on the form:

Name: Field0  
RowSourceType: Table/Query  
RowSource: Customers  
ColumnCount: 1  
LimitToList: Yes

3. View the form in Form view.

4. Select any name in the combo box.

5. Delete the selected name in the combo box and then press ENTER. You receive the error message stated above.

#### REFERENCES

=====

For more information about the NotInList event, search for "NotInList" then "NotInList Event" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: FmsCmbo



**PRB: Cannot Connect DDE from Word 6.0 to Microsoft Access**  
**Article ID: Q109948**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Word version 6.0 for Windows
- 

SYMPTOMS

=====

When you are running a Mail Merge in Microsoft Word version 6.0 for Windows, you cannot connect to a Microsoft Access database using dynamic data exchange (DDE) because the DDE option is not available in the Confirm Data Source dialog box.

CAUSE

=====

This problem occurs because Microsoft Access and its database extension (.MDB) is not registered with Microsoft Windows.

RESOLUTION

=====

Follow the steps below to register Microsoft Access and its database extension with Windows:

1. In Microsoft Windows File Manager, choose Associate from the File menu.
2. Type "MDB" (without the quotation marks) in the Files With Extension box.
3. Choose the Browse button.
4. Select your Access directory in the Directories box, and then select MSACCESS.EXE in the File Name box.
5. Choose OK.
6. Exit and restart Microsoft Windows.

Additional reference words: 1.00 1.10 2.00 Interoperability

KBCategory: kbinterop

KBSubcategory: IntpPrtm

## **PRB: Cannot Delete Admin User**

**Article ID: Q114838**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you attempt to delete the user Admin, you get the error message:

Can't perform this operation

### CAUSE

=====

In Microsoft Access version 2.0, you cannot delete any of the default user or group accounts (Admin, Guest, Admins, Guests, and Users).

### RESOLUTION

=====

With the security enhancements in Microsoft Access version 2.0, it is not necessary to delete the default users and groups. Instead, you can create alternate users and groups and then disable the default users and groups.

For example, follow the steps below to create a new database administrator:

1. Add a new user to the Admins group.
2. Log in as the new user.
3. Remove the Admin user from the Admins group, leaving the Admin user in the Users group.

NOTE: Step 3 is optional as long as you do not create any databases or database objects while logged in as the Admin user. You may want to assign the Admin user a unique password so that you can log in as another user, and to keep others from logging in as the Admin user on the SYSTEM.MDA that was used to create the database.

### REFERENCES

=====

For more information about security, search for "security: object ownership" then "Viewing and Changing Ownership" using the Microsoft Access Help menu. Or, search for "security: basics" then "Securing a Database."

Additional reference words: 2.00 Security

KBCategory: kbusage

KBSubcategory: ScrtOthr

## **PRB: Cannot Delete Attached Table with Relationship**

Article ID: Q119481

---

The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

If you try to delete an attached dBASE, FoxPro, Paradox, or Btrieve table, you may receive the following error message even if no relationship appears to exist between the attached table and any other tables in your database:

Can't delete table <tablename>. It is participating in one or more relationships.

### CAUSE

=====

At one time, a non-enforceable relationship was established between the attached table and another table in your database. This attachment is no longer valid (for example, the file containing the attached table has been moved or deleted).

### RESOLUTION

=====

If the file containing the attached table was moved, use the Attachment Manager to refresh the relationship. After you refresh the relationship, remove the relationship in the Relationships window and then delete the attached table.

If the file containing the attached table was deleted, you cannot refresh and then remove the relationship. Instead, you can create an Access Basic procedure to remove any relationship in which the attached table is participating and then delete the attachment. The advantage of this approach is that you can delete the relationship even though no relationship appears in the Relationships window.

### MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

The following example demonstrates how to create and use an Access Basic procedure to delete an attached table.

NOTE: In the following sample code, an underscore ( \_ ) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Create a new module and enter the following line in the Declarations section:

```
Option Explicit
```

2. Create the following procedure in the module:

```
Function DeleteAttachment()

Dim MyDB As Database, TableName As String, I As Integer
On Error GoTo DeleteErr
Set MyDB = DBEngine.WorkSpaces(0).Databases(0)

' Prompt for the name of the attached table.
TableName = InputBox$("Enter the name of the attached_
 table you want to delete.")

' First, determine if the table is an attached table. If so,
' examine all relationships in the database.
If Not ((MyDB.TableDefs(TableName).Attributes And_
 DB_ATTACHEDTABLE) = 0) Then

 For I = 0 To MyDB.Relations.Count - 1
 ' If the attached table is part of the current
 ' relationship, delete the relationship.
 If (MyDB.Relations(I).Table = TableName) Or_
 (MyDB.Relations(I).ForeignTable = TableName) Then
 MyDB.Relations.Delete MyDB.Relations(I).Name
 End If
 Next I

 ' After all relationships involving this table have been
 ' deleted, delete the attached table.
 MyDB.TableDefs.Delete TableName
End If

DeleteExit:
 Exit Function

DeleteErr:
 If Err = 3265 Then
 MsgBox "There is no table named " & TableName
 Else
 MsgBox "An unexpected error (" & Err & ") occurred: "_
 & Error(Err)
 End If
 Resume DeleteExit

End Function
```

3. From the View menu, choose Immediate Window.

4. In the Immediate window, type the following line and then press ENTER:

```
? DeleteAttachment()
```

NOTE: After you delete a table using this method, you may need to refresh

the Database window. To do so, choose the Query button in the Database window, then choose the Table button.

Additional reference words: 2.00

KBCategory: kb3rdparty kberrmsg

KBSubcategory: IsmXbase

**PRB: Cannot Display More Than Four Graph Objects at Once**  
**Article ID: Q114729**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

Attempting to open a form or preview a report that contains more than four Microsoft Graph version 5.0 objects results in the error message:

Not enough memory to complete the operation on the OLE object.

NOTE: You may have been able to open the form or preview the report in Microsoft Access version 1.x without any error messages.

CAUSE

=====

In Microsoft Access version 2.0, displaying each unbound Graph 5.0 object requires approximately 10% of system resources. This limits the number of graphs that can be displayed at one time to three or four (the actual limit is determined by the resources available in a particular system).

Activating a graph object in-place takes approximately 15% of the available system resources. For performance reasons, once a graph is activated in-place, the instance of Graph 5.0 that was started to support the editing remains in memory until the form is closed. This limits the number of graphs that can be activated to three.

RESOLUTION

=====

Limit the number of graphs on a single page to four. Reduce that number to three in forms where in-place activation of the graphs will take place.

If the graphs are not based on data that will change, convert them to pictures, or store them in tables. More than four bitmaps or bound graphs can be displayed on a single page. If multiple graphs overlap on a form or report but only one is shown at a time, consider using only one graph object and setting its row source with code each time a different set of data is shown.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Create a new report.
2. Add a Graph 5.0 object to the report.

3. Copy and paste the Graph 5.0 object to create a total of six graphs.
4. Preview the report. The error message stated above will occur. If you choose OK, the report will be displayed, but one or more of the graph objects will be blank.

#### Workaround

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

The following example demonstrates how to modify the Sales By Product form in the sample database NWIND.MDB so that you can choose between a graph showing sales by month for 1992 and a graph showing sales by month for 1993:

1. Open the Sales By Product form in Design view.
2. Place a command button on the form. Change the button's Caption property to "1992" (without quotation marks).
3. Create an event procedure for the button's OnClick property by clicking in the OnClick property field, and then choosing the Build button to the right of the field. Select Code Builder, and then add the following code.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Me!Embedded13.RowSource = "SELECT DISTINCTROW_
Format([Order Date], ""MMM \ 'YY"") AS [Sales for 1992], _
SUM([Sales by Product].[Product Amount]) AS [Product Amount]_
FROM [Sales by Product] WHERE [Order Date] BETWEEN _
#1/1/92# AND #12/31/92# GROUP BY Year([Order Date])*12_
+Month([Order Date])-1, Format([Order Date], ""MMM \ 'YY"");"
```

HINT: This query was created by copying the RowSource property of the graph on the Sales By Product form. The year in the criterion and chart title was changed and extra quotation marks were added to the second argument of the Format() function.

4. Choose the Compile Loaded Modules button on the toolbar, and then close the module.
5. Add another command button to the form. Set the new command button's Caption property to "1993" (without quotation marks).
6. Create the following event procedure for the button's OnClick property.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
Me!Embedded13.RowSource = "SELECT DISTINCTROW_
```

```
Format([Order Date], ""MMM \'YY"") AS [Sales for 1993], _
SUM([Sales by Product].[Product Amount]) AS [Product Amount]_
FROM [Sales by Product] WHERE [Order Date] BETWEEN _
#1/1/93# AND #12/31/93# GROUP BY Year([Order Date])*12_
+Month([Order Date])-1, Format([Order Date], ""MMM \'YY"");"
```

7. Choose the Compile Loaded Modules button on the toolbar and then close the module.
8. View the form in Form view. Choose the 1992 button to change the graph's row source so that it displays sales for 1992.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 19, "Using Pictures, Graphs, and Other Objects," pages 478-483

Additional reference words: 2.00 chart memory

KBCategory: kbtool

KBSubcategory: IntpGrph



**PRB: Cannot Enforce Referential Integrity in Relationship**  
**Article ID: Q112111**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

You cannot enforce referential integrity when you are defining a relationship between tables.

CAUSE

=====

There are several reasons why you might not be able to enforce referential integrity when you define a relationship, including the following:

- Existing data in the tables violates referential integrity. For example, you have a child record that does not have a matching parent in the primary table.
- You can define a relationship with an attached table, but Microsoft Access will not enforce referential integrity between two tables unless both tables are in the same database and the database user has permissions to create the relationship in that database.
- You started dragging from the wrong table or field. You must drag the primary key field (or uniquely indexed field) from the primary table to the related table. For example, to enforce referential integrity between the Customers and Orders tables in the sample database NWIND.MDB, you must drag the Customer ID field (primary key) from the Customers table to the Customer ID field in the Orders table. If you did not start dragging from the primary table, choose Cancel in the Relationships dialog box and start again.
- You chose the correct fields, but they are not appropriately indexed. Before adding relationships, open the primary table in Design view and set a primary key or create a unique index for the fields you want to use in an enforced relationship.
- Although you can create relationships using queries as well as tables, referential integrity is not enforced with queries.

MORE INFORMATION

=====

Referential integrity is a set of rules that preserves the defined relationships between tables when you enter or delete records. If you enforce referential integrity, you cannot add a record to a related table when there is no associated record in the primary table, change a value in a primary table that would result in an orphan record in a related table, or delete a record from a primary table when there is a matching related record.

If you select the Cascade Update Related Fields or Cascade Delete Related Records option for a relationship, Microsoft Access will change or delete related records to ensure that the referential integrity rules are enforced when you change or delete records.

#### REFERENCES

=====

For more information about referential integrity, search for "referential integrity" then "Defining Relationships Between Tables" using the Microsoft Access Help menu.

Additional reference words: 2.00 tables ri relationships

KBCategory: kbusage

KBSubcategory: RltRef

**PRB: Cannot Open Multiple Instances of the Same Form**  
**Article ID: Q101312**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you try to open a second instance of an open form with an OpenForm macro action, the instance of the form that is already open is activated.

CAUSE

=====

You can have only one copy of a form open at a time.

RESOLUTION

=====

NOTE: This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

The following information describes how to create and use an Access Basic function called OpenFormInstance(), which allows you to open more than one copy of the same form in Microsoft Access. To make OpenFormInstance work, you must first make <x> number of copies of your original form, giving each copy a numeric suffix greater than zero. That is, if your form is called "Customers," make copies of the form renaming them to "Customers1," "Customers2," "Customers3," and so forth.

When OpenFormInstance() is invoked with "Customers," it checks to see if a copy of the form is available for display. It returns the following error message if all instances are currently open:

```
Could not open form "Customers"
Please close another instance and try again.
```

To use the OpenFormInstance() function, create a new module with the following Access Basic functions:

NOTE: In the following sample code, an underscore ( \_ ) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

```
'*****
' DECLARATION SECTION
'*****
Option Explicit
```

```

'*****
' FUNCTION: OpenFormInstance()
'
' PURPOSE:
' Allows you to open more than one copy of the same form
' on screen at a time.
'
' PREREQUISITES:
' You must make X number of copies of the the
' form, renaming them with a numeric suffix greater than zero.
' Example:
'
' "Customers1," "Customers2," "Customers3," ...
'
' REQUIRES:
' Function IsLoaded() - Determines if form is loaded.
' Function GetFormNames() - Supplied in this module.
'
' ARGUMENTS:
' FormName - The generic name of the form to open.
' Example:
'
' "Customers"
'
' RETURNS:
' True - if an instance of FormName was opened.
' False - if no instance of FormName was found or all available
' instances are active.
'
' NOTES:
' You can customize the OpenForm action in the code below to add
' additional (optional) parameters, such as a Where condition or
' or alternate view.
'
'*****
Function OpenFormInstance (ByVal FormName As String)
 Dim Count, i, Msg As String, InstanceCount
 ReDim Names(0) As String

 ' Get a list of all the forms in the current database.
 Count = GetFormNames(Names())

 ' Loop through the list of forms for a match with
 ' .. the requested form.
 For i = 0 To Count - 1
 ' Is the name of the form, minus the suffix, the same?
 If FormName = Left(Names(i), Len(FormName)) Then

 ' .. is the suffix a number greater than zero?
 If Val(Mid(Names(i), Len(FormName) + 1)) > 0 Then

 ' Count the number of instances.
 InstanceCount = InstanceCount + 1

 ' If the form is NOT loaded,load it.
 If Not IsLoaded(Names(i)) Then
 DoCmd OpenForm Names(i)
 End If
 End If
 End If
 Next i
End Function

```

```

 OpenFormInstance = True
 Exit Function
 End If
End If
End If
Next i

' No form was found or all instances are being used.
OpenFormInstance = False

' If no instance was found, just return.
If InstanceCount = 0 Then Exit Function

' If all instances are being used, display error message.
Msg = "Couldn't open form "" & FormName & ""."
Msg = Msg & Chr$(13) & Chr$(13)
Msg = Msg & "Please close another instance and try again."
MsgBox Msg, 48

End Function

'*****
' FUNCTION: GetFormNames()
'
' PURPOSE:
' Fills the string array with a list of forms names.
'
' ARGUMENTS:
' Names - A single dimensional array of type string.
'
' RETURN:
' The number(zero based) of names stored in the Names array.
'
' NOTES:
' This function uses information stored in the
' MySysObject table of the currently opened database.
' The system tables are undocumented and are subject to
' change in future versions of Microsoft Access.
'*****
Function GetFormNames (Names() As String)
 Dim db As Database, ss As Snapshot
 Dim Count, SQL
 Dim Msg As String

 SQL = "Select Name,Type from MSysObjects Where Type="
 SQL = SQL & "-32768 And Left(Name,1)<>'~' Order By Name;"
 Set db = CurrentDB()
 Set ss = db.CreateSnapshot(SQL)

 ss.MoveLast
 If ss.RecordCount > 0 Then
 ReDim Names(0 To ss.RecordCount - 1)
 Else
 GetFormNames = 0
 Exit Function
 End If

```

```

 ss.MoveFirst
 Count = 0
 Do While Not ss.EOF
 Names(Count) = ss![name]
 Count = Count + 1
 ss.MoveNext
 Loop

 GetFormNames = ss.RecordCount
End Function

```

#### How to Use the OpenFormInstance() Function

---

The following steps assume there is a need to see more than one customer form on the screen at a given time in the sample database NWIND.MDB:

1. Open NWIND.MDB and choose the Form button in the Database window.
2. Click, 1 time, on the Customers form, and choose Copy from the Edit menu.
3. Choose Paste from the Edit menu. Type "Customers1" (without the quotation marks) in the Form Name box and then press ENTER.
4. Repeat step 3 for however many instances you expect your users to need ("Customers2," "Customers3," and so on).
5. Open the module containing the OpenFormInstance() function in Design view, and then choose Immediate Window from the View menu.
6. Type the following and press ENTER:

```
? OpenFormInstance("Customers")
```

NOTE: The first instance of the Customers form opens.

7. Repeat step 6.

NOTE: The second instance of the Customers form opens.

8. Repeat step 6 until you receive an error message indicating that no more instances are available.

#### Problems to Consider When Using OpenFormInstance()

---

Since you are not actually opening the same form multiple times (you are opening copies), it is important that any macros and module code that make direct references to the form using "Forms!FormName" references be reconsidered.

If you have more than one instance of the form opened, it is best to use the following techniques when you reference controls on the form.

- User Screen.ActiveForm form notation wherever possible.

```
MsgBox Screen.ActiveForm![ControlName]
```

- When Screen.Active form is not appropriate, pass the name of the form to the Access Basic function to call.

```
Function MyFunction(MyForm)
 MsgBox Forms(MyForm)![ControlName]
End Function
```

- Since you cannot pass arguments to macros, consider reworking the macro as an Access Basic function. You could call the function above, MyFunction(), using a RunCode macro action as follows:

```
RunCode
 Function Name: =MyFunction(Screen.ActiveForm.FormName)
```

STATUS  
=====

This behavior is by design.

REFERENCES  
=====

Microsoft Access, "Introduction to Programming," version 1.1, Chapter 7, "Forms and Report Variables"

Microsoft Access, "Building Applications," version 2.0, Chapter 4, "Using Forms to Collect, Display, and Filter Information"

Additional reference words: 1.00 1.10 2.00 open twice  
KBCategory: kbusage  
KBSubcategory: FmsOthr

## PRB: Cannot Open Paradox 4.x Table

Article ID: Q114082

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you try to import or attach a Paradox version 4.x file, you receive the following error message:

Can't open this Paradox 4.x table because ParadoxNetStyle is set to 3.x in msacc20.ini.

### CAUSE

=====

The ParadoxNetStyle entry in the [Paradox ISAM] section of your MSACC20.INI file is set to 3.x. When this entry is set to 3.x, you can only import or attach Paradox version 3.x files.

### RESOLUTION

=====

Set the ParadoxNetStyle setting to 4.x instead of 3.x. When this setting is set to 4.x, you can import or attach Paradox version 3.x or 4.x files.

Note that all the users of a shared database must have the same ParadoxNetStyle settings. If users of a shared database need to access both Paradox 3.x and 4.x files, all the users should use the 4.x ParadoxNetStyle setting.

The [Paradox ISAM] section of your MSACC20.INI file should contain:

```
[Paradox ISAM]
ParadoxNetStyle=4.x
```

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

NOTE: For this example to work, you must have the ParadoxNetStyle setting set to 3.x in the [Paradox ISAM] section of your MSACC20.INI file.

1. Start Microsoft Access version 2.0 and open a database.
2. From the File menu, choose Import.
3. In the Data Source box, select Paradox 4.x, and then choose OK.
4. In the File Name box, select the Paradox 4.x file you want to import.



5. Choose the Import button.

REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 9, "Importing, Exporting, and Attaching," pages 159-194

For more information about importing or attaching files, search for "Import" or "Attach Table" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmPdox

**PRB: Cannot Perform OpenTable Method on an Attached Table**  
**Article ID: Q94036**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

If you try to use an OpenTable method on an attached table, you will receive one of the following error messages:

Can't perform operation; it is illegal.

-or-

Invalid Operation.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Attach any valid table.
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
'table's 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 execute the sub. You will receive one of the

error messages stated above.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg kberrmsg

KBSubcategory: PgmObj

**PRB: Cannot Process Close Action in Report's OnOpen Property**  
**Article ID: Q94926**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You receive the following error message when you attempt to process a Close action in a macro that is assigned to a report's OnOpen property:

Can't run this action while processing a report event (in Microsoft Access version 1.x)

-or-

Can't run this action while processing a form or report event (in Microsoft Access version 2.0)

RESOLUTION

=====

Do not use a Close action in a macro that is assigned to a report's OnOpen property.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Create the following new macro called CloseForm:

| Macro Name | Action |
|------------|--------|
|------------|--------|

-----

|           |       |
|-----------|-------|
| CloseForm | Beep  |
|           | Close |

CloseForm Actions

-----

Close

Object Type: Form

Object Name: Form1

2. Create the following new macro called CloseReport:

| Macro Name | Action |
|------------|--------|
|------------|--------|

-----

CloseReport    Beep  
                  Close

CloseReport Actions

-----  
Close  
  Object Type: Report  
  Object Name: Report1

3. Create a new form. Assign the CloseForm macro to the form's OnOpen property. Save the form as Form1.
4. View the form in Form view.
5. Create a new report. Assign the CloseReport macro to the report's OnOpen property. Save the report as Report1.
6. Preview the report. Your computer will beep and you will receive one of the error messages stated above. Note that you did not receive an error message in step 4, when you opened the form.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: RptProp

## **PRB: Cannot Run Setup for Solutions Pack on Workstation**

**Article ID: Q114878**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Solutions Pack version 1.0
- 

### SYMPTOMS

=====

On a computer running a network installation of Microsoft Access version 2.0, running Setup for the Microsoft Access Solutions Pack version 1.0 results in the following message:

Setup could not find Microsoft Access version 2.0 installed locally on your system. Please install Microsoft Access version 2.0 before installing this application.

### CAUSE

=====

The Microsoft Access Solutions Pack requires that Microsoft Access be installed on the local computer and will not accept a network location for the Microsoft Access directory.

The same message occurs if the Solutions Pack setup cannot find MSACC20.INI.

### RESOLUTION

=====

If the MSACC20.INI file is missing, either recreate the MSACC20.INI file manually or reinstall Access 2.0 to recreate it. Then run the Solutions Pack setup again.

Otherwise, either install Microsoft Access 2.0 locally on the computer, or use the following steps to force Setup to believe that Microsoft Access is installed locally.

NOTE: The technique documented in this article does not work if you are running on a computer that does not have a local hard drive. Also, this technique does not work for the ADT. Access MUST be installed on a local hard disk to install the ADT and run the Setup wizard.

1. On the local hard drive, create a directory called ACC20 that you will cause Setup to believe is the local Microsoft Access directory.
2. In the directory that you created in step 1, create a file called UTILITY.MDA. You can create the empty file with any text editor (such as Windows Notepad).
3. In your local Windows directory, rename the MSACC20.INI file to MSACC20.OLD.

4. Using any text editor (such as Windows Notepad) create a file containing the following lines:

```
[Options]
UtilityDB=C:\ACC20\UTILITY.MDA
```

Save the file in your local Windows directory as MSACC20.INI. You should now be able to run Setup successfully for the Microsoft Access Solutions Pack.

5. Once Setup is finished, delete the MSACC20.INI file in your local Windows directory. Rename the MSACC20.OLD file that you created in step 3 to MSACC20.INI.

Additional reference words: 2.00 solpack solution  
KBCategory: kbsetup kberrmsg  
KBSubcategory: StpOthr

**PRB: Cannot Search for Name with Apostrophe in Find Customers**  
**Article ID: Q121260**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you are using the Find Customers form in the sample database SOLUTION.MDB, you cannot search for a customer whose name contains an apostrophe ('). For example, if you try to search for a customer with the name "B's Foods," you receive the following error message:

Syntax error in query expression '[Company Name] Like 'B's Foods\*''

RESOLUTION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

The following steps demonstrate how to modify the AddToWhere Sub procedure so that you can search for customer names containing apostrophes.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Quit Microsoft Access if it is running.
2. Copy the SOLUTION.MDB file to a file called MYSOLUT.MDB.
3. Start Microsoft Access and open the MYSOLUT.MDB database.
4. In the Select A Category Of Examples box, select Sample Forms. In the Select An Example box, select Find Customers. Choose OK.
5. View the form in Design view.
6. From the View menu, choose Code.
7. In the Procedure box on the toolbar, select AddToWhere.
8. Place an apostrophe (') at the beginning of the line that reads:

```
Mycriteria = (Mycriteria & FieldName & " Like " & Chr(39) & _
Fieldvalue & Chr(42) & Chr(39))
```

Placing an apostrophe at the beginning of the line comments the line out.



9. Enter the following code just below the line that you commented out in step 8:

```
If InStr(1, fieldvalue, "'") Then
 Mycriteria = (Mycriteria & FieldName & " Like " & Chr(39) &
 Left(Fieldvalue, InStr(1, Fieldvalue, "'") - 1) & "' " & _
 Mid(Fieldvalue, InStr(1, Fieldvalue, "'")) & Chr(42) & Chr(39))
Else
 Mycriteria = (Mycriteria & FieldName & " Like " & Chr(39) & _
 Fieldvalue & Chr(42) & Chr(39))
End If
```

10. Close the module and then view the form in Form view.
11. Search for a company whose name contains an apostrophe.
12. From the File menu, choose Save Form.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsEvt

**PRB: Cannot Share Some Microsoft Access Wizards**

Article ID: Q121662

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

If you try to run the Table Wizard, the Combo Box Wizard, the List Box Wizard, the Option Group Wizard, the Field Builder, or the Database Documentor add-in while someone else is already using it, you will receive one of the following error messages:

- For the Database Documentor add-in or the Print Definitions command:

Another user or another instance of Microsoft Access is currently Printing Definitions or using the Database Documentor.

- For the Table Wizard or Field Builder:

Another user or another instance of Microsoft Access is currently using the Table Wizard or the Field Builder.

- For the Combo Box Wizard or the List Box Wizard:

Unable to launch the Wizard. Make sure that there is only one instance of the List Box or Combo Box Wizard running.

- For the Option Group Wizard:

Another user or another instance of Microsoft Access is using the Option Group Wizard.

CAUSE

=====

When you install Microsoft Access on a network server, Setup installs the Wizard library databases in the ACCESS\WORKDIR subdirectory on the server, making the library databases available to any user with permissions for that subdirectory. The Wizard library databases are databases that, like any other database, contain modules, tables, forms, and so on. When two or more users open the same Wizard at the same time, those users are sharing that library database. The Wizards, builder, and add-in listed in the "Symptoms" section above require exclusive access to tables in the library database, and cannot therefore be run by more than one person at a time.

The library databases that exhibit this behavior are:

| Wizard, Builder, or Add-in | File name   |
|----------------------------|-------------|
| Table Wizard               | WZTABLE.MDA |
| Field Builder              | WZTABLE.MDA |
| Database Documentor        | WIZLIB.MDA  |

|                     |            |
|---------------------|------------|
| Combo Box Wizard    | WZBLDR.MDA |
| List Box Wizard     | WZBLDR.MDA |
| Option Group Wizard | WZBLDR.MDA |

## RESOLUTION

=====

Even though this behavior is by design, there is a way to work around it. To avoid receiving the error messages stated above, open a different copy of the .MDA file. You can do this by copying the .MDA files either to your workstation's local drive or to a personal directory on the server, and then modifying your MSACC20.INI file to point to the new location for the files.

Note that copying the .MDA files to a local drive uses less disk space on the server. If several users each copy the .MDA files to a personal directory on the server, the library database files could use up much of the available disk space on the server.

To copy the .MDA files to your workstation's local drive, follow these steps:

1. Assuming that the server that Microsoft Access is installed on is connected to your workstation as drive F, and that Microsoft Access is installed on your workstation in the ACCESS directory on drive C, type the following lines at an MS-DOS command prompt to copy the .MDA files:

```
copy f:\access\workdir\wztable.mda c:\access
copy f:\access\workdir\wzblldr.mda c:\access
copy f:\access\workdir\wzlib.mda c:\access
```

2. Open your MSACC20.INI file in any text editor (such as Windows Notepad). Modify the following lines

```
[Libraries]
F:\ACCESS\WORKDIR\WZLIB.MDA=RW
F:\ACCESS\WORKDIR\WZTABLE.MDA=RW
F:\ACCESS\WORKDIR\WZBLDR.MDA=RW
F:\ACCESS\WORKDIR\WZQUERY.MDA=RW
F:\ACCESS\WORKDIR\WZFRMRPT.MDA=RW
```

to read:

```
[Libraries]
C:\ACCESS\WZLIB.MDA=RW
C:\ACCESS\WZTABLE.MDA=RW
C:\ACCESS\WZBLDR.MDA=RW
F:\ACCESS\WORKDIR\WZQUERY.MDA=RW
F:\ACCESS\WORKDIR\WZFRMRPT.MDA=RW
```

3. Save and then close the MSACC20.INI file.

## REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 15, "Using

Library Databases and Dynamic-Link Libraries," pages 345-376

Additional reference words: 2.00 multiuser

KBCategory: kbusage kberrmsg

KBSubcategory: Gn1Mu

## **PRB: Cannot Sum Calculated Controls in Forms or Reports**

**Article ID: Q113354**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

"#Name?" is displayed in a text box in a form footer, or a parameter box appears requesting the field specified in the sum expression in a report.

### CAUSE

=====

The text box is trying to perform a sum on a calculated control in the detail section of the form or report.

You cannot sum a calculated control on a form or report because the calculation is not stored anywhere for use in a later computation. The calculation is computed and then output to the screen for display only.

### RESOLUTION

=====

There are two solutions to this problem.

1. Repeat the calculation being computed in the calculated control in the sum expression. If the calculated control on the form or report has the expression

```
=[Unit Price] * [Quantity]
```

repeat this calculation in the sum expression as:

```
=SUM([Unit Price] * [Quantity])
```

2. Base the form or report on a query and calculate the expression as a calculated field in the query rather than as a calculated control on the form or report. Since the calculation is being computed in the query, the result will be available for other computations. This method is considerably faster than the first method as the computation does not have to be done twice. If the calculated control on the form or report has the expression

```
=[Unit Price] * [Quantity]
```

then the new calculated field in the query will have the following in the Field row of the query grid:

```
Extended Price: [Unit Price] * [Quantity]
```

If the form or report is based on a query with this calculated field, Extended Price will appear in the field list and should be used in place

of the calculated control on the form or report. The sum expression in the form or report footer would look like the following:

```
=SUM([Extended Price])
```

#### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Open the form Orders Subform in Design view.
3. Place a text box with the following properties in the detail section:

```
ControlName: My Extended Price
ControlSource: =[Unit Price] * [Quantity]
```

4. Alter the Order Subtotal text box ControlSource property from

```
=Sum([Extended Price])
```

to:

```
=Sum([My Extended Price])
```

NOTE: The original expression will correctly display the sum of the Extended Price field since this field is being computed as a calculated field in the Order Details2 query the form is based on. The Extended Price calculated field in the Order Details2 query looks like the following:

```
Extended Price: CLng([Order Details].[Unit Price]*[Quantity]*(1-[Discount])*100)/100
```

5. Change the form's ViewsAllowed property from Datasheet to Both.
6. View the form in Form view. "#Name?" will appear in the Order Subtotal box in the form footer.

#### Work Around 1: Repeating the Calculation in the Sum Expression

-----

NOTE: These steps continue from the "Steps to Reproduce Behavior" section above.

Alter the Order Subtotal ControlSource property from

```
=Sum([My Extended Price])
```

to:

```
=Sum([Unit Price] * [Quantity])
```

Note that the calculation correctly displays in the Order Subtotal box in

the form footer.

## Work Around 2: Repeating the Calculation in the Sum Expression

---

1. Open the Order Details2 query in Design view.
2. In an empty query grid column, type the following in the Field row:  

My Extended Price: [Order Details].[Unit Price]\*[Quantity]
3. Remove the My Extended Price calculated control in the detail section.

The calculation correctly displays in the Order Subtotal box in the form footer.

## REFERENCES

---

For more information, search for "totals" then "Calculating Totals in Forms" using the Microsoft Access Help menu.

Microsoft Access "User's Guide," version 1.0, Chapter 11, "Using Expressions in Forms," pages 295-298

Microsoft Access "User's Guide," version 1.1, Chapter 11, "Using Expressions in Forms," pages 299-302

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsSubf

**PRB: Cannot Use Counter to Determine If Record Is New Record**  
**Article ID: Q112109**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

The status of the counter field cannot be used to determine if the current record is the new record.

CAUSE

=====

In Microsoft Access version 1.x, the counter field was null until the record was saved (that is, you moved to the next record, or chose Save Record from the File menu). Therefore, if you checked the counter field (using the IsNull function) and it was null, then the record being edited was a new one. If not, then it was an existing record. The usual method was to put the following expression in the form's BeforeUpdate property:

```
IsNull([<counterfieldname>])
```

In Microsoft Access version 2.0, however, the counter field is updated as soon as you begin inserting a new record, which invalidates the method described above.

RESOLUTION

=====

To determine if the current record is a new record using a counter field in Microsoft Access version 2.0, you can check the OldValue property. OldValue will return null for a new record.

To do this, use the following expression in the form's BeforeUpdate property:

```
IsNull([<counterfieldname>].OldValue)
```

For another technique that does not require the use of a counter field, query on the following words here in the Microsoft Knowledge Base:

Determine and New and Record and Bookmark

REFERENCES

=====

For more information about the OldValue property, search for "OldValue" then "OldValue Property" using the Microsoft Access Help menu.

Additional reference words: 2.00 Forms

KBCategory: kbusage

KBSubcategory: FmsHowto





**PRB: Cannot Use SQL Pass-Through Command for Subreport**  
**Article ID: Q111780**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

You receive the error message

Can't use a subform or subreport bound to a crosstab query  
in which the column headings can vary

when you try to use a SQL-specific pass-through query as the RecordSource  
for a subreport.

CAUSE

=====

Reports require a list of fields to create a subreport, and a list of  
fields is not stored in a SQL pass-through query.

STATUS

=====

This behavior is by design.

REFERENCES

=====

For more information on SQL pass-through commands, search for "Pass-  
Through" then "Pass-Through Command" using the Microsoft Access Help menu.

Additional reference words: 2.00 Queries

KBCategory: kbusage

KBSubcategory: RptOthr

**PRB: Change Option Affects Only One Instance of MS Access**  
**Article ID: Q90986**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When two instances of Microsoft Access are run on the same computer, 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 computer, only the routines in the instance where the preferences are changed are called. The other instance does not have its memory copies updated.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Many options are stored in memory so that they do not have to be read from the preferences table every time they are accessed. The mechanism by which the memory and disk copies are kept in synchronization is the array of routines that is called whenever a preference is changed.

When there are two instances of Microsoft Access running, only the routines in the instance changing preferences are called. The other instance does not have its memory copies updated, but the View options shows them as being changed.

In order to make the preference take effect in another instance on the same system, the user either needs to shutdown and reopen this instance, or change the preference and resave it.

This is not an issue for users on separate systems, since their preferences are stored per user.

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, chapter 25, "Administering a Database System"

Microsoft Access "User's Guide," version 1.0, Appendix D, "Setting Up

Microsoft Access on a Network"

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: Gn1Glob1

**PRB: Changing Control Properties Affects All Records in a Form**  
**Article ID: Q119992**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Changing a control's property setting (such as the BackColor property setting) in a continuous form causes all instances of that control in all records to be changed.

RESOLUTION

=====

You may be able to use the Format property to alter the color of numeric data in a text box based on the data's value (positive, negative, zero, or null), instead of changing the control's property setting.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Microsoft Access stores a single set of property settings for each control on a form. In order for a control to appear differently from record to record on a form, Microsoft Access would have to store and maintain property settings for each control for every record in the form.

If a control has approximately 50 property settings, a form with 1000 records would need to maintain 50,000 property settings. Maintaining this much information for controls would severely impact performance.

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create the following new macro and then save the macro as Highlight:

| Macro Name | Action   | Description                |
|------------|----------|----------------------------|
| On         | SetValue | Change BackColor to Yellow |
| Off        | SetValue | Change BackColor to White  |

Highlight Actions

-----  
SetValue  
Item: [Screen].[ActiveControl].[BackColor]  
Expression: 8454143

SetValue

Item: [Screen].[ActiveControl].[BackColor]  
Expression: 16777215

3. Create a new form based on the Customers table. Add three or four fields from the table to the form. Make sure the form's DefaultView property is set to Continuous.

4. Set the following properties for every control on the form:

OnEnter: Highlight.On  
OnExit: Highlight.Off

5. Save the form as Test.

6. View the form in Form view. Use the TAB key to move among the controls on the form. Note that as you move from control to control, the BackColor property for all controls, not just the active control, is changed.

#### REFERENCES

=====

For more information about using color in the Format property, search for "Format" then "Format Property" using the Microsoft Access version 2.0 Help menu, then select the "Number and Currency Data Types" topic.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsProp

**PRB: Changing Printer Port Resets Printer Setup Information**  
**Article ID: Q121375**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The printer settings for forms and reports (such as a custom paper size or paper source) are reset if you change the printer's port in the Printers dialog box in Windows Control Panel.

CAUSE

=====

Microsoft Access saves printer port information along with the form or report. Changing the printer port causes Microsoft Access to reset the printer settings for that object.

RESOLUTION

=====

Make sure that all users are using the same port for the default printer, so that there is no need to change ports.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Start Microsoft Access, open the sample database NWIND.MDB, and then open the Products On Order report in Design view.
2. From the File menu, choose Print Setup.
3. In the Paper Size box, select "Legal 8 1/2 x 14 in," and then choose the OK button.
4. Save and then close the report.
5. Start Windows Control Panel and then double-click the Printers icon.
6. In the Printers dialog box, choose the Connect button. Select a different port in the Ports box, and then choose OK. Choose the Close button.

7. In Microsoft Access, open the Products On Order report in Design view.
8. From the File menu, choose Print Setup. Note that the paper size has been reset to "Letter 8 1/2 x 11 in."

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprint

KBSubcategory: PtrSetup



**PRB: Choosing a New Printer Resets Some Printer Settings**  
**Article ID: Q92741**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Microsoft Access retains certain printer settings with form and report objects.

For instance, if you have a specific printer specified under the Print Setup dialog and then select a new printer your Paper and Option settings change to the default values for the new printer chosen.

CAUSE

=====

This functionality is controlled by the Windows Common Dialog DLL (Dynamically Linked Library.) The same behavior appears within Microsoft Write.

STATUS

=====

This behavior is by design.

Additional reference words: 1.00 1.10 2.00 defaults

KBCategory: kbprint

KBSubcategory: PtrOthr

**PRB: Closing Forms, Tables, or Queries Is Slow**  
**Article ID: Q109398**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SYMPTOMS

=====

Closing forms, tables, or queries based on attached Sybase SQL Server tables can take a long time.

CAUSE

=====

When you close the table, Microsoft Access issues an ODBC API SQLCancel() or SQLFreeStmt() call. This tells the server to cancel processing of the current query. However, some network libraries are not able to cancel a query request in an efficient manner. Instead, they continue to read the query results from the server until the process is complete, discarding the results.

RESOLUTION

=====

If you are using standard IPX, append the string "URGENT" (without the quotation marks) to the end of the SQUERY or the <MYDATASOURCE> line in the WIN.INI file. Substitute the name of the SQL Server for <MYDATASOURCE>. This tells the network library that your SQL Server supports in-band attention signals. The new line in the WIN.INI file should look like:

```
<MYDATASOURCE> = 111.111.111.111,3801,URGENT
```

Sybase SQL Server is manufactured by Sybase Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.10 2.00

KBCategory: kbusage

KBSubcategory: ObcOthr

## **PRB: Compile Error When Running ADT Application**

**Article ID: Q119754**

-----  
The information in this article applies to:

- Microsoft Access Distribution Kit version 2.0  
-----

### SYMPTOMS

=====

When you run a run-time application that you created with the Microsoft Access Developer's Toolkit version 2.0, you receive a compile error. The same application runs correctly when you run it with the retail version of Microsoft Access.

### CAUSE

=====

This behavior can occur if you make a reference in your application's code to functions or variables defined in the WZLIB.MDA file (which cannot be legally distributed with your application), or to any other library database that is not loaded in your application.

### RESOLUTION

=====

Remove or modify the code in your application that refers to the WZLIB.MDA file, or to any other file that is not loaded.

### MORE INFORMATION

=====

The WZLIB.MDA library database contains routines and utilities used by the Microsoft Access Wizards and by other code. For example, the Reattach NWIND module in the SOLUTION.MDB database (a sample database shipped with Microsoft Access version 2.0) includes a function that refers to WZLIB.MDA.

If you include code in your application that refers to the WZLIB.MDA file, and then run your application with the retail version of Microsoft Access, the application will run correctly. However, if you run the Setup Wizard to create distribution disks for your application, and correctly modify the initialization (.INI) file for the application to remove the Wizard files from the [Libraries] section, you will receive a compile error when you run the application as a run-time application. Note that error-trapping routines that you may have included in your code will not be called, because these routines cannot trap compile errors.

To discover which section of your code is causing the compile error, run your application with the retail version of Microsoft Access, but with your application's custom .INI file. For example, to run an application called MYAPP.MDB with a custom .INI file called MYAPP.INI, create an icon with the following command line.

NOTE: This example assumes that you have Microsoft Access installed in the ACCESS directory on drive C, and that the MYAPP.MDB application is in the

MYAPP directory on drive C.

```
C:\ACCESS\MSACCESS.EXE C:\MYAPP\MYAPP.MDB /INI C:\MYAPP\MYAPP.INI
```

This command will run the MYAPP.MDB application with only the libraries specified in the [Libraries] section of the custom MYAPP.INI file. Compile your application's code at this point to locate the cause of the compile error.

NOTE: If you open a module in Design view and then choose Compile Loaded Modules from the Run menu, code in any form or report modules will not be compiled. To compile code in form and report modules, the forms and reports must be open in Design view.

#### REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0, Chapter 1, "Developing Run-Time Applications: an Overview," page 10, and Chapter 5, "Creating Wizards, Builders, and Add-ins"

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: PrgrmLib

## **PRB: Concatenation of Memo Fields Creates Text Field**

**Article ID: Q92892**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### 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

=====

#### Steps To Reproduce:

-----

1. Create a table with two memo fields:

Table: OldTable

-----

Field Name: Mem01

Data Type: Memo

Field Name: Mem02

Data Type: Memo

2. Create a make-table query based on OldTable. Use the following SQL statement for the query:

```
SELECT DISTINCTROW
 [Mem01]&[Mem02]
AS Concat
INTO NewTable
FROM OldTable;
```

After running the query, note that the NewTable table contains a text field with only 255 characters.

## Workaround

1. Create another table containing one memo field:

Table: ExistingTable

-----  
Field Name: ExistingField

Data Type: Memo

2. Create an append query based on OldTable. Use the following SQL statement for the Query:

```
INSERT INTO ExistingTable
 (ExistingField)
SELECT DISTINCTROW
 [Memo1]&[Memo2]
AS Concat FROM OldTable;
```

When you run this query, ExistingTable will contain the two memo fields concatenated.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryMktbl

**PRB: Creating Multiple New Import/Export Specifications**  
**Article ID: Q93136**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Creating multiple new Import/Export specifications using the Import/Export Setup dialog box requires that you either exit the dialog or delete or change each entry in the dialog that does not apply. The dialog does not provide a method to clear or reset each option in the dialog.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open any database or create a new one.
2. From the File menu, choose Imp/Exp Setup.
3. Enter information in several fields.
4. From the File menu, choose Save As.

If you want to create another specification at this point, you must either exit this dialog or manually delete all of the irrelevant information.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr

## **PRB: Crosstab Query Returns "<>" as a Column Heading**

**Article ID: Q119074**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

One of the column headings returned by your crosstab query is "<>" even though you do not have a field labeled "<>".

### CAUSE

=====

The field specified in the crosstab query's Column Heading column contains null values. Microsoft Access returns "<>" as the field name if any values in the Column Heading field are null.

### RESOLUTION

=====

There are two ways to prevent a Column Heading field that contains null values from returning "<>" as its field name:

- Use the ColumnHeading property to explicitly define your column heading names. (In Microsoft Access version 1.x, use the FixedColumnHeadings property.)

-or-

- Set the Column Heading field's criterion to:

Is Not Null

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create a new query and add the Orders table.
3. From the Query menu, choose Crosstab.
4. Add the following fields to the query:

Field: Employee ID  
Total: Group By  
Crosstab: Row Heading  
Field: Ship Region  
Total: Group By



Crosstab: Column Heading  
Field: Freight  
Total: Sum  
Crosstab: Value

5. Run the query. Since the Ship Region field in the Orders table contains null values, "<>" will appear as a column heading in the crosstab query's results.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, pages 283-285

Microsoft Access "User's Guide," version 1.1, pages 156-158

Microsoft Access "User's Guide," version 1.0, pages 152-154

Additional reference words: 1.00 1.10 2.00 cross tab

KBCategory: kbusage

KBSubcategory: QryCrsstb

**PRB: Data Not Sorted In Index Field Order**

**Article ID: Q114081**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you use an index on a table without a primary index, the table does not sort according to the order of the fields in the index.

CAUSE

=====

Opening a table in Datasheet view is, internally, the same as opening a recordset. If there is no primary index on the table, Microsoft Access uses any field with a unique index to sort the recordset. If none or more than one of the fields has a unique index, Microsoft Access chooses one at random. The order of the indexes shown in the Index window has no bearing on which index is used to sort the recordset.

RESOLUTION

=====

To sort a recordset on a particular field, either make that field the primary index for the table (by setting its PrimaryIndex property to Yes), or use a query to sort the data.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Start Microsoft Access and open any database.
2. Create a table with three fields. Make one of the fields a Counter field, and make the other two Text fields.
3. Do not specify a primary key for the table. Define an ascending compound index on the two text fields. For example:

| Index Name | Field Name | Sort Order |
|------------|------------|------------|
| FieldText  | Field1     | Ascending  |
|            | Field2     | Ascending  |

Set all index properties to No.

4. Save the table, and then enter the following data in the table:

| Counter | Field1 | Field2 |
|---------|--------|--------|
| 1       | zzzz   | dddd   |
| 2       | dddd   | yyyy   |
| 3       | mmmm   | iiii   |

5. Close the table, and then open it in Datasheet view. It is sorted as:

| Counter | Field1 | Field2 |
|---------|--------|--------|
| 2       | dddd   | yyyy   |
| 3       | mmmm   | iiii   |
| 1       | zzzz   | dddd   |

6. Switch to Design view and add the Counter field to the index:

| Index Name | Field Name | Sort Order |
|------------|------------|------------|
| Counter    | Counter    | Ascending  |
| FieldText  | Field1     | Ascending  |
|            | Field2     | Ascending  |

Leave all the index properties set to No.

7. Save and then close the table, then open it in Datasheet view. It is sorted as:

| Counter | Field1 | Field2 |
|---------|--------|--------|
| 2       | dddd   | yyyy   |
| 3       | mmmm   | iiii   |
| 1       | zzzz   | dddd   |

Instead of sorting first on the Counter field, and then on Field1 and Field2 as you might expect, the sorting remains unchanged.

8. Set the Counter field's UniqueIndex property to Yes. The table is sorted as:

| Counter | Field1 | Field2 |
|---------|--------|--------|
| 1       | zzzz   | dddd   |
| 2       | dddd   | yyyy   |
| 3       | mmmm   | iiii   |

Note that the data is sorted on the Counter field.

#### REFERENCES =====

Microsoft Access "User's Guide," version 2.0, Chapter 8, "Changing and Customizing Tables," pages 151-154

Additional reference words: 2.00 Indexes Indexing Sorting Ordering  
KBCategory: kbusage  
KBSubcategory: RptSort

## **PRB: Data Outline Control Events Not Triggered**

**Article ID: Q122714**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

### SYMPTOMS

=====

Code assigned to a data outline control's event (such as the DoRowClick, DoRowDbClick, or DoKeyPress event) does not run when the event occurs.

### CAUSE

=====

The data outline control's SendMouseEvents and SendKeyboardEvents properties are not set to True. The default for both properties is False.

### RESOLUTION

=====

You must set the data outline control's SendMouseEvents and SendKeyboardEvents properties to True to enable the firing of mouse and keyboard events by the data outline control. To set the SendMouseEvents and SendKeyboardEvents properties to True, follow these steps:

1. Open in Design view the form containing the data outline control.
2. Click the data outline control with the right mouse button.
3. From the shortcut menu, choose Data Outline Control Object, and then choose Properties from the menu that appears.
4. Select the SendMouseEvents and SendKeyboardEvents check boxes, and then choose OK.

Note that you need only select the appropriate property for the type of events you want the data outline control to process. For example, if you want the data outline control to process only mouse events, select only the SendMouseEvents check box.

### REFERENCES

=====

For more information about the SendMouseEvents property, search for "SendMouseEvents" then "SendMouseEvents Property" using the OUTL1016.HLP Help file shipped with the Microsoft Access Developer's Toolkit.

For more information about the SendKeyboardEvents property, search for "SendKeyboardEvents" then "SendKeyboardEvents Property" using the OUTL1016.HLP Help file shipped with the Microsoft Access Developer's Toolkit.

Additional reference words: 2.00  
KBCategory: kbinterop  
KSubcategory: IntpCstm

## **PRB: Data Type Error Inserting Data Outline Control**

Article ID: Q120481

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Developer's Toolkit version 2.0
- 

### SYMPTOMS

=====

You receive the following error message when you are defining relationships with the data outline control:

One of the specified RecordSources is invalid

### CAUSE

=====

The fields for which you are trying to define a relationship have different data types.

### RESOLUTION

=====

Make sure to define relationships only for fields that have the same data type.

### MORE INFORMATION

=====

The data outline control is included with the Microsoft Access Developer's Toolkit (ADT). You can use this control to define relationships between fields so that you can expand or collapse data linked on those fields. The Data Outline Control Wizard does not prevent you from defining relationships for fields with different data types, but when you try to expand such a level in the data outline control, you will receive an error message.

### Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create a blank, new, unbound form. Make sure that the Control Wizards button in the toolbox is selected (the button should appear sunken).
3. From the Edit menu, choose Insert Object. In the Insert Object dialog box, select the Insert Control option button, then select Data Outline Control in the Control Type box. Choose OK.
4. In the "For each level, what data do you want the Data Outline Control to display?" screen, select the Categories table in the Table/Query box for level 1, and then choose the Next Level button.

5. For level 2, select the Orders table in the Table/Query box, and then choose OK when you are prompted "You must define a relationship between the data on this level and the previous level."
6. Choose the Define Relationship button. In the Categories box, select Category ID. In the Orders box, select Freight. Choose OK.
7. Choose the Finish button.
8. View the form in Form view.
9. Expand any of the levels in the control. You will receive the error message stated above.

#### REFERENCES

=====

For more information about the data outline control, see the Overview topic in the Data Outline Control Help file that is shipped with the ADT.

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: GnlApp



**PRB: DateAdd() "w" Interval Does Not Work As Expected**  
**Article ID: Q115489**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you add days to a date using the "w" interval, the DateAdd() function adds the total number of days you specified to the date, instead of adding the appropriate number of workdays to the date as you expect.

CAUSE

=====

The intervals in the DateAdd() function are the same as those used in the DatePart() function. The DatePart() function uses the "w" interval to return the weekday, or day of the week, from a date. The value returned from the DatePart() function is an integer from 1 to 7 representing the days of the week (Sunday through Saturday). When you add days using the "w" interval, weekdays (which include all the days of the week) are added.

To add workdays (Monday through Friday) to a date, use the sample function described in the "Resolution" section below.

RESOLUTION

=====

You can use the following sample user-defined function to add workdays, rather than weekdays, to a date.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

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

'=====
```

```
'The DateAddW() function provides a workday substitute
'for DateAdd("w", number, date). This function performs
'error checking and ignores fractional Interval values.
'=====
```

```
Function DateAddW (ByVal TheDate, ByVal Interval)

 Dim Weeks As Long, OddDays As Long, Temp As String

 If VarType(TheDate) <> 7 Or VarType(Interval) < 2 Or _
 VarType(Interval) > 5 Then
 DateAddW = TheDate
 ElseIf Interval = 0 Then
```

```

 DateAddW = TheDate
ElseIf Interval > 0 Then
 Interval = Int(Interval)

' Make sure TheDate is a workday (round down).
 Temp = Format(TheDate, "ddd")
 If Temp = "Sun" Then
 TheDate = TheDate - 2
 ElseIf Temp = "Sat" Then
 TheDate = TheDate - 1
 End If

' Calculate Weeks and OddDays.
 Weeks = Int(Interval / 5)
 OddDays = Interval - (Weeks * 5)
 TheDate = TheDate + (Weeks * 7)

' Take OddDays weekend into account.
 If (DatePart("w", TheDate) + OddDays) > 6 Then
 TheDate = TheDate + OddDays + 2
 Else
 TheDate = TheDate + OddDays
 End If

 DateAddW = TheDate
Else
 'Interval is < 0
 Interval = Int(-Interval) 'Make positive & subtract later.

' Make sure TheDate is a workday (round up).
 Temp = Format(TheDate, "ddd")
 If Temp = "Sun" Then
 TheDate = TheDate + 1
 ElseIf Temp = "Sat" Then
 TheDate = TheDate + 2
 End If

' Calculate Weeks and OddDays.
 Weeks = Int(Interval / 5)
 OddDays = Interval - (Weeks * 5)
 TheDate = TheDate - (Weeks * 7)

' Take OddDays weekend into account.
 If (DatePart("w", TheDate) - OddDays) < 2 Then
 TheDate = TheDate - OddDays - 2
 Else
 TheDate = TheDate - OddDays
 End If

 DateAddW = TheDate
End If

End Function

```

#### How to Use the DateAddW() Function

---

Use the DateAddW() function wherever you would use the DateAdd() function.

For example, instead of

```
DateAdd("w",[StartDate],10)
```

use:

```
DateAddW([StartDate],10)
```

To test the DateAddW() function, enter the following line in the Immediate window, and then press ENTER:

```
? DateAddW(#2/2/93#,10)
```

Note that the result is 2/16/93 (10 workdays).

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

Enter the following line in the Immediate window, and then press ENTER:

```
? DateAdd("w",#2/2/93#,10)
```

Note that the result is 2/12/93 (10 days), not 2/16/93 (10 workdays).

REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 102-107 and 492-493

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmOthr

## PRB: DateDiff "W" Option Does Not Work

Article ID: Q95977

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

When calculating the number of weekdays between two dates using the "w" Weekday option, DateDiff() returns the number of weeks, not the number of work days. The "w" option is supposed to function the same as "d" for DateDiff(). It is provided as an option for compatibility with the DatePart() function.

### RESOLUTION

=====

If you are using the DateDiff() function to return the number of days, substitute "d" for "w". You can use the Access Basic code in this article to return the number of work days rather than the number of days.

The code below shows how to calculate the number of work days in a date range. To understand the following information, you need to be familiar with Access Basic, know how to create and use Access Basic procedures, and know how to use the Immediate window.

The following code provides a function, DateDiffW(), that calculates the number of work days between two dates.

NOTE: In the following sample code, an underscore (\_) is used as a line continuation character. Remove the underscore when re-creating this code in Access Basic.

```

'Declaration section of the module

Option Explicit

Function DateDiffW(BegDate, EndDate)
 Const SUNDAY = 1
 Const SATURDAY = 7
 Dim NumWeeks As Integer

 If BegDate > EndDate Then
 DateDiffW= 0
 Else
 Select Case Weekday(BegDate)
 Case SUNDAY : BegDate = BegDate + 1
 Case SATURDAY : BegDate = BegDate + 2
 End Select
 Select Case Weekday(EndDate)
 Case SUNDAY : EndDate = EndDate - 2
```

```
 Case SATURDAY : EndDate = EndDate - 1
 End Select
 NumWeeks = DateDiff("ww", BegDate, EndDate)
 DateDiffW= NumWeeks * 5 + Weekday(EndDate) - Weekday(BegDate)
End If
End Function
```

#### How to Use the DateDiffW() Function

-----

Use the DateDiffW() function wherever you would use DateDiff().  
Instead of

```
DateDiff("W",[StartDate],[EndDate])
```

use the following:

```
DateDiffW([StartDate],[EndDate])
```

#### Steps to Test the DateDiffW() Function

-----

In the Immediate window, type:

```
?DateDiffW(#2/2/93#,#2/18/93#)
```

The result is 12, the number of work days.

#### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

In the Immediate window, type:

```
? DateDiff("W",#2/2/93#,#2/18/93#)
```

The result is 2 (the number of weeks), not 16 (the number of days) or  
12 (the number of work days).

#### REFERENCES

=====

For more information on the DateDiff() and WeekDay() functions, search for  
"DateDiff" and "WeekDay" using the Microsoft Access Help menu

Microsoft Access "Language Reference," version 1.0, pages 102-107 and 492-  
493

Additional reference words: 1.00 1.10 2.00 dates work day diff part

KBCategory: kbprg

KBSubcategory: PgmOthr

## PRB: DateDiff() Function Returns Incorrect Value

Article ID: Q104973

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

When you use the DateDiff() function on a large interval where the time periods are specified in minutes or seconds, the result is not correct.

### CAUSE

=====

The DateDiff() function returns a long integer. If the result is greater than the allowed range for long integers (-2,147,483,648 to 2,147,483,647), Microsoft Access truncates the value and returns the truncated value.

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. Create a new module. Enter the following in the module's Immediate window, and then press ENTER:

```
? DateDiff("n",#12-31-100#, #12-31-9999#)
```

The returned value, 911,403,103 minutes, is truncated because the actual value exceeds 2,147,483,647.

2. Enter the following calculation, which is equivalent to the function above, and then press ENTER:

```
x=#12-31-100#
y=#12-31-9999#
? (y-x)*24*60
```

The returned value, 5,206,370,400 minutes, is correct. Note that Microsoft Access uses double-precision real numbers when doing immediate calculations.

Additional reference words: 1.00 1.10 2.00 date

KBCategory: kbusage

KBSubcategory: ExrOthr

**PRB: DeleteObject Action Causes "Illegal Function Call" Msg**  
**Article ID: Q122712**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you use the DeleteObject action in Access Basic code to delete a module, you receive the error message "Illegal function call." When you use this action in a macro, the module is deleted correctly.

CAUSE

=====

When Microsoft Access runs an Access Basic function, it loads all modules into memory. If you add or delete a function, Microsoft Access must recompile the code, but it cannot do so while code is running.

RESOLUTION

=====

Use the SelectObject and SendKeys actions to delete the module instead of using the DeleteObject action.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Create a new module.
2. Enter the following line in the module's Declarations section:

```
Option Explicit
```

3. Enter the following function in the module:

```
Function DelModule()
 DoCmd DeleteObject A_MODULE, "Delete Test"
End Function
```

4. Save the module as Test.
5. Create another new module.
6. Enter the following line in the module's Declarations section:

```
Option Explicit
```

7. Enter the following function in the module:

```
Function TestFunction()
 MsgBox "Test Function"
End Function
```

8. Save the module as Delete Test.
9. From the View menu, choose Immediate Window.
10. In the Immediate window, type the following line and then press ENTER:

```
?DelModule()
```

You receive the error message stated above.

To work around this behavior, change the DelModule() function to:

```
Function DelModule()
 DoCmd SelectObject A_MODULE, "Delete Test", True 'Select the module
 SendKeys "{DEL}" 'Delete the module
End Function
```

Additional reference words: 2.00

KBCategory: kbprg kberrmsg

KBSubcategory: MdlProb



**PRB: Detail Section Blank When Form Is Printed or Previewed**  
**Article ID: Q117172**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you print or preview a form, the detail section is blank, even though controls are visible in the detail section when you view the form in Form view. Other sections, such as the form header and form footer, appear correctly when you print or preview the form.

CAUSE

=====

This behavior occurs if the form's underlying table or query returns no records, or if the detail section's DisplayWhen property is set to Screen Only.

RESOLUTION

=====

Make sure the underlying table or query returns at least one record, or place necessary controls in other sections, such as the form header or form footer. If the detail section's DisplayWhen property is set to Screen Only, change it to Always or Print Only.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

Example 1: No Records in Underlying Table:

1. Open the sample database NWIND.MDB.
2. Create a new table with several fields. Do not add any records to the table.
3. Save the new table as Table1 and then close it.
4. Create a new form based on Table1. Place the fields from the table in the form's detail section.
5. Place some controls, such as labels or buttons, in the form header and form footer.
6. View the form in Form view. Note that you can see all the controls.
7. Print or preview the form. Note that the detail section is blank.

Example 2: DisplayWhen Property Set to Screen Only:

1. Open the sample database NWIND.MDB.
2. Preview the Categories form. Note that you can see the controls in the form's detail section.
3. View the form in Design view.
4. Change the detail section's DisplayWhen property to Screen Only.
5. Preview the form again. Note that the detail section is blank.

Additional reference words: 2.00 missing fields

KBCategory: kbusage

KBSubcategory: FmsOthr

**PRB: DFirst() and DLast() Functions Return Unexpected Records**  
**Article ID: Q109380**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The DFirst() and DLast() functions do not return the first and last records of the specified domain as you expect. If the domain is a query, the DFirst() and DLast() functions appear to ignore the sort order of the query.

CAUSE

=====

The DFirst() and DLast() functions return values from the first and last records according to the order of the records in the domain. If the domain is an indexed table, the functions will return the first and last records in the indexed order. If there is no index, the functions will return the first and last records in the order in which they were entered in the underlying table.

RESOLUTION

=====

The following two sample Access Basic functions, GetFirst() and GetLast(), can be used in place of the DFirst() and DLast() functions to return the first and last records in the sorted domain as you expect.

NOTE: In the following sample code, an underscore (\_) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

Option Explicit

```
Function GetFirst (Expr$, Domain$, Criteria$)
 Dim MyDB As Database
 Dim MyDyna As Dynaset

 On Error GoTo Err_GetFirst

 Set MyDB = CurrentDB()
 Set MyDyna = MyDB.CreateDynaset(Domain$)

 If Len(Criteria$) > 0 Then
 MyDyna.Filter = Criteria$
 Set MyDyna = MyDyna.CreateDynaset()
 End If

 MyDyna.MoveFirst
 GetFirst = MyDyna(Expr$)
```

```

Bye_GetFirst:
 Exit Function

Err_GetFirst:
 GetFirst = Null
 Resume Bye_GetFirst

End Function

Function GetLast (Expr$, Domain$, Criteria$)
 Dim MyDB As Database
 Dim MyDyna As Dynaset

 On Error GoTo Err_GetLast

 Set MyDB = CurrentDB()
 Set MyDyna = MyDB.CreateDynaset(Domain$)

 If Len(Criteria$) > 0 Then
 MyDyna.Filter = Criteria$
 Set MyDyna = MyDyna.CreateDynaset()
 End If

 MyDyna.MoveLast
 GetLast = MyDyna(Expr$)

Bye_GetLast:
 Exit Function

Err_GetLast:
 GetLast = Null
 Resume Bye_GetLast

End Function

```

STATUS  
=====

This behavior is by design.

MORE INFORMATION  
=====

Steps to Reproduce Behavior  
-----

1. Open the sample database NWIND.MDB.
2. Open the Employee List query. Note that the employee names are sorted by last name, with the name "Buchanan, B. L." listed first, with an Employee ID of 5.
3. Open the Introduction To Programming module in Design view, and then choose Immediate Window from the View menu.
4. Type the following in the Immediate window, and then press ENTER:

? DFirst("[Employee ID]","Employee List")

Employee ID 1 is returned, not Employee ID 5, even though Employee ID 5 is first in the sort order. Employee ID 1 is the first indexed record in the table that the query is based on.

#### REFERENCES

=====

Microsoft Access "Language Reference," pages 139-140

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ExrOthr

**PRB: Editing Allowed Although AllowEditing Set to Unavailable**  
**Article ID: Q100129**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Even though a form's AllowEditing property is set to Unavailable, you can edit, add, or delete records in the form.

RESOLUTION

=====

This behavior is by design. To set a form so that records cannot be edited, set the DefaultEditing property to Read Only and the AllowEditing property to Unavailable.

MORE INFORMATION

=====

The form's DefaultEditing property determines whether you can modify records in the form. You can set the DefaultEditing property to Read Only, Data Entry, or Allow Edits.

The AllowEditing property on a form determines whether the Allow Editing command on the Records menu is available when you open the form. If the Allow Editing command is available, you can override the form's DefaultEditing property.

REFERENCES

=====

For more information on the AllowEditing and DefaultEditing properties, search for "allow editing" then "AllowEditing, DefaultEditing Properties" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 ui

KBCategory: kbusage

KBSubcategory: FmsProp

**PRB: Empty Table List when Attaching Btrieve Table**  
**Article ID: Q94828**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you attach a Btrieve table, you select the Xtrieve dictionary file (FILE.DDF) in the Select File box. When you choose OK, Microsoft Access opens the dictionary file and displays a list of tables you can import or attach. However, the list is empty. In Microsoft Access version 2.0, you may receive the error message:

Btrieve: Can't find file FIELD.DDF

CAUSE

=====

You do not have the FIELD.DDF file that Microsoft Access requires to attach a Btrieve table.

RESOLUTION

=====

Microsoft Access requires the following three .DDF files in order to attach a Btrieve table: FILE.DDF, INDEX.DDF, and FIELD.DDF. Make sure that you have all three .DDF files before you try to attach a Btrieve table.

MORE INFORMATION

=====

When Microsoft Access cannot find the FIELD.DDF file, it creates a 6144-byte dummy FIELD.DDF file in the same directory in which FILE.DDF appears. If you find this dummy file, you know that you are missing the required FIELD.DDF file. Use Xtrieve or another third-party utility to create the required .DDF files.

Btrieve and Xtrieve are manufactured by Novell, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding these products' performance or reliability.

Additional reference words: 1.00 1.10 2.00 Xtrieve Btrieve index  
KBCategory: kb3rdparty  
KBSubcategory: IsmBtrv

**PRB: Err Msg "Can't Set Value" Adding Record to Subform**  
**Article ID: Q123653**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you try to add a new record to a subform, you receive the error message "Can't set value."

CAUSE

=====

This error can occur if the subform's LinkChildFields property contains the name of a form property.

RESOLUTION

=====

Change the name contained in the subform's LinkChildFields property, and the name of the associated field in the underlying table, to something other than the name of a form property.

MORE INFORMATION

=====

When you add a record to a linked subform, Microsoft Access sets the value from the field on the main form specified in the subform's LinkMasterFields property to the field on the subform specified in the subform's LinkChildFields property. If the field specified in the LinkChildFields property is the name of a form property, Microsoft Access tries to set the value of that property for the form instead of the field.

Steps to Reproduce Behavior

-----

1. Start Microsoft Access and create a new database.
2. Create the following two tables:

Table Name: OneSideTable

-----

Field Name: AName

Data Type: Text

Set the AName field as the primary key for the OneSideTable table.

Table Name: ManySideTable

-----

Field Name: Num

Data Type: Number

Field Name: Parent



Data Type: Text

Note that "Parent" is the name of a form property.

3. Create a one-to-many relationship between the OneSideTable table and the ManySideTable table using the OneSideTable table as the primary table. Join the AName field with the Parent field to create the relationship. Enforce referential integrity.
4. Use the Main/Subform Form Wizard to create a main form and subform. Use the OneSideTable table as the main form, and the ManySideTable table as the subform.
5. Open the main form in Form view and then try to add a record to the subform. You receive the error message stated above. Once you choose the OK button you can add the new record.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.1, Chapter 10, "Creating Forms Within Forms," pages 274-292

Microsoft Access "User's Guide," version 2.0, Chapter 17, "Creating Forms Based on More Than One Table," pages 421-446

For more information about the LinkChildFields and LinkMasterFields properties, search for "LinkChildFields" then "LinkChildFields, LinkMasterFields Properties" using the Microsoft Access Help menu.

Additional reference words: 2.00 1.10 1.00 setvalue

KBCategory: kbusage kberrmsg

KBSubcategory: FmsSubf

**PRB: Err: "Data Source Not Found, No Default Driver Specified"**  
**Article ID: Q95050**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you are trying to attach a SQL Server table, or use a table that has already been attached, you receive the following error message:

Data source not found and no default driver specified.

CAUSE

=====

You may have a missing, damaged, or incorrect ODBC.INI file.

Other situations may indirectly cause this error message. For more information, query on the following words here in the Microsoft Knowledge Base:

debugging and troubleshooting and tips and Q102437

RESOLUTION

=====

Verify the following:

- There is an ODBC.INI file in your WINDOWS subdirectory.
- There is only one copy of the ODBC.INI file available on the computer you are using.
- The entries in the ODBC.INI file are correct for the SQL database server to which you want to connect.

MORE INFORMATION

=====

The contents of your ODBC.INI file should resemble the following:

```
[ODBC Data Sources]
MYSERVER=SQL Server
YOURSRVR=SQL Server
```

```
[MYSERVER]
Driver=C:\WINDOWS\SYSTEM\SQLSRVR.DLL
Description=SQL Server on server MYSERVER
OemToAnsi=No
Network=dbnmp3
Address=\\MYSERVER\PIPE\SQL\QUERY
```

```
[YOURSVR]
Driver=C:\WINDOWS\SYSTEM\SQLSRVR.DLL
Description=SQL Server on server YOURSVR
OemToAnsi=No
Network=dbnmp3
Address=\\YOURSVR\PIPE\SQL\QUERY
```

You will receive the error message if there is no section that refers to the SQL database server to which you are trying to attach or if any of the following are true:

- The [MYSERVER] section is missing.
- The Driver= line is missing.
- The .DLL file on the Driver= line is missing its .DLL extension.
- The .DLL file is missing, damaged, or outdated.

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbusage kberrmsg kbtshoot  
KBSubcategory: ObcOthr

**PRB: Error In Help Files for Custom Control Refresh Method**  
**Article ID: Q115192**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

The Refresh method syntax in the Help files for the Scroll Bar and Calendar custom controls is incorrect. These custom controls and their Help files are included with the Microsoft Access Developer's Toolkit (ADT) version 2.0.

RESOLUTION

=====

The correct syntax for the Refresh method for the Scroll Bar and Calendar custom controls is:

```
calendar.object.[refresh]
```

-or-

```
scroll.object.[refresh]
```

Make sure to use brackets ([]) around the word "refresh."

MORE INFORMATION

=====

The ADT includes a Help file for each of the three custom controls shipped with the ADT. Only the Data Outline custom control Help file has the correct syntax for the Refresh method.

Using the incorrect syntax listed in the other Help files will generate the error message:

```
Method not applicable for this object
```

REFERENCES

=====

For more information about the Refresh method, search for "Refresh" then "Refresh Method" by double-clicking the Data Outline Control Help icon in the Microsoft ADT program group in Program Manager.

Additional reference words: 2.00 kbdocerr

KBCategory: kbole

KBSubcategory: IntpOle

**PRB: Error Message "Couldn't Open File" in Shared Environment**  
**Article ID: Q103254**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You cannot open multiple instances of a Microsoft Access database even though multiuser options have been set. You receive either the error message

    Couldn't Open File <database name>

or the error message:

    <Database name> This file is already in use. Use a new filename or close the file in use by another application.

CAUSE

=====

This problem can occur if you attempt to open a shared database by choosing it from the list at the bottom of the File menu, instead of using the Open Database command on the File menu. When you open a database by choosing it from the bottom of the File menu, Microsoft Access opens the database with the same settings as the last time it was opened.

RESOLUTION

=====

Running a single instance of Microsoft Access, open that particular database with the Open Database command on the File menu. Then close the database, thereby making the correct settings to the database.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

This problem occurs when Microsoft Access is being shared on a network, but can be reproduced on a stand-alone computer with the following steps:

1. Open the sample database NWIND.MDB by choosing Open Database from the File menu. Select the Exclusive check box and clear the Read Only check box before choosing OK.
2. From the View menu, choose Options.
3. In the Category box, choose Multiuser.

4. In the Items box, set Default Mode For Open Databases to Shared, and then choose OK.
5. Quit Microsoft Access.
6. Restart Microsoft Access, and then open the sample database NWIND.MDB by choosing Open Database from the File menu.
7. Start a second instance of Microsoft Access.
8. Open the sample database NWIND.MDB by choosing the database from the bottom of the File menu.

The following error message occurs:

Couldn't Open File.

If steps 6 and 8 are reversed the following error message occurs:

This File Already In Use.

#### REFERENCES

=====

For more information on problems opening databases in a shared environment, query on the following words here in the Microsoft Knowledge Base:

couldn't and open and file

-or-

file and already and use

Additional reference words: 1.00 1.10 2.00 UI err msg recent multi user  
KBCategory: kbusage  
KBSubcategory: GnlMu

## **PRB: Error Message "Unable to Connect to Data Source"**

**Article ID: Q95012**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

You receive the following error message when you are trying to access a SQL Server:

Unable to connect to data source

### CAUSE

=====

The SQL Server you are trying to access does not have a correct entry in the [SQLSERVER] section of the WIN.INI file, or the SQL server is down, overloaded, or otherwise unavailable.

### RESOLUTION

=====

To correct this situation, verify that the SQL Server is up and available and that the entry in the [SQLSERVER] section of the WIN.INI file looks like this:

```
[SQLSERVER]
MY_SERVER=DBNMP3,\\MY_SERVER\PIPE\SQL\QUERY
```

If there are any spaces in this entry or if the server name in the \\MY\_SERVER\PIPE\SQL\QUERY part of the statement is misspelled, you will receive the error message.

### MORE INFORMATION

=====

It is possible that there are other situations which may indirectly cause this error message. Here are several other things you can check to troubleshoot this problem:

1. Use File Manager to search for multiple, corrupted, or outdated copies of these files:

- ODBC.DLL - 44736 bytes, dated 10-16-92 or later
- DBNMP3.DLL - 8241 bytes, dated 9-15-92 or later
- SQLSRVR.DLL - 135792 bytes, dated 10-16-92 or later
- NETAPI.DLL - depends on the network being used
- COMMDLG.DLL - 89248 bytes, dated 10-25-92 or later

NOTE: The sizes and dates referenced are correct if the files were installed by Microsoft Access version 1.0.

2. Make sure the files listed above are in appropriate locations. All files except NETAPI.DLL are usually installed in the Windows System directory. The NETAPI.DLL file is a network specific driver, so it's usually located in the network's program directory, which should be in the path.
3. Use another application, such as Q+E, SAF, or PowerBuilder, on the same computer to attempt to attach to the same data source. If you cannot access the data source from another application, there is probably a problem with the network (NETAPI.DLL) or with the server, not with Microsoft Access or ODBC.
4. Attempt to attach to another data source, if available, such as another SQL server. If you can attach to another server, then the original server may be down or you may need to reconfigure the entry for that server using the ODBC Administration Utility.
5. Attempt to attach to the same data source from another computer. If you cannot access the SQL server from another computer, there may be a network wide problem or the server may be down or not communicating.
6. Try increasing the ODBC timeout settings in the ODBC section of the MSACCESS.INI file.
7. Check with the system administrator to see if the SQL server is up, available for transactions, and operating correctly.
8. Reinstall ODBC after performing a clean boot and renaming the ODBC.DLL, DBNMP3.DLL, and SQLSRVR.DLL files.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Appendix D, "Setting Up Microsoft Access on a Network," pages 660-662

Additional reference words: 1.00 1.10 2.00 ODBC

KBCategory: kbusage kberrmsg

KBSubcategory: ObcOthr



**PRB: Error Message When Importing Informix Delimited Text File**  
**Article ID: Q99406**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you attempt to import an Informix delimited text file that uses the pipe (|) symbol as a delimiter, you receive the following error message:

Name " contains invalid characters

CAUSE

=====

When you export in Informix as text delimited and you specify the pipe symbol as the delimiter, Informix adds an extra pipe symbol at the end of the line. When you import this file to Microsoft Access by specifying "First row contains field names," Microsoft Access assumes there is another field after the last pipe symbol. Since the value is null, Microsoft Access returns the above error message.

RESOLUTION

=====

- Open the file in a text editor and add a dummy name after the last pipe symbol. After you import the file, you can delete this dummy field.
- Clear the First Row Contains Field Names check box when you import the file. After you import the file, delete the first row of the table (which contains the field names instead of data). Then, switch to Design view and modify the table's field names.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Create a text file in Microsoft Windows Notepad as follows:

```
A|B|C|
1|2|3|
```

Place a pipe symbol between each character. The pipe symbol is usually found on the backslash (\) key on your keyboard.

2. Save the file as INFORMIX.TXT.
3. Open a database in Microsoft Access.

4. From the File menu, choose Import.
5. Select Text (delimited) as your data source and choose Import.
6. Select the INFORMIX.TXT file that you created in step 1. Choose OK.
7. The Import Options dialog box appears. Select the First Row Contains Field Names check box, then choose the Options button.
8. In the Field Separator combo box, type the pipe symbol and choose OK. The above error message appears.

Informix is manufactured by Informix Software, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00 Import\Export Informix  
KBCategory: kb3rdparty  
KBSubcategory: IsmTxd

**PRB: Error Message When Opening Second Instance of .MDB File**  
**Article ID: Q93139**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you open a database exclusively, and then attempt to open the same database as shared in another instance of Microsoft Access, you receive the following error message:

'Database.mdb' This file is already in use. Use a new filename or close the file in use by another application.

In Microsoft Access version 1.x, when you open a database as shared, then attempt to open the same database exclusively in another instance of Microsoft Access, you receive the error message:

Couldn't open file '<path><filename>'.

This message is not very helpful, in that it does not state why the file could not be opened.

In Microsoft Access version 2.0, you receive the error message:

The Database is opened by user '<username>' on machine '<machine name>'. You can't open this database exclusively.

RESOLUTION

=====

Open the database as shared in the second instance of Microsoft Access, or close the database in the first instance of Microsoft Access.

STATUS

=====

This behavior is by design.

REFERENCES

=====

For more information on opening a database for shared access, search for "opening databases" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 lock network multiuser

KBCategory: kbusage kberrmsg

KBSubcategory: GnlMu

**PRB: Error Message: "This Action Cannot Be Completed..."**  
**Article ID: Q121678**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you update a Microsoft Word 6.0 object in an OLE control on a form, the computer seems to hang, or repeat the following error message:

- This action cannot be completed because the other application is busy. Choose Switch To to activate the busy application and correct the problem.

-or-

- This action cannot be completed because the WINWORD application (Microsoft Word) is not responding. Choose Switch To to activate WINWORD and correct the problem.

When you choose Switch To, or Retry, the error message is displayed again.

CAUSE

=====

This behavior may occur because of an AutoClose macro that calls a WordBasic command that is not available when an OLE object is selected.

RESOLUTION

=====

Remove the WordBasic command from the macro.

MORE INFORMATION

=====

If the AutoClose macro calls the FileSummaryInfo command, you will receive the following error message when you update the Word 6.0 OLE object:

WordBasic Err=509

The FileSummaryInfo command is not available because the current selection is an OLE object.

If the Word application is hidden when you in-place edit a Word 6.0 object in an OLE control, it may seem that Microsoft Access is locked into an error message loop. Microsoft Access expects you to switch to Word 6.0 to correct the problem, but Microsoft Windows cannot make Word 6.0 active because the application window is hidden.

To avoid this situation, make sure that AutoClose macros that call such commands are not part of the global template, or of any template that is used for an OLE object.

Additional reference words: 2.00  
KBCategory: kbole kberrmsg  
KSubcategory: IntpOle

**PRB: Error Msg. "Error Occurred in WriteSTF" in Setup Wizard**  
**Article ID: Q117767**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0  
-----

SYMPTOMS

=====

When you run the Microsoft Access Developer's Toolkit version 2.0 Setup Wizard, you receive the following error message when the Setup Wizard creates the disk images:

Error occurred in WriteSTF

CAUSE

=====

This error message can occur if any of the files you included with your application has a name that contains an apostrophe (').

RESOLUTION

=====

Rename the file that has an apostrophe in its name to a new name that does not contain an apostrophe, and then run the Setup Wizard again.

MORE INFORMATION

=====

After you see the "Error occurred in WriteSTF" error message, the following error message will appear:

File '<disk-image path>\Disk1\setup.stf' is opened for exclusive access by another application.

Although the Setup Wizard will seem to finish successfully, the Setup disks that are created will not work correctly. When you try to install your application from these Setup disks, you will receive the following error message and then Setup will terminate:

Object ID 23: Group Object: Bad Data Value.

Steps to Reproduce Behavior

-----

1. Use any text editor (such as Windows Notepad) to create a text file with the name TEST'1.TXT. (Note the apostrophe in the filename.)
2. Start the Setup Wizard. In the Files To Include box, include only the TEST'1.TXT file, then choose Next.
3. In the Available Options box, do not select any options. Choose Next.

4. In the Application Name box, type "TestIt" (without quotation marks). In the Default Installation Directory box, type "C:\TESTIT" (without quotation marks). Choose Next.
5. Do not type anything in the Executable File Name box or the Command Line box. Choose Next.
6. In the Application Setup Directory box, type "C:\SETUP" (without quotation marks). Select the Network Setup (Flat) option button. Choose Finish. The error messages stated above will appear.

#### REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0,  
Chapter 2, "Creating a Custom Setup Program"

Additional reference words: 2.00 adt  
KBCategory: kbsetup kberrmsg  
KBSubcategory: StpOthr

**PRB: Error Msg: "Can't Open File 'C:\ACCESS\WZLIB.MDA'"**  
**Article ID: Q120147**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you start Microsoft Access, you receive the error message

Can't open file 'C:\ACCESS\WZLIB.MDA'

followed by similar messages for the other Microsoft Access Wizard files.  
Finally you receive the error message:

Some library modules could not be loaded. Click OK or press ENTER to  
continue

CAUSE

=====

Microsoft Access needs to open the Wizard files (files with names beginning with "WZ and ending with ".MDA") in read/write mode. If these files cannot be opened in read/write mode, Microsoft Access returns the error messages stated above.

RESOLUTION

=====

Make sure that the Wizard files can be opened in read/write mode by checking the following items:

- Make sure that the Read Only property is not set for any of the Wizard files using File Manager.
- Make sure that the Wizard files are not in a read-only drive or directory.
- Make sure that you have write permission on the Wizard files.

Note that when Microsoft Access is installed on a network, Setup places files that cannot be read-only in the WORKDIR subdirectory.

REFERENCES

=====

For more information about using Microsoft Access on a network, see the following topics in the ACREADME.HLP file:

- Network Installations
- Setting Up Microsoft Access on a Network File Server

Additional reference words: 2.00 system.mda wzquery.mda wztable.mda



KBCategory: kbsetup kberrmsg  
KBSubcategory: StpNetinst

## **PRB: Error Occurs in Print Preview Without Driver**

**Article ID: Q95901**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

You receive the following error message when you preview a form or report:

An error occurred while attempting to retrieve printer information for the <printer name> on <printer port>.

### CAUSE

=====

This error occurs when the printer driver in use when the form or report was created no longer exists on the system. If you print the form or report rather than preview it, you will receive the following error message from COMMDLG.DLL:

This document was previously formatted for the printer <previous printer name> on <PORT>, but that printer is not available. Use the default printer <default printer name> on <PORT>.

### RESOLUTION

=====

Open the form or report in Design view and then make sure that the selected printer is correct. If Specific Printer is selected in the Print Setup dialog box, you must have the printer driver that was in use when the form or report was created installed on the system. Save the form or report before you preview or print it.

Selecting Default Printer in the Print Setup dialog box before you save a new form or report makes the form or report more portable, since the form or report will use the default printer on any system when that form or report is previewed or printed.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. Create a new form.
2. Before you save the new form, choose Print Setup from the File menu.

Select the Specific Printer option, and then select a printer.

3. Save the form and then close it.
4. Open Windows File Manager.
5. Locate the printer driver for the printer you selected in step 2.  
(Printer drivers are usually in the WINDOWS\SYSTEM subdirectory.)
6. Rename the printer driver file.
7. Open the form you created in step 1.
8. Preview the form.

The error message stated above will occur.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprint

KBSubcategory: PtrPrev

**PRB: Errors Using Recordset Methods on Data Outline Control**  
**Article ID: Q120568**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

When you are writing code for the data outline control, you can create a Recordset-like object using the GetRecordSetClone method on the data outline control object. However, when you use standard Recordset methods on this object, you receive compile errors.

RESOLUTION

=====

In your code, enclose the methods used on the Recordset-like object in brackets. For example, assuming an object called RecSet that you created with the GetRecordSetClone method, use

```
RecSet.[movefirst]
```

instead of:

```
RecSet.movefirst
```

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create a blank, unbound form called TestOutline. Make sure that the Control Wizards button in the toolbox is selected (the button should appear sunken).
3. From the Edit menu, choose Insert Object. In the Insert Object dialog box, select the Insert Control option button, then select Data Outline Control in the Control Type box. Choose OK.
4. In the "For each level, what data do you want the Data Outline Control to display?" screen, select the Customers table in the Table/Query box for level 1, and then choose the Finish button.
5. From the View menu, choose Code.

6. In the Object box on the toolbar, select Embedded0 (the data outline control). In the Procedure box, select Enter.

7. Enter the following code in the module:

```
Sub Embedded0_Enter ()

 Dim DOC As Object 'Data outline control
 Dim RecSet As Object 'Recordset

 Set DOC = Forms!TestOutline!Embedded0.Object
 Set RecSet = DOC.GetRecordsetClone(1)

 RecSet.MoveFirst

End Sub
```

8. Compile the code by choosing Compile All from the Run menu. You will receive the compile error message "Method not applicable for this object."

#### REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0,  
Chapter 6, "Using OLE Custom Controls"

Additional reference words: 2.00

KBCategory: kbprg kberrmsg

KBSubcategory: MdlRcd

## **PRB: Every Other Page Blank in Printed Reports**

**Article ID: Q95920**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

When you print a report generated in Microsoft Access, a blank page appears between every printed page of the report or a blank mailing label appears between every printed label.

### CAUSE

=====

Blank pages or mailing labels are generated when a report or label is wider or longer than the specified page. This condition can be caused by one or more of the following problems:

- The total width of the form or report, plus the widths of the left and right margins, exceeds the page width (which is usually 8.5 inches).
- The page size exceeds the user-defined paper size.
- You used the Groups/Totals ReportWizard, but did not select the Fit All Fields On One Page check box. Because of this, the ReportWizard may have located fields or controls off the page.
- In the Print Setup box under Item Size, the Same As Detail check box has been cleared.

### RESOLUTION

=====

The following resolutions correspond to each of the four causes described above:

- Reduce the width of the form or report, reduce the left or right margins, or reduce all three so that the total width of the form or report, plus the width of the margins, does not exceed the page width. Use the following formula:

$$(\text{Form or Report Width}) + (\text{Left Margin}) + (\text{Right Margin}) \leq (\text{Page Width})$$

Use the form or report's property sheet to view or change the Width property. Before you reduce the width of a form or report, move the fields and controls on it to match the new size.

To view the left and right margin values, choose Print Setup from the File menu.

If problems persist, use the following procedure to check the Print Setup dimensions:

1. From the File menu, choose Print Setup.
2. Choose More.
3. Under Item Size, verify that the Same As Detail box is selected. If so, the width and height listed will match the dimensions of the detail section of the report. If not, these settings will override the detail settings of the report and may cause blank pages to print.

NOTE: Item Size is usually used for multiple-column reports. You must ensure that all the columns fit on the report form when you set these values.

- If the page size is user-defined, Microsoft Access reads the settings that you specified for that printer driver with the Printers utility in the Windows Control Panel. If your report page exceeds this user-defined paper size, your report will include blank pages.

A new page size will not be available in Microsoft Access until you set one up in the Printers utility in the Control Panel.

The following steps describe how to change the page size with the Printers utility:

1. Open Control Panel, usually found in the Main program group of Microsoft Windows Program Manager.
2. Double-click the Printers icon to start the Printers utility.
3. Select the printer driver you want to change and then choose the Setup button.
4. In the Paper Size box, select User-Defined. A dialog box will be displayed in which you can adjust the settings.

NOTE: Only dot-matrix printer drivers have a user-definable page option. Laser printer drivers do not.

- When you are using the Groups/Totals ReportWizard, select the Fit All Fields On One Page check box on the last screen before you switch to Design view or print or preview the report. Once you select this check box, the ReportWizard assumes a page 8.5 inches wide, then sizes the fields and controls accordingly. If the check box is cleared, the resulting report width may exceed the page size.
- The Same As Detail option is useful when you create a detailed, multiple-column report and want the detail section to be a different size than the other sections. Clearing this check box can cause blank pages if the width defined for the detail section of the report is greater than the page size specified. This box, found under Item Size in the Print Setup dialog box, is selected by default. The Width property is set to match the report's Width property.

STATUS

=====

This behavior is by design.

Additional reference words: 1.00 1.10 2.00 skipping blank page

KBCategory: kbusage kberrmsg

KBSubcategory: RptLayout



**PRB: Export Table to Other Database Type Does Not Create Index**  
**Article ID: Q97773**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you export a Microsoft Access table with a primary key field to any of the supported database formats, an index is not created. Also, if you import a supported foreign database table having an index or primary key, the resulting Microsoft Access table will not have a primary key defined.

CAUSE

=====

The Microsoft Access import and export utility only imports and exports data, not key or index definitions.

RESOLUTION

=====

When you export a Microsoft Access table to an external database format, you will need to use the native application for that file type to define the index or key fields. Likewise, after a foreign database is imported to your Microsoft Access database, you will need to open the table in Design view and define the primary key field or fields.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

The following database formats can be imported or exported by Microsoft Access. After a table is exported from Microsoft Access, use the native product for each file type to create your indexes or key fields.

- FoxPro 2.x
- Paradox 3.x
- Paradox 4.x (Microsoft Access version 2.0 only)
- dBASE III +
- dBASE IV
- Btrieve
- SQL Server

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr

**PRB: Exporting Date Fields to Text Includes Time Format**  
**Article ID: Q103174**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

If you export a table that includes a date field to a text file, Microsoft Access will automatically add the time format to the date in the exported text file. The information will be formatted as "mm/dd/yy hh:nn:ss".

CAUSE

=====

In Microsoft Access, dates are stored with both date and time formats. During the export process Microsoft Access creates both the date and time format in the exported text file. If there is no data for the time format it will show up as all zeros (indicating 12 A.M.).

RESOLUTION

=====

There are two methods you can use to work around this problem:

Method 1

-----

Create a query with a calculated field to format the date field appropriately, and then export the results of the query. The details of this method are listed below in the "workaround" section.

Method 2

-----

Microsoft Access can create a fixed width text file, using the "Text (Fixed Width)" export format. Using this method, you can set the size of the date field to 8 characters in the Import/Export specification. This will truncate the time portion of the date field during export.

NOTE: Common text exports require the data to be delimited using the "Text (Delimited)" format. Method 2 will not be useful in these cases. For more information on this method, see the Microsoft Access "User's Guide," pages 78-79.

STATUS

=====

This behavior is by design. This feature is under review and will be considered for revision in a future release.

MORE INFORMATION

=====  
Steps to Reproduce Behavior  
-----

1. Create a table with a new field called Birthday with Date/Time chosen as the data type.
2. From the File menu, choose Save. Save the table with a unique name, such as "MyTable".
3. From the View menu, choose Datasheet. Type "11/15/67" (without the quotation marks) in the first record, and then close the table.
4. From the File menu, choose Export.
5. Select "Text (Delimited)" from the list of available export formats, and then choose OK.
6. Select the table saved in step 2 from the list of available tables, and then choose OK.
7. Type a unique name in the File Name box, and then choose OK.
8. In the Export Text Options dialog box, select OK.
9. Open the exported text file in any text editor (for example, the Windows Notepad text editor), and note that the date is formatted as "11/15/67 00:00:00".

Workaround (Using a Query to Export Date Fields Appropriately)  
-----

1. Create a new query and add the name of the table to be exported.
2. Include all desired fields in the query grid except for the date field.
3. For a date field called "Birthday", type the following in an empty column's FieldName box:

```
Birthday2: Format([Birthday], "mm/dd/yy")
```

NOTE: The new calculated field name cannot match the name of any other field that is included in the query.

4. Repeat step 3 for all other date fields.
5. From the Query menu, choose Make Table. In the Table Name box, type a unique table name, and then choose OK.
6. From the Query menu, choose Run to execute the query.

The query will run and the results will be stored in the new table named in step 5.

7. Close the query. Make sure to save the query with a unique name if

you will want to use it again.

8. Activate the Database window, and then export the new table using the Export command on the File menu.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, pages 78-79 and 162-167

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr

**PRB: Exporting to Fixed-Width Text File Left Justifies Numbers**  
**Article ID: Q98663**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you export a table containing a Number or Currency field to a fixed-width text file, the numbers in the table become left justified.

CAUSE

=====

Since the numbers are being exported to a fixed-width text file, Microsoft Access pads zeroes to the right of the number until it meets the proper field width.

RESOLUTION

=====

To make the numbers align with the decimal and pad zeros to the left instead of the right, you first must determine the maximum length of the Number or Currency field. Use this number of zeros in the Format() function below. Next, you must determine the number of decimal places the field in question uses. Use this value to determine how many zeros to place after the decimal point in the Format() function below. Finally, create a query based on the following Microsoft Access SQL statement:

```
SELECT DISTINCTROW
 Format([MyNumber],"0000.00")
AS [Expr1]
INTO [MyNewTable]
FROM [MyOldTable];
```

For this example, this SQL statement assumes that you have the following table:

```
Table: MyOldTable

FieldName: MyNumber
DataType: Currency
```

It also assumes that the longest value in the MyNumber field is 7 characters in length (including the decimal and decimal places) and that each value has two decimal places. The above Microsoft Access SQL statement, when typed in the SQL window in the query-by-example (QBE) grid, makes a new table called MyNewTable with the MyNumber field padded to the left with zeros.

When you use the above Microsoft Access SQL statement in a query, the data from MyOldTable is formatted with decimal justification and

placed in a new table called MyNewTable.

STATUS  
=====

This behavior is by design.

MORE INFORMATION  
=====

Steps to Reproduce Behavior  
-----

1. Create the following table:

Table: MyOldTable  
-----  
FieldName: MyNumber  
DataType: Currency

2. Enter the following numbers in the table:

34.5  
123.56  
4578.90

3. From the File menu, choose Export and select the Text (Fixed Width) option. Export MyOldTable.

4. Open the exported text file with a utility, such as Microsoft Windows Notepad, and view the numbers. They appear as follows:

34.5000  
123.560  
4578.90

Additional reference words: 1.00 1.10 2.00 Importing/Exporting  
Justification  
KBCategory: kb3rdparty  
KBSubcategory: IsmTxtfx

**PRB: Field Object Type Property Returns "Invalid Operation"**  
**Article ID: Q115901**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

When you assign a variable to a property (such as Type) of a field object of an Index object, you receive the error message "Invalid Operation." This occurs for all properties except Name and Attributes.

CAUSE

=====

Field objects of an Index object have a subset of the normal properties. The only valid properties are Name and Attributes, and the Attributes property has special values. For an Index object's field object, the Attributes property represents an ascending or descending sort order.

RESOLUTION

=====

If you need to refer to other properties of the field object, use the TableDef field object. For example, to get the Type property you could use

```
x = tabledef.fields("myfield").properties("Type")
```

instead of:

```
x = tabledef.indexes("Myindex").fields("myfield").properties("Type")
```

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB and create a new module.
2. Enter the following line in the Declarations section of the module:

```
Option Explicit
```

3. Enter the following code in the module:

```
Function FieldType ()
 Dim ws As Workspace
 Dim db As Database
 Dim tdef As TableDef
 Dim result As String
 Dim i As Integer
```

```
' Set up the environment.
Set ws = DBEngine.Workspaces(0)
Set db = ws.Databases(0)

' Main routine
Set tdef = db.tabledefs("Employees")

' The line below results in "Invalid Operation."
result = tdef.indexes("PrimaryKey").fields(0).properties("Type")
Debug.Print result

' Cleanup
db.Close
ws.Close
End Function
```

4. From the View menu, choose Immediate Window.

5. In the Immediate window, type the following line and then press  
ENTER:

```
? FieldType()
```

Additional reference words: 2.00

KBCategory: kbprg kberrmsg

KBSubcategory: PgmObj



**PRB: Field Properties Not Inherited with Make-Table Query**  
**Article ID: Q121564**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You can use a make-table query to create a new table from a query's dynaset. Although the new table will inherit the original table's data types and FieldSize property settings, the new table will not inherit the original table's Format, InputMask, Caption, DefaultValue, ValidationRule, ValidationText, Required, or Indexed property settings.

CAUSE

=====

The new table only inherits the original table's data types and FieldSize property settings. All other properties in the new table are set to default values.

RESOLUTION

=====

After you run a make-table query, open the new table in Design view and re-create the indexes and property settings. You may also want to define a primary key for the table at this time.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

---

1. Open the sample database NWIND.MDB.
2. Create a new query based on the Order Details table.
3. Drag all the fields from the Order Details field list to the query grid.
4. From the Query menu, choose Make Table.
5. In the Query Properties dialog box, type "Order Details Compare" (without quotation marks) in the Table Name box, and then choose the OK button.
6. Run the query.

7. Open the Order Details and the Order Details Compare tables in Design view. Compare the fields in the two tables and note that the Format property settings are missing in the Order Details Compare table.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 3, "Changing and Customizing Tables," pages 39-42, and Chapter 7, "Designing Action Queries and Parameter Queries," pages 163-167

Microsoft Access "User's Guide," version 1.1, Chapter 3, "Changing and Customizing Tables," pages 39-42, and Chapter 7, "Designing Action Queries and Parameter Queries," pages 167-171

Microsoft Access "User's Guide," version 2.0, Chapter 13, "Changing Data with Action Queries," pages 301-303

Additional reference words: 1.00 1.10 2.00 maketable

KBCategory: kbusage

KBSubcategory: QryMaktbl

**PRB: Form Name Error Message Using Data Outline Control**  
**Article ID: Q118793**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

When you double-click a row in your data outline control, you receive the error message

Action requires a Form Name argument.

CAUSE

=====

The data outline control's FormName property for that level is blank.

RESOLUTION

=====

You can create an Access Basic procedure to check whether a FormName property is blank. To use this method, add the following sample code to the data outline control's RequestFormOpen event:

```
Sub Outline_RequestFormOpen (Cancel As Integer, Level As Integer)
 Dim L As Object
 Set L = Me![Outline].Object.Levelinfos
 If L(Level).Formname = "" Then Cancel = True
End Sub
```

This procedure prevents the data outline control from trying to open the form if the FormName property is blank.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

NOTE: This example assumes that you have the Data Outline Control Wizard loaded as a library.

1. Open the sample database NWIND.MDB.
2. Create a new, blank form. Make sure that the Control Wizards button in the toolbox is selected (the button should appear sunken).
3. From the Edit menu, choose Insert Object.
4. In the Insert Object dialog box, select the Insert Control option button.

5. In the Control Type box, select Data Outline Control. Choose OK.
6. In the Table/Query box, select the Categories table.
7. Press the TAB key to move to the Associated Form box, then press the DEL key to remove Categories from the Associated Form box.
8. Choose the Finish button.
9. View the form in Form view and double-click any row in the data outline control.

#### REFERENCES

=====

For more information about the RequestFormOpen event, search for "RequestFormOpen" then "RequestFormOpen Event" using the Data Outline Control Help file that is included with the Microsoft Access Developer's Toolkit version 2.0.

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0, Chapter 6, "Using OLE Custom Controls," pages 163-175

Additional reference words: 2.00 ab

KBCategory: kbinterop kberrmsg

KBSubcategory: IntpCstm

## **PRB: Form Opened in Form View Is Completely Blank**

**Article ID: Q93261**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

You open a form in Form view, and it is completely blank. There are no controls or text. Everything is blank.

### CAUSE

=====

Typically, this happens because the query returns no records or privileges have been limited.

You can also see a blank form if the header and footer are occupy so much space that there's no room for the detail section.

### RESOLUTION

=====

If the form is based on a query, run the query. If the query returns no datasheet cells in which to enter data, then the query has a problem (such as unjoined tables). A good query returns empty cells even if the tables are empty.

Also controls located in form headers or footers are always visible--even when privilege limits or empty dynasets prevent the display of controls located in the detail section. Therefore, when you design a form, place necessary controls such as command buttons and check boxes in the form header or footer section. Also, you should ensure that each dynaset contains at least one record.

Finally, try reducing the size of the header and footer to allow more room for the detail section.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

The following situations may cause a blank form to display:

- The form is based on a query that contains a one-to-many join, and the query returns no records.
- The form is not updatable at all. It is a read-only form with the AllowUpdating property set to No Tables, or it is a static form that

displays results by using a GROUP BY clause or a DISTINCT operator.

- The form is based on a query that contains a one-to-one join, and one of the tables is in a read-only database.
- The form is based on a query that contains a many-to-one inner join, and the query does not contain all the joined columns of the many-side table.
- The form is based on a query that includes an updatable remote table but does not include all the columns that make up the primary key.
- The form is read-only and is based on an empty table.

#### Steps to Reproduce Situation 1

-----

1. Change the Catalog query in the NWIND sample database placing IS NULL in the criteria cell for the field [Category Name].
2. Create a new form based on the modified query, and place some fields and a command button in the detail section.
3. Switch to Form view.

Result: The blank form is displayed.

#### Steps to Reproduce Situation 6

-----

1. Create a new table. Do not enter any data.
2. Create a new form based on the table. Add a command button and make the form read-only.
3. Switch to form view.

Result: A blank form is displayed.

The form displays blank in all situations because the dynaset is empty. The form would display as expected if you were adding new records, but the setting for the form or the joins in the query prevent you from using the form to add new records.

To work around this issue, place necessary controls or command buttons in the form header and footer. Headers and footers display even when the detail section is blank.

Additional reference words: 1.00 1.10 2.00 object has no value

KBCategory: kbusage

KBSubcategory: FmsOthr

## **PRB: Form Size Changes Not Retained**

**Article ID: Q105661**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

If you alter the size of a form in Design view, and then save and close the form, when you open the form from the Database window, the size of the form will not reflect the changes you made.

### CAUSE

=====

You sized the window that contains the form, not the form itself. A form has its own dimensions independent of the size of the window the form is displayed in, so that the form can be printed correctly no matter what size the window is.

### RESOLUTION

=====

There are several methods of changing a form's size, including the following:

- Set the form's AutoResize property to No. This will cause the Form window to open to the size you set it to, rather than snapping to the size of the form contained within it.
- Set the form's Width property to the width you want.
- Set the form section's Height property to the height you want for that section.
- Use the mouse to drag the right side of the form to make the form the width you want.
- Use the mouse to drag the bottom of the form or section to make the form or section the height you want.

Note that you cannot size the width of the form narrower than the reach of the rightmost control on the form. You cannot size the height of a section shorter than the reach of the last control in the section.

NOTE: If you view the form in Form view directly from Design view, the Form window will remain the same size it was in Design view. You can force the Form window to snap to the size of the form by choosing Size To Fit Form from the Window menu when you are viewing the form in Form view.

If you open the form in Form view from the Database window, the Form window will snap to the size of the form automatically.

## MORE INFORMATION

=====

### Steps to Reproduce Behavior

-----

1. Create a new, blank form. Note that the white area of the detail section is 5 inches wide and 1 inch tall.
2. Drag the Form window to make it narrower than 5 inches wide.
3. Save the form with any unique filename, and then close the form.
4. In the Database window, select the form and then choose Open.

The form will open 5 inches wide and 1 inch tall.

## REFERENCES

=====

For more information on sizing forms, search for "sizing forms" using the Microsoft Access Help menu.

Microsoft Access "Getting Started," version 1.1, pages 75 and 84

Microsoft Access "User's Guide," version 1.1, pages 261 and 263

Additional reference words: 1.00 1.10 2.00 resize

KBCategory: kbusage

KBSubcategory: FmsProp



**PRB: Forms and Reports Print Slowly, Large Temporary Files**  
**Article ID: Q103430**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You run out of temporary disk space when you print forms and reports, or your reports print very slowly.

CAUSE

=====

Forms and reports containing lines, bordered fields, or opaque fields print much more slowly than the same form or report containing no graphic objects. This behavior holds true for Microsoft Windows applications in general.

Because lines, bordered fields, and opaque fields are treated as bitmaps, a great deal of data is sent to the printer at once. Large temporary files are needed to print the form or report.

If you specify an incorrect or low setting for printer memory in Control Panel, Windows will "band" complex print jobs. That is, it sends small blocks of graphics to the printer, then clears them after the printer has processed them. Since Windows cannot band text, it sends TrueType fonts to the printer as graphics. When this occurs, the files become extremely large.

When you print your document in landscape format, your print speed may decrease and your temporary files may increase in size. This problem occurs primarily because Microsoft Access versions 1.0 and 1.1 print lines as graphics.

Printing a line as a graphic is acceptable if the line is horizontal, since graphics are sent to the printer one row of pixels at a time. However, vertical lines can be a problem. In large graphics, a record is sent to the printer for each pixel in the line. When you print in landscape format, the printer interprets these horizontal lines as vertical lines.

MORE INFORMATION

=====

Microsoft Access version 2.0 has a FastLaserPrinting property for forms and reports. If you set the FastLaserPrinting property setting to Yes, lines and rectangles in your form or report are replaced by rules when you print the form or report. This property speeds up printing only on certain laser printers; it has no effect on dot-matrix printers. The FastLaserPrinting property also cannot speed up early versions of laser printers that do not support rules.

REFERENCES

=====

For more information about the FastLaserPrinting property in Microsoft Access version 2.0, search for "FastLaserPrinting" then "FastLaserPrinting Property" using the Microsoft Access version 2.0 Help menu.

Additional reference words: 1.00 1.10 2.00 printing

KBCategory: kbprint

KBSubcategory: PtrOthr

**PRB: GP Fault Attaching Tables with Old CTL3D.DLL File**  
**Article ID: Q109391**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
- 

SYMPTOMS

=====

A general protection (GP) fault may occur when you attach tables using ODBC. Or, you may notice that the borders of the ODBC dialog boxes appear unusual.

CAUSE

=====

The GP fault may be due to an older CTL3D.DLL file. This file is usually treated as a shared DLL and is installed in the WINDOWS\SYSTEM directory. However, some applications will install the CTL3D.DLL file in the Windows directory. Older versions of the CTL3D.DLL file in the Windows directory can cause problems with ODBC.

RESOLUTION

=====

Search your hard drive for CTL3D.DLL files. If there is more than one, rename all the files except the one with the latest date to some other filename. Make sure that the CTL3D.DLL file with the latest date is in the WINDOWS\SYSTEM directory. Note that the CTL3D.DLL file that is shipped with Microsoft Access is dated 5/21/93 or later with a size of 14,480 bytes.

Additional reference words: 1.10 2.00 GPF

KBCategory: kbusage

KBSubcategory: ObcOthr

**PRB: GPF with Diamond Viper VESA Local Bus Video Driver V1.01**  
**Article ID: Q100172**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you are using version 1.01 of the Diamond Viper VESA local bus video driver for Microsoft Windows version 3.1, you receive a general protection (GP) fault when you attempt to run Microsoft Access. The GP fault occurs immediately after you try to open a database.

CAUSE

=====

This problem is caused by a fault in version 1.01 of the Diamond Viper VESA local bus video driver for Windows 3.1.

RESOLUTION

=====

According to Diamond Computer Systems, later versions of the driver correct this problem. To obtain this driver, call Diamond Computer Systems Technical Support or download the latest VPRxxx.EXE file by modem from the Diamond Computer Systems bulletin board service (BBS) at (408) 730-1100.

The Diamond Viper VESA local bus video driver is manufactured by Diamond Computer Systems, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00 WinIss video driver GP fault GPF  
KBCategory: kbenv  
KBSubcategory: EvnOs

**PRB: Imported MS Excel Carriage Returns Become Vertical Bars**  
**Article ID: Q115576**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Excel versions 2.x, 3.x, 4.x, and 5.0
- 

SYMPTOMS

=====

When you import a Microsoft Excel spreadsheet into Microsoft Access, carriage return (CR) characters appear as small vertical bars.

For example, when you import a Microsoft Excel spreadsheet mailing list with complete addresses stored in single cells formatted with carriage returns, the addresses appear in Microsoft Access as single lines with small vertical bars between the address items.

CAUSE

=====

The CR character used in Microsoft Excel (ALT+ENTER) differs from that used in Microsoft Access (CTRL+ENTER). As a result, the CR characters in Microsoft Excel spreadsheets are not parsed into carriage returns in Microsoft Access, but into small vertical bars.

RESOLUTION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Building Applications" manual.

You can use the following sample user-defined Access Basic function to convert the CR character used in Microsoft Excel to the CR character used in Microsoft Access.

1. Start Microsoft Access and open any database.
2. Enter the following user-defined function in a module.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Function ChangeStr (s As Variant, a As String, n As String, _
c As Integer) As Variant
 ' This function changes all substrings "a" and string "s"
 ' to "n." The parameter "c" has the same function as in the
 ' InStr() function.
 Dim temp As String, pos As Integer

 temp = ""
```

```

If IsNull(s) Then
 ChangeStr = Null
 Exit Function
End If
If a = "" Or s = "" Then
 ChangeStr = s
 Exit Function
End If
pos = InStr(1, s, a, c)
While pos > 0
 temp = temp & Mid$(s, 1, pos - 1) & n
 s = Right$(s, Len(s) - pos - Len(a) + 1)
 pos = InStr(1, s, a, c)
Wend
ChangeStr = temp & s
End Function

```

3. Create a new update query. Drag the field that you want converted to the query grid.

4. In the Update To row of the query grid, enter the following expression:

```
ChangeStr([<fieldname>],Chr$(10),Chr$(13) & Chr$(10),0)
```

Substitute the name of the field that you want converted for <fieldname> in the expression above.

5. Run the query. Text that contains small vertical bars will be separated into separate lines with carriage returns.

STATUS  
=====

This behavior is by design. Changes to this behavior are being reviewed and will be considered for inclusion in a future release of Microsoft Access.

MORE INFORMATION  
=====

Steps to Reproduce Behavior  
-----

1. In Microsoft Excel, create a spreadsheet with the following data:

```

Fred Test
111 Main
Anytown, USA

```

```

Joe Test
222 Broadway
Anytown, USA

```

2. Save the spreadsheet.

3. Start Microsoft Access and open any database.

4. From the File menu, choose Import.

5. In the Data Source box, select either Microsoft Excel 2.0-4.0 or Microsoft Excel 5.0, depending on the version of Microsoft Excel you are using, and then choose OK.
6. In the File Name box, select the spreadsheet you saved in step 2, and then choose the Import button.
7. In the Table Options group, select the Create New Table option button, then choose OK.
8. When you get a message stating that two records were imported successfully, choose OK. Choose Close.
9. Open the new table in Datasheet view. Note the small vertical bars between the data elements in the table.

#### REFERENCES

=====

For more information about importing, search for "Import" then "Import Command (File Menu)" using the Microsoft Access Help menu.

Additional reference words: 2.00 pipes lf

KBCategory: kbinterop

KBSubcategory: IsmExl5

**PRB: Incorrect Catalog Stored Procedures Error Message**  
**Article ID: Q96013**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The following error message may be displayed when you attempt to attach a SQL table in Microsoft Access:

[Microsoft][ODBC SQL Server Driver] The ODBC catalog stored procedures installed on your server <servername> are version xx.xxxx; version xx.xx.xxxx is required to ensure proper operation. Please contact your system administrator.

CAUSE

=====

To perform its functions, Microsoft Access requires certain information from a back-end database. This information is provided by a series of catalog stored procedures that reside on the back-end database. You need to use the correct version of the catalog procedures for the version of Microsoft Access and SQL Server that you are running.

This error indicates that you have an incorrect version of the catalog stored procedures installed.

RESOLUTION

=====

Reinstall the stored procedure file. Refer to the following paragraphs to determine which script file to use with your version of Microsoft Access and SQL Server.

Microsoft Access Version 1.0

-----

If you run Microsoft Access version 1.0 with Microsoft or Sybase SQL Server, use the correct script file from the table below:

| SQL Server<br>And Version                    | Catalog Stored Procedure<br>Script File Requirements                                                                                                  |
|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Microsoft SQL Server<br>versions 1.x and 4.2 | Use the INSTCAT.SQL file from the Microsoft Access ODBC disk.                                                                                         |
| Microsoft SQL Server<br>version 4.2a         | INSTCAT.SQL is not required, but some Microsoft Access problems are resolved with it that are not in the Microsoft SQL Server 4.2a stored procedures. |



Sybase SQL Server            Use the INSTCAT.SQL file from the Microsoft  
versions before 4.8        Access ODBC disk.

Sybase SQL Server            Use the INSTCAT.48 file from the MSACCESS  
versions 4.8-4.9x         forum on CompuServe. Go to the ODBC/SQL  
Server library, and download the SYBASE.ZIP  
file.

#### Microsoft Access Versions 1.1 and 2.0

-----

In Microsoft Access version 1.1, both the Sybase and Microsoft SQL servers use the same INSTCAT.SQL file.

To install the catalog stored procedures, use the SQL Server ISQL (Interactive SQL) tool. You can run the ISQL tool from the MS-DOS or OS/2 command prompt.

**IMPORTANT:** Do not use the SAF utility provided with Microsoft SQL Server. Microsoft SAF for MS-DOS and OS/2 is limited to 511 lines of code in an SQL script.

The syntax to install INSTCAT.SQL using ISQL is:

```
isql /U <sa loginname> /n /P <password> /S <SQL servername>
/i <drive:\path\INSTCAT.SQL> /o <drive:\path\outputfilename>
```

**NOTE:** Enter the preceding two lines as one line at the command prompt, and do not include the angle brackets < >.

The following are the command line parameters for the ISQL command:

```
/U Login name for the system administrator.
/n Eliminates line numbering and prompting for user input.
/P Password (case sensitive) used for the system administrator.
/S Name of the driver to set up.
/i Provides the drive and fully qualified path for INSTCAT.SQL.
/o Provides ISQL with an output file destination for results,
 including errors.
```

#### Example

-----

```
isql /U sa /n /P skier /S DUMMY_SERVER /i D:\SQL\INSTCAT.SQL /o
D:\SQL\OUTPUT.TXT
```

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: ObcOthr

## **PRB: Incorrect Results Received with DSum() Function**

Article ID: Q97000

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

Incorrect results received from the Dsum() function.

### CAUSE

=====

The Dsum() function returns a set of values in a specified set of records (domain). Typically you might use this function in a text box control on a report to total the values from a field in a table or query.

If you omit the surrounding quotation marks from the expression argument, you will not see the expected error message. Instead, you see a value that is actually derived by taking the last value of the field from the report and summing it x number of times (once for each record in the table or query).

### RESOLUTION

=====

Surround each argument in the Dsum() function with quotation marks.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

The following example is based on the sample database NWIND.MDB:

1. Open the Freight Charges report in Design view.
2. Add a text box to the Company Name footer section:

Object: Text Box

-----  
ControlName: MyDsum

ControlSource: =DSum([Freight],"Freight Charges")

3. Preview the report. The amount showing in this text box is incorrect (compare it to the text box control in the report footer

section named Grand Total). Notice that the optional criteria expression is not included in this example. Therefore, you would expect to see the total of the Freight field from the Freight Charges query.

4. Change the ControlSource to:

```
=DSum("[Freight]","Freight Charges")
```

6. Preview the report. Now the amount in the text box is correct.

#### REFERENCES

=====

For more information on the Dsum() function, search for "Dsum" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 expressions

KBCategory: kbusage

KBSubcategory: FmsHowto

**PRB: Index Missing for Attached FoxPro or dBASE Table**  
**Article ID: Q122561**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you open an attached dBASE or FoxPro table in Design view, the table's index is missing even though you specified an index when you attached the table.

RESOLUTION

=====

When you attach a dBASE or FoxPro table, make sure that the index you specify is based on a single field or a simple concatenation of multiple text fields. An index based on a more complex expression may not be visible in Microsoft Access.

MORE INFORMATION

=====

A dBASE or FoxPro index based on a complex expression (anything other than a single field or a simple concatenation of text fields) may cause the Microsoft Jet database engine used in Microsoft Access to form incorrect joins due to incorrect assumptions about the order of the keys. For example, an index based on "string1 + function(string2)" may appear to the Microsoft Jet database engine to be based on "string1, string2." The Microsoft Jet database engine will not use such indexes, and will not display them in the Indexes window.

The indexed sequential access method (ISAM) driver used by Microsoft Access will, however, attempt to maintain an associated dBASE or FoxPro index while the ISAM is updating the dBASE or FoxPro table even though the index is not visible in Microsoft Access.

dBASE is manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 9, "Importing, Exporting, and Attaching," pages 164-166

Additional reference words: 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmXbase

## PRB: Input Mask Character (!) Does Not Work Correctly

Article ID: Q114880

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SYMPTOMS

=====

The exclamation point character (!) in an input mask does not cause the input mask to fill from right to left as you expect. Instead, it fills from left to right.

For example, the input mask

```
!\(999") "000\-0000
```

fills characters from left to right, instead of from right to left.

### CAUSE

=====

This behavior is caused by a design limitation in the way input mask characters are validated.

### STATUS

=====

This behavior is by design. Changes to input mask character validation are being reviewed, and will be considered for possible inclusion in a future release of Microsoft Access.

### MORE INFORMATION

=====

You can work around this behavior to some extent by combining a more generic input mask and a validation rule. For example, using the sample input mask listed in the "Symptoms" section of this article, you have to use the mouse or the keyboard to bypass the area code if you want to enter a 7-digit phone number. If you use the following sample input mask and validation rule instead, you can type digits anywhere in the mask, even though the digits still fill from left to right:

```
Input Mask: !\(999") "999\-9999
Validation Rule: Like "#####" Or Like "#####"
Validation Text: You must enter a 7- or 10-digit phone number
```

NOTE: The input mask does not contain any required digits. The required digits are handled by the validation rule.

### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 8 "Changing and

Customizing Tables," pages 140-141

Additional reference words: 2.00 validating masking

KBCategory: kbusage

KBSubcategory: Gn1Valid

**PRB: Invalid Path Error Message Using UNC with 10+ Characters**  
**Article ID: Q105514**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

If you attempt to use a universal naming convention (UNC) path in Microsoft Access in which the server name is longer than 10 characters or contains spaces, you get the following error message

```
\\<servername>\<sharename> isn't a valid path
```

even though valid UNC server names can be up to 15 characters and can contain spaces.

For example, the path

```
SystemDB=\\s234567890\ANYSHARE\SYSTEM.MDA
```

works correctly, but the following UNC paths cause the above error message to appear:

```
SystemDB=\\s2345678901\ANYSHARE\SYSTEM.MDA
SystemDB=\\s2345 7890\ANYSHARE\SYSTEM.MDA
```

CAUSE

=====

This problem is not caused directly by Microsoft Access, but rather by an external host, such as Microsoft Windows for Workgroups, or by a network problem. Microsoft Access does not parse UNC paths; it hands them off to external processes.

RESOLUTION

=====

One way to work around this problem is to connect the server to a logical drive before using Microsoft Access and then reference the logical drive instead of the server.

For example, you want to use a server named \\MY SERVER\MYSHARE, and you are using Microsoft LAN Manager. To reference this server, type:

```
NET USE F: \\MY SERVER\MYSHARE
```

In your MSACCESS.INI file, reference it as follows:

```
SystemDB=F:\SYSTEM.MDA
```

Additional reference words: 1.00 1.10 2.00 WinIss attaching

KBCategory: kbenv  
KBSubcategory: EvnOs



**PRB: Invalid Reference to Field '<Value>' Using SetValue**  
**Article ID: Q96580**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

If the Expression argument of a SetValue macro action evaluates to a text expression beginning with a letter, you may receive the following alert message when a SetValue macro action attempts to set the value of a control:

! Invalid reference to field '<value>'

CAUSE

=====

An Expression argument of a SetValue macro action that begins with a leading equal sign (=) is understood to be a field name.

RESOLUTION

=====

Remove the leading equal sign in the Expression argument.

Steps to Reproduce Behavior

-----

1. Create a macro named My Macro with one action as follows:

| Macro Name | Action   |
|------------|----------|
| My Macro   | SetValue |

My Macro Actions

-----

SetValue  
Item: Field0  
Expression: ="A023bbb"

2. Create a new unbound blank form.
3. Place a command button on the form by dragging the macro named My Macro from the Database window onto the form.
4. Create a text box on the form. Verify that the ControlName is Field0.
5. Switch to Form view.
6. Click the command button to run the macro.

You will receive the following error message:

Invalid reference to field 'A023bbb'

REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, pages 538-539

Additional reference words: 1.00 1.10 2.00 setvalue error

KBCategory: kbusage kberrmsg

KBSubcategory: McrArg

**PRB: Irregular Characters in Attached dBASE IV Memo Field**  
**Article ID: Q88647**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

If you have an attached dBASE IV table with a memo field, a strange character resembling an "i" (ASCII 161) appears randomly throughout the memo field when viewed in Microsoft Access.

CAUSE

=====

This problem is the result of the way dBASE IV handles its memo fields and how this translates when you view the memo field in Microsoft Access.

When you open a memo field in dBASE IV, you are presented with an editor that is fixed at 65 characters per line and cannot be adjusted. In order to format your memo data so that it fits comfortably in the memo editor, dBASE IV inserts invisible characters that act as carriage returns. These characters become an actual part of the data in the .DBF file. Microsoft Access does not use these characters to format the memo data because the size of a memo control is adjustable, so they are ignored and consequently appear in the memo control on your form.

RESOLUTION

=====

You can create an Access Basic procedure to remove these characters from the memo field. The procedure should go through each record in the table and copy the memo field, character by character, to a temporary holding area, ignoring all instances of Chr(161). Then copy the corrected string back into the memo field.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

dBASE IV is manufactured by a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmXbase



**PRB: LimitToList Enforced Even When Set to No**  
**Article ID: Q122242**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you type a unique value in a combo box and then press ENTER, you receive the following error message even though the combo box's LimitToList property is set to No:

The text you enter must match an entry in the list

CAUSE

=====

Even if the combo box's LimitToList property is set to No, Microsoft Access limits entries to those in the combo box if the first column displayed in the combo box is not the bound column. This behavior occurs because Microsoft Access stores the bound column value when you enter a nonbound column selection. If you could enter a new value for the nonbound column, Microsoft Access would have no value to store for the bound column.

RESOLUTION

=====

In order to allow unique entries in a column box, make the first nonhidden column in the combo box the bound column.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create a new, unbound form.
3. Add a combo box with the following properties to the form:

RowSourceType: Table/Query  
RowSource: Product Names and IDs  
ColumnCount: 2  
BoundColumn: 2  
LimitToList: No

Note that the bound column is column 2, which contains the Product ID field. The Product Name field will appear in the first column.

4. View the form in Form view.
5. Type "New Cola" (without quotations marks) in the combo box and then press ENTER. You will receive the error message stated above.

#### REFERENCES

=====

For more information about the LimitToList property, search for "LimitToList" then "LimitToList Property" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: FmsCmbo

## **PRB: Locking First Record Also Locks Last Record**

Article ID: Q109347

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

If you set the Default Record Locking option to Edited Record in the Multiuser category of the Options dialog box, editing the first record may lock both the first and the last record in the table.

### CAUSE

=====

Some data from the first record may be stored in the last data page in the table. The entire data page is locked, which locks both the first and the last record.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. Start Microsoft Access on two different computers and open the sample database NWIND.MDB.
2. On both computers, choose Options from the View menu.
3. On both computers, select Multiuser from the Category box.
4. On both computers, select Edited Record in the Default Record Locking box.
5. On one of the computers, open the Customers table. Edit the first record by adding enough characters to each field so that each field is the maximum allowed length. This will force Microsoft Access to store the data on the last data page.
6. On the other computer, open the Customers table, move to the last record, and try to edit the record.

Note that compacting your database periodically will keep the data in order on the data pages, which should help prevent this problem from occurring.

Note also that if all the records in a table are locked when you choose the Edited Record option it means that all the records fit on one data page,

and therefore if any record is locked, then all the records are locked.

#### REFERENCES

=====

For more information on record locking, query on the following words here in the Microsoft Knowledge Base:

record and locking

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: GnlMu



**PRB: Mail Label Wizard Cuts Off First Field at Top of Page**  
**Article ID: Q90858**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The first line on each page in a mailing label created with the ReportWizard may not print or show in Print Preview. Forms and reports may also exhibit this behavior.

CAUSE

=====

All printers have non-printable areas. Microsoft Access reads the default Windows printer driver to determine what to print or show in Print Preview. If the label layout exceeds the printable area, the labels will be truncated and the first line will not print.

RESOLUTION

=====

If you experience this problem, you should increase the height of your page header to align your labels. In some cases the first row of labels will be unusable.

STATUS

=====

This behavior is by design.

REFERENCES

=====

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

Additional reference words: 1.00 1.10 2.00 wizards

KBCategory: kbusage

KBSubcategory: WzOthr

**PRB: Make-Table Query Generates "Invalid" Error Message**  
**Article ID: Q109365**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you are running a make-table query, an error message similar to

'queryname' isn't a valid name

will be displayed if you supply an invalid name for the table to be created.

CAUSE

=====

The name of the table to be created by the make-table query is invalid.

The error message is misleading in stating that the name of the query is invalid. It is the name of the table to be created that is invalid.

RESOLUTION

=====

To correct the problem, supply a valid table name in the Query Properties dialog box. Valid names for objects in Microsoft Access can be up to 64 characters long, and can contain any combination of letters, numbers, spaces, and special characters except the period (.), exclamation point (!), and bracket ([]) characters. Valid names cannot include leading spaces or control characters (ASCII characters 0-32).

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Open the Customer List query in Design view.
3. From the Query menu, choose Make Table.
4. Enter the name "5!" (without the quotation marks) in the Table Name box.
5. Save the query as Test Query.
6. Close the query.
7. Run the Test Query query.

An error message will be displayed.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryMktbl

## **PRB: Maximizing One Form Maximizes All Forms**

**Article ID: Q121410**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

When you maximize a form, all other open forms are also maximized. You cannot maximize one form independent of the other open forms.

### CAUSE

=====

Microsoft Access is a multiple document interface (MDI) application. The default behavior for an MDI document is for all child windows to be maximized when one is maximized. This behavior occurs in many applications. For example, if you maximize one group window in Windows Program Manager, any other group window that you select (using the Window menu) will also be maximized. Or, if you maximize a document window in Microsoft Word for Windows, all other document windows will be maximized as well.

### RESOLUTION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual in version 2.0.

There are two ways to work around this behavior:

- Forms whose PopUp property is set to Yes are not MDI child windows. Pop-up forms float on top of other forms, and are not maximized when an MDI child window is maximized.
- You can simulate maximizing a form by sizing it as large as possible in a restored state. The following example demonstrates how to create and use a sample Sub procedure called MaximizeRestoredForm to restore a form if it is maximized, and then move the form to the upper left corner of the Microsoft Access client area window and size it as large as possible.

NOTE: In the following sample code, an underscore ( `_` ) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

1. Create a new module and enter the following lines in the Declarations section of the module:

Option Explicit

```

Type Rect
 x1 As Integer
 y1 As Integer
 x2 As Integer
 y2 As Integer
End Type

Declare Sub GetWindowRect Lib "User" (ByVal hWnd As Integer, _
 lpRect As Rect)
Declare Function IsZoomed Lib "User" (ByVal hWnd As Integer) _
 As Integer
Declare Sub ShowWindow Lib "User" (ByVal hWnd%, ByVal nCmdShow%)
Declare Sub MoveWindow Lib "User" (ByVal hWnd As Integer, _
 ByVal X As Integer, ByVal Y As Integer, ByVal nWidth As _
 Integer, ByVal nHeight As Integer, ByVal bRepaint As Integer)
Declare Function GetParent Lib "User" (ByVal hWnd As Integer) _
 As Integer

Const SW_MAXIMIZE = 3
Const SW_SHOWNORMAL = 1

```

2. Enter the following Sub procedure in the module:

```

Sub MaximizeRestoredForm (F As Form)
 Dim MDIRect As Rect

 ' If the form is maximized, restore it.
 If IsZoomed(F.hWnd) <> 0 Then
 ShowWindow F.hWnd, SW_SHOWNORMAL
 End If

 ' Get the screen coordinates and window size of the
 ' MDIClient window.
 GetWindowRect GetParent(F.hWnd), MDIRect

 ' Move the form to the upper left corner of the MDIClient
 ' window (0,0) and size it to the same size as the
 ' MDIClient window.
 MoveWindow F.hWnd, 0, 0, MDIRect.x2 - MDIRect.x1, _
 MDIRect.y2 - MDIRect.y1, True
End Sub

```

3. To automatically simulate maximizing a form when the form is opened, set the form's OnLoad property to the following event procedure:

```

Sub Form_Load()
 MaximizeRestoredForm Me
End Sub

```

4. To simulate maximizing a form called MyForm, use the following statement in a function or subroutine:

```

MaximizeRestoredForm Forms!MyForm

```

STATUS  
=====

This behavior is by design.

#### REFERENCES

=====

For more information about pop-up forms, search for "PopUp" then "Creating Pop-Up Forms and Dialog Boxes" using the Microsoft Access version 2.0 Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsProp

## **PRB: Memo and OLE Fields in a Snapshot Are Not Static**

Article ID: Q109717

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

Data contained in a Snapshot that includes a Memo or OLE field can be changed if the data in the underlying table for the Snapshot changes.

### CAUSE

=====

Microsoft Access stores the contents of Memo and OLE fields differently than other data types. Memo and OLE field data is not stored in the data page holding the actual record. Instead, the data page holding the record contains an address pointer that specifies the data page(s) containing the data.

When you create a Snapshot, a static copy of the data page holding the record is copied. This copy contains only the address of the Memo or OLE information. If the actual Memo or OLE information is changed, the Snapshot will look to the same data page and will find the changed information.

### STATUS

=====

This behavior is by design. The actual Memo or OLE information is not loaded into the Snapshot in order to avoid performance degradation.

### MORE INFORMATION

=====

A Snapshot is a copy of the record source data at the point in time when the Snapshot was created. Unlike the data in a Dynaset, Snapshot data is not linked to data in the underlying tables. The Snapshot data does not change as its underlying tables change.

### Steps to Reproduce Behavior

-----

You can use the following sample Access Basic function to demonstrate that a Memo field contained in a Snapshot can be altered:

1. Open the sample database NWIND.MDB.
2. Create the following new function in a module:

```
Function TestSnapShot ()
 Dim SS As Snapshot
 Dim DS As Dynaset
```

```

Dim DB As Database

Set DB = CurrentDB()

' Create a Snapshot and Dynaset of the Categories table.
Set SS = DB.CreateSnapshot("Categories")
Set DS = DB.CreateDynaset("Categories")

' Move to the first record.
SS.MoveFirst
DS.MoveFirst

' Print the contents of the Category Name and Description
' .. fields from both record sets. Category Name is a
' .. Text field and Description is a Memo field.
Debug.Print
Debug.Print "Snapshot:"
Debug.Print " Category Name: " & SS![Category Name]
Debug.Print " Description: " & SS![Description]
Debug.Print "Dynaset:"
Debug.Print " Category Name: " & DS![Category Name]
Debug.Print " Description: " & DS![Description]

' Alter the Description and Category Name fields using
' .. the dynaset.
DS.Edit
 DS![Description] = "My Description"
 DS![Category Name] = "My Category"
DS.Update

' Print the new contents of the recordsets.
Debug.Print
Debug.Print "Snapshot:"
Debug.Print " Category Name: " & SS![Category Name]
Debug.Print " Description: " & SS![Description]
Debug.Print "Dynaset:"
Debug.Print " Category Name: " & DS![Category Name]
Debug.Print " Description: " & DS![Description]

End Function

```

3. Type the following in the module's Immediate window, and then press ENTER:

```
? TestSnapShot()
```

The following text will be displayed in the Immediate window:

```

Snapshot:
 Category Name: Beverages
 Description: Soft drinks, coffees, teas, beer, and ale
Dynaset:
 Category Name: Beverages
 Description: Soft drinks, coffees, teas, beer, and ale

Snapshot:
 Category Name: Beverages

```



Description: My Description  
Dynaset:  
Category Name: My Category  
Description: My Description

Note that the second Description for the Snapshot reads "My Description," which is the same as the change made by the Dynaset. Note also that the second Category Name for the Snapshot is unchanged, as Text fields are static in a Snapshot.

#### REFERENCES

=====

Microsoft Access "Language Reference," versions 1.0 and 1.1, page 91

Additional reference words: 1.00 1.10 2.00

KBCategory: kbprg

KBSubcategory: PgmObj

## **PRB: Memory or Disk Space Problems Using MS-DOS 6.0**

**Article ID: Q103991**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

When you are working in Microsoft Access, you receive an error message stating that you are out of memory. The message may be accompanied by other messages stating that you have insufficient space in your TEMP directory, or that you have insufficient disk space.

### CAUSE

=====

If you are running MS-DOS 6.0, these error messages may be due to the way the memory on your computer is configured, or to the use of file compression on your hard disk.

### RESOLUTION

=====

Check each of the items described below and make any necessary changes.

### MORE INFORMATION

=====

#### DoubleSpace

-----

MS-DOS 6.0 includes its own disk compression program called DoubleSpace. This program makes your hard disk seem larger than it actually is by compressing the files on the disk. The DoubleSpace program has an option to let you specify the compression ratio that will be used. You can increase the default compression ratio that DoubleSpace sets up, but increasing this ratio usually does not result in additional usable space on your hard disk.

The average file compression ratio is about 1.5 to 1. With this ratio, a 20 MB disk could actually yield about 30 MB of usable space. If you set the DoubleSpace compression ratio to 16 to 1, the disk will show about 320 MB of usable space, but will still only actually provide about 30 MB of space.

If you have set up a high compression ratio, the hard disk may show far more free space than can actually be used. If you attempt to install Microsoft Access to such a disk, you can run out of disk space during the installation process.

If you are using DoubleSpace, you can check your compression ratio by typing "dblspac" (without the quotation marks) at the MS-DOS prompt. From the Drive menu, choose Info. If the compression ratio is set larger than the estimated compression ratio, choose Ratio and set the new estimated compression ratio to match the estimated compression ratio. This will give

you the most accurate information about the amount of actual usable free disk space.

#### MemMaker

-----

MS-DOS 6.0 includes a memory-management utility called MemMaker that helps optimize the configuration of your computer's memory. This utility may alter two commands that can affect Microsoft Access.

The first item is the FILES=<xx> line in your CONFIG.SYS file. Microsoft Access requires that <xx> be 50 or greater, but the MemMaker utility sometimes sets this line to less than 50. Check your CONFIG.SYS file with any text editor and make sure that the line reads FILES=50.

The second item is the DEVICE=EMM386.EXE line in your CONFIG.SYS file. The MemMaker utility sometimes adds the HIGHSCAN parameter to this line, but this parameter can cause conflicts on some computers. You may want to remove this parameter and any i=<xxxx>-<xxxx> parameters, and make sure that the NOEMS switch is present. For example, the line might read DEVICE=EMM386.EXE NOEMS.

#### Undelete

-----

If you are running the Delete Sentry option of the Microsoft Undelete program, your hard disk may show an incorrect amount of disk space. This is because Delete Sentry makes a hidden copy of all deleted files. These hidden files are stored for seven days, and are not counted when you use the DIR command to show available disk space.

If you run out of disk space when you are installing Microsoft Access, use the DIR /A command to show all files. This command will show you any hidden files, including hidden files stored by the Delete Sentry. You can delete the hidden files to free up disk space, but these files will no longer be guaranteed to be recoverable. \\* \\* If you run into unusual problems, one means of troubleshooting may \\* be to switch the SET TEMP= statement to point to the uncompressed \\* drive.

#### REFERENCES

=====

Microsoft MS-DOS 6.0 "User's Guide"

Additional reference words: 1.00 1.10 2.00 WinIss kberrmsg

KBCategory: kbenv

KBSubcategory: EvnOs

**PRB: MS Access Converts Extended Chars in Attached FoxPro DBF**  
**Article ID: Q113303**

-----  
The information in this article applies to:

- Microsoft Access versions 1.1 and 2.0
  - FoxPro for Windows versions 2.5a and 2.5b
- 

SYMPTOMS

=====

Extended characters (ANSI characters greater than 128) that you enter in an attached FoxPro table are converted to other characters when you commit the record.

CAUSE

=====

The ISAM that Microsoft Access uses to attach FoxPro databases reads a Code Page flag in the header of the FoxPro database. This flag tells the ISAM how to handle OEM to ANSI conversions. In this case, the flag is set to convert ANSI characters to OEM characters.

RESOLUTION

=====

To correct this problem, add the following line to the FoxPro configuration file CONFIG.FPW:

```
CODEPAGE=AUTO
```

This line tells the ISAM not to perform ANSI to OEM conversions.

MORE INFORMATION

=====

The resolution above will not correct character conversions that were made prior to your changing the CONFIG.FPW file. To correct conversions that have already been made, port the data from the existing FoxPro database to a new FoxPro database. Consult the FoxPro documentation for information on how to port data from an existing database to a new database.

REFERENCES

=====

Microsoft Access "Language Reference," version 1.1, Appendix A, page 505

Additional reference words: 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmXbase

**PRB: Network Problems May Cause "Segment Load Failure" Error**  
**Article ID: Q109730**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You receive a "Segment Load Failure" error message or another memory-conflict error message.

CAUSE

=====

This type of error can occur if Microsoft Access is running on a network or is accessing data on a network, and the network server stops functioning correctly.

A segment load failure implies that there was a problem moving information from secondary storage, such as a network server disk, to main memory. If a network server is no longer functioning correctly, expected information may no longer be available to Microsoft Access.

RESOLUTION

=====

The following suggestions may help you to diagnose the problem:

- Make a backup of all critical data and run disk diagnostic software, such as the MS-DOS CHKDSK command, to check the integrity of the workstation and server disk drives.
- If the workstation has a complete Microsoft Access installation, quit and then restart Microsoft Access on the workstation and see if it operates correctly without using the network for any services. If the problem continues with Microsoft Access operating in isolation, begin checking the workstation for memory conflicts.
- If the database is being accessed off a network server, consider copying it to the local workstation to isolate whether the problem is workstation or server specific.
- If the server is required, just start the server and the Microsoft Access workstation to test Microsoft Access. If the problem does not occur at this point, you can start and diagnose other workstations one by one.

REFERENCES

=====

For more information on how to diagnose segment load failures, query on the following words here in the Microsoft Knowledge Base:

segment and load and failure and troubleshoot

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr

**PRB: New Label Controls Inherit FormWizards' Label Text Align**  
**Article ID: Q89604**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

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.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsProp

**PRB: No AutoRepeat with GoToRecord/Next and DoMenuItem**  
**Article ID: Q95645**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The AutoRepeat property for command buttons does not work in conjunction with the GoToRecord/Next macro action for moving to successive records on a form. In other words, if your form uses command buttons instead of the record selector buttons to move from record to record, the AutoRepeat property does not repeatedly execute a macro that performs a GoToRecord/Next action.

Also, the AutoRepeat property fails under the same circumstances when using the DoMenuItem/Form/Records/GoTo/Next macro action. To re-execute the macro, you must click the command button again.

CAUSE

=====

This is correct behavior for these macro actions. Here are two points that need clarification before the cause can be explained:

1. OnPush property: The macro under the OnPush property of a command button normally will not execute until you push and let up the command button with the mouse pointer in the Command Button's region. The exception is when the command button's AutoRepeat property is set to Yes. In this case, the macro under the OnPush property of the command button executes when the button is pushed and the mouse pointer is in the Command Button's region.
2. Focus: Several macro actions change the focus. For example, GoToControl, GoToRecord, OpenForm, OpenTable, and ApplyFilter all change the focus.

Therefore, if a command button's OnPush property macro changes the focus to, for example, the next record, then the command button has lost the focus. That is, when you click the command button, the click does not apply to that command button anymore because the focus changed. It's as if you had moved the mouse pointer off the command button while pressing the mouse button, as if the command button moved out from under the mouse pointer right before you pressed the mouse button.

STATUS

=====

This behavior is by design.

MORE INFORMATION



=====

## Steps to Reproduce Behavior

-----

1. Start Microsoft Access.
2. Open the sample NWIND.MDB database.
3. Create a new macro with two actions (OpenTable and GoToRecord). Use the following as a guide to assign action arguments to each action:

### OpenTable

-----  
Table Name: Categories  
View: Datasheet  
Data Mode: Edit

### GoToRecord

-----  
Object Type: Table  
Object Name: Categories  
Record: Next

4. Close, save, and name the macro.
5. Open a new form and click the Blank Form button.
6. Drag the macro you just created onto the blank form to create a command button.
7. Set the AutoRepeat property of the command button to Yes. Notice the command button's OnPush property is set to the name of the macro.
8. Change to Form view to browse the form. As a result, you should see a single command button on the form.
9. Click the command button. Now you should see the Categories table.
10. Size the Categories table so that both the form and the table fit on the screen and you can see the VCR button on the table.

Each time you click the command button, the record selector moves to the next record. When you click and hold down the command button, you would expect Microsoft Access to scroll rapidly to the last record in the table, but this does not happen. Instead, the record selector moves to the next record only, regardless of the AutoRepeat property setting of the command button.

However, if you place a macro that uses a different macro action such as Beep on the form, the AutoRepeat action works as expected. This is because Beep does not change the focus.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: McrActn



**PRB: No Password Prompt Using Microsoft Access .MDB with ODBC**  
**Article ID: Q112346**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You are not prompted for a user name and password as you expect when you are using ODBC from an application such as Microsoft Query (which is included with Microsoft Excel) to access data in a Microsoft Access database.

CAUSE

=====

You have not specified a Microsoft Access SYSTEM.MDA file to be used with the Microsoft Access ODBC data source. The SYSTEM.MDA file contains a list of users, groups, and passwords.

If you do not specify a SYSTEM.MDA file, ODBC will open your Microsoft Access database (.MDB) file without a password as a member of the Users group, with the permissions of the Users group.

RESOLUTION

=====

Edit the Microsoft Access ODBC data source to use the appropriate Microsoft Access SYSTEM.MDA file. Use the following steps to alter a Microsoft Access ODBC data source to use a SYSTEM.MDA file:

1. Start Microsoft Windows Control Panel.
2. Double-click the ODBC icon.
3. In the Data Sources box, select the Microsoft Access data source you are using, and then choose the Setup button.
4. Choose the Options button.
5. In the System Database box, select the Database option button instead of the None option button.
6. Choose the Select Database button.
7. Find and select the appropriate SYSTEM.MDA file. This file will probably be in your Microsoft Access directory.
8. Choose the OK button.

MORE INFORMATION

=====

If ODBC opens your Microsoft Access database without a password, this does not mean that you have circumvented Microsoft Access security.

Microsoft Access employs user-level security on databases, not share-level security. In user-level security, you grant permissions to users of your database resources (such as tables and forms). In share-level security, you password-protect your resources.

If ODBC opens your Microsoft Access database without a password, you will log on to the database as a member of the Users group and will only have permissions to objects in the database that are available to members of the Users group.

If members of the Users group do not have permissions to edit data in a table, you will not be able to edit that table using ODBC with no SYSTEM.MDA file specified.

In user-level security, the combinations of users and their permissions define database security. Information about users, groups, and passwords is stored in a SYSTEM.MDA file. Information about who has permissions on database objects is stored with each database in the Microsoft Access database (.MDB) file.

Passwords are stored in a SYSTEM.MDA file, and are used only to prevent a user from logging on to a database as another user who has different permissions on database objects. Members of the Users group may not have permissions to modify data in tables, but a user with a password may have permissions to modify data in the tables.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 25, "Administering a Database System," pages 612-624

Microsoft Access "User's Guide," version 1.1, Chapter 25, "Administering a Database System," pages 616-633

For more information about security, search for "security: an overview," or "security: basics" then view each of the topics using the Microsoft Access Help menu.

For information on the highly recommended "SecurityWizard and White Paper" Application Note, query on the following words here in the Microsoft Knowledge Base:

securitywizard and white and paper

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ScrtOthr

**PRB: No Results When Calling ORACLE 7.0 Stored Procedure**  
**Article ID: Q120491**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Visual Basic version 3.0
- 

SYMPTOMS

=====

Although your Microsoft Access SQL pass-through query calls a valid ORACLE version 7.0 stored procedure, no results are returned.

CAUSE

=====

This behavior occurs because the current ORACLE ODBC driver uses ODBC specification 1.0, and not 2.0. ODBC specification 2.0 is required to support IN-OUT parameters and the ability to return values in an interoperable fashion.

RESOLUTION

=====

If the stored procedure that you are calling requires only IN parameters, you should be able to get results back, since the ODBC 1.0 specification supports this functionality. Stored procedures requiring OUT or IN-OUT parameters cannot be called from Microsoft Access or Microsoft Visual Basic.

MORE INFORMATION

=====

ORACLE version 7.0 stored procedures are PL/SQL blocks (that is, they are not queries). PL/SQL procedures are similar to PASCAL procedures. They, like SQL Server procedures, can accept IN, OUT, or IN-OUT parameters, and can contain functions that return values.

The only procedures that you can call in ORACLE version 7.0 from Microsoft Access are those that require IN parameters. To do this, construct the statement as

```
{call StoredProcXX(Arg1, ...)}
```

by concatenating the name of the procedure and its arguments.

There is no concept of a stored function in the ORACLE ODBC specification, therefore you cannot call stored functions. The same is true for ORACLE packages which are ADA-like constructs (data and methods wrapped in a package). To invoke a procedure in a package in ORACLE version 7.0, preface the procedure or function with the package name. For example:

```
PackageAA.ProcedureXX(Arg1, Arg2,...)
```

This is not possible with the current ORACLE ODBC driver.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: ObcOracl

**PRB: No Way to Specify Page Range Using Output To Command**  
**Article ID: Q120059**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

You can use the Output To command in Microsoft Access to save the output of a table, query, form, report, or module in one of the following file formats:

- Microsoft Excel (.XLS)
- Rich Text Format (.RTF)
- MS-DOS Text (.TXT)

When you use the Output To command with a report, there is no way to specify the range of pages that you want. The entire report is output to the file.

RESOLUTION

=====

Base the report on a query that limits the report's output to the records you want. For example, use a query that limits the records in the report to the top 10 records.

MORE INFORMATION

=====

When you choose Output To from the File menu, the All and Selection option buttons are unavailable, since specifying a range for the output is not supported.

REFERENCES

=====

For more information about the Output To command, search for "Output To" then "Output To Command" using the Microsoft Access Help menu.

Additional reference words: 2.00 export grayed

KBCategory: kbusage

KBSubcategory: FmrHowto

## **PRB: NotInList Event Is Not Triggered**

**Article ID: Q123867**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

### SYMPTOMS

=====

The Access Basic code or macro specified in a combo box's OnNotInList property setting is not triggered when you enter a value in the combo box that is not in the combo box's list.

### CAUSE

=====

The NotInList event is triggered only if the value entered in the combo box cannot be uniquely associated with any item in the combo box's list, and if the combo box's LimitToList property is set to Yes.

### RESOLUTION

=====

Make sure the combo box's LimitToList property is set to Yes.

Also, check to see if the value you entered in the combo box can be uniquely associated with an item in the combo box's list. For example, assume a combo box with the following list:

Beverages  
Condiments  
Confections  
Dairy Products

If you type the letter "D" in the combo box, the "Dairy Products" item will be selected in the combo box and the NotInList event will not be triggered. If you type the letter "C" in the combo box, the NotInList event will be triggered, because "C" is not uniquely associated with an item in the list. If you type "COND" in the combo box, the "Condiments" item will be selected and the NotInList event will not be triggered.

### MORE INFORMATION

=====

#### Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create a blank, new form. Place a combo box with the following properties on the form:

RowSourceType: Table/Query  
RowSource: Categories



ColumnCount: 2  
ColumnWidths: 0;1  
BoundColumn : 2  
LimitToList: Yes  
AutoExpand: No

3. Set the combo box's OnNotInList property to the following event procedure:

```
Sub Field0_NotInList (NewData As String, Response As Integer)
 MsgBox "This item is not in the combo box."
End Sub
```

4. View the form in Form view. Type "CON" (without quotation marks) in the combo box and then press the TAB key. The "This item is not in the combo box" message box appears as you expect.
5. Type "COND" (without quotation marks) in the combo box and then press TAB. Note that "Condiments" is automatically selected in the combo box and that the NotInList event is not triggered.

#### REFERENCES

=====

For more information about the OnNotInList property, search for "OnNotInList" then "OnNotInList Property" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsCmbo

**PRB: Office Manager 4.2 Will Not Load Microsoft Access 2.0**  
**Article ID: Q113931**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

After you install the Microsoft Office Manager version 4.2, the button for Microsoft Access generates the following error message:

Cannot find 'MSACCESS.EXE'. Please start the application another way. While the application is running, click on the toolbar button or choose the menu command again.

RESOLUTION

=====

If Microsoft Access will not start properly after you follow the suggestion to start Microsoft Access and then click the Office Manager button, contact the Microsoft Office Product Support group to receive up-to-date information about this issue. You can reach the Microsoft Office Product Support group at (206) 635-7056.

There is also a Microsoft Office Setup section in the Microsoft Excel forum on CompuServe where you can get answers to questions about this issue.

Additional reference words: 2.00

KBCategory: kbenv

KBSubcategory: EvnOs

**PRB: OLE MS Excel, Word Do Not Close When Form Closed**  
**Article ID: Q96465**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Microsoft Access will not close some OLE servers when you close a table, form, or report while editing an OLE object.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Microsoft Access will not close Multiple Document Interface (MDI) applications, even if they were started as an OLE server for an object stored in Microsoft Access. An MDI application can edit and display multiple documents at once.

Microsoft Excel version 3.0 and Microsoft Word version 2.0 for Windows are examples of applications that will be left running if you double-click an existing embedded object in Microsoft Access and then switch back to Microsoft Access and close the table, form, or report without closing the embedded object.

Steps to Reproduce Behavior

-----

1. Open a form in Microsoft Access, and choose Insert Object from the Edit menu. In the list of available OLE servers, select Microsoft Excel Worksheet.
2. Create the object (worksheet) you want to embed. Once the object is completed, close Microsoft Excel. Save the Access form and close it.
3. Reopen the form and double-click the embedded object that you just created. Once Microsoft Excel opens and the object is displayed, minimize Microsoft Excel and close the Microsoft Access form. Microsoft Excel will still be running. You must activate Microsoft Excel and close it.

If you repeat these steps with applications such as Paintbrush, Microsoft Draw, or Note-It (these are not MDI applications), you will note that these servers are closed when the object is closed.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbole

KBSubcategory: IntpOle



**PRB: OnExit Macro Runs with AfterUpdate Macro in Option Group**  
**Article ID: Q109317**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Both the macro in the AfterUpdate property and the macro in the OnExit property of an option group will run when you update the option group if the macro action in the AfterUpdate property is an ApplyFilter action.

RESOLUTION

=====

Do not use a macro in the OnExit property of an option group if the option group's AfterUpdate property runs a macro that contains an ApplyFilter action.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create a new form based on the Employees table. Add the Employee ID and Last Name fields to the form.
3. Create an option group on the form with two toggle buttons and the following properties:

Option Group: Test  
  AfterUpdate: Macro1  
  OnExit: Macro2  
Option Button: Button1  
  Option Value: 1  
Option Button: Button2  
  Option Value: 2

4. Save the form as Form1.
5. Create the following two new macros:

| Macro Name | Condition              | Action      |
|------------|------------------------|-------------|
| Macro1     | Forms![form1]![test]=1 | ApplyFilter |
|            | Forms![form1]![test]=2 | ApplyFilter |
| Macro2     |                        | MsgBox      |

Macro1 Actions

-----

```
ApplyFilter
 Where: [employee id]=5
ApplyFilter
 Where: [employee id]=7
```

Macro2 Actions

```

MsgBox
 Message: Test
```

6. Toggle between the two option buttons. In Microsoft Access version 1.x, both the ApplyFilter and MsgBox actions are run. In Microsoft Access version 2.0, the following error message is displayed:

Field or record can't be saved while it's being validated.

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbusage  
KBSubcategory: DcmNwd

**PRB: Only Admins Group Members Can List Groups They Belong To**  
**Article ID: Q124240**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

You can use Access Basic code to determine to which groups a user belongs, but only members of the Admins group can view this information. For example, running the following sample function will return the current user's name and a list of the groups to which the current user belongs:

```
Function GetGroups ()
 Dim W As Workspace, US As String, U As User, i As Integer, j As Integer

 Set W = DBEngine.Workspaces(0)
 US = CurrentUser()
 Set U = W.Users(US)

 Debug.Print U.Name
 For j = 0 To U.Groups.Count - 1
 Debug.Print " " & U(j).Name
 Next
End Function
```

If the current user is not a member of the Admins group, the following error message will be generated:

```
Record(s) can't be read; no Read Data permissions on 'MSysGroups'
```

RESOLUTION

=====

To work around this behavior, you must log in to a new workspace as a member of the Admins group and pass the name of the user whose groups you want to list. The following sample function demonstrates this method:

```
Function GetGroups2 ()
 Dim W As Workspace, US As String, U As User, i As Integer

 Set W = DBEngine.CreateWorkspace("NewWS", "MyAdmin", "MyPassword")
 US = CurrentUser()
 Set U = W.users(US)

 Debug.Print U.Name
 For i = 0 To U.Groups.Count - 1
 Debug.Print " " & U(i).Name
 Next
End Function
```

NOTE: This function assumes the existence of a user named MyAdmin who is a member of the Admins group in the current system database with an account

password of MyPassword.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 14,  
"Securing Your Applications," pages 340-344

Additional reference words: 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: ScrPerm



**PRB: OpenRecordset on Recordset Causes "Invalid Operation"**  
**Article ID: Q120914**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you use the OpenRecordSet method on a Recordset created with the RecordSetClone property of a form, you receive the error message:

Invalid operation

RESOLUTION

=====

Instead of creating the Recordset using the RecordSetClone property of the form, base the Recordset on the table or query that is the form's record source. Note that if you opened the form with a filter or a Where clause, or if you later apply a filter to the form, the changes will not be reflected in the Recordset. In such a case, you might want to save the Where condition in a global variable, and then use the variable with the Filter property of the Recordset object.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Open the Customers form in Form view.
3. Create a new module and enter the following sample function:

```
Function TestRecordset ()

 Dim recordset1 As Recordset
 Dim recordset2 As Recordset

 Set recordset1 = Forms!Customers.recordsetclone
 Set recordset2 = recordset1.OpenRecordset()

End Function
```

4. From the View menu, choose Immediate Window.
5. Type the following line in the Immediate window and then press ENTER:

```
?TestRecordset()
```

You will receive the error message stated above.

## REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 11,  
"Working with Sets of Records," pages 241-267

Additional reference words: 2.00

KBCategory: kbprg kberrmsg

KBSubcategory: MdlRcd

**PRB: Operation Stops when Editing Attached SQL Tables**  
**Article ID: Q96897**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Microsoft Access may return an inconsistent value for a floating point field in an SQL table when it checks the value of the field before editing and before updating. This can result in the error message

Data Has changed: Operation Stopped

when you are trying to update a record in an attached SQL table.

CAUSE

=====

The problem occurs if an SQL field data type is set to Float and the table does not contain a TimeStamp field, or when an SQL indexed field data type is set to either DateTime or Float.

RESOLUTION

=====

1. Remove the index from any fields with the DateTime or Float data types.

To obtain information on any index within a SQL table, you can run the following system stored procedure within the System Administrator Facility (SAF):

```
sp_helpindex <Table Name>
```

To remove an index, use the following command:

```
DROP INDEX <Table_Name.Index_Name>
```

Note that you must be logged into the SQL Server as either the table owner or the System Administrator.

2. If other fields in the table have the Float data type, you must insert a TimeStamp field in the table. If a field with this data type is present, Microsoft Access returns a consistent value for fields of this data type and will use the value in this field to verify whether the record has been modified.

This can be accomplished by performing the following command within SAF:

```
ALTER TABLE <Table Name> ADD TimeStamp timestamp
```

Note that you must be logged in as either the database owner or the System Administrator to run the above command.

Additional reference words: 1.00 1.10 2.00 SQL

KBCategory: kbusage

KBSubcategory: ObcOthr

**PRB: ORA-00942 Error Opening Attached ORACLE Table**  
**Article ID: Q121833**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you attach an ORACLE table using one login ID and password, and then attach a second ORACLE table using a different login ID and password, you may receive the following error messages when you try to open the second attached ORACLE table:

- Couldn't execute query; couldn't find linked table
- [Page Ahead][ODBC Oracle Driver][Oracle OCI]ORA-00942: table or view does not exist.[#942]
- Can't open table in Datasheet view

CAUSE

=====

This behavior is caused by the way the Microsoft Jet database engine shares connections to remote databases. You will receive the error messages stated above under the following conditions:

1. You have two user accounts set up on the ORACLE server. The first account does not have select rights granted on at least one of the tables created by the second user account.
2. You attach and open any table using the first user account.
3. Using the second user account, you close the first table and attach a table that is not accessible by the first user account.

When you try to open the second table, Microsoft Access will try to open the table using the first account's login information.

RESOLUTION

=====

Create another ODBC data source for the same server and database. When you attach the second table, select this new data source. Choosing the new data source will create a separate connection to the same server, preventing the Microsoft Jet database engine from sharing the connection.

STATUS

=====

This behavior is by design. Changes to this behavior are being reviewed and will be considered for inclusion in a future release of Microsoft Access.

MORE INFORMATION

=====

ORACLE is manufactured by Oracle Corporation, a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 2.00 attaching

KBCategory: kbinterop kberrmsg

KBSubcategory: ObcOracl

**PRB: Outer Join with Where Clause Returns Unexpected Records**  
**Article ID: Q124152**

-----  
The information in this article applies to:

- Microsoft Access version 2.0  
-----

SYMPTOMS

=====

When you run a query based on two tables (in this example, called table A and table B) with an outer join to include all the records from table A and apply a Where clause to table B, only those records from table B that have a matching value in table A are returned. This seems to be the reverse of what you expected.

CAUSE

=====

When you run the query the outer join is performed first, creating all the records from table A. Then, the Where clause from table B is applied to all the records, eliminating records from the query. The result set does not contain all the records from table A, but contains only those where the condition is met for table B.

RESOLUTION

=====

If you want to return all the records in table A, you can use two queries. The first query should apply the Where clause to table B, and the second query should combine table A with the first query to perform the outer join.

STATUS

=====

This type of query processing is by design and complies with the ANSI SQL 92 specification.

MORE INFORMATION

=====

Steps to Reproduce Behavior  
-----

1. Open the sample database NWIND.MDB
2. Create a new query based on the Categories and the Products tables. The two tables should be joined on the Category ID field.
3. Double-click the join line. In the Join Properties box, select the "Include ALL records from 'Categories'..." option button, and then choose the OK button.
4. Drag the Category Name field from the Categories field list to the

first column of the query grid.

5. Drag the Product Name field from the Products field list to the second column of the query grid.
6. In the Criteria row of the Product Name column, type "Like A\*" (without quotation marks).
7. Run the query. Note that not all the records from the Categories table are returned. Instead, only those records with a Product Name starting with the letter "A" are returned.

#### Steps to Return the Expected Recordset

-----

1. Open the sample database NWIND.MDB.
2. Create a new query based on the Products table.
3. Drag the Category ID and Product Name fields from the Products field list to the query grid.
4. In the Criteria row for the Product Name field, type "Like A\*" (without quotation marks).
5. Save the query as Query1 and then close it.
6. Create a new query based on the Categories table and the Query1 query.
7. Drag the Category Name field from the Categories field list to the query grid.
8. Drag the Product Name field from the Query1 field list to the query grid.
9. Double-click the join line. In the Join Properties box, select the "Include ALL records from 'Categories'..." option button, and then choose OK.
10. Run the query. Note that all the records from the Categories table are returned, with the Product Name field filled in for records with a matching record in the Products table.

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: QryJoin



**PRB: Parameter Limit for Access Basic Functions in Validation**  
**Article ID: Q105127**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You receive the error message "Expression Too Complex" when you are creating a ValidationRule property for a table or control.

CAUSE

=====

You can pass a maximum of 28 parameters to an Access Basic function when you call the function from the ValidationRule property of a table or control. Attempts to pass more than 28 parameters will result in the error message stated above.

RESOLUTION

=====

You can pass more than 28 parameters by concatenating parameters. Each parameter will be separated by a character (for example, a semicolon). After passing the individual parameters as one long parameter, you can parse out the individual parameters by searching for the separator character (the semicolon), using the InStr() and Mid() functions.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access. For more information on Access Basic, please refer to the "Introduction to Programming" manual in Microsoft Access version 1.x, or the "Building Applications" manual, Chapter 3, "Introducing Access Basic" in version 2.0.

NOTE: In the following sample code, an underscore ( \_ ) is used as a line-continuation character. Remove the underscore when re-creating this code in Access Basic.

Steps to Reproduce Behavior

-----

1. Open a new module and enter the following:

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

Option Explicit

```
'=====
' The following function CanPass28 will be accepted when entered
' into a validation rule
'=====
```

```
Function CanPass28 (a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, _
 p, q, r, s, t, u, v, w, x, y, z, a27, b28)
 Dim Parm28, Parm29 'These 3 lines
 Parm28 = Mid(b28, 1, InStr(b28, ";") - 1) 'explained in
 Parm29 = Mid(b28, InStr(b28, ";") + 1) 'step 5.
End Function
```

```
'=====
'The following function CannotPass29 will give an error message
' when used in a validation rule
'=====
```

```
Function CannotPass29 (a, b, c, d, e, f, g, h, i, j, k, l, m, n, _
 o, p, q, r, s, t, u, v, w, x, y, z, a27, b28, c29)
End Function
```

2. Open a new form. Add a text box to the form.
3. Enter the following expression for the ValidationRule property of the text box, and then press ENTER:

```
-1 = CannotPass29 (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16, _
 17,18,19,20,21,22,23,24,25,26,27,28,29)
```

The error message "Expression Too Complex" will occur.

4. Change the expression for the ValidationRule property to the following, and then press ENTER:

```
-1 = CanPass28 (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16, _
 17,18,19,20,21,22,23,24,25,26,27,28)
```

No error message will occur.

5. Enter the following expression, which concatenates the last two parameters to one parameter, for the ValidationRule property. Press ENTER:

```
-1 = CanPass28(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16, _
 17,18,19,20,21,22,23,24,25,26,27,28&";"&29)
```

The last two parameters are separated inside the function by searching for the semicolon with the following lines from the function created in step 1:

```
Parm28 = Mid(b28, 1, InStr(b28, ";") - 1)
Parm29 = Mid(b28, InStr(b28, ";") + 1)
```

Additional reference words: 1.00 1.10 2.00 ADK  
KBCategory: kbusage  
KBSubcategory: GnlValid

**PRB: Paste Unavailable After Record Is Copied to Clipboard**  
**Article ID: Q95323**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

If two users are simultaneously modifying the same record in the same table, the user who attempts to save the record last receives the following error message:

Data has changed; Operation Stopped

When you choose the OK button to close this message, you receive the following error message:

This record has been changed by another user since you started editing it. If you save the record you will overwrite the changes the other user made. Copying the changes to the Clipboard will let you look at the values the other user entered, and then paste your changes back in if you decide to make changes.

You are given the option to copy the changes the other user made to the Clipboard and then view the contents in the Clipboard; however, the Paste command is unavailable in Microsoft Access.

CAUSE

=====

Microsoft Access does not support pasting records when the focus is in a control.

RESOLUTION

=====

You can choose Paste Append to append the record to the table; or, if you select the record you want to paste to, you can then use the Paste command (or CTRL+V).

STATUS

=====

This behavior is by design.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage kberrmsg

KBSubcategory: GnlMu

**PRB: Permissions Command Unavailable When Logged in as Guest**  
**Article ID: Q115572**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you log into Microsoft Access as Guest (when Guest is assigned the default permissions) and then create a database, you cannot view or change permissions on any of the objects in the database.

CAUSE

=====

The Permissions command is always unavailable when you are logged in as Guest, even if you give the Guest user permissions to view or change object permissions.

This behavior occurs because the Guests group does not inherit read data permission on the MSysUserList query in the SYSTEM.MDA file. To prevent permission-related errors, the Permissions command is always disabled for users logged in as Guest.

STATUS

=====

This behavior is by design. Changes to this behavior are being reviewed and will be considered for inclusion in a future release of Microsoft Access.

MORE INFORMATION

=====

Steps to Reproduce Problem

-----

1. Start Microsoft Access and open any database.
2. Enable security (if it is not already enabled) by assigning a password to the Admin user. To do this, choose Change Password from the Security menu, and then enter a password for the Admin user (the Admin user is the default user).
3. Quit Microsoft Access and then restart it. Log in as Guest (no password required).
4. Create a new database.
5. Create a new table, and then create a query based on the table.
6. From the Security menu, try to choose Permissions. Note that the Permissions command is unavailable.

Additional reference words: 2.00 security greyed grayed  
KBCategory: kbusage  
KBSubcategory: ScrtPerm

## **PRB: Printing on Avery 5267 labels**

**Article ID: Q103171**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

When you print Avery 5267 labels, the text may appear to "creep" down the page.

### CAUSE

=====

Avery 5267 labels are not cut in precise .5-inch increments on the page. Because the labels are slightly larger than .5 inch, if you print in .5-inch increments, the text appears to creep down the page.

### RESOLUTION

=====

When you are placing the controls in a report, allow room for the label discrepancy. The following steps print three lines of text on Avery 5267 labels. This example uses a function that concatenates the fields for optimal use of space on each label.

1. From Print Setup, set the top and bottom margins to .5 inch and set the left and right margins to .25 inch. Press the More button and set Items Across to 4, Row Spacing to 0, and Column Spacing to 0.3 inches. Under Item Size, select the Same As Detail option.
2. Make the width of the report 1.75 inches.
3. Set the Height property for the Detail section to .5 inch and set the CanGrow and CanShrink properties to No.
4. Locate one text box control and set the properties manually as follows:

Control: Text Box

Top: .01

Height: .48

CanGrow: Yes

CanShrink: Yes

Font: Arial

FontSize: 8

ControlSource:

=AddressBlock([First],[Last],[Address],[City],[State],[Zip])

5. Create a new Access Basic module, and type the following code into it:

NOTE: In the following sample code, an underscore (\_) is used as a

line-continuation character. Remove the underscore when re-creating this code in Access Basic.

Option Explicit

```
Function AddressBlock$ (First, Last, Address, City, State, Zip)
 Dim A1$, A2$, A3$, CR$

 CR$ = Chr(13) & Chr(10) 'Carriage return and line feed
 A1$ = IIf(ISB(First),"",First & " " & IIf(ISB(Last),"",Last _
 & CR$))
 A2$ = IIf(ISB(Address),"",Address & CR$)
 A3$ = City & ", " & State & " " & Zip
 AddressBlock = A1$ & A2$ & A3$ 'Concatenate the strings.
End Function

Function ISB (V) As Integer
 If IsNull(V) or V = "" Then ISB = True Else ISB = False
End Function
```

The spacing above creates a .01 reserve above and below the text control. Variations of this layout may also print correctly.

NOTE: The Mailing Label ReportWizard creates two controls, each with a height of .17, leaving a reserve below and above the text for label adjustments.

To see another example, query on the following words in the Microsoft Knowledge Base:

white and space and canshrink and reports

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbusage  
KBSubcategory: RptLabel

**PRB: Prompt to Save Libraries After Replace Operation**  
**Article ID: Q114680**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

You may be prompted to save libraries if you perform a global search and replace operation using the Loaded Modules search option.

CAUSE

=====

Microsoft Access replaces text in library modules if the DebugLibraries option is set to True in the [Options] section of the MSACC20.INI file.

This new option was added in Microsoft Access version 2.0 to make debugging libraries easier. It allows you to edit loaded library modules along with user-created modules. The default for this option is False.

RESOLUTION

=====

If you accidentally replace text that you did not want to replace in a library, you can either reinstall Microsoft Access, or decompress the library from your original Microsoft Access disks.

If you need to replace text in multiple modules, first remove the DebugLibraries option from the MSACC20.INI file and then restart Microsoft Access. After the replacement is done, add the option back to the MSACC20.INI file.

MORE INFORMATION

=====

The decompression program, DECOMP.EXE is on disk 1 of your original Microsoft Access disks. Copy the DECOMP.EXE file to your hard disk so that it is available no matter what disk is in the floppy disk drive.

The syntax for the DECOMP.EXE command is as follows:

DECOMP.EXE <source filename> <destination filename>

For example, use the following command to decompress the WZLIB.MD\_ file to the Microsoft Access directory:

DECOMP.EXE A:\WZLIB.MD\_ C:\ACCESS\WZLIB.MDA

The following libraries ship with Microsoft Access 2.0:

| Library | File on disk |
|---------|--------------|
| -----   |              |



|              |              |
|--------------|--------------|
| WZBLDR.MDA   | WZBLDR.MD_   |
| WZFRMRPT.MDA | WZFRMRPT.MD_ |
| WZLIB.MDA    | WZLIB.MD_    |
| WZQUERY.MDA  | WZQUERY.MD_  |
| WZTABLE.MDA  | WZTABLE.MD_  |

REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 15

Additional reference words: 2.00 debuglibraries find

KBCategory: kbprg

KBSubcategory: PgmOthr

**PRB: Query by Form Returns No Rows When All Rows Expected**  
**Article ID: Q94027**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

While specifying criteria for a query in a previously designed form, you leave one of the fields empty because you want all the records, not just the ones that match a certain value. But instead of getting all the records, as you expected, you get none at all.

CAUSE

=====

Queries that contain empty criteria fields may return unexpected results because an empty criteria field results in the following condition in the WHERE clause:

Like Null

This condition is always false because any operation that includes a null value returns a null result.

Queries can reference form fields through implicit parameters--making a query by form. For example, an application developer might put the following criteria on the CustomerName field:

Like Forms!CriteriaDialog!CustomerName

Here CriteriaDialog is a form that holds the criteria fields. Someone using the application might omit a customer name entry in the CriteriaDialog form in the hopes of seeing all the rows. However, the query actually returns no rows because of the null reference in the resulting WHERE clause.

RESOLUTION

=====

Application developers can work around this potential problem by adding the following function to a module:

```
Function CNulls (v As Variant, subs As Variant) As Variant
 If (IsNull(v)) Then
 CNulls = subs
 Else
 CNulls = v
 End If
End Function
```

The CNulls() function converts null values to a given value. Essentially, if the first argument to the function is null, the second

argument is returned. Otherwise, the first argument is returned unchanged.

After you add the function to a module, change the criteria to read like this:

```
Like CNulls(Forms!CriteriaDialog!CustomerName, "*")
```

Now, if the user does not supply a customer name, the CNulls() function will return the asterisk, and the condition will return all rows that contain data.

Additional reference words: 1.00 1.10 2.00 handling nulls as param values

KBCategory: kbusage

KBSubcategory: QryOthr

## **PRB: Record Sort Order Is Different Than Expected**

**Article ID: Q105978**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

When you view data in a datasheet, combo box, or list box, the record order is different than you expect. For example, text fields may appear out of alphabetical order or numeric fields may appear out of sequence.

### CAUSE

=====

Microsoft Access has two default sort orders, insertion order and primary key order. These sort orders are used at different times, causing records to appear to be sorted differently than you expect.

### RESOLUTION

=====

If you want to see the insertion order for a table with a primary key field, remove the primary key from the table. For more information on removing a primary key, search for "primary key" then "Deleting the Primary Key" using the Microsoft Access Help menu.

If you want to store records in a different order than insertion order, build a new table with a custom sort. To do this, use a make-table query with a specified sort order. For more information on make-table queries, see the Microsoft Access "User's Guide," Chapter 7.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

Insertion order is the order in which records are entered and stored in a table. This order may not be in alphabetical or numeric sequence, since records are often entered in a random order.

Primary key order is based on the field or fields in a primary key. This order has no effect on the way records are stored in a table; the records are still stored in insertion order. Primary key order changes only the way the records are displayed.

If the underlying table for a table, form, subform, or query datasheet has no primary key, the data will be displayed in insertion order. If the table has a primary key, the data will be displayed in primary key order.

There are two exceptions to this rule. For query datasheets, insertion order will be used for a query that has specific criteria but no specific sort order, even if the underlying table has a primary key. For combo boxes and list boxes, insertion order is used even if the underlying table has a primary key.

#### Steps to Reproduce Behavior

-----

The following four examples demonstrate the two default record orders and the two exceptions described above. Note that all the examples use the sample database NWIND.MDB.

#### Viewing Records in Insertion Order:

1. Make a copy of the Categories table and save the copy as Categories2.
2. Open the Categories2 table in Datasheet view.
3. Add the following record:

```
Category ID: APPZ
Category Name: Appetizers
Description: <leave blank>
Picture: <leave blank>
```

Note the record order in the Category ID field. BEVR is the first record and APPZ is the last. This is the order in which the records were added to the table.

#### Viewing Records in Primary Key Order:

1. Create a new query based on the Categories2 table as follows:

```
Field name: Category ID
Sort: <leave blank>
Show: True
Criteria: <leave blank>
```

2. From the Query menu, choose Run.

Note that the query datasheet is sorted alphabetically by Category ID, which is the primary key field.

#### Viewing Records in Insertion Order Despite a Primary Key:

1. Create a new query based on the Categories2 table as follows:

```
Field name: Category ID
Sort: <leave blank>
Show: True
Criteria: a* or s*
```

2. From the Query menu, choose Run.

The following records will be displayed:

Category ID

-----

SEAF

APPZ

Note that SEAF appears before APPZ even though Category ID is a primary key field. Microsoft Access is using insertion order because the query includes specific criteria but no specific sort order.

Viewing Combo Box Items in Insertion Order:

1. Create the following new form:

Form: Test1

ControlSource: Unbound

2. Add the following combo box to the Test1 form:

Combo box: Field1

ControlSource: Unbound

RowSourceType: Table/Query

RowSource: Categories2

ColumnCount: 2

ColumnWidth: 1 in; 1 in

3. View the form in Form view.

4. View the items listed in the combo box.

Note that the items are displayed in the combo box in insertion order, even though the underlying table has a primary key field.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: TblPriky

**PRB: RecordCount Property Incorrect Using OpenTable Method**  
**Article ID: Q117167**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The RecordCount property of a table may be incorrect.

CAUSE

=====

If you open a table, begin a transaction, add a new record, and then roll back the transaction, the record count returned on the table may be incorrect because the previous value of the RecordCount property is not restored. This behavior can occur when you use the OpenTable method, or when you use the DB\_OPEN\_TABLE Type property setting with the OpenRecordset method on a TableDef object using data access objects (DAO).

RESOLUTION

=====

To retrieve the correct record count, create a dynaset based on the table, use the MoveLast method to move to the last record in the dynaset, and print the dynaset's RecordCount property. The following example demonstrates how to do this:

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open a new module and enter the following sample code.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
' *****
' Declarations section of the module
' *****

Option Compare Database
Option Explicit

' *****
' The RightCount() function creates a dynaset based on the
' Employees table, prints the table's record count, starts a new
' transaction, adds a record, rolls back the transaction, and then
' prints the dynaset's record count.
' *****

Function RightCount ()
 Dim db As Database
 Dim MyDyna As Dynaset
```

```

Set db = CurrentDB()
Set MyDyna = db.CreateDynaset("Employees")
MyDyna.MoveLast
Debug.Print "BEFORE Transaction: Employee Record Count = " & _
 MyDyna.recordcount

BeginTrans
 MyDyna.AddNew
 MyDyna![Last Name] = "Doe"
 MyDyna![First Name] = "John"
 MyDyna.Update
Rollback
MyDyna.Close
Set MyDyna = db.CreateDynaset("Employees")
MyDyna.MoveLast
Debug.Print "AFTER Transaction: Employee Record Count = " & _
 MyDyna.recordcount

MyDyna.Close
End Function

```

3. From the View menu, choose Immediate Window.

4. In the Immediate window, type the following line and then press ENTER:

```
?RightCount()
```

Compare the record count returned after the transaction is rolled back with the actual number of records in the table and note that they are the same.

STATUS  
=====

This behavior is by design.

MORE INFORMATION  
=====

When you use the OpenTable method, or the DB\_OPEN\_TABLE Type property setting with the OpenRecordset method, the base table is opened directly. The RecordCount property retrieved from the table is only an approximate value, and is not always accurate. When a transaction is rolled back, the previous value of the RecordCount property is not restored.

Steps to Reproduce Behavior  
-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open a new module and enter the following sample code.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```

' *****
' Declarations section of the module
' *****

```



```
Option Compare Database
Option Explicit
```

```
' *****
' The WrongCount() function opens the Employees table, prints the
' record count, starts a new transaction, adds a record, rolls
' back the transaction, and then prints the table's record count.
' *****
```

```
Function WrongCount ()
 Dim db As Database
 Dim MyTable As Table
 Set db = CurrentDB()
 Set MyTable = db.OpenTable("Employees")
 MyTable.MoveLast
 Debug.Print "BEFORE: Employee Record Count = " & _
 MyTable.RecordCount

 BeginTrans
 MyTable.AddNew
 MyTable![Last Name] = "Doe"
 MyTable![First Name] = "John"
 MyTable.Update
 Rollback
 MyTable.Close
 Set MyTable = db.OpenTable("Employees")
 MyTable.MoveLast
 Debug.Print "AFTER: Employee Record Count = " & _
 MyTable.RecordCount

 MyTable.Close
End Function
```

3. From the View menu, choose Immediate Window.
4. Type the following line in the Immediate window, and then press ENTER:

```
?WrongCount()
```

Compare the record count returned after the transaction is rolled back with the actual number of records in the table and note that they are different. The record count that is returned after the transaction is rolled back is not correct.

Additional reference words: 1.00 1.10 2.00  
KBCategory: kbprg  
KBSubcategory: PgmObj

**PRB: RecordCount Property Returns Incorrect Number of Records**  
**Article ID: Q105976**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The RecordCount property, when used with a dynaset or snapshot, returns a recordset containing an incorrect number of records.

CAUSE

=====

For dynasets and snapshots, Microsoft Access does not automatically return the number of records that exist in the recordset. It returns the number of records accessed.

RESOLUTION

=====

To determine the exact number of records in a dynaset or snapshot, use the MoveLast method before checking the RecordCount property.

MORE INFORMATION

=====

This article assumes that you are familiar with Access Basic and with creating Microsoft Access applications using the programming tools provided with Microsoft Access.

The following Access Basic function, MyWrongRecordCount(), returns the number 1 for the Customers table in the sample database NWIND.MDB because only one record has been accessed. The MyRightRecordCount() function uses the MoveLast method first to access all records in the dynaset and then to return the RecordCount value.

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create a new module.
3. Add the following line of code to the Declarations section:

Option Explicit

4. Create the following functions:

```
'=====
' The following function, MyWrongRecordCount(), demonstrates
' the incorrect way to use the RecordCount property to count
```

```

' records in a dynaset.
'=====
Function MyWrongRecordCount ()

 Dim MyDB As Database, MyDyna As Dynaset

 Set MyDB = CurrentDB()
 Set MyDyna = MyDB.CreateDynaset("Customers")
 MyWrongRecordCount = MyDyna.RecordCount
 MyDyna.Close

End Function

'=====
' The following function, MyRightRecordCount(), demonstrates
' the correct way to use the RecordCount property to count
' records in a dynaset.
'=====
Function MyRightRecordCount ()

 Dim MyDB As Database, MyDyna As Dynaset

 Set MyDB = CurrentDB()
 Set MyDyna = MyDB.CreateDynaset("Customers")
 MyDyna.MoveLast
 MyRightRecordCount = MyDyna.RecordCount
 MyDyna.Close

End Function

```

5. From the View menu, choose Immediate Window.
6. In the Immediate window, type the following and then press ENTER:

```
?MyWrongRecordCount()
```

Note that the function returns 1.

7. In the Immediate window, type the following and then press ENTER:

```
?MyRightRecordCount()
```

Note that the function returns the correct number of records in the Customers table.

#### REFERENCES

=====

Microsoft Access "Language Reference," versions 1.0 and 1.1, page 390

For more information on the RecordCount property, search for "RecordCount" then "RecordCount Property" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 record count move last

KBCategory: kbprg

KBSubcategory: PgmObj



**PRB: Removing ADT Does Not Unregister Custom Controls**  
**Article ID: Q123399**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

When you remove the Microsoft Access Developer's Toolkit (ADT) from your computer by choosing the Remove All button in the ADT Setup program, the custom controls are not recognized as shared components and are not removed from the registration database.

RESOLUTION

=====

After you remove the ADT, manually remove the custom controls from the registration database by following these steps:

1. In Program Manager, choose Run from the File menu.
2. In the Command Line box, type "regedit.exe" (without quotation marks) and then choose OK.
3. In the Registered File Types box, select the custom control you want to remove.
4. From the Edit menu, choose Delete File Type.
5. Repeat steps 3-4 for each custom control you want to remove.
6. When you are finished, quit and then restart Microsoft Windows.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

CAUTION: Following these steps will remove the ADT from your computer.

1. Start the ADT Setup program by double-clicking the MS ADT Setup icon in the ADT program group.
2. Choose the Remove All button. When you are prompted "Are you sure you want to remove this application?" choose Yes.
3. When Setup is finished, start Microsoft Access and then open the sample database NWIND.MDB.
4. Create a blank, new form.
5. From the Edit menu, choose Insert Object.

6. Select the Insert Control option button. Note that the custom controls (the calendar, data outline, and scroll bar controls) appear as if they are available because they have not been removed from the registration database.

#### REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0,  
"Introduction to the Developer's Toolkit," page xi

Additional reference words: 2.00 ADT un-install uninstall

KBCategory: kbenv

KBSubcategory: EvnTemp

## **PRB: Replace Command Does Not Find NULL Values**

**Article ID: Q120730**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you choose the Replace command from the Edit menu, Microsoft Access does not locate any null values even though they do exist in the table.

### RESOLUTION

=====

To replace null values in a table, create and run an update query.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

If you choose the Find command from the Edit menu to search for null values, the values are located correctly. It is only the Replace command that does not locate null values. NOTE: In Microsoft Access version 1.x, neither the Find nor the Replace command will locate null values.

### Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB and then open the Orders table in Datasheet view.
2. Position the insertion point in the first row of the Ship Region column.
3. From the Edit menu, choose Replace.
4. In the Find What field, type "NULL" (without quotation marks).
5. In the Replace With field, type "TEST" (without quotation marks).
6. Choose the Find Next button. Note that Microsoft Access does not locate any null values in the field, even though such values do exist.

### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 5, "Finding and Sorting Data," pages 84-90

For more information about the Replace command, search for "Replace" then "Replace Command" using the Microsoft Access Help menu.

For more information about update queries, search for "update query" then "Updating Records as a Group" using the Microsoft Access Help menu.

Additional reference words: 2.00

KBCategory: kbui

KBSubcategory: UiOthr



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

Article ID: Q92687

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### 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 occurs when you attempt to open a table in Design view and the table is the record source for a form or report, and one of the following conditions is also true:

1. A report is open in Print Preview or a form is open in Form view that has the same record source, or is based on a query with the same record source.
2. A form is opened in Form 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 view. (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 Form 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.

### REFERENCES

=====

For more information, search for "error messages: reference" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00 locking

KBCategory: kbusage

KBSubcategory: TblOthr

**PRB: Rounding Errors Importing/Exporting dBASE IV Values**  
**Article ID: Q96905**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Some floating-point values (single and double precision) have rounding errors when exported to or imported from a dBASE IV database. For example, if the floating-point number .0807 is exported from a Microsoft Access database to a dBASE IV database and then imported back to Microsoft Access, after the import the value reads .08069.

CAUSE

=====

This problem is caused by the floating-point number being converted from IEEE floating-point numeric format (used by Microsoft Access and several other Microsoft products) to ASCII character values required by dBASE IV. The Microsoft Access dBASE ISAM converts IEEE floating-point values to ASCII, and conversion of binary fractions to decimal is inherently inexact.

Typically, such rounding errors are not noticeable or relevant because they occur at digits far to the right of the decimal point and are not displayed. However, sometimes the rounding error may appear in the visible portion of the number.

STATUS

=====

This behavior is by design.

Additional reference words: 1.00 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmXbase

## **PRB: Rounding Errors Using Floating-Point Numbers**

**Article ID: Q111781**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

### SYMPTOMS

=====

When you calculate using floating-point numbers, the result is not always what you expect. For example, if you use the CInt() (convert to integer) function in the calculation

```
CInt(3.555 * 100)
```

you get 356 as the result, as you expect. However, the calculation

```
CInt(4.555 * 100)
```

results in 455, instead of 456 as you expect.

### CAUSE

=====

Errors similar to the example above occur in any programming language that uses floating-point numbers. This is because decimal fractions do not always have exact binary equivalents, which can result in rounding errors.

### RESOLUTION

=====

There are several methods you can use to avoid rounding errors when you are using floating-point numbers:

- If you are using Single numbers, convert them to Doubles and use formatting to hide the extra digits. If there are any rounding errors, they will occur at the end of the numbers where they will not affect the visible data.
- Use the Currency data type. If you use currency values of four decimal places or less, rounding errors will not occur since these values are actually scaled integers rather than floating-point numbers. In Access Basic code, use the at sign (@) character to use a Currency data type. For example, the calculation

```
CInt(4.555@ * 100)
```

results in a value of 456.

- Break the calculation into two or more steps. When you list a calculation in one long line, Access Basic stores the intermediate values internally. For example, in the calculation

```
CInt(4.555 * 100)
```

the value 455.5 is temporarily stored and used in the CInt() function. If you break the calculation into the two steps

```
x = 4.555 * 100
CInt(x)
```

you avoid this internal storage and thus avoid the floating-point rounding error.

Additional reference words: 2.00 AB Roundoff

KBCategory: kbprg

KBSubcategory: PgmOthr

**PRB: SCSI Drives Can Generate Corrupt Error Loading Database**  
**Article ID: Q100159**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you are loading a Microsoft Access database file from a SCSI drive, you receive the following error message:

<filename> is corrupted or is not a database file. Attempt to repair?

Attempts to repair the database do not solve the problem.

CAUSE

=====

There is probably nothing wrong with the database file. The actual problem is that Microsoft Windows version 3.1 is having trouble accessing the SCSI drive.

RESOLUTION

=====

Add the following line to the CONFIG.SYS file on your drive. If you already have a line in your CONFIG.SYS file that references SMARTDRV.EXE, modify it to match the following line. After adding or modifying this line, restart your computer.

```
DEVICE=C:\WINDOWS\SMARTDRV.EXE /double_buffer
```

NOTE: If the computer is running MS-DOS 6.0 or a later version, use C:\DOS instead of C:\WINDOWS to load the newest version of SMARTDRV.EXE.

MORE INFORMATION

=====

Microsoft Windows 3.1 requires the double\_buffer switch in the SMARTDRV.EXE line to start in 386 enhanced mode on most SCSI drives. On a computer where Windows is not located on the SCSI drive, it is possible that Windows has started in 386 enhanced mode but that the double buffer is not in use.

REFERENCES

=====

Microsoft Windows "User's Guide," version 3.1, Chapter 14

Additional reference words: 1.00 1.10 2.00 WinIss

KBCategory: kbenv

KBSubcategory: EvnOs



**PRB: Seek Method Is Faster Than Find Method**

Article ID: Q98806

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

When you are using Access Basic to find a record in an indexed field, the Seek method may be faster than the Find method, especially in a large table.

CAUSE

=====

This behavior is by design.

RESOLUTION

=====

Use the Seek method on indexed fields to optimize your search speed.

MORE INFORMATION

=====

When you perform a Seek, you are opening a table directly and moving to the record based on an index value. When you create a dynaset and perform a Find, you are checking the value of the field in every record until you find a match.

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Create a new module and enter the following two functions:

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

'=====
```

```
'The following function uses the Seek method.
'It quickly returns the Customer ID where the
' PrimaryKey field = 11070.
'=====
```

```
Function Faster ()
 Dim db As Database, tbl As Table

 Set db = CurrentDB()
 Set tbl = db.OpenTable("Orders")
```

```

tbl.Index = "PrimaryKey"
tbl.Seek "=", 11070
Debug.Print tbl("customer id")
tbl.Close
End Function

'=====
'The following function uses the Find method.
'It is slightly slower in returning the Customer
' ID where the PrimaryKey field = 11070.
'=====
Function Slower()
 Dim Criteria As String, MyDB As Database, Myset As Dynaset

 Set MyDB = CurrentDB()
 Set Myset = MyDB.CreateDynaset("Orders")

 Criteria = "[Order id] =" & 11070
 Myset.FindNext Criteria
 Debug.Print Myset("Customer id")
End Function

```

3. From the View menu, choose Immediate Window.

4. In the Immediate window, type

```
?Faster()
```

and press ENTER. Then type

```
?Slower()
```

and press ENTER.

The Faster() function, which uses Seek, is slightly faster than the Slower() function, which uses FindNext.

#### REFERENCES

=====

Microsoft Access "Language Reference," version 1.0, pages 187-188 and 426-427

Additional reference words: 1.00 1.10 2.00 seeking finding

KBCategory: kbprg

KBSubcategory: PgmObj



**PRB: SendKeys Macro Action Does Not Work on Toggle Keys**  
**Article ID: Q101125**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

A SendKeys macro action will not work correctly on the toggle keys CAPS LOCK or SCROLL LOCK. You may see the CAPS LOCK or SCROLL LOCK lights flash on the keyboard, but the lights will not remain on and the keys will not be correctly activated after the SendKeys action is completed. Note that SendKeys also will not work correctly with the ALT+PRINT SCREEN key combination.

CAUSE

=====

SendKeys keystrokes are sent to applications at a high level. Microsoft Windows traps toggle keys and the PRINT SCREEN key at a lower level, keeping them from your application.

RESOLUTION

=====

There is a workaround for the CAPS LOCK problem. To force input in a field into uppercase, set the control's Format property to ">". The UCase() function can also be used to force entries in a field to uppercase.

Although a SendKeys macro action cannot toggle keys, a Windows API SetKeyboardState() function call can toggle keys. The following example demonstrates how to use the Windows API SetKeyboardState() function:

```
Declare Sub SetKeyboardState Lib "User" (lpKeyState As Any)
```

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

The following SendKeys action will not toggle the CAPS LOCK key to on:

```
MacroName Action

Macrol SendKeys

Macrol Actions

SendKeys
 Keystrokes: {CAPSLOCK}
```

Wait: NO

REFERENCES

=====

Microsoft Access "Language Reference," pages 432 and 485

Additional reference words: 1.00 1.10 2.00 send key num

KBCategory: kbusage

KBSubcategory: McrOthr

**PRB: Setting KeyCode=0 for KeyDown Does Not Ignore All Keys**  
**Article ID: Q120827**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

Setting KeyCode to 0 in the event procedure for a form or form control's OnKeyDown property does not trap all keys and key combinations.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Setting KeyCode to 0 causes the following keys and key combinations to be trapped:

|           |          |    |
|-----------|----------|----|
| TAB       | F6       | F7 |
| SHIFT+TAB | SHIFT+F6 | F2 |

The following keys and key combinations are not trapped:

|                |                 |             |
|----------------|-----------------|-------------|
| F1             | F4              | SHIFT+DEL   |
| CTRL+SHIFT+TAB | CTRL+TAB        | ALT+TAB     |
| CTRL+C         | CTRL+F          | CTRL+H      |
| CTRL+N         | CTRL+O          | CTRL+P      |
| CTRL+S         | CTRL+V          | CTRL+X      |
| CTRL+Z         | CTRL+SHIFT+HOME | END         |
| CTRL+ENTER     | BACKSPACE       | CTRL+INSERT |
| SHIFT+INSERT   | ALT+BACKSPACE   | F11         |
| F12            |                 |             |

The following keys are not trapped in the Microsoft Access Developer's Toolkit (ADT):

F11 F12 SHIFT+F12

These keys also cannot be trapped in the AutoKeys macro.

Steps To Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. Select the Customers table in the Database window, then click the AutoForm button on the toolbar.

3. View the new form in Design view. Set the Contact Name text box's OnKeyDown property to the following event procedure:

```
Sub Contact_Name_KeyDown (KeyCode As Integer, Shift As Integer)
 KeyCode = 0
End Sub
```

4. View the form in Form view and press some of the keys and key combinations listed earlier in this article.

Note that if you press TAB, nothing happens, since this key does not get trapped. If you press CTRL+TAB, the next control is selected, since this key combination does not get trapped.

#### REFERENCES

=====

Microsoft Access "Building Applications," version 2.0, Chapter 8, "Events," page 200

For more information about the KeyDown event, search for "KeyDown" then "KeyDown, KeyUp Events" using the Microsoft Access Help menu.

Additional reference words: 2.00 keyup

KBCategory: kbusage

KBSubcategory: FmsEvt

**PRB: Setup Wizard Cannot Find VSHARE.386 with NT**  
**Article ID: Q121782**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

When you are using the Setup Wizard to create installation disks for your custom application on a computer running Microsoft Windows NT, you may receive the following error message:

Can't find file 'C:\NT\SYSTEM\VSHARE.386'

CAUSE

=====

VSHARE.386 is a file-locking management file that is shipped with Microsoft Access version 2.0. The Setup Wizard requires that this file be included on your application's custom Setup disks in order for your run-time application to work on a computer running Microsoft Windows version 3.1 or Microsoft Windows for Workgroups version 3.1 or 3.11. However, since Microsoft Windows NT provides its own form of file locking, Microsoft Access Setup does not install the VSHARE.386 file to computers running this operating system.

RESOLUTION

=====

To work around this behavior, install the VSHARE.386 file from your original Microsoft Access disks to your computer's NT\SYSTEM subdirectory. Follow these steps to install the VSHARE.386 file:

1. Copy the DECOMP.EXE file from Disk 1 of your Microsoft Access disks to your Windows NT directory.
2. Insert Disk 4 in a floppy drive, type the following line, and then press ENTER:

```
DECOMP <drive>:\VSHARE.38_ <drive>:\<directory>
```

For example, to decompress the file from a disk in drive A to the NT\SYSTEM subdirectory on drive C, you would type the following line:

```
DECOMP A:\VSHARE.38_ C:\NT\SYSTEM\VSHARE.386
```

The size of the VSHARE.386 file when expanded should be 14,933.

Additional reference words: 2.00

KBCategory: kbsetup kberrmsg

KBSubcategory: StpFail



**PRB: Setup Wizard Templates Lost Installing Service Pack**  
**Article ID: Q124183**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

When you install the new Setup Wizard that is shipped with the Microsoft Access version 2.0 Service Pack, the templates you had already defined in the Setup Wizard are lost.

CAUSE

=====

The new version of the Setup Wizard overwrites the older one when you unpack it with the Decompile utility included with the Service Pack. When the older version of the Setup Wizard is overwritten, the existing templates contained in it are overwritten as well.

RESOLUTION

=====

To work around this behavior, you can import the templates in the older Setup Wizard into the new version. To do so, follow these steps.

NOTE: You must follow these steps before installing the new Setup Wizard. If you have already installed the new Setup Wizard (and overwritten the older Setup Wizard) these steps will not work.

1. Rename the older Setup Wizard file from SETUPWIZ.MDB to SETUPWIZ.BAK.
2. Install the new Setup Wizard following the instructions in the Service Pack.
3. Start Microsoft Access. While pressing the SHIFT key, open the SETUPWIZ.MDB file. (Pressing the SHIFT key while opening a database prevents the database's Autoexec macro from running.)
4. From the File menu, choose Import. In the Data Source box, select Microsoft Access, and then choose the OK button.
5. In the File Name box, select SETUPWIZ.BAK, and then choose OK.
6. In the Import Objects dialog box, select the zws\_SetupTemplates table and then choose the Import button.
7. Repeat step 6 for the zws\_TemplateFiles table.
8. After the tables are imported, choose the Close button.
9. Delete the zws\_SetupTemplates and zws\_TemplateFiles tables. These are the empty tables from the new Setup Wizard.

10. Rename the zws\_SetupTemplates1 table to zws\_SetupTemplates. Rename the zws\_TemplateFiles1 table to zws\_TemplateFiles.

11. Quit Microsoft Access and delete the SETUPWIZ.BAK file.

#### REFERENCES

=====

For more information about the Microsoft Access version 2.0 Service Pack, please see the following article in the Microsoft Knowledge Base:

ARTICLE-ID: Q122927

TITLE : WX1124: Microsoft Access Version 2.0 Service Pack

Additional reference words: 2.00 jet25 jet 2.5

KBCategory: kbusage

KBSubcategory: DcmOthr



**PRB: Shortcut Menus Cannot Be Disabled In Microsoft Access**  
**Article ID: Q113930**

---

The information in this article applies to:

- Microsoft Access version 2.0
  - Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

You cannot disable, or turn off, the shortcut menus in the Microsoft Access version 2.0 Database window. Shortcut menus are displayed by clicking the secondary mouse button.

RESOLUTION

=====

While you cannot disable shortcut menus in the Database window, you can disable them in forms by setting the form's ShortCutMenu property to No.

The shortcut menus are also disabled when a Microsoft Access application is run using the run-time version of Microsoft Access provided in the Microsoft Access Developer's Toolkit (ADT).

To prevent users from seeing the built-in toolbars, set the Built-In Toolbars Available option to No in the Options dialog box. Note that this will not prevent users from seeing any available custom toolbars.

To prevent users from changing custom toolbars, set the Can Customize Toolbars option to No in the Options dialog box.

STATUS

=====

This behavior is by design. Changes in this functionality are being reviewed for possible revision in a future version of Microsoft Access.

REFERENCES

=====

For more information about toolbar options, search for "Options command - general" then "Options Command (View Menu)" using the Microsoft Access Help menu.

Additional reference words: 2.00 Menus Short Cut

KBCategory: kbui

KBSubcategory: UifOthr

**PRB: Some Labels Skipped in Mailing Label Report**  
**Article ID: Q119476**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you print a mailing label report, Microsoft Access skips some of the labels, even though all of the records are printed.

CAUSE

=====

This behavior can occur if you set the CanGrow property for the report's detail section and for all the controls in the detail section to Yes.

RESOLUTION

=====

Set the CanGrow property for the detail section, or for one or more of the controls in the detail section, to No.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

If the CanGrow property is set to Yes for both the detail section and for all the controls in the detail section, one or more of the labels may grow on some pages. If a label grows, the detail section also grows, forcing some labels to be printed on the next page.

Steps to Reproduce Behavior

-----

1. Open the sample database NWIND.MDB.
2. In the Database window, choose the Reports button, then choose New.
3. In the Select A Table/Query box, select the Customer table, and then choose the Report Wizards button.
4. In the Which Wizard Do You Want? screen, select Mailing Label and then choose OK.
5. Design the mailing label as follows, and then choose the Next button:

Contact Name  
Contact Title

Address  
City, Region Postal Code

6. In the "Which fields do you want to sort by?" screen, choose Next.
7. In the Label Type box, select the Sheet Feed option button.
8. Select Avery label number 5164, then choose the Finish button.
9. View the report in Design view, and make sure that the CanGrow property for either the detail section or one or more of the controls in the detail section is set to No.
10. Preview the report. Note that no labels on the report are skipped.
11. View the report in Design view. Set the CanGrow property for the detail section, and for all the controls in the detail section, to Yes.
12. Preview the report. Note that the labels on some pages are skipped.

#### REFERENCES

=====

For information about the CanGrow property, search for "CanGrow" then "CanGrow, CanShrink Properties" using the Microsoft Access Help menu.

Microsoft Access "User's Guide," version 2.0, Chapter 20, "Report Basics," pages 485-504, and Chapter 21, "Designing Reports," pages 505-545

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: RptLabel

**PRB: Stored procedure 'PROCEDURE' not found -- Error Message**  
**Article ID: Q95061**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You receive the error message:

Stored procedure 'PROCEDURE' not found

The 'PROCEDURE' noted in the error message is the name of a stored procedure that cannot be found but is necessary to gain access to the SQL Server.

CAUSE

=====

The SQL Server you are attempting to access is not set up correctly. It is missing the required stored procedures.

RESOLUTION

=====

To correct this situation, install the necessary stored procedures on the SQL Server by using the INSTCAT.SQL SQL script file supplied with the Microsoft Access disk set.

MORE INFORMATION

=====

Run INSTCAT.SQL to configure Microsoft SQL Server for use with ODBC, the open database connectivity protocol used by Microsoft Access to attach SQL Server tables. INSTCAT.SQL is the SQL script file that ships with with ODBC. You need to run it to set up the stored procedures that provide catalog information used by ODBC.

To install the catalog stored procedures using INSTCAT.SQL, run INSTCAT.SQL using the SQL Server facility ISQL (Interactive SQL) from the MS-DOS command line.

Following is the syntax for this procedure. Note that you need to enter the two lines as one continuous line, and do not include the angle braces <> in the command:

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

Following is a description for each switch:

```
/U Gives the system administrator's login name
/n Eliminates line numbering and prompting for user input.
```

/P Gives the system administrator's Password -- case sensitive  
/S Gives the name of the server to set up.  
/i Gives the drive and fully qualified path for INSTCAT.SQL  
/o Provides the output file destination for results of the process including errors.

Here is a sample ISQL command line:

```
ISQL /U sa /n /P SA_Password /S SQL_SERVER /i d:\SQL\INSTCAT.SQL
/o d:\SQL\output.txt
```

After running INSTCAT.SQL, run the RECONFIGURE command against the MASTER database by using the SQL Administration Facility (SAF) from an MS-DOS client or the server, or run the SQL Administrator program from a Microsoft Windows client.

#### Other Possible Causes for this Error Message

-----  
It is possible that other situations can indirectly cause this error message. Use the following steps to troubleshoot the problem:

1. Use File Manager to search for multiple, corrupted, or outdated copies of these files:

ODBC.DLL - dated on or after 10-16-92 containing 44736 bytes  
DBNMP3.DLL - dated on or after 9-15-92 containing 8241 bytes  
SQLSRVR.DLL - dated on or after 10-16-92 containing 135792 bytes  
NETAPI.DLL - depends on the network being used  
COMMDLG.DLL - dated on or after 10-25-92 containing 89248 bytes

NOTE: The sizes and dates referenced are correct if the files are installed by Microsoft Access version 1.0.

2. Make sure the files listed above are in appropriate locations. All the files except NETAPI.DLL are usually installed in the Windows System directory. The NETAPI.DLL file is a network specific driver that is usually located in the network's program directory, which should be in the path.
3. Use another application, such as Q+E, SAF, or PowerBuilder, on the same machine to attempt to attach the same data source. If you cannot access the data source from another application, there is probably a problem with the network. That is there may be a problem with the NETAPI.DLL file or a problem with the SQL database server, not with Microsoft Access or ODBC.
4. Attempt to attach another data source such as another SQL server. If you are successful, the original server may be down or you may need to reconfigure the entry for that server by using the ODBC Administration Utility.
5. Attempt to attach to the same data source from another machine. If you cannot gain access to the SQL database server from another machine, there could be a network wide problem or the server may be down or not communicating.

6. Try increasing your ODBC timeout settings in the ODBC section of the MSACCESS.INI file.
7. Check with the system administrator to see if the SQL database server is up, available for transactions, and operating correctly.
8. Reinstall ODBC after restarting the server and renaming the ODBC.DLL, DBNMP3.DLL, and SQLSRVR.DLL files.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Appendix D, "Setting up Microsoft Access on a Network," pages 660-662

Microsoft SQL Server "Administrator's Guide," version 4.2, pages 205-212

Additional reference words: 1.00 1.10 2.00 ODBC kbtshoot errsg

KBCategory: kbusage

KBSubcategory: ObcOthr

**PRB: Strange Characters Appear in Imported dBASE IV Database**  
**Article ID: Q99399**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0  
-----

SYMPTOMS

=====

Strange characters, specifically characters Chr(236) and Chr(10), appear approximately every 65th character in a memo field in an imported dBASE IV database.

CAUSE

=====

These characters are used by dBASE IV so that its memo editor word wraps automatically. The dBASE IV memo editor has a fixed line length of 65 characters, and the mechanism is stored in the data so the memo editor does not have to force the word wrap.

RESOLUTION

=====

Use the following procedure and sample code to remove the Chr(10) and Chr(236) characters from the data. This procedure will not damage the data, and you can export the data back to dBASE IV later if you want to:

1. Open a new module and enter the following line in the Declarations section:

```
Option Explicit
```

2. Enter the following code in the module:

```
Sub DB4MemoFix (TableName As String, FieldName As String)
 Dim D As Database, T As Table
 Dim oldmemo As String, newmemo As String
 Dim i As Integer
 Set D = CurrentDB()
 Set T = D.OpenTable(TableName)
 T.MoveFirst
 Do Until T.EOF
 If T(FieldName) <> "" Then
 oldmemo = Trim(T(FieldName))
 newmemo = ""
 For i = 1 To Len(oldmemo)
 If Mid(oldmemo, i, 2) = (Chr(236) & Chr(10)) Then
 i = i + 1
 Else
 newmemo = newmemo & Mid(oldmemo, i, 1)
 End If
 Next
 T.Edit
```

```
 T(FieldName) = newmemo
 T.Update
 End If
 T.MoveNext
Loop
D.Close
Debug.Print
Debug.Print "Done"
End Sub
```

3. Save the module.
4. From the View menu, choose Immediate Window.
5. In the Immediate window, type the following:

```
DB4MemoFix "<tablename>", "<fieldname>"
```

Substitute the name of the imported dBASE IV table for <tablename>, and the name of the memo field for <fieldname>.

For example, for a table named CUST with a memo field called NOTES, you would type:

```
DB4MemoFix "CUST", "NOTES"
```

When the process is complete, the word "Done" will appear in the Immediate window.

#### MORE INFORMATION

=====

dBASE IV is manufactured by Borland International, Inc., a vendor independent of Microsoft; we make no warranty, implied or otherwise, regarding this product's performance or reliability.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr



## **PRB: Subform Not Updated**

**Article ID: Q92676**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

As you scroll through records in the main form, your subform is not updated. The records on the main form and the subform are not synchronized.

### CAUSE

=====

Your subform is not linked to the main form.

### RESOLUTION

=====

You should use the LinkMasterFields and LinkChildFields form properties to link your main form and subform automatically. You may manually update the subform by pressing the F9 (recalculate) key.

### STATUS

=====

This behavior is by design.

### MORE INFORMATION

=====

Recalculation occurs automatically for controls that reference other fields on the same form, or fields in subforms. Recalculation does not occur automatically for subform controls that only reference fields on the main (master) form, or in other subforms.

This is because subforms notify the master form of any changes, but the master form does not notify the subforms of changes. Nor do subforms on the same main form notify one another of any changes.

### REFERENCES

=====

For more information, search for "Link: main form and subform" then "function keys" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: FmsSubf

**PRB: Subreport or Subform Data Missing from "Output To" Result**  
**Article ID: Q112752**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

Information in subforms and subreports is missing from files created with the Output To command in Microsoft Access version 2.0.

CAUSE

=====

The Output To command does not support subforms and subreports.

RESOLUTION

=====

To output forms or reports containing subforms or subreports, open the subform or subreport as a separate form or report and use the Output To command to print that form or report independently.

If you must have the subform or subreport print with the main form or report, you can direct the output to a text file using the Windows Generic print driver attached to the FILE port.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open the Sales Summaries report in Design view.
3. Verify that there are two subreports in the detail section.
4. From the File menu, choose Output To. Select MS-DOS Text (\*.txt) as the file type and then choose OK.
5. Choose OK.
6. Choose OK when you get the message "The Output To command will not process any subreports or subforms contained in your report. Proceed anyway?"

7. Open the text file created by the Output To command. It contains only:

Sales Summaries  
15-Mar-94

Employee Sales:

Category Sales:

#### REFERENCES

=====

For a complete list of the functionality of each file type created by the Output To command, search for "Output To" using the Microsoft Access Help menu.

For more information about how to output Microsoft Access forms and reports to text files, query on the following words here in the Microsoft Knowledge Base:

print and file and generic

Additional reference words: 2.00 OutputTo

KBCategory: kbinterop

KBSubcategory: IntpOthr

**PRB: Syntax Error on Page 196 of "Advanced Topics" Manual**  
**Article ID: Q120886**

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0  
-----

SYMPTOMS

=====

Page 196 of the Microsoft Access Developer's Toolkit version 2.0 "Advanced Topics" manual contains the following sample code.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Sub PostRecords_Click ()
 Dim MyWS As Workspace, MyDB As Database
 On Error GoTo TransferFailed

 Set MyWS = DBEngine.Workspaces(0)
 Set MyDB = MyWS.Databases(0)

 MyWS.BeginTrans
 MyDB.Execute ("INSERT INTO RmtOrdersEmpty SELECT * from _
 LclOrders", DB_FAILONERROR)
 MyDB.Execute ("INSERT INTO RmtOrderDetailsEmpty SELECT * from _
 LclOrderDetails", DB_FAILONERROR)
 MyDB.Execute ("Delete from LclOrders")
 MyDB.Execute ("Delete from LclOrderDetails")
 MyWS.CommitTrans
 Me.Requery
 Exit Sub

TransferFailed:
 MsgBox Error$
 MyWS.Rollback
 Exit Sub
End Sub
```

When you compile this code, you receive the error message:

Expected: )

CAUSE

=====

The sample code incorrectly includes parentheses in the syntax of the Execute method.

RESOLUTION

=====

Remove the parentheses from the syntax of the Execute method in the sample

code. The corrected code is listed below.

NOTE: In the following sample code, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this code in Access Basic.

```
Sub PostRecords_Click()
 Dim MyWS As Workspace, MyDB As Database
 On Error GoTo TransferFailed

 Set MyWS = DBEngine.workspaces(0)
 Set MyDB = MyWS.Databases(0)

 MyWS.BeginTrans
 MyDB.Execute "INSERT INTO RmtOrdersEmpty SELECT * from _
 LclOrders", DB_FAILONERROR
 MyDB.Execute "INSERT INTO RmtOrderDetailsEmpty SELECT * from _
 LclOrderDetails", DB_FAILONERROR
 MyDB.Execute "Delete from LclOrders"
 MyDB.Execute "Delete from LclOrderDetails"
 MyWS.CommitTrans
 Me.Requery
 Exit Sub

TransferFailed:
 MsgBox Error$
 MyWS.Rollback
 Exit Sub
End Sub
```

#### REFERENCES

=====

Microsoft Access Developer's Toolkit "Advanced Topics," version 2.0,  
Chapter 7, "Developing Client/Server Applications," page 196

Additional reference words: 2.00 adt  
KBCategory: kbusage kbdocerr kberrmsg  
KBSubcategory: DcmMnl

**PRB: Syntax Errors in "Advanced Topics" Sample Functions**  
**Article ID: Q120909**

---

The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

SYMPTOMS

=====

Pages 36-37 of the Microsoft Access Developer's Toolkit version 2.0 "Advanced Topics" manual contains a sample function called ReattachTables(). The code for the ReattachTables() function contains the following lines:

```
On Error GoTo ReattachTables_Err
ReattachTables_Err:
```

The syntax of these lines is incorrect.

RESOLUTION

=====

Change the lines in the ReattachTables() function to be:

```
On Error GoTo ReattachTables_Err
ReattachTables_Err:
```

MORE INFORMATION

=====

The CheckAttached() and ReattachTables() functions are meant to be run from an extended installation database at the end of a custom Setup program to check and correct attachments in the application database installed by the Setup program. As they are listed in the "Advanced Topics" manual, however, these functions check the attachments in the current database instead of in an external database.

The syntax corrections in this article cause the functions to check and correct attachments in an external database as they were designed to do.

REFERENCES

=====

For more information about extended installation databases, please see the Microsoft Access Developer's Toolkit "Advanced Topics" manual, version 2.0, Chapter 2, "Creating a Custom Setup Program," page 31.

Additional reference words: 2.00

KBCategory: kbusage kbdocerr

KBSubcategory: DcmMnl

**PRB: System Error When Sharing Database on LAN Man/Workgroups**  
**Article ID: Q100160**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

If you try to save a Microsoft Access database larger than 2000 bytes (approximately 2 MB) on a Microsoft LAN Manager server with a boot drive that uses the file allocation table (FAT) format, the following error message appears:

System error... Network error on drive <x>

This error occurs only on Microsoft Windows for Workgroups workstations if you attempt to open a database already in use by a Microsoft LAN Manager workstation or another Windows for Workgroups workstation.

CAUSE

=====

The Microsoft LAN Manager server responds incorrectly to a request by Windows for Workgroups to share the file. The server responds with a message that is too long for Windows for Workgroups to process correctly, resulting in an error from the NetBIOS.

RESOLUTION

=====

The following are two ways to work around this problem:

- Change the format of the Microsoft LAN Manager server boot drive from FAT to high-performance file system (HPFS), then place the Microsoft Access database you want to share on the boot drive.

-or-

- Move the database to a Windows for Workgroups server.

Additional reference words: 1.00 1.10 2.00 W4W FAT HPFS NetBIOS boot drive

KBCategory: kbusage

KBSubcategory: GnlOthr

**PRB: Type Conversion Failure when Importing Excel Data**  
**Article ID: Q109376**

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
  - Microsoft Excel for Windows, version 4.x
- 

SYMPTOMS

=====

When you are importing a Microsoft Excel spreadsheet to a new table in a Microsoft Access database, you receive the error message "Type Conversion Failure."

CAUSE

=====

When you import a Microsoft Excel spreadsheet to a new table, Microsoft Access checks the first row of the spreadsheet to determine the field types for the new table. If the first row of the spreadsheet contains a numeric value in a column that allows alphabetic and numeric characters, Microsoft Access will define the field type in the table for that column as numeric. If there are alphabetic characters in other rows in this column in the spreadsheet, the error message stated above will occur.

RESOLUTION

=====

To work around this behavior, insert an extra row of data as the first row in the spreadsheet. Insert alphabetic characters in any columns that are defined as alphanumeric columns. Enter numeric characters in any columns that are defined as numeric columns. Once you have imported the spreadsheet into Microsoft Access, delete the extra row of data from the spreadsheet and the table.

MORE INFORMATION

=====

Steps to Reproduce Behavior

---

1. Open the sample database NWIND.MDB.
2. Export the Employees table as a Microsoft Excel spreadsheet named EMPLOYEE.XLS.
3. Open EMPLOYEE.XLS in Microsoft Excel.
4. Go to the first row of the Postal Code column. Type 98122 (without quotation marks) and press ENTER.
5. Close the spreadsheet and save your changes.
6. In Microsoft Access, import the spreadsheet as a new table. Make sure



to select the First Row Contains Field Names check box in the Import Spreadsheet Options dialog box.

7. After the import is complete, an Import Errors table will appear in the Database window.
8. Open the Import Errors table. Note the "Type Conversion Failure" error for all postal codes that contain alphabetic characters.

#### REFERENCES

=====

For more information on setting the field type when you are importing files into Microsoft Access, query on the following words here in the Microsoft Knowledge Base:

import and files and first and line

Additional reference words: 1.00 1.10 2.00

KBCategory: kbinterop

KBSubcategory: IsmExl4

## **PRB: Unable to Change Permissions (Security)**

**Article ID: Q93718**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

When you try to open an object or change permissions for an account, you receive one of the following error messages:

You can't view this object's permissions.

-or-

No permission for <object name>.

### CAUSE

=====

Only the database creator, or a member of the Admins group in the workgroup in which the database was created, or (in version 2.0) a user with administrator permission on a particular object has full permissions to database objects.

### RESOLUTION

=====

Contact your database administrator. If you are the database administrator, check to make sure you are logged in as a user who is a member of the Admins group. If you still have the original user Admin, make sure it is a member of the Admins group and try again.

### STATUS

=====

This behavior is by design.

### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 25, "Administering a Database System," pages 609-629

Microsoft Access "User's Guide," version 2.0, Chapter 14

For more information search for "Error Messages: Reference" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: ScrtOthr



**PRB: Unable to Import Portion of Fixed-Width Text Files**  
**Article ID: Q94417**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

You cannot import any portion of a fixed-width text file.

CAUSE

=====

Each record in a fixed-width text file must be separated from the next by a carriage return and line feed, and all the records must contain the same number of characters. The text file you are trying to import does not meet these criteria.

RESOLUTION

=====

Modify the text file to ensure that all records meet these requirements:

- Each record is on a different line.
- Each record contains the same number of characters.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Microsoft Access cannot import fixed-width text records that are not delimited by a carriage return or line feed.

Mainframe users may be familiar with long strings of data where the first x number of characters are the first record, the next x number of characters are the second record, and so on. In order to import these records into Microsoft Access, you must ensure that each record is on its own line with a carriage return and line feed at the end of each record.

For example, consider the following string of characters:

lastnamefirstnameagelastnamefirstnameagelastnamefirstnameage

You cannot import these records into Microsoft Access without modification. If you try to import the above string as a fixed-width text file, you will either create an empty table or you will be unsuccessful and may need to repair your database.

To modify the file, use Access Basic and follow these steps:

1. Read in the file in Binary mode with a fixed-length string that is the same length as the record.
2. Write each string to a new file that is open for sequential output.
3. Repeat steps 1 and 2 until you reach the end of file.

Microsoft Access can import the following records:

```
lastnamefirstnameage
lastnamefirstnameage
lastnamefirstnameage
```

To import a file containing records like these, follow these steps:

1. From the File menu, choose Imp/Exp Setup.
2. Change the Text Delimiter from " to {none}.
3. Enter the following settings for Field Information:

| Field Name | Data Type | Start | Width |
|------------|-----------|-------|-------|
| Last       | Text      | 1     | 8     |
| First      | Text      | 9     | 9     |
| Age        | Number    | 18    | 3     |

4. Choose OK, and save this specification as TEST.
5. From the File menu, choose Import.
6. Select Text (Fixed-Width), then choose OK.
7. Select the text file you want to import, then choose Import.
8. In the Import Text Options dialog box, select the TEST specification. Choose OK.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 1.0, Chapter 4, "Importing, Exporting, and Attaching," pages 75-81

Additional reference words: 1.00 1.10 2.00

KBCategory: kb3rdparty

KBSubcategory: IsmTtxtfx

**PRB: UndoCurrentRecord In BeforeUpdate Property Causes Error**  
**Article ID: Q97525**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTONS

=====

If you create a macro that contains a DoMenuItem action that chooses Undo Current Record from the Edit menu, and you run the macro from a form's BeforeUpdate property, Microsoft Access generates the following message when you edit data in the form:

Command not available: UndoCurrentRecord.

However, if you select the Edit menu, Undo Current Record is available and works properly.

CAUSE

=====

The Undo Current Record command is not available for use with a macro run from a form's BeforeUpdate property.

RESOLUTION

=====

In your macro, enter CancelEvent as the first action, then follow that with the SendKeys action to send an ESC key. (This has the same effect as choosing the Undo Current Field command from the Edit menu.)

MORE INFORMATION

=====

Page 48 of the Microsoft Access "Language Reference," versions 1.0 and 1.1, discusses uses for this type of macro in the BeforeUpdate property, and indicates that calling Undo Current Record works correctly in such a macro. This is not the case. Calling Undo Current Record causes the error message above and should not be used.

REFERENCES

=====

Microsoft Access "Language Reference," versions 1.0 and 1.1, "BeforeUpdate, AfterUpdate Properties," pages 47-48

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage kberrmsg kbdocerr

KBSubcategory: McrActn

**PRB: Unexpected Cursor Behavior in Text Box with Input Mask**  
**Article ID: Q114507**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you click in a text box that has a defined InputMask property, the insertion point remains where it was in the text box when you clicked the mouse button, instead of moving to the beginning of the text box.

RESOLUTION

=====

Press F2 to place the cursor at the beginning of the text box, or use the TAB key to navigate to the text box instead of clicking in the text box.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

An input mask uses predefined place holders in the text box. If you tab to the text box, the insertion point is placed at the beginning of the input mask. If you enter the text box by clicking in it, the insertion point remains where it was when you clicked the mouse button.

You can make the insertion point move to the beginning of the text box by automatically pressing F2 when a text box is entered. To do this, create a macro with the following action and call it from the text box's OnEnter property:

```
SendKeys
 Keystrokes: {F2}
 Wait: No
```

REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 24, "Macro Basics," and Chapter 26, "Using Macros with Forms"

Additional reference words: 2.00

KBCategory: kbusage

KBSubcategory: FmsProp

**PRB: Unexpected Results with Make-Table or Append Query**  
**Article ID: Q90879**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Using a make-table (SQL Select...Into) or an append (SQL Insert...Into) query with criteria that have no matching records causes an empty table to be created. In Microsoft Access version 1.x, if the empty table contains a counter field, the first record added to the table will have a counter value of 1. In Microsoft Access version 2.0, the first record added will have a counter value of 2.

CAUSE

=====

This behavior is by design.

REFERENCES

=====

For more information about make-table or SELECT INTO queries, search for "make-table query" or "SELECT INTO (SQL)" using the Microsoft Access Help menu.

For more information about append or INSERT INTO queries, search for "append queries" or "INSERT INTO (SQL)" using the Microsoft Access Help menu.

Additional reference words: 1.00 1.10 2.00

KBCategory: kbusage

KBSubcategory: QryMktbl



**PRB: Unknown Reference or Function in Validation Expression**  
**Article ID: Q113298**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you try to save a table that has a field name, control name, or user-defined function in a validation rule or default value, you receive the error message:

Unknown or invalid reference '<field>' in validation expression or default value in table '<Table>'.

-or-

Unknown function '<function>' in validation expression or default value in '<Table>'.

CAUSE

=====

Validation rules are enforced by the Microsoft Access Jet database engine in Microsoft Access version 2.0. There are some rules allowed in Microsoft Access versions 1.0 and 1.1 that are not allowed at the field level in Microsoft Access 2.0. In version 2.0, validation rules at the field level cannot refer to other fields (whether in the current table or another table), controls on a form, or user-defined functions.

RESOLUTION

=====

To get the validation rule you want, create the rule at the table level instead of the field level. Note that validation rules on a form can refer to other controls on the form.

STATUS

=====

This behavior is by design.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Start Microsoft Access and either create a new database or open one of your choosing.
2. Create a new table.

3. Add two text fields named ABC and DEF to the table.

3. Set the ValidationRule property of the DEF control to:

<[ABC]

4. Save the table (do not define a primary key). You will receive one of the error messages stated above.

Additional reference words: 2.00 Tables

KBCategory: kbusage

KBSubcategory: TblDsign

## **PRB: Update Query Based on Totals Query Fails**

Article ID: Q116142

---

The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

### SYMPTOMS

=====

When you run an update query, you receive the following error message:

Operation must use an updatable query.

### CAUSE

=====

This error message occurs when the values in the query's Update To row are fields in either a crosstab query or a select query or subquery that contains totals or aggregate functions. The error message indicates that the records in the totaling query cannot be updated.

### RESOLUTION

=====

There are three methods of working around this behavior. The first method processes one record at a time, instead of using a totaling query. The second method uses a temporary, or intermediate, table to hold the results of the totaling query. The third method uses a domain function instead of a totaling query.

#### Method 1: Processing One Record at a Time

---

The following example demonstrates how to add a new value to an existing value and place the result back into a field.

**CAUTION:** Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open the Products table in Design view.
3. Add a new field called Sales So Far with a Currency data type to the table. Save and then close the table.
4. Create a new query based on the Products and Order Details tables. The two tables will be joined automatically on the Product ID field.
5. From the Query menu, choose Update (or click the Update Query button on the toolbar).
6. Drag the Sales So Far field from the Products table to the query

grid.

7. In the Update To row of the Sales So Far column, enter the following expression.

NOTE: In the following sample expression, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this expression.

```
IIf(IsNull([Sales So Far]),[Order Details].[Unit Price]*_
[Order Details].[Quantity],[Sales So Far]+_
([Order Details].[Unit Price]*[Order Details].[Quantity]))
```

8. Run the query. When you are prompted "2813 row(s) will be updated" choose OK.

#### Method 2: Using a Temporary Table

-----

This method stores the results of the totaling query in an intermediate table, which is then used in the update query. The following example demonstrates this method.

CAUTION: Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open the Products table in Design view.
3. Add a new field called Sales So Far to the table. Give the field a Currency data type. Save and then close the table.
4. Create a new query based on the Order Details table. Add the following fields to the query:

```
Field: Product ID
Total: Group By
```

```
Field: SumPerProduct: [Unit Price]*[Quantity]
Total: Sum
```

NOTE: To see the Total field, click the Totals button on the toolbar, or choose Totals from the View menu.

From the Query menu, choose Make Table. In the Make New Table box, type "SalesSoFar Test" (without quotation marks). Save the query as Query1, and then run the query.

5. Create a new query based on the SalesSoFar Test and the Products tables.
6. Add the Sales So Far field (from the Products table) to the query grid.
7. From the Query menu, choose Update.
8. In the Update To field in the Sales So Far column, enter the following

line:

[SumPerProduct]

9. Run the query. You will receive a message stating "77 row(s) will be updated." Choose OK. The SumPerProduct information will be written to the Sales So Far field in the Products table.

#### Method 3: Using a Domain Function to Calculate the Update To Values

---

This method uses a domain function instead of a totaling query and does not require the use of an intermediate table. This method may take longer than the second method if the table being totaled is large. The following example demonstrates this method.

**CAUTION:** Following the steps in this example will modify the sample database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open the Products table in Design view.
3. Add a new field called Sales So Far to the table. Give the field a Currency data type. Save and then close the table.
4. Create a new query based on the Products table, and then choose Update from the Query menu. Add the following field to the query.

**NOTE:** In the following sample field, an underscore (\_) at the end of a line is used as a line-continuation character. Remove the underscore from the end of the line when re-creating this field.

Field: Sales So Far

Update To: DSum("[Quantity]\*[Unit Price]","Order \_  
Details","[Product ID]=" & [Product ID])

5. Run the query. You will receive a message stating "77 row(s) will be updated." Choose OK. The information will be written to the Products table.

STATUS

=====

This behavior is a design limitation. Changes to this behavior are being reviewed and will be considered for inclusion in a future release of Microsoft Access.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

**CAUTION:** Following the steps in this example will modify the sample

database NWIND.MDB. You may want to back up the NWIND.MDB file, or perform these steps on a copy of the NWIND database.

1. Start Microsoft Access and open the sample database NWIND.MDB.
2. Open the Products table in Design view.
3. Add a new field called Sales So Far to the table. Give the field a Currency data type. Save and then close the table.
4. Create a new query based on the Order Details table. Add the following fields to the query, and then save the query as Query1:

Field: Product ID  
Total: Group By

Field: SumPerProduct: [Unit Price]\*[Quantity]  
Total: Sum

NOTE: To see the Total field, click the Totals button on the toolbar, or choose Totals from the View menu.

This query creates a list of all the products sold (grouped by the Product ID), and the total dollar amount from those sales.

5. Create a new query based on Query1 and the Products table.
6. Add the Sales So Far field (from the Products table) to the query grid.
7. From the Query menu, choose Update.
8. In the Update to field in the Sales So Far column, enter the following line:

[SumPerProduct]

9. Run the query. You will receive the error message stated above.

#### REFERENCES

=====

Microsoft Access "User's Guide," version 2.0, Chapter 13, "Changing Data with Action Queries," pages 311-312

For additional information about using domain functions, please see the following articles in the Microsoft Knowledge Base:

ARTICLE-ID: Q103403  
TITLE : INF: Custom Domain Functions Similar to DFirst() and DLast()

ARTICLE-ID: Q108098  
TITLE : INF: DLookup() Usage, Examples, and Troubleshooting Tips

Additional reference words: 1.00 1.10 2.00 querying updating updateable  
KBCategory: kbusage kberrmsg  
KBSubcategory: QryUpdat



**PRB: User Needs Read/Write Privileges for WORKDIR Directory**  
**Article ID: Q117912**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SYMPTOMS

=====

When you start Microsoft Access on a network workstation, you receive the following error messages:

Could not open SYSTEM.MDA

-and-

Can't find SYSTEM.MDA

Or, you receive the following compile error messages while Microsoft Access is loading the add-ins:

Compile time error

-and-

Tried to load module with duplicate procedure name

CAUSE

=====

Microsoft Access was installed on a network server with the WORKDIR directory set to Read Only.

RESOLUTION

=====

Make sure that all Microsoft Access users have at least Read and Write privileges for the WORKDIR directory.

MORE INFORMATION

=====

When you start Microsoft Access on a network workstation, you must have write privileges for the following .MDA files in the WORKDIR directory on the server: SYSTEM.MDA, WZLIB.MDA, WZTABLE.MDA, WZBLDR.MDA, WZFRMRPT.MDA, and WZQUERY.MDA (the Wizard files). The file UTILITY.MDA is installed in the parent directory of the WORKDIR directory (whether this is an Office install or Microsoft Access install) and should be read only.

To verify that the problem you are having is due to incorrect privileges for the WORKDIR directory, follow these steps:

1. Set the attributes for the WORKDIR directory (not just the individual files) to Read/Write.



2. Use a supervisory-level login to start Microsoft Access on the workstation.

REFERENCES

=====

For more information about network installations, see the "Network Installations" topic in the ACREADME.HLP file (in the ACCESS directory).

Additional reference words: 2.00

KBCategory: kbinterop

KBSubcategory: IntpOthr

**PRB: Validation Rules Evaluate Expressions in Unexpected Ways**  
**Article ID: Q104972**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

Microsoft Access evaluates expressions with logical operators differently than you expect. For example, entering zero in a field with the validation rule "=true and not true" produces the error message:

The value you entered is prohibited by the validation rule set for this field.

CAUSE

=====

Microsoft Access evaluates expressions in parts, and compares each part to the value in the field being validated. The expression "=true and not true" is evaluated as:

Field=True AND Field<>True

This results in the error message stated above if you enter zero in the field because the validation fails.

RESOLUTION

=====

Use parentheses in the expression to change the way it is evaluated. The correct syntax for the example above is:

=(True AND NOT True)

Microsoft Access will interpret this expression as:

Field=(True AND NOT True)

Zero is an acceptable value for a field with the above validation rule.

MORE INFORMATION

=====

Steps to Reproduce Behavior

-----

1. Open a new, blank form in Design view.
2. Add a text box with the following validation rule to the form:

Text box: Field1  
Validation rule: =True and not True

3. View the form in Form view.

4. Enter a zero in the text box and then press ENTER. An error message will occur.

Additional reference words: 1.00 1.10 2.00 Boolean bitwise math

KBCategory: kbusage

KBSubcategory: GnlValid

**PRB: White Lines or Missing Data When Printing Reports**  
**Article ID: Q103180**

-----  
The information in this article applies to:

- Microsoft Access versions 1.0, 1.1, and 2.0
- 

SYMPTOMS

=====

The following problems can occur when you are printing from Microsoft Access using the Canon BJC800 or BJ10E printer drivers:

- Horizontal white lines appear on the report.
- Portions of the text are not printed.
- You can only print the report using a lower resolution than normal.
- You can only print color reports in monochrome.
- You can only print the report one page at a time, rather than printing the whole document at once.

NOTE: These problems can occur with other printer drivers also.

CAUSE

=====

These problems are due to an interaction between the way Microsoft Windows sends data to the printer, and the amount of time the printer needs to interpret the data. Microsoft Access treats lines or borders as a bitmap or large graphic, and the printer needs more time to interpret the data.

RESOLUTION

=====

Change the default settings for the printer in the Microsoft Windows Control Panel. Choose the Printers icon in the Control Panel, and then choose Connect. Make the following entries in the Timeouts (seconds) box:

| Option              | Setting |
|---------------------|---------|
| Device Not Selected | 15      |
| Transmission Retry  | 90      |

The default setting for Device Not Selected is 15, and Transmission Retry is 45. These settings may need to be increased even further, to 30 and 120 or higher for printing complex graphics images, or for printing over a network.

Additional reference words: 1.00 1.10 2.00 PtrIss whitespace  
KBCategory: kbprint

KBSubcategory: PtrProb

**WX0994: Form and Report Questions and Answers**

**Article ID: Q114809**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

The "Form and Report Questions and Answers" (WX0994) Application Note contains questions and answers about Microsoft Access version 2.0 forms and reports.

You can obtain this Application Note from the following sources:

- Microsoft FastTips Technical Library
- Microsoft Product Support Services

For complete information, see the "To Obtain This Application Note" section at the end of this article.

THE TEXT OF WX0994

=====

Microsoft(R) Product Support Services Application Note (Text File)  
WX0994: FORM AND REPORT QUESTIONS AND ANSWERS

=====

Revision Date: 3/94  
No Disk Included

The following information applies to Microsoft Access(R) version 2.0.

-----

|  |                                                                     |  |
|--|---------------------------------------------------------------------|--|
|  | INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY     |  |
|  | ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |  |
|  | Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER      |  |
|  | EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED      |  |
|  | WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR       |  |
|  | PURPOSE. The user assumes the entire risk as to the accuracy and    |  |
|  | the use of this Application Note. This Application Note may be      |  |
|  | copied and distributed subject to the following conditions: 1) All  |  |
|  | text must be copied without modification and all pages must be      |  |
|  | included; 2) If software is included, all files on the disk(s)      |  |
|  | must be copied without modification (the MS-DOS(R) utility          |  |
|  | diskcopy is appropriate for this purpose); 3) All components of     |  |
|  | this Application Note must be distributed together; and 4) This     |  |
|  | Application Note may not be distributed for profit.                 |  |
|  |                                                                     |  |
|  | Copyright (C) 1994 Microsoft Corporation. All Rights Reserved.      |  |
|  | Microsoft, Microsoft Access, and MS-DOS are registered trademarks   |  |
|  | and Windows is a trademark of Microsoft Corporation.                |  |

-----

1. Q. Why is the data sorted in my query but not in my report?
  - A. Reports create their own internal queries to present the data.

If you want to have the data in your report presented in a particular order, you must explicitly set the sort order in the Sorting And Grouping dialog box. To do this, open the report in Design view, then choose Sorting And Grouping from the View menu.

2. Q. How can I keep a group of records together on a report?

A. The new KeepTogether property for groups in Microsoft Access version 2.0 gives you the ability to keep groups of like information together. This property is available in the Sorting And Grouping dialog box for reports. Using this property, you can keep an entire group together (including the group header, all records, and the group footer), or keep the group header with the first record.

For more information about sorting and grouping, please see the Microsoft Access "User's Guide," version 2.0, Chapter 22, "Sorting and Grouping Data."

3. Q. Why is every other page of my report blank, and how can I correct this problem?

A. This problem occurs when the total width of your report exceeds the width of the paper specified in the Print Setup dialog box. For example, blank pages print if your report form is 8 inches wide and your left and right margins are 1 inch wide for a total width of 10 inches, and if the paper size specified in the Print Setup dialog box is only 8.5 inches wide.

Using this example, if controls (such as text boxes) extend beyond 8.5 inches, the controls are printed on a second page. Otherwise, you receive a warning message stating that some pages may be blank. Blank pages generated after the warning are not counted in the total pages of your report.

For additional information about preventing blank pages, you can order item number Q95920 by selecting the FastTips Technical Library option from the FastTips Main Menu.

4. Q. What is an event procedure? Can I still call an Access Basic function from my form or report?

A. Event procedures are Access Basic functions that respond to events (such as a mouse click) that occur on forms and reports. Event procedures are stored in form or report modules attached to the form or report, and are incorporated into the form or report's design structure. You can also call an Access Basic function stored in a separate module from an event property.

For more information about event procedures, order the fax or mail copy of this script.

- Event procedures are private procedures--only the form or report to which an event procedure is connected can call or use that event procedure. If you want another form or report to use the same procedure, make the procedure a global

function by storing it in a module. Functions stored in modules are available to all forms and reports.

- Since event procedures are private, or locally scoped, you can use the Me property to refer to the form or report, rather than using the full syntax (such as Forms![form name]...) to refer to the form or report.
- Form and report modules are loaded only when the form or report is open. If you want to compile the procedure or search in the module, you must first open the form or report.

To call an Access Basic function from a property, enter the following in the property:

```
=MyFunctionName()
```

NOTE: The equal sign and parentheses are required.

To create an event procedure, choose the Build button in the property sheet, choose the Code Builder, and write the code for that event. When you complete and close the code, you will see [Event Procedure] displayed in the property sheet, which indicates that the event has code written for it.

For more information about writing and using event procedures, see the Microsoft Access "Building Applications" manual, version 2.0, Chapter 3, "Introduction to Access Basic," and Chapter 5, "Access Basic Fundamentals."

5. Q. How can I print a page number that includes the full number of pages in the report on each page of a report?
- A. You can use the Page and the Pages properties to create a "Page X of Y" expression. The following example, used as a text box's ControlSource property, will print a "Page X of Y" page number on each page of the report:

```
="Page " & Page & " of " & Pages
```

6. Q. Why do I get the message "#Error" in some controls on my form or report? How do I reference a control on a subform or subreport?
- A. Microsoft Access displays the "#Error" message in a field or text box when it cannot find necessary information, execute an expression, or store a value within the field's defined limits. "#Error" is not the only possible error message for these conditions. For additional information about troubleshooting "#Error" messages, you can order item number Q112103 by selecting the FastTips Technical Library option from the FastTips Main Menu.
7. Q. Do form validation rules override table validation rules? When are the validation rules on a form evaluated?



- A. In Microsoft Access 2.0, table validation rules are always enforced, no matter how you add or edit data. Form validation rules do not override table validation rules, but can be used in addition to table validation rules.

When you enter or edit data in a form and then move the insertion point to a different field or record, the form validation rule is evaluated first. If the data passes this validation test, the table validation is evaluated. Therefore, you can use form validation to refine table validation.

Since validation rules are evaluated only when data is added or edited, you may want to use the new Required property to force users to enter a value. The Required property prevents users from tabbing out of and not changing a field.

For additional information about the Required property, search for "Required" then "Required Property" using the Microsoft Access Help menu.

- 8. Q. When is it appropriate to use an exclamation point (!) versus a period (.) for identifying objects and properties in an expression?
  - A. Use an exclamation point before anything you specifically name, such as the name of a form or a control on a form, and use a period before anything Microsoft Access names, such as a property.

In Microsoft Access version 1.x, you can use a period to reference a field name, such as "MyTable.Name" to refer to the Name field in the table MyTable. If, however, you use this same expression in Microsoft Access 2.0, you refer to the Name property, rather than the field called Name. To prevent this, use "MyTable!Name" in Microsoft Access 2.0.

- 9. Q. How do I turn off the ControlWizards?
  - A. ControlWizards help you create list boxes, combo boxes, option groups, and command buttons. Once you are comfortable working with these controls, you may want to turn off the ControlWizards instead of having to cancel them each time you add one of these controls. To turn off the ControlWizards, either choose ControlWizards from the View menu to clear the check mark, or choose the ControlWizards button in the toolbox to toggle the button off.

- 10. Q. How can I check for duplicate records immediately after I enter a value in a primary key field?
  - A. Normally, Microsoft Access does not check the values in primary key fields for duplicates until you move to the next record. If you want to check for duplicate values immediately after entering a value in a primary key field, use a macro in the field's AfterUpdate property. The macro should use the DLookup() function to check for duplicates and then display an appropriate message.

For additional information about checking for duplicate primary keys, you can order item number Q102527 by selecting the FastTips Technical Library option from the FastTips Main Menu.

-----  
TO OBTAIN THIS APPLICATION NOTE  
=====

- You can have this Application Note mailed or faxed to you from the automated Microsoft FastTips Technical Library, which you can call 24 hours a day, 7 days a week at (800) 936-4100. NOTE: The FastTips Technical Library is available only to customers within the U.S. and Canada.
  
- If you are unable to access the source(s) listed above, you can have this Application Note mailed or faxed to you by calling Microsoft Product Support Services Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time at (206) 635-7050. If you are outside the United States, contact the Microsoft subsidiary for your area. To locate your subsidiary, call Microsoft International Customer Service at (206) 936-8661.

Additional reference words: 2.00  
KBCategory: kbusage kbfasttip kbappnote  
KBSubcategory: Fstqa

## WX0995: Table and Query Questions and Answers

Article ID: Q114810

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

The "Table and Query Questions and Answers" (WX0995) Application Note contains questions and answers about Microsoft Access version 2.0 tables and queries.

You can obtain this Application Note from the following sources:

- Microsoft FastTips Technical Library
- Microsoft Product Support Services

For complete information, see the "To Obtain This Application Note" section at the end of this article.

THE TEXT OF WX0995

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0995: TABLE AND QUERY QUESTIONS AND ANSWERS  
=====

Revision Date: 3/94  
No Disk Included

The following information applies to Microsoft Access(R) version 2.0.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |  
| PURPOSE. The user assumes the entire risk as to the accuracy and |  
| the use of this Application Note. This Application Note may be |  
| copied and distributed subject to the following conditions: 1) All |  
| text must be copied without modification and all pages must be |  
| included; 2) If software is included, all files on the disk(s) |  
| must be copied without modification (the MS-DOS(R) utility |  
| diskcopy is appropriate for this purpose); 3) All components of |  
| this Application Note must be distributed together; and 4) This |  
| Application Note may not be distributed for profit. |  
| |  
| Copyright (C) 1994 Microsoft Corporation. All Rights Reserved. |  
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |  
| and Windows is a trademark of Microsoft Corporation. |  
|-----

1. Q. How have queries changed in Microsoft Access version 2.0?

A. Queries are now more flexible and are easier to create. The new

QueryWizards--Crosstab, Find Duplicates, Find Unmatched, and Archive--help you quickly create complex queries for managing your data.

There are also three new SQL-specific queries. These are union queries, data-definition queries (queries that create, change, or delete tables and indexes in Microsoft Access databases), and SQL pass-through queries (queries that pass SQL statements directly to a SQL database, providing improved connectivity in client-server relationships).

In addition, improvements in updating in multiple-table queries and the addition of TopValues queries and subqueries give you greater query functionality.

2. Q. When I add two tables to my query that do not have a defined relationship, Microsoft Access automatically joins them. Can I prevent this from happening?
- A. Microsoft Access 2.0 automatically joins two tables in a query if the tables meet the following criteria:
- There is no relationship defined between the tables.
  - Each table contains at least one field whose name and data type matches the name and data type of a field in the other table.
  - One of the tables has a primary key defined on the matching field.

Only one AutoJoin is automatically created between two tables. Even if there is more than one join possible between the two tables, a join is created only between the first fields that meet the above criteria. If you add three tables that meet the above criteria to a query, three joins are created--one for each table pair.

You cannot turn this functionality off. You must either delete the join line after it is created or manually define a relationship between the two tables.

3. Q. How does the new Relationships window work? What does it mean when I am prompted to save the layout?
- A. The new Relationships window gives you a graphical display of the tables in your database and their relationships to each other. You can customize the Relationships window and view as many or as few tables as you want. You can also create or change table relationships in this view. For more details about the Relationships window, order the fax or mail copy of this script.
- To create a relationship, drag the related field from the primary table to the related table. Microsoft Access will then display the Relationships dialog box. When you choose OK, the relationship is created and saved.

- Removing (deleting) a table from the Relationships window does not delete the table or any of its relationships from the database; it simply hides that particular table from view.
- To delete a relationship, select the join line and press the DEL key.
- To make sure that you are seeing all the defined relationships in your database, choose Show All from the Relationships menu.
- When you close the Relationships window, you are prompted to save the window's layout. Choosing Yes or No does not affect whether your relationships are saved, but only the positions of the visible tables in the Relationships window.
- To show only the relationships for a specific table, use the following three steps:
  1. From the Edit menu, choose Clear Layout, and then choose OK when you are prompted. Note that this clears the layout only and does not delete existing relationships.
  2. From the Relationships menu, choose Add Table. In the Table/Query box, select the table whose relationships you want to see, and then choose Add. Choose Close.
  3. From the Relationships menu, choose Show Direct.

NOTE: If you do not clear the layout before choosing Show Direct, the relationships for the table you selected will be displayed in addition to any other relationships that are already being displayed.

4. Q. Why can I update more fields in my query than I could in Microsoft Access version 1.x?
- A. In Microsoft Access 2.0, when a query includes fields from more than one related table, you can update data on both sides of the join. This means that in a query that combines data from two tables, you can update data from both of the tables in the query.

If you want to prevent users from updating fields in a multiple-table query, create a form based on the query and then set the Locked property for the fields you do not want users to update.

For more information about updating queries, search for "queries: updating underlying tables" and then view both of the related topics using the Microsoft Access Help menu.

5. Q. How have validation rules and default values changed?
- A. In Microsoft Access 2.0, validation rules are always enforced, no matter how you add or edit data. Because default values and

validation rules are always enforced, the following items are not valid in default values or field-level validation rules:

- References to fields or controls. Instead, use record validation rules when you are comparing two or more fields. For more information about record validation rules, search for "validation: rules and text" then "Validating Data in a Table" using the Microsoft Access Help menu.
- User-defined functions.
- Microsoft Access domain functions.
- Aggregate functions.
- CurrentUser() or Eval() functions.

6. Q. Why can't I set referential integrity?

A. Referential integrity is a set of rules that preserves the defined relationships between tables when you enter or delete records. If you enforce referential integrity, you cannot add a record to a related table when there is no associated record in the primary table, change a value in a primary table that would result in an orphan record in a related table, or delete a record from a primary table when there is a matching related record.

If you select the Cascade Update Related Fields or Cascade Delete Related Records option for a relationship, Microsoft Access will change or delete related records to ensure that the referential integrity rules are enforced when you change or delete records.

There are several possible reasons why you can't enforce referential integrity when you are defining a relationship. For additional information, you can order item number Q112111 by selecting the FastTips Technical Library option from the FastTips Main Menu.

7. Q. How can I optimize my queries?

A. Microsoft Access 2.0 uses Rushmore, a data-access technology that permits sets of records to be queried very efficiently. Here is a summary of query performance tips:

- Index the fields used in sorts and criteria.
- Index the fields used in joins in both tables.
- Use multiple-field indexes on fields where there are multiple-column joins between the tables.
- If a table has a single-field primary key, do not add a separate index to the primary key field. If a table has a multiple-field primary key, it may help to have a separate index on each field.

- Use outer joins only when necessary--outer joins limit the options for the query optimizer.

For more information about query performance, search for "Rushmore technology" then "Optimizing Queries with Rushmore Technology" using the Microsoft Access Help menu. Or, you can order item number Q112112 by selecting the FastTips Technical Library option from the FastTips Main Menu.

8. Q. Why do I see a number instead of "(counter)" for my counter field?

A. Microsoft Access 2.0 enters a counter value when you start to edit a new record. In Microsoft Access 1.x, this value was entered after you saved the record.

Since this value is now provided earlier, if you start editing a new record and then cancel the record, the counter value is still used, even though no record is stored with the value. For example, when you add a new record to a table containing two records, the counter value is 3. If you cancel this new record and then later add another new record, the counter value is 4 for the new record.

Counter values are not reused when you delete records. For example, if in a table of 15 records you delete the last three records and then add a new record, the counter value for the new record is 16. To reset the next available counter value, compact the database. After the database is compacted, the next available counter is set to one higher than the last counter value in the table.

For additional information about counters, you can order item number Q112160 by selecting the FastTips Technical Library option from the FastTips Main Menu.

9. Q. While I was creating a query, when I switched from a Select query to a SQL Specific query, the SQL statement disappeared. When I switched back to a Select query, my query disappeared. Where did my query go?

A. When you switch to a SQL Specific query, the SQL statement from the existing query is deleted. If you want to start with a Select query to help create the SQL statement for a SQL Specific query (to save some typing) use the following six steps:

1. Create the Select query.
2. From the View menu, choose SQL.
3. Select the entire SQL statement, except for the ending semicolon.
4. Copy the selected text to the Clipboard by pressing CTRL+C, or by choosing Copy from the Edit menu.

5. From the Query menu, choose SQL Specific, then choose a query type.
  6. Paste the query text into the Query window by pressing CTRL+V, or by choosing Paste from the Edit menu.
10. Q. How do I remove the underline characters in my input masks?
- A. An input mask consists of up to three parts, separated by semicolons. The first part specifies the input mask itself, the second part specifies whether Microsoft Access stores the literal display characters in the table when you enter data, and the third part specifies the character that Microsoft Access displays in the input mask for spaces.

If you omit the third part, Microsoft Access uses an underline character for the input mask. For example, in the Customers table in the sample database NWIND.MDB, the input mask on the Customer ID field is:

```
>LLLLL
```

When you enter data in a new record, five underlines are displayed in the field. You can replace these underlines with spaces by using the following for the input mask:

```
>LLLLL; ;" "
```

For more information about input masks, search for "input mask" then "Adding an Input Mask" using the Microsoft Access Help menu.

---

-----

TO OBTAIN THIS APPLICATION NOTE

=====

- You can have this Application Note mailed or faxed to you from the automated Microsoft FastTips Technical Library, which you can call 24 hours a day, 7 days a week at (800) 936-4100. NOTE: The FastTips Technical Library is available only to customers within the U.S. and Canada.
- If you are unable to access the source(s) listed above, you can have this Application Note mailed or faxed to you by calling Microsoft Product Support Services Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time at (206) 635-7050. If you are outside the United States, contact the Microsoft subsidiary for your area. To locate your subsidiary, call Microsoft International Customer Service at (206) 936-8661.

Additional reference words: 2.00  
KBCategory: kbusage kbfasttip kbappnote  
KBSubcategory: Fstqa





## WX0996: Setup and Conversion Questions and Answers

Article ID: Q114811

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

The "Setup and Conversion Questions and Answers" (WX0996) Application Note contains questions and answers about Microsoft Access version 2.0 Setup and conversions.

You can obtain this Application Note from the following sources:

- Microsoft FastTips Technical Library
- Microsoft Product Support Services

For complete information, see the "To Obtain This Application Note" section at the end of this article.

THE TEXT OF WX0996

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0996: SETUP AND CONVERSION QUESTIONS AND ANSWERS  
=====

Revision Date: 3/94  
No Disk Included

The following information applies to Microsoft Access(R) version 2.0.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |  
| PURPOSE. The user assumes the entire risk as to the accuracy and |  
| the use of this Application Note. This Application Note may be |  
| copied and distributed subject to the following conditions: 1) All |  
| text must be copied without modification and all pages must be |  
| included; 2) If software is included, all files on the disk(s) |  
| must be copied without modification (the MS-DOS(R) utility |  
| diskcopy is appropriate for this purpose); 3) All components of |  
| this Application Note must be distributed together; and 4) This |  
| Application Note may not be distributed for profit. |  
| |  
| Copyright (C) 1994 Microsoft Corporation. All Rights Reserved. |  
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |  
| and Windows is a trademark of Microsoft Corporation. |  
|-----|

1. Q. Is there an easy way for me to remove all or part of Microsoft Access from my computer?

- A. Microsoft Access Setup now has a maintenance mode for adding or removing Microsoft Access components. After you originally install Microsoft Access, Setup automatically runs in maintenance mode. To remove the entire Microsoft Access program from your hard disk, start Setup and then choose the Remove All button. Setup will remove the Microsoft Access program and the associated icons in Program Manager, but will not remove the Microsoft Access directory, the Program Manager group, user-created .MDB and .LDB files, or shared .DLL files in the WINDOWS\SYSTEM directory.

To change specific components, choose the Add/Remove button in Setup. Select the check box for any component you want to add, or clear the check box for any component you want to remove. Setup prompts you to verify the components you are removing. If a selected component is already installed, Setup does not reinstall it. To reinstall a component, you must first remove the component and then install it. An example is available in the fax or mail copy of this script.

For example, to reinstall the sample database NWIND.MDB, use the following 10 steps:

1. Start Microsoft Access Setup.
  2. Choose Add/Remove.
  3. In the Options box, select Sample Apps, and then choose Change Option.
  4. Clear the Northwind Traders check box, and then choose OK.
  5. Choose Continue, and then choose Yes when you are prompted to confirm that you want to remove the component.
  6. Start Setup again after it finishes.
  7. Choose Add/Remove.
  8. In the Options box, choose Sample Apps, and then choose Change Option.
  9. Select the Northwind Traders check box, and then choose OK.
  10. Choose Continue.
2. Q. Why does the Microsoft Access Setup program not complete successfully?
- A. There are several things that may prevent the Microsoft Access Setup program from completing successfully. Four troubleshooting tips are available in the fax or mail copy of this script.
1. Make sure you have enough free disk space. Disk-compression

utilities, such as DoubleSpace, make your hard disk seem larger than it actually is by compressing the files on the disk. However, if you set the DoubleSpace compression ratio for your hard disk to 16 to 1, you might see 160 MB of free space where there is actually only 10 MB of space available.

2. Make sure that your problem is not due to a bad installation disk by copying the disk to an empty directory on your hard disk. If the disk can be copied with no errors, then the disk is not bad. If you do receive an error message when you are copying the disk, obtain a replacement disk by calling Microsoft Sales Information Center at (800) 426-9400.
3. Make sure that the PATH statement in your AUTOEXEC.BAT file includes the complete path to your Windows directory, including the drive letter. For example, if your Windows directory is named WINDOWS and is on drive C, your PATH statement must include:

```
path=c:\windows;c:\dos
```

IMPORTANT: If you change the PATH statement, you must restart your computer before the change takes effect.

4. In your CONFIG.SYS and AUTOEXEC.BAT files, include only the devices or drivers necessary to run Windows and your computer. Remove any third-party memory managers and virus-checking software. In Windows, close all other applications, including virus-checking software and screen savers, before you run Microsoft Access Setup.

For additional information about configuring your CONFIG.SYS and AUTOEXEC.BAT files and troubleshooting Setup problems, you can order item number Q96109 by selecting the FastTips Technical Library option from the FastTips Main Menu.

3. Q. Can I install Microsoft Access 2.0 over my current installation of Microsoft Access version 1.x?

- A. Microsoft Access 2.0 is designed to install over Microsoft Access 1.x files. However, to avoid any potential naming conflicts among database files, you may want to install Microsoft Access 2.0 in a new directory, or at least back up your existing database files before running Setup. Install Microsoft Access 2.0 in a different directory if you want to retain your previous version of Microsoft Access.

If you plan to create or modify databases that are compatible with Microsoft Access 1.x, you should install Microsoft Access 2.0 in a different directory. Once you convert a version 1.x database to version 2.0, you will not be able to open that database in version 1.x again. Microsoft Access 2.0 can open version 1.x databases and modify the data, but will not allow you to create new objects or modify existing objects. To do this, you must convert the database to version 2.0.

4. Q. How do I set up Microsoft Access 2.0 for shared use on a

network?

- A. Network installation is a two-step process. First, install Microsoft Access on the network file server by running the administrator Setup command `SETUP /A` from the Setup disk. Next, set up each workstation by running Setup from the network file server. Note that there is no `SETUP /N` command to set up workstations as there was in Microsoft Access version 1.x.

When you are setting up the workstations, you can choose to install Microsoft Access on each workstation's hard disk, or to set up the workstations to run Microsoft Access from the file server. If your network supports the universal naming convention (UNC), it is best to start the workstation Setup using the UNC path. Doing this eliminates possible future problems with incorrect network drive letter references. In addition, you can create custom scripts to set up workstations "silently," or without input from the users.

For detailed network installation instructions, see the `ACREADME.HLP` file that is shipped with Microsoft Access.

5. Q. My validation rules worked correctly in Microsoft Access version 1.x, but when I convert my version 1.x database to version 2.0, I receive an error message and a Convert Error table is created. Why is this?
- A. In Microsoft Access 2.0, validation rules are enforced whenever you add or edit data. Because validation rules are always enforced, the following items are not valid in default values or field-level validation rules:
- References to fields or controls.
  - User-defined functions.
  - Microsoft Access domain functions.
  - Aggregate functions.
  - `CurrentUser()` or `Eval()` functions.

For additional information about validation rules, you can order item number Q<srx940128000575> by selecting the FastTips Technical Library option from the FastTips Main Menu.

6. Q. Are there any changes that I should be aware of when I convert my database from version 1.x to version 2.0?
- A. Microsoft Access 2.0 contains many changes and new features. For specific information about these items, refer to:
- Microsoft Access "User's Guide", version 2.0, "What's New in Version 2.0" section
  - Microsoft Access "User's Guide", version 2.0, Chapter 1, "Setting Up Microsoft Access"

- Microsoft Access "Building Applications", version 2.0, Appendix C

- The ACREADME.HLP file that is shipped with Microsoft Access

7. Q. Why am I unable to create new objects when I open my Microsoft Access version 1.x database in Microsoft Access 2.0?

A. When you open a Microsoft Access version 1.x database in Microsoft Access 2.0, you can add or modify data, but you cannot create new objects or modify the design of existing objects.

To create new objects or modify the design of existing objects, you must first convert your version 1.x database to version 2.0. To do this, make sure that there are no databases open in Microsoft Access, then choose Convert from the File menu. You will be prompted for the name of the database to convert and a new name for the converted database. Note that if you convert your database to version 2.0, it cannot be opened again in version 1.x.

For additional information about database conversion, see the Microsoft Access "User's Guide," version 2.0, Chapter 1, "Setting Up Microsoft Access."

8. Q. Can Microsoft Access version 1.x read a version 2.0 database?

A. Microsoft Access 1.x cannot read a version 2.0 database. Also, you cannot convert a Microsoft Access 2.0 database to version 1.x format. You can export your version 2.0 tables to a 1.x database, but because of all the new features in version 2.0, you cannot convert any other objects. Therefore, be sure to keep 1.x versions of your databases until you have upgraded all of your users to version 2.0.

For more information about using version 1.x databases in version 2.0, see the Microsoft Access "User's Guide", version 2.0, Chapter 1, "Setting Up Microsoft Access."

9. Q. How much hard disk space is required to install Microsoft Access 2.0?

A. The Microsoft Access Setup program has three installation options, each having different disk space requirements. The space needed for each option is listed below:

- Typical. For a Typical installation, you need approximately 19 MB of disk space (15 MB in the Microsoft Access directory, plus 4 MB in the Windows directory).

NOTE: The Typical installation installs all of the Microsoft Access components except for Open Database Connectivity (ODBC). To install ODBC, you must choose the Complete/Custom option.

- Complete/Custom. For a Complete installation, you need approximately 20 MB of disk space (15 MB in the Microsoft Access directory, plus 5 MB in the Windows directory).
- Laptop/Minimum. For a Laptop/Minimum installation, you need approximately 5 MB of disk space (4.5 MB in the Microsoft Access directory, plus 0.5 MB in the Windows directory).

-----  
TO OBTAIN THIS APPLICATION NOTE  
=====

- You can have this Application Note mailed or faxed to you from the automated Microsoft FastTips Technical Library, which you can call 24 hours a day, 7 days a week at (800) 936-4100. NOTE: The FastTips Technical Library is available only to customers within the U.S. and Canada.
- If you are unable to access the source(s) listed above, you can have this Application Note mailed or faxed to you by calling Microsoft Product Support Services Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time at (206) 635-7050. If you are outside the United States, contact the Microsoft subsidiary for your area. To locate your subsidiary, call Microsoft International Customer Service at (206) 936-8661.

Additional reference words: 2.00  
KBCategory: kbusage kbfasttip kbappnote  
KBSubcategory: Fstqa

## **WX0997: Interoperability Questions and Answers**

**Article ID: Q114812**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

The "Interoperability Questions and Answers" (WX0997) Application Note contains questions and answers about Microsoft Access version 2.0 interoperability.

You can obtain this Application Note from the following sources:

- Microsoft FastTips Technical Library
- Microsoft Product Support Services

For complete information, see the "To Obtain This Application Note" section at the end of this article.

THE TEXT OF WX0997  
=====

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0997: INTEROPERABILITY QUESTIONS AND ANSWERS  
=====

Revision Date: 3/94  
No Disk Included

The following information applies to Microsoft Access(R) version 2.0.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |  
| PURPOSE. The user assumes the entire risk as to the accuracy and |  
| the use of this Application Note. This Application Note may be |  
| copied and distributed subject to the following conditions: 1) All |  
| text must be copied without modification and all pages must be |  
| included; 2) If software is included, all files on the disk(s) |  
| must be copied without modification (the MS-DOS(R) utility |  
| diskcopy is appropriate for this purpose); 3) All components of |  
| this Application Note must be distributed together; and 4) This |  
| Application Note may not be distributed for profit. |  
| |  
| Copyright (C) 1994 Microsoft Corporation. All Rights Reserved. |  
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |  
| and Windows is a trademark of Microsoft Corporation. |  
|-----

1. Q. How do I send Microsoft Access database objects to other software applications?



- A. There are three commands on the File menu and two toolbar buttons that send database objects to other applications.
- The Export command exports data only from a table or query. To export the layout, use the Output To command.
  - The Print Definition command creates a report about an object's structure.
  - The Output To command exports data to a file, preserving the layout. You enter the output format and name of the file to create. To view the file, start the appropriate application.
  - The Publish It With MS Word and Analyze It With MS Excel buttons save the file to the current directory and start Microsoft Excel or Word. For information about adding these buttons to your toolbar, search for "customizing toolbars" then "Adding or Removing Buttons on Toolbars" using the Microsoft Access Help menu.

2. Q. How can Visual Basic access Microsoft Access version 2.0 databases?

- A. A mapping layer has been created so that Visual Basic version 3.0 can communicate with Microsoft Access 2.0 databases. This mapping layer works with both Microsoft Access version 1.1 and 2.0 databases. The mapping layer will be available in the Microsoft Access Developer's Toolkit (ADT), and in the Microsoft Office Developer's Kit (ODK).

For additional information about using Microsoft Access 2.0 databases with Visual Basic, you can order item number Q112104 by selecting the FastTips Technical Library option from the FastTips Main Menu.

3. Q. Has the ability to use my Microsoft Access data in a Microsoft Word for Windows mail merge changed in version 2.0?

- A. Using Microsoft Access data for a Word for Windows mail merge is easier in version 2.0. The new Microsoft Word Mail Merge Wizard helps you create a dynamic data exchange (DDE) link between Microsoft Access and Microsoft Word version 6.0 for Windows. To use the Microsoft Word Mail Merge Wizard, use either of the following methods:
- Select your table or query in the Database window, and then choose the Merge It button on the toolbar.
  - Create a new report based on your table or query. Choose the ReportWizards button, then choose Microsoft Word Mail Merge in the ReportWizards dialog box.

For more information about using Microsoft Access data in a Word for Windows merge, search for "Word for Windows" then "Merging Microsoft Access Data with a Word for Windows Document" using the Microsoft Access Help menu.

4. Q. Do I need to change anything so that Microsoft Word version 6.0 for Windows or Microsoft Excel version 5.0 can use my Microsoft Access 2.0 databases?
- A. To use Microsoft Access version 2.0 databases with Word 6.0 for Windows and Microsoft Excel 5.0, you need the Microsoft Access 2.0 ODBC driver for Microsoft Office. For additional information about this driver, you can order item number Q112062 by selecting the FastTips Technical Library option from the FastTips Main Menu.

5. Q. How can I obtain the Oracle ODBC driver?

- A. Microsoft Access 2.0 supplies an ODBC driver for connecting to SQL Server, but does not supply a driver for connecting to Oracle 6. Customer research showed that few people used the Oracle driver we shipped with Microsoft Access version 1.1. However, if you have the Microsoft Access 1.1 Oracle driver, you can use it with your Microsoft Access 2.0 applications.

For additional information about obtaining the Oracle ODBC driver, you can order item number Q112105 by selecting the FastTips Technical Library option from the FastTips Main Menu.

6. Q. How can I execute a stored procedure using an SQL pass-through query?

- A. You can use SQL pass-through queries to send commands directly to an ODBC database server (such as Microsoft SQL Server). You must send the commands using the syntax required by the particular server. With pass-through queries, you work directly with the tables on the remote server instead of attaching them.

For an example of running a SQL Server stored procedure, order the fax or mail copy of this script.

For example, to run the stored procedure `sp_configure`, which returns SQL Server configuration information, use the following steps:

1. Create a new query.
2. From the Query menu, choose SQL Specific, then choose Pass-Through.
3. In the SQL Pass-Through Query window, type the following:

```
sp_configure
```

When you run the query, you will be prompted for the ODBC data source. The configuration information for the SQL Server will be returned.

For more information about the syntax of SQL pass-through queries, please consult the documentation for your server.

7. Q. When I attach a SQL Server table, I cannot edit any of the

data, although I have rights to edit the data. Why is this and how can I edit the data?

- A. When you attach any table with ODBC, the table must have a unique index before Microsoft Access can update the data. In Microsoft Access 2.0, you can create a temporary unique primary index on the attached table using a data definition query (DDL). This temporary index will not be created on the server, but will allow you to update the data in Microsoft Access. The following is a DDL query to create a temporary primary index:

```
CREATE UNIQUE INDEX index1 On <tablename>(<uniqueIDfieldname>)
```

For more information about DDL queries, search for "Data Definition" then "Data Definition Command" using the Microsoft Access Help menu.

8. Q. How do I link a Microsoft Access table to a Microsoft Excel spreadsheet using dynamic data exchange (DDE)?

- A. The easiest way is to use DDE to create a paste link between Microsoft Access and Microsoft Excel. To do this, use the following steps:

1. In the Microsoft Access Database window, select the table you want to link to Microsoft Excel.
2. From the Microsoft Access Edit menu, choose Copy.
3. From the Microsoft Excel Edit menu, choose Paste Special.
4. Select the Paste Link option button, then select either Text (unformatted) or CSV (formatted). Choose OK.

The Microsoft Excel formula resulting from the Paste Link command will be similar to:

```
{=MSAccess|'<DATABASENAME>.MDB;Table <Tablename>'!All}
```

For more information about linking Microsoft Access data to data in other applications, see the Microsoft Access "User's Guide," version 2.0, Chapter 19, "Using Pictures, Graphs, and Other Objects."

9. Q. Has dynamic data exchange (DDE) changed in Microsoft Access version 2.0?

- A. DDE functionality has not changed from Microsoft Access version 1.1 to 2.0. No new DDE commands were added.

For additional DDE information and examples, you can order item number Q89586 by selecting the FastTips Technical Library option from the FastTips Main Menu.

For more information about DDE, search for "DDE server" then view each of the topics using the Microsoft Access Help menu.

-----  
TO OBTAIN THIS APPLICATION NOTE  
=====

- You can have this Application Note mailed or faxed to you from the automated Microsoft FastTips Technical Library, which you can call 24 hours a day, 7 days a week at (800) 936-4100. NOTE: The FastTips Technical Library is available only to customers within the U.S. and Canada.
- If you are unable to access the source(s) listed above, you can have this Application Note mailed or faxed to you by calling Microsoft Product Support Services Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time at (206) 635-7050. If you are outside the United States, contact the Microsoft subsidiary for your area. To locate your subsidiary, call Microsoft International Customer Service at (206) 936-8661.

Additional reference words: 2.00  
KBCategory: kbusage kbfasttip kbappnote  
KBSubcategory: Fstqa

## WX0998: Macro and Module Questions and Answers

Article ID: Q114813

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

The "Macro and Module Questions and Answers" (WX0998) Application Note contains questions and answers about Microsoft Access version 2.0 macros and modules.

You can obtain this Application Note from the following sources:

- Microsoft FastTips Technical Library
- Microsoft Product Support Services

For complete information, see the "To Obtain This Application Note" section at the end of this article.

THE TEXT OF WX0998

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0998: MACRO AND MODULE QUESTIONS AND ANSWERS  
=====

Revision Date: 3/94  
No Disk Included

The following information applies to Microsoft Access(R) version 2.0.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |  
| PURPOSE. The user assumes the entire risk as to the accuracy and |  
| the use of this Application Note. This Application Note may be |  
| copied and distributed subject to the following conditions: 1) All |  
| text must be copied without modification and all pages must be |  
| included; 2) If software is included, all files on the disk(s) |  
| must be copied without modification (the MS-DOS(R) utility |  
| diskcopy is appropriate for this purpose); 3) All components of |  
| this Application Note must be distributed together; and 4) This |  
| Application Note may not be distributed for profit. |  
| |  
| Copyright (C) 1994 Microsoft Corporation. All Rights Reserved. |  
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |  
| and Windows is a trademark of Microsoft Corporation. |  
|-----

1. Q. What is DAO?

A. Data access objects (DAO) and collections provide a framework

for using code to create and manipulate components of your database system. Objects and collections have properties that describe the characteristics of database components and methods. DAO provides the functionality of the dynasets, Snapshots, and QueryDef objects you used in version 1.x, while giving you more structure and features. All collections use the same syntax.

For more information about DAO, please refer to the following:

- Search for "data access objects" then "Data Access Objects and Collections" using the Microsoft Access Help menu.
- Search for "converting databases to version 2.0" then "Converting Macros and Code from Version 1.x to 2.0" using the Microsoft Access Help menu.
- See Microsoft Access "Building Applications," Chapter 7, "Objects and Collections."
- See Microsoft Access "Building Applications," Chapter 11, "Working with Sets of Records."

2. Q. How can I use data access objects (DAO)?

A. You can use DAO methods to create or modify different parts of your database, including TableDef objects, security, relationships, and so on. Three common tasks that you can accomplish with DAO are:

- Indexing an existing field in a table.
- Assigning permissions to a user.
- Adding a user to a group.

For additional information about indexing an existing field in a table, you can order item number Q112107 by selecting the FastTips Technical Library option from the FastTips Main Menu.

For additional information about assigning permissions to a user, you can order item number Q112106 by selecting the FastTips Technical Library option from the FastTips Main Menu.

For additional information about adding a user to a group, you can order item number Q112063 by selecting the FastTips Technical Library option from the FastTips Main Menu.

3. Q. How do I create an SQL pass-through query using Access Basic?

A. Use the CreateQueryDef method to create an SQL pass-through query using Access Basic. Set the properties of your QueryDef as follows:

- ConnectString: This sets the ODBC connection string, and must be at least "ODBC;". If the connection string does not

include at least "ODBC;" the query is not an SQL pass-through query and you will receive a syntax error message. If you do not include the data source name, you will be prompted for it when you run the query. For more information about connection strings, search for "ODBC connection string" then "ODBCConnectionString Property" using the Microsoft Access Help menu.

- SQL statement: This is the SQL statement that is passed to the server. For additional information about the syntax of the SQL statement, please refer to your server's documentation.

For additional information and an example of an SQL pass-through query, you can order item number Q112108 by selecting the FastTips Technical Library option from the FastTips Main Menu.

4. Q. How do I determine if the record being edited is a new record?

- A. In Microsoft Access version 1.x, the counter field is null until the record is saved. So, to check for a new record, you would check to see if the counter field is null. In Microsoft Access version 2.0, the counter field is updated as soon as you begin inserting a new record. However, the OldValue property for the counter will still be null. You can use the following expression to determine whether the record being edited is a new record:

```
IsNull([<counterfieldname>].OldValue)
```

For additional information about checking for new records in Microsoft Access 2.0, you can order item number Q112109 by selecting the FastTips Technical Library option from the FastTips Main Menu.

5. Q. Why doesn't my version 1.x Access Basic code work correctly in Microsoft Access 2.0?

- A. Microsoft Access 2.0 introduces many changes and new features to Access Basic. When you convert your database from version 1.x to 2.0, your Access Basic code is not automatically converted. Because of the changes and new features, you must modify your code for it to work correctly in version 2.0. Some of the changed items include the SendKeys and DoMenuItem actions, and field name references. For more details about these changes, order the fax or mail copy of this script.

- The SendKeys action: Some menus in Microsoft Access 2.0 have changed. One of the common uses for the SendKeys action, hiding and showing the toolbar, has been replaced with the new ShowToolbar action. To access the options in the Options dialog box (such as Show Status Bar), use the new SetOption and GetOption methods with the Application object. For more information about the SetOption method, search for "SetOption" then "GetOption, SetOption Methods" using the

Microsoft Access Help menu.

- The DoMenuItem action: Because some menus have changed, there is now a fifth argument, <Version>, that you must supply when you use the DoMenuItem action in Access Basic. For more information about the DoMenuItem action, search for "DoMenuItem" then "DoMenuItem Action" using the Microsoft Access Help menu. You can also order item number Q112065 by selecting the FastTips Technical Library option from the FastTips Main Menu.
  
- Using a period to reference field names: In Microsoft Access 1.x, you can reference fields using a period before the field name. For example, you can use "MyTable.Name" to refer to the Name field in the MyTable table. If you use the same expression in Microsoft Access 2.0, you reference the Name property instead of the field called Name. Use "MyTable!Name" to refer to the Name field in Microsoft Access 2.0.

For more information about other changes and new features in Access Basic, search for "converting databases to version 2.0" then "Converting Macros and Code from Version 1.x to 2.0" using the Microsoft Access Help menu.

6. Q. Why doesn't a new table created using data access objects (DAO) show in the Database window?

A. Database changes made using DAO are not automatically synchronized with the Database window in order to avoid affecting system performance. To view changes made using DAO, refresh the Database window by choosing a different object button, and then choose the object button for the object type you were working with. You can automate this refresh method in your code by using two SelectObject actions.

When you make a change without using DAO, it is not reflected in the corresponding DAO object unless you call the Refresh method on the collection containing that object. For example, if you delete a table in the Database window, use the following expression to remove the table from the TableDefs collection:

```
<MyDatabase>.TableDefs.Refresh
```

7. Q. How do I include a variable in the WHERE clause of my SQL statement?

A. The syntax for including a variable in the WHERE clause of an SQL statement depends on the variable's data type. Numeric variables do not require delimiters, string variables should be enclosed in single quotation marks, and date variables should be enclosed in number signs (#). Concatenate the variable and the appropriate delimiter, if required, as shown in the three examples available in the fax or mail copy of this script.

For a numeric variable, use the following syntax:



```
<myq>.sql = "select * from <table> where
[<field>]= " & <mynum> & ";"
```

For a string variable, use the following syntax:

```
<myq>.sql = "select * from <table> where
[<field>]=' ' & <mytext> & "';"
```

For a date variable, use the following syntax:

```
<myq>.sql = "select * from <table> where
[<field>]=#" & <mydate> & "#;"
```

For additional information about concatenating variables, you can order item number Q96576 by selecting the FastTips Technical Library option from the FastTips Main Menu.

8. Q. Will the macros I created in Microsoft Access version 1.x work in version 2.0?
- A. Macro syntax has not changed from version 1.x to 2.0, so macros created in version 1.x should work correctly. However, if your macros use the SendKeys action, you may need to make some changes to reflect the menu changes and new features in Microsoft Access 2.0.
9. Q. Can I call a function stored in my form module from a regular module?
- A. Functions stored in form or report modules are private to that module's form or report, and can only be called from that module's form or report. If you want to use the function in a different form, report, or global function, make it a global function by storing it in a regular module.

For more information about writing and using event procedures, see Microsoft Access "Building Applications," Chapter 3, "Introducing Access Basic," and Chapter 5, "Access Basic Fundamentals."

10. Q. How do I edit a library? Do I have to unload the library first as in Microsoft Access version 1.x?
- A. Debugging libraries in Microsoft Access version 1.x requires you to unload the library and restart Microsoft Access each time you find an error in the library. In Microsoft Access 2.0, you can debug libraries without unloading the library and restarting Microsoft Access.

To debug or edit a library in Microsoft Access 2.0, add the following line to the [Options] section of the MSACC20.INI file:

```
DebugLibraries=True
```

Once you have added this line to the MSACC20.INI file and

restarted Microsoft Access, you can edit the functions in your library. If this line is not in your MSACC20.INI file, or is set to False, you will not be able to edit your library.

-----  
TO OBTAIN THIS APPLICATION NOTE  
=====

- You can have this Application Note mailed or faxed to you from the automated Microsoft FastTips Technical Library, which you can call 24 hours a day, 7 days a week at (800) 936-4100. NOTE: The FastTips Technical Library is available only to customers within the U.S. and Canada.
- If you are unable to access the source(s) listed above, you can have this Application Note mailed or faxed to you by calling Microsoft Product Support Services Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time at (206) 635-7050. If you are outside the United States, contact the Microsoft subsidiary for your area. To locate your subsidiary, call Microsoft International Customer Service at (206) 936-8661.

Additional reference words: 2.00  
KBCategory: kbusage kbfasttip kbappnote  
KBSubcategory: Fstqa

## WX0999: MS Access Dev. Toolkit Questions & Answers

Article ID: Q115622

-----  
The information in this article applies to:

- Microsoft Access Developer's Toolkit version 2.0
- 

The "Microsoft Access Developer's Toolkit Questions and Answers" (WX0999) Application Note contains questions and answers about the Microsoft Access Developer's Toolkit version 2.0.

You can obtain this Application Note from the following sources:

- Microsoft FastTips Technical Library
- Microsoft Product Support Services

For complete information, see the "To Obtain This Application Note" section at the end of this article.

THE TEXT OF WX0999

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX0999: MICROSOFT ACCESS(R) DEVELOPER'S TOOLKIT QUESTIONS AND ANSWERS  
=====

Revision Date: 5/94  
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 (C) 1994 Microsoft Corporation. All Rights Reserved. |  
| Microsoft, Microsoft Access, and MS-DOS are registered trademarks |  
| and Windows is a trademark of Microsoft Corporation. |  
-----

1. Q. Can I include the OLE custom controls that come with the Microsoft Access Developer's Toolkit (ADT) with my application?
  - A. You can include the three OLE custom controls that come with the ADT (the Calendar, Scroll Bar, and Data Outline controls)

with your application, royalty-free.

2. Q. How can I create my own OLE custom controls?

A. You can create your own OLE custom controls using the Custom Controls Development Toolkit add-on to Microsoft Visual C++ that is scheduled for release in summer 1994.

3. Q. What is the best way to distribute updates of my application?

A. To make updating your application easy, begin by creating your application as two databases. One database (DATA.MDB) is for storing the application's data, and the other database (PROGRAM.MDB) is for storing the application's queries, forms, reports, macros, and modules. To distribute updates of your application, simply distribute a new PROGRAM.MDB database. You can use this method to update the application without disturbing your users' data.

All the objects in the PROGRAM.MDB database should be based on attached tables from the DATA.MDB database. Note that when you create a database with attached tables, the path to the attached tables is stored in the database. If the path to the attached tables becomes invalid, an error message will occur when the database is opened. For information about verifying attachments, open the Solutions sample application (SOLUTION.MDB). In the Select A Category Of Examples box, select Use Multiple Databases, then in the Select An Example box, select Attach Tables At Startup.

4. Q. How can I prevent users from seeing the code in my application?

A. You can control what users of your application see by taking the following precautions:

- Use form and report modules.
- Use forms and macros to control users' movements within your application.
- Secure all the objects in your database.
- Encrypt your database.

5. Q. How do I distribute the ODBC SQL Server driver with my application?

A. To enable your run-time application to connect to a SQL Server database using ODBC, select ODBC With SQL Server Driver in the Available Options box in the Setup Wizard.

6. Q. How do I include my own SYSTEM.MDA file with my application?

A. To include your own SYSTEM.MDA file, add your SYSTEM.MDA file in the Files To Include box in the Setup Wizard. Setup will not overwrite a file named SYSTEM.MDA.

NOTE: If you do not provide a SYSTEM.MDA file with your run-time application, Setup will create a new one.

7. Q. Why can't I access my Calendar control's Font and Color properties?
- A. The OC1016.DLL file was not automatically registered in the Microsoft Windows REG.DAT file when you installed the ADT. The OC1016.DLL file must be registered in the REG.DAT file in order for you to access the Calendar control's Font and Color properties.

For information about how to register the OC1016.DLL file in the REG.DAT file, you can order item number Q115125 by selecting the FastTips Technical Library option from the FastTips Main Menu.

8. Q. Can I add my own bulletin boards to Setup? Can I change the Setup bitmap to something else?
- A. You cannot currently add your own bulletin boards to, or modify the bitmap for, custom Setup programs. If you want this level of control in creating your custom Setup program, you may want to consider using a third-party tool. Note, however, that third-party tools cannot be used to modify an existing custom Setup program developed with the ADT.
9. Q. Can I distribute the Microsoft Access Wizards with my application?
- A. You cannot distribute the Microsoft Access Wizards. However, you can create and distribute your own Wizards. The ADT includes commented Wizard code as a learning aid for creating your own Wizards.
10. Q. Can I use custom toolbars in my run-time application? If so, how?
- A. You can use custom toolbars in your run-time application. To do so, use the ShowToolbar macro action in your application to display the custom toolbars. Note that by default, no toolbars (built-in or custom) are displayed in run-time applications. You must use the ShowToolbar action to display any toolbar in your application.

For information about how to create custom toolbars, search for "custom toolbars" then "Creating Custom Toolbars" using the Microsoft Access Help menu.

-----  
TO OBTAIN THIS APPLICATION NOTE

=====

- You can have this Application Note mailed or faxed to you from the automated Microsoft FastTips Technical Library, which you can call 24 hours a day, 7 days a week at (800) 936-4100. NOTE: The FastTips Technical Library is available only to customers within the U.S. and Canada.

- If you are unable to access the source(s) listed above, you can have this Application Note mailed or faxed to you by calling Microsoft Product Support Services Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time at (206) 635-7050. If you are outside the United States, contact the Microsoft subsidiary for your area. To locate your subsidiary, call Microsoft International Customer Service at (206) 936-8661.

Additional reference words: 2.00

KBCategory: kbusage kbfasttip kbappnote

KBSubcategory: Fstqa

**WX1051: Security Wizard and White Paper App. Note 2.0**  
**Article ID: Q122036**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

The "Security Wizard and White Paper for Version 2.0" (WX1051) Application Note contains in-depth information about the Microsoft Access security model to help you understand and use Microsoft Access security effectively.

You can obtain this Application Note from the following sources:

- CompuServe, GENie, and Microsoft OnLine
- Microsoft Download Service (MSDL)
- The Internet (Microsoft anonymous ftp server)
- Microsoft Product Support Services

For complete information, see the "To Obtain This Application Note" section at the end of this article.

THE TEXT OF WX1051

=====

=====  
Microsoft(R) Product Support Services Application Note (Text File)  
WX1051: SECURITY WIZARD AND WHITE PAPER FOR VERSION 2.0  
=====

Revision Date: 10/94  
1 Disk Included

The following information applies to Microsoft Access, version 2.0.

-----  
| INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY |  
| ACCOMPANY THIS DOCUMENT (collectively referred to as an Application |  
| Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER |  
| EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED |  
| WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR |  
| PURPOSE. The user assumes the entire risk as to the accuracy and |  
| the use of this Application Note. This Application Note may be |  
| copied and distributed subject to the following conditions: 1) All |  
| text must be copied without modification and all pages must be |  
| included; 2) If software is included, all files on the disk(s) |  
| must be copied without modification (the MS-DOS(R) utility diskcopy |  
| is appropriate for this purpose); 3) All components of this |  
| Application Note must be distributed together; and 4) This |  
| Application Note may not be distributed for profit. |  
| |  
| Copyright (C) 1994 Microsoft Corporation. All Rights Reserved. |  
| |  
| Microsoft and MS-DOS are registered trademarks and Windows is a |  
| trademark of Microsoft Corporation. |  
-----

The disk included with this document contains two files. The SECURE20.TXT file is a white paper that contains in-depth information about the Microsoft Access security model to help you understand and use Microsoft Access security effectively. The SECURE20.MDA file contains the Microsoft Access Security Wizard, which is a tool for easily securing your databases. The white paper includes complete instructions for using the Security Wizard.

#### To install the Security Wizard

---

1. Insert the WX1051 disk in the appropriate disk drive.
2. Start Microsoft Access and open any database.
3. From the File menu, choose Add-ins, and then choose Add-in Manager from the submenu.
4. In the Add-in Manager dialog box, choose the Add New button.
5. In the Drives box, select the drive containing the Application Note disk. In the File Name box, select SECURE20.MDA, then choose the OK button.
6. When the Security Wizard appears in the Available Libraries box, choose the Close button.
7. Quit and then restart Microsoft Access.

The Security Wizard runs only in Microsoft Access version 2.0. It works with Microsoft Access version 1.x databases provided that they are first converted to Microsoft Access version 2.0 database format. If you try to use the Security Wizard in Microsoft Access version 1.x, you will receive an "Out-of-date database format" error message, followed by the message "Some library modules could not be loaded" when you start Microsoft Access. Databases created by the Security Wizard are in version 2.0 format.

The Security Wizard may take some time to run. For example, securing the sample database NWIND.MDB on a 486/50 computer takes approximately 13 minutes. The Security Wizard's About dialog box gives further information about what the Security Wizard does, as does the text file included on the WX1051 disk.

#### The Security White Paper

---

The security white paper is a text document written by the Microsoft Access product group. It contains in-depth information on the security model used by Microsoft Access, details on how and why Microsoft Access security acts the way it does, and answers to some common security questions. The file is in text (.TXT) format and is readable by any word processor.

---

TO OBTAIN THIS APPLICATION NOTE



- =====
- On CompuServe, GENie, and Microsoft OnLine, Application Notes are located in the Microsoft Software Library. You can find WX1051 in the Software Library by searching on the word WX1051, the Q number of this article, or S14978. WX1051 is a compressed, self-extracting file. After you download WX1051, run it to extract the file(s) it contains.
  - Application Notes are available by modem from the Microsoft Download Service (MSDL), which you can reach by calling (206) 936-6735. This service is available 24 hours a day, 7 days a week. The highest download speed available is 14,400 bits per second (bps). For more information about using the MSDL, call (800) 936-4100 and follow the prompts. To obtain WX1051, download WX1051.EXE. WX1051.EXE is a compressed, self-extracting file. After you download WX1051, run it to extract the file(s) it contains.
  - On the Internet, Application Notes are located on the Microsoft anonymous ftp server, which you can reach by typing "ftp ftp.microsoft.com" (without the quotation marks) at the ">" command prompt.
  - If you are unable to access the sources listed above, you can have this Application Note mailed or faxed to you by calling Microsoft Product Support Services Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time at (206) 635-7050. If you are outside the United States, contact the Microsoft subsidiary for your area. To locate your subsidiary, call Microsoft International Customer Service at (206) 936-8661.

Additional reference words: 2.00 appnote  
KBCategory: kbusage  
KBSubcategory: ScrtOthr

**WX1124: Microsoft Access Version 2.0 Service Pack**  
**Article ID: Q122927**

-----  
The information in this article applies to:

- Microsoft Access version 2.0
- 

SUMMARY

=====

The Microsoft Access version 2.0 Service Pack contains files that replace existing files in your current Microsoft Access version 2.0 installation. The Service Pack includes the latest versions of:

- The Microsoft Jet database engine (version 2.5).
- The OLE dynamic-link libraries (version 2.02).
- The Btrieve, Paradox, and xBASE installable ISAM drivers.

The Microsoft Jet database engine has been modified to enhance database repair functionality and to correct the erroneous setting of an internal database flag that in some cases results in the message "Couldn't open SYSTEM.MDA." The OLE dynamic-link libraries (DLLs) have been modified to allow the use of OLE custom controls created by third-party vendors and also to correct problems with diminished system resources when printing reports containing embedded Microsoft Graph objects. The installable ISAM drivers have been modified to correct various minor problems.

You can obtain the Service Pack by downloading the files from one of the services listed below. Or, you can obtain the Service Pack by calling the Microsoft FastTips system and having disks mailed to you.

MORE INFORMATION

=====

Download Services

-----

The Service Pack is available as a self-extracting file called ACCSVC.EXE in the Microsoft Software Library (MSL) on the following services:

- CompuServe
  - GO MSL
  - Search for ACCSVC.EXE
  - Display results and download
- Microsoft Download Service (MSDL)
  - Dial (206) 936-6735 to connect to MSDL
  - Download ACCSVC.EXE
- Internet (anonymous FTP)
  - ftp ftp.microsoft.com
  - Change to the softlib/mslfiles directory

Get ACCSVC.EXE

Microsoft FastTips

-----

FastTips is an interactive telephone system that you can call 24 hours a day, 7 days a week for information on many Microsoft products. You can order the Service Pack to be sent by mail for no charge by dialing the FastTips line at 1-800-936-4100 and following these steps:

1. Press 4 for Microsoft Access.
2. Press 1 for version 2.x.
3. Press 1 for Express Order Service.
4. When you are prompted, press 2 for a mail delivery method.
5. At the tone, enter the number 1124 for the Item ID.
6. Press # to verify the order is correct.
7. When you are prompted, leave your name and mailing address in the voice mailbox.
8. When you are prompted, press \* to finish your order.

Additional reference words: 2.00 appnote jet25 jet 2.5

KBCategory: kbusage kbfile

KBSubcategory: Softlib

**More Information About Knowledge Base (KB) Help**  
**Article ID: Q199990**

The KB Help file is a snapshot of the Knowledge Base written by Product Support Engineers to help support Microsoft customers at the time this file was created.

More recent KB articles can be found on Internet, CompuServe, and GENIE. After you log on to one of these services with your account information, follow these steps to find the KB:

- On Internet, look in DESKAPPS\ACCESS\KB\README.TXT
- On CompuServe, type "GO MSKB"
- On GENie, type "M 505"

OTHER SOURCES OF MICROSOFT SUPPORT INFORMATION

Microsoft offers Electronic Information Services as an alternative to person-to-person support. All U.S. and Canadian Microsoft customers have access 24 hours a day, 7 days a week to a range of free or low-cost service options including:

Microsoft FastTips

Toll-free, automated information about key Microsoft products. Customers have access to technical articles (including common questions and answers) via voice, fax, and U.S. Mail. The four categories are:

|                                         |                |
|-----------------------------------------|----------------|
| FastTips for Desktop Applications       | (800)-936-4100 |
| FastTips for Personal Operating Systems | (800)-936-4200 |
| FastTips for Development Products       | (800)-936-4300 |
| FastTips for Advanced Systems           | (800)-936-4400 |

Microsoft Knowledge Base

This is the primary Microsoft product information source--a database of more than 25,000 detailed articles containing technical information, bug lists, fix lists, and more. Access via CompuServe, the Internet, and GENie.

Microsoft Download Service

Direct access to Microsoft Download Library (MSDL) by modem through the Microsoft BBS service at (206) 936-6735. Connect information: 1200, 2400, or 9600 baud, no parity, 8 data bits, and 1 stop.

Microsoft Forums on CompuServe

Microsoft facilitates a variety of interactive dialog forums for its worldwide community of customers. Access by typing: GO MICROSOFT. CompuServe connect charges apply. To establish

an information service account, contact CompuServe directly at (800) 848-8199.

#### Microsoft Information Network

##### TechNet

Part of the support professionals program, the TechNet CD contains resource kits, customer solution profiles, case studies, and tips and techniques for integration. For TechNet subscriptions, call (800) 344-2121 ext. 035.

##### Microsoft Developers' Network (MSDN)

Level 1 CD is a Development Library of comprehensive development information for Win16 and Win32 APIs. Level 2 CD contains all software needed to develop Windows-based applications. For Microsoft Developers Network subscriptions, call (800) 759-5474.

##### The Internet

Access the Microsoft Software Library and the Microsoft Knowledge Base by anonymous ftp. The Microsoft Internet FTP archive host supports anonymous login. The name of the server is FTP.MICROSOFT.COM; the IP address is 198.105.232.1.

## Microsoft Knowledge Base Help File Survey

Article ID: Q199991

### Microsoft Knowledge Base Survey

To better meet your needs, please give us your feedback on the Microsoft Knowledge Base (KB) Help file. We appreciate your input.

There are several Help files available; which files have you downloaded?

- |                                                                          |                                               |
|--------------------------------------------------------------------------|-----------------------------------------------|
| <input type="checkbox"/> Access 2.x or Access 1.x                        | <input type="checkbox"/> Word 6.0 articles    |
| <input type="checkbox"/> Excel 5.0 Most Useful articles                  | <input type="checkbox"/> Windows 3.1 articles |
| <input type="checkbox"/> Visual Basic for Applications                   | <input type="checkbox"/> Windows Printing     |
| <input type="checkbox"/> Project 4.0 articles                            | <input type="checkbox"/> Publisher            |
| <input type="checkbox"/> Windows for Workgroups articles                 | <input type="checkbox"/> Office               |
| <input type="checkbox"/> Windows for Workgroups 100 Most Useful articles |                                               |
| <input type="checkbox"/> Multimedia Helpfile                             |                                               |

How did you receive this KB Help file?

- ftp.microsoft.com (Internet), softlib directory
- CompuServe user forums or MSL
- MSDL
- FastTips
- Mailed from PSS
- From a local network or BBS, coworker, or friend

Did you find a resolution to your problem in the KB Help file?

Yes / No

How do you prefer to receive your KB information?

- A KB Help file with the most useful articles
- A KB Help file with all articles (even if the file is several MB)
- Several KB Help files, each with articles from a major category
- Individual articles, such as those available in the KB on Internet and CompuServe
- Other \_\_\_\_\_

Where do you normally get your technical information?

Please rank each source from 1-8:

- KB Help file
- KB on ftp.microsoft.com or CompuServe
- MSDN or TechNet CDs
- CompuServe Technical Support Forums
- Calling Microsoft FastTips 800 number
- Calling Microsoft Product Support (PSS)
- Calling PSS and receiving technical items by fax/mail/both
- Other \_\_\_\_\_

Do you have a modem? Yes / No

May we follow up this survey with a phone call or E-mail if we need additional information? Yes / No

Name \_\_\_\_\_

Address, E-mail name, or Internet address \_\_\_\_\_

Phone \_\_\_\_\_

Occupation \_\_\_\_\_

Type of business/industry \_\_\_\_\_  
Microsoft products you use \_\_\_\_\_

To submit this survey, please print the Help topic and fax your responses to Microsoft KB Survey, (206) 704-4412.

NOTE: We are unable to respond to technical support questions at this phone number.

