Microsoft Access 2.x Frequently Asked Quetions
Prepared 02/21/95

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 Frequently Asked Quetions

Microsoft Access 2.x Frequently Asked Quetions

Microsoft Access 2.x Frequently Asked Quetions

Categories and Keywords for All Knowledge Base Articles

Microsoft Access KB Categories and Subcategories

Knowledge Base Articles

- INF: How to Create a Multiuser Custom Counter
- INF: ODBC Setup for Microsoft Access and SQL Server
- INF: Using Date/Time Data Type and Calculating Elapsed Time
- INF: Fill New Record with Data from Prev. Record Automatically
- INF: Topics Supported by Microsoft Access as a DDE Server
- PRB: "Can't Bind Name '[XXX]'" Error with Crosstab Query
- INF: Using Column Property of Combo Box to Update Text Box
- PRB: "Couldn't Update; Currently Locked by User" Error Message
- PRB: Every Other Page Blank in Printed Reports
- INF: How to Use the Query-by-Form (QBF) Technique
- INF: Troubleshooting Setup and Installation Problems
- INF: Requirements to Connect to SYBASE SQL Server
- INF: Error Messages When Concatenating Variables or Controls
- INF: Round or Truncate Values to Desired Number of Decimals
- INF: How to Create Synchronized Combo Boxes
- INF: Sending a MAPI Mail Message Using Access Basic
- INF: Four Ways to Move to a Record from a Combo Box Selection
- INF: Connecting to SYBASE SQL Server from an ODBC Application
- PRB: "Record Lock Threshold Exceeded" with Large Action Query
- INF: How to Check for Duplicate Values in Primary Key Fields
- INF: Showing All Records (Including Null) in a Parameter Query
- PRB: "Out of Memory" or "Query Too Complex" Running Report
- INF: How to Make a Combo Box Default to First Item in List
- INF: "n Parameters Were Expected, but Only 0 Were Supplied"
- INF: DLookup() Usage, Examples, and Troubleshooting Tips
- INF: How to Delete Duplicate Records from a Table
- PRA: Error Message "Internal Database Error -1310"
- INF: Questions and Answers About .LDB Files
- INF: Troubleshooting Tips for Error Values
- INF: Using Microsoft Access 2.0 Databases with Visual Basic
- INF: Microsoft Access and ORACLE ODBC Drivers
- INF: How to Create an SQL Pass-Through Query in Access Basic
- PRB: Cannot Enforce Referential Integrity in Relationship
- INF: How to Optimize Microsoft Access Version 2.0 Queries
- INF: How to Optimize Microsoft Access Version 2.0 Performance
- INF: Overview of Conversion from Version 1.1 to 2.0 Issues
- INF: Using Version 1.1 Databases in Version 2.0
- INF: Using SHARE.EXE and VSHARE.386 with Microsoft Access 2.0
- INF: Using "Name" in Expressions

- INF: Tips for Improving Combo and List Box Performance
- INF: Tips for Improving Subform Performance
- INF: How to Use In-Place Activation with OLE Objects
- INF: How to Filter a Subform by Changing the Record Source
- INF: How to Use DDE to Pass Information to MS Access 2.0
- INF: How to Refer to a Control on a Subform or Subreport
- PRB: Office Manager 4.2 Will Not Load Microsoft Access 2.0
- PRB: Cannot Display More Than Four Graph Objects at Once
- 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
- PRB: Cannot Run Setup for Solutions Pack on Workstation
- PRA: Error "Setup Cannot Create SYSTEM.MDA File (-1809)"
- PRA: Mailing Label Wizard Creates One-Page Report
- INF: How to Find a Record Using a Bound Control (2.0)
- PRB: Adding Records to AutoLookup Form Generates Error
- INF: Query by Form (QBF) Using Dynamic QueryDef
- INF: Understanding Event Order in MS Access 2.0
- PRA: Error (-1310) Running Crosstab or Totalling Query
- INF: How to Bring a Subtotal from a Subform to a Main Form
- INF: How to Create a Multiple-Selection List Box
- PRA: Error Message "Couldn't Open SYSTEM.MDA"
- INF: OLE 2.02 Required to Use Custom Controls

Sources of More Information

User Survey

Access 2.x Frequently Asked Quetions

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

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:

```
kbbuglist
```

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

```
kbbuglist or kbfixlist
```

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

```
kbtshoot 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)	FmsEvnt
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	FmrTlbx
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)	IsmTxtd
ISAM	Import/Export Text (fixed width)	IsmTxtfx
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	PtrDialg
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	PgmPrcs
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: 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

To following steps outline in general how to create a custom counter. Further below is a more detailed description on how to create a custom counter.

1. Create a separate table that will maintain the next available custom counter. This table will have one field and one record, with the value of the next available counter in this one record.

2. From within Access Basic, open this counter table and retrieve the value stored there.

3. Increment the value retrieved and store the number back into the database.

4. Close the table and use the value in the appropriate table as the next available counter.

In a single-user environment, the steps described above can be done with macro actions. In a multiuser environment, Access Basic is needed to handle the event where this counter table is locked. Macro actions do not provide the locking control needed in the multiuser environment.

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: 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: 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 = CVDate("1/3/92 4:05:06")
   Time2 = CVDate("2/2/92 5:02:10")
   HoursTest = Format(Int(Abs(Time1 - Time2) * 24), "###:") + _
```

```
         Format(Abs(Time1 - Time2), "nn:ss")
   End Function
```

The HoursTest() function uses the CVDate() function to convert the string expression to a date value. The HoursTest() function then takes the absolute value of the difference between the two date/time values and multiplies this new value by 24:

```
   Abs(Time1 - Time2) * 24
```

HoursTest() uses the Int() function to return only the integer portion of the date/time value. The resulting integer is the total number of hours elapsed:

```
   Int(Abs(Time1 - Time2) * 24)
```

The first Format() function returns the elapsed hours. The second Format() function takes the absolute value of the difference between the two time values and uses the "nn:ss" argument to capture only the minutes and seconds. HoursTest() concatenates the results of two Format() functions, as follows:

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

The second example, ElapsedTest(), adds the number of elapsed days to the resulting value:

```
   Function ElapsedTest()
   ' This function displays the elapsed time as the total number of
   ' days, hours, minutes, and seconds. For example: 4 Days 01:04:05 Hours.
      Time1 = CVDate("1/31/92 4:01:05")
      Time2 = CVDate("2/4/92 5:05:10")
      ElapsedTime = Abs(Time1 - Time2)
      ElapsedDays = Int(ElapsedTime)
      ElapsedPartialDay = (ElapsedTime - ElapsedDays)
      ElapsedTest = ElapsedDays & " Days " & Format(ElapsedPartialDay,_
         "HH:NN:SS") & " Hours"
   End Function
```

The ElapsedTest() function uses a process similar to that used by HoursTest(), but displays its result in a different format.

Microsoft Excel Dates vs. Microsoft Access Dates

Microsoft Excel also uses double-precision numbers to store date/time values, but Microsoft Excel values are limited to the range 1/1/1900 through 12/31/2078. The Microsoft Excel date and time system begins with the double-precision number 1.0, which represents 1/1/1900 00:00:00 (12:00:00 A.M.).

Originally, Microsoft Access supported the same date range as Microsoft Excel (1900-2078). But when Microsoft extended the date range in Microsoft Access to 100-9999, the 1900 leap year anomaly was corrected. This means that for dates between 1/1/1900 and 2/29/1900, the internal integer portion of the date values for Microsoft Access

and Microsoft Excel differ by one.

Additional reference words: 1.00 1.10 2.00 date/time fractional
KBCategory: kbusage
KBSubcategory: TblDatyp

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: 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:

0	Invalid
1	True/False (non-NULL)
2	Unsigned byte (Byte)
3	2-byte signed integer (Integer)
4	4-byte signed integer (Long)
5	8-byte signed integer (Currency)
6	4-byte single-precision floating point (Single)
7	8-byte double-precision floating point (Double)
8	Date/Time (date is integer, time is fraction)
9	Binary data, 255-byte maximum
10	ANSI text, not case sensitive, 255-byte maximum (Text)
11	Long binary (OLE Object)
12	Long text (Memo)

NextRow	The data in the next row in the table or query. When you first open a channel, NextRow returns the data in the first row. If the current row is the last record and you execute NextRow, the request fails.

PrevRow	The data in the previous row in the table or query. If PrevRow is the first request on a new channel, the data in the last row of the table or query is returned. If the first record is the current row, the request 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

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

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

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: 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:

  (Form or Report Width)+(Left Margin)+(Right Margin)<=(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: RptLayou

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: 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 $p$g
path=c:\windows
set temp=c:\<valid path>
```

4. Exit from the text editor.

5. With the clean, bootable floppy disk in drive A, reboot your system.

6. If you still experience Setup problems, try starting Windows in standard mode by typing

```
win /s
```

at the A prompt. Windows does not try to create any virtual memory in standard mode. If running in standard mode resolves your problem, contact Windows Product Support for further assistance.

For more information on the clean booting your system, query on the following words in the Microsoft Knowledge Base:

```
clean and boot and WINDOWS
```

kbsetup
Additional reference words: 1.00 1.10 2.00
KBCategory: kbsetup kbtshoot
KBSubcategory: StpOthr

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: 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: 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: 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: 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
================

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.

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()
   ```

   NOTE: In version 2.0, Use OnClick instead of OnPush.

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 MAPISendMail 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 MAPISendMail function
'   is called to send the message.
'
' INPUT PARAMETERS:
'   sSubject: The text to appear in the subject line of the message
'   sTo:      Semicolon-delimited list of names to receive the
'             message
'   sCC:      Semicolon-delimited list of names to be CC'd
'   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 MAPISendMail 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 MAPISendMail 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 = MAPISendMail(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: 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 Combo1 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: Combo1
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: =Combo1
   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]![Combo1]
```

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: Combo1
   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: Combo1

By changing the value in Combo1, 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: 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 contains 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

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

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",_         MsgBox
               "[customer id]=Form.[customer id]")<>0
               ...                                          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: 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

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

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: "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: 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: 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

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

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: GnlMu

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: 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: 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: 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

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

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 Impove 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. Exressions 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: 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: 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: 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 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 3.11, 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 "Name" in Expressions

Article ID: Q112130

The information in this article applies to:

- Microsoft Access version 2.0

SUMMARY
=======

All Microsoft Access version 2.0 objects have a Name property, but Microsoft Access version 1.x objects do not. A Microsoft Access 1.x database will not run correctly in Microsoft Access 2.0 if you refer to Name as an ambiguously qualified object in the 1.x database. To correct this, Microsoft Access treats Name as a special case when you are using a 1.x database.

MORE INFORMATION
================

With the advent of Microsoft Access 2.0 and data access objects (DAO), all objects have a Name property. Name is also a very common field name. This creates a conflict when you are running a 1.x database in Microsoft Access 2.0. When you refer to an ambiguously qualified object, both Microsoft Access 1.x and 2.0 have a predetermined search path to locate the object. The search order is as follows:

   Properties, controls, fields

If a field has the same name as a property, the value of the property will be returned instead of the value of the field. In order to deal with the new Name property, the search order when referring to Name changes to:

   Controls, fields, properties

Note that this only applies to Name and not to any other property.

Listed below is acceptable ambiguous syntax in Microsoft Access 1.x and unambiguous syntax in Microsoft Access versions 1.x and 2.0. Microsoft Access 2.0 will handle the ambiguous syntax correctly when you are using an Access 1.x database.

Acceptable (Ambiguous) 1.x Syntax

- In a text box on a report:

      ControlSource: =[Name]

- In a macro:

      SetValue
         Item: [Name]
         Expression: UCase([Name])

- In a DLookup() expression:

```
DLookup("[ID]","MyTable","[Name]=Form.Name")
```

- In Access Basic using ListTables(), ListFields(), and so on:

```
Set MySnap = MyDB.ListTables()
Debug.Print MySnap.Name
```

Unambiguous 1.x and 2.0 Syntax

- In a text box on a report:

```
ControlSource: =Report![Name]
```

- In a macro:

```
SetValue
   Item: Form![Name]
   Expression: UCase(Form![Name])
```

- In a DLookup() expression:

```
DLookup("[ID]","MyTable","[Name]=Form!Name")
```

- In Access Basic using ListTables(), ListFields(), and so on:

```
Set MySnap = MyDB.ListTables()
Debug.Print MySnap!Name
```

Note the explicit use of the form or report object and the use of the exclamation point (!) instead of the period (.) to refer to fields. This tells Microsoft Access to search controls and fields and to ignore properties. If you use the period (.), properties are assumed and searched before controls and fields.

Using Naming Conventions

Ambiguously referenced objects can also be correctly referenced by using a naming convention that includes a standard prefix or suffix that can be attached to object names. For example, ctlName could indicate a control, and fldName could indicate a field.

Converting to Microsoft Access 2.0

When you are converting a Microsoft Access 1.x database to 2.0 format, you must use the unambiguous method to reference fields and controls that have the same names as properties.

REFERENCES
==========

For more information about identifiers in expressions, search for "identifiers in expressions" using the Microsoft Access 2.0 Help menu.

Additional reference words: 2.00 dao
KBCategory: kbusage
KBSubcategory: ExrOthr

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: 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 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 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 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

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: 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

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.

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.

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.

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.

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 "ODBCConnectString 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

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

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: 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 stop
KBCategory: kbusage
KBSubcategory: RptLbl

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

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, one 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
>
> -or-
>
> Field can't be edited

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

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: 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

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

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 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 multiselection
KBCategory: kbusage
KBSubcategory: FmrCodeb

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

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

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 alternativeto 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?

____ Access 2.x or Access 1.x
____ Excel 5.0 Most Useful articles
____ Visual Basic for Applications
____ Project 4.0 articles
____ Windows for Workgroups articles
____ Windows for Workgroups 100 Most Useful articles
____ Multimedia Helpfile
____ Word 6.0 articles
____ Windows 3.1 articles
____ Windows Printing
____ Publisher
____ Office

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.