

ODBC Drivers

You can use ODBC drivers supplied with Microsoft Access to connect to SQL databases and import, export, or attach the data in these databases. In Microsoft Access version 1.0, the Microsoft SQL Server driver was supplied. In version 1.1, the SQL Server driver also supports Sybase SQL Server, and there is a new ORACLE Server driver.

Installing and Administering ODBC Drivers

The following topics describe how to install the ODBC drivers and set them up for use with Microsoft Access. Once you have installed the ODBC drivers you want, you need to set up a data source for each SQL database for which you want to import, export, and attach data.

- Installing ODBC Drivers and Setting Up Data Sources explains how to install ODBC drivers and set up data sources from the ODBC Setup disk and from a network server.
- ODBC Drivers and Built-in Drivers describes the drivers that Microsoft Access uses for importing, exporting, and attaching data from various database formats, spreadsheets, and text files.
- Network Configurations and ODBC Drivers describes the network protocols and supported network configurations you can use when you connect a Microsoft Access database to an SQL database using an ODBC driver.

The ODBC Help Files

In addition to the Help for Microsoft Access, version 1.1 includes a separate Help file for each ODBC driver and for the ODBC Control Panel option.

- The SQL Server Driver Help (DRVSSRVR.HLP) provides detailed information on using the SQL Server driver and setting up data sources for SQL Server databases. These topics apply to both Microsoft SQL Server and Sybase SQL Server databases.
- The ORACLE Server Driver Help (DRVORACL.HLP) provides detailed information on using the ORACLE Server driver and setting up data sources for ORACLE Server databases. You can also read the ORACLE.TXT file in your Microsoft Access directory for specific information on how to configure ORACLE Server components for use with Microsoft Access and the ORACLE Server driver.
- The ODBC Control Panel Option Help (ODBCINST.HLP) explains how to use the ODBC Control Panel option (called the ODBC Administrator if you have Microsoft Windows version 3.0) to add, modify, and delete ODBC drivers and data sources.

Using the ODBC Help Files

Because the ODBC Help files are separate from the Microsoft Access Help file and are meant to be used with a number of Microsoft applications, you should keep the following information in mind when you use these files:

- The SQL Server Driver and ORACLE Server Driver Help files are installed on your computer only if you have installed the associated ODBC driver. If you can't open one of these Help files, it probably means you haven't installed this ODBC driver. The Help file for the ODBC Control Panel option is installed when you run the Setup program on the ODBC Setup disk.
- The SQL Server Driver and ORACLE Server Driver Help files include For Advanced Users and For Programmers topics, which contain information that may not be applicable to Microsoft Access. The For All Users topics provide most of the information you need to use the ODBC drivers with Microsoft Access. If you need more information (for example, you are developing a Microsoft Access application and need to call ODBC functions in your Access Basic code, or you need to use SQL statements to query data in an SQL database), look for this information in the more advanced topics.
- You must use the Back button at the top of the Help window to return to Microsoft Access Help from one of these Help files. If you choose the Contents button, you will display the Contents screen for that ODBC Help file.

ODBC Drivers and Built-in Drivers

In Microsoft Access, you can import, export, and attach data in a number of different database formats and from spreadsheets and text files. To connect to a particular type of data, Microsoft Access uses either an ODBC driver or a built-in driver.

ODBC Drivers

The following ODBC drivers have been tested and verified for use with Microsoft Access:

- SQL Server (Microsoft and Sybase)
- ORACLE Server

To use one of these drivers, you must install the driver and set up a data source for each database of this type whose data you want to use with Microsoft Access.

Built-in Drivers

You can use built-in drivers to import, export, or attach the following types of data:

- Other Microsoft Access databases
- FoxPro version 2.0 and 2.5 database files
- Paradox version 3.x tables
- dBASE III and IV files
- Btrieve tables
- Microsoft Excel and Lotus 1-2-3 spreadsheets
- Fixed-width and delimited text files
- Microsoft Word for Windows mail merge data files (export only)

You do not need to install the drivers for these types of data; they are built into Microsoft Access.

Note Products from other vendors and other Microsoft products also contain ODBC drivers, including drivers for the applications in the lists above. These drivers may have been installed on your computer. If you want to know whether these drivers have been tested and verified for use with Microsoft Access, contact the driver vendor.

Installing ODBC Drivers and Setting Up Data Sources

Before you can use an [SQL database](#) for importing, exporting, or attaching, the appropriate [ODBC driver](#) for this type of database must be installed, and you must set up a [data source](#) for the data.

You can install one or more ODBC drivers and set up data sources for these drivers using the ODBC Setup program after you first install Microsoft Access. You can also add, modify, and delete ODBC drivers and data sources at a later time using the ODBC Control Panel option (called the ODBC Administrator if you are using Microsoft Windows version 3.0).

Using the ODBC Setup Program to Install the ODBC Files

You must first use the ODBC Setup program to install the ODBC driver files and other related files on your computer or on a network server.

To install the ODBC files on your computer from the ODBC Setup disk

- 1 Insert the ODBC Setup disk into drive A.
- 2 Start Windows.
- 3 In the Windows Program Manager, choose Run from the File menu.
Windows displays the Run dialog box.
- 4 In the Command Line box, type **a:setup**
- 5 Choose the OK button, and then follow the instructions on the screen.

To install the ODBC files from a network server

- 1 Have your network administrator copy all the files on the ODBC Setup disk to the network server.
- 2 Connect to this network server.
- 3 Start Windows.
- 4 In the Windows Program Manager, choose Run from the File menu.
Windows displays the Run dialog box.
- 5 In the Command Line box, type **path\setup** where *path* is the server directory to which the ODBC files were copied; for example, type **f:\access\odbc\setup**
- 6 Choose the OK button, and follow the instructions on the screen.

For more information on setting up Microsoft Access on your computer or on a network server, see the Installing Microsoft Access (Version 1.1 Upgrade) card included with version 1.1 of Microsoft Access.

Note If you want to install the same ODBC drivers and data sources for a number of computers on a network, you can use the automatic setup (**setup /auto**) option. This option is described in Appendix D, "Setting Up Microsoft Access on a Network," in the *Microsoft Access User's Guide*.

The ODBC Setup program installs the ODBC files on your computer. These include:

- The ODBC Driver Manager (ODBC.DLL), a dynamic-link library (DLL) that loads ODBC drivers for use by Microsoft Access or other applications.
- The ODBC.INI file, which stores information about each data source you set up.
- The ODBC drivers you select. Each driver is a DLL that enables Microsoft Access to connect to a database server, retrieve data from the server, and return any errors that occur during this process.
- The Help files for any ODBC drivers you select.
- The ODBC Control Panel option (ODBCINST.DLL) or the ODBC Administrator (ODBCADM.EXE) and the Help file for the program (ODBCINST.HLP).

Each ODBC driver has specific requirements for installing the driver and setting up data sources. When you run the ODBC Setup program, you can press the Help button on each screen to get Help on how to enter the information on this screen. If you need more information about a particular driver, consult the [SQL Server Driver Help](#) or the [ORACLE Server Driver Help](#).

Using the ODBC Control Panel Option to Install ODBC Drivers and Set Up Data Sources

You can use the ODBC Control Panel option (or the ODBC Administrator) to add, modify, and delete ODBC drivers and data sources at any time after you have used the ODBC Setup program to install the ODBC files. Note that when you use the ODBC Setup program, you don't need to install all the drivers you want or set up the data sources. You can do this later using the ODBC Control Panel option (or the ODBC Administrator).

To start the ODBC Control Panel option

- 1 In the Main group in the Program Manager window, double-click the Control Panel icon.
- 2 In the Control Panel window, double-click the ODBC icon.

Note If you're using Windows version 3.0, start the ODBC administrator by double-clicking the ODBC Administrator icon in the Microsoft ODBC group.

When you start the ODBC Control Panel option (or the ODBC Administrator), you can press the Help button on each screen to get Help on how to enter the information on this screen. The [ODBC Control Panel Option Help](#) provides detailed information on how to use this program.

Network Configurations and ODBC Drivers

You need to use specific network protocols and network configurations when you connect a Microsoft Access database to an SQL database using an ODBC driver. The following protocols and configurations have been tested for use with Microsoft Access and ODBC drivers.

ORACLE Server

Microsoft Access has been tested with the ORACLE Server driver on the following platforms.

Server platform	Network protocol
ORACLE6/7 (Sun OS)	Microsoft LAN Manager TCP/IP version 1.0
ORACLE6 (OS/2)	Microsoft Named Pipes

For other supported platforms, please consult your ORACLE SQL*Net documentation.

SQL Server

Microsoft Access has been tested with the SQL Server driver on the following platforms.

Server platform	Network protocol
Sybase SQL Server version 4.9.1 (Sun OS)	Microsoft LAN Manager TCP/IP version 1.0
Microsoft SQL Server version 4.2a (OS/2)	Microsoft Named Pipes

For other supported platforms, please consult your Sybase SQL Server documentation.

What's New in Microsoft Access Version 1.1

Common Questions and Answers

Microsoft Access version 1.1 includes the following new features and enhancements to existing features:

New Maximum Database Size

The maximum database size is now 1 gigabyte instead of 128 megabytes. You can convert your version 1.0 databases to the new format.

Exporting to Word for Windows Mail Merge Data Files

You can create Word for Windows mail merge data files using your Microsoft Access data.

Improved Connectivity to Microsoft FoxPro Databases

You can import and attach data from and export data to FoxPro version 2.0 and FoxPro version 2.5 databases.

New Connectivity to ORACLE Server and Sybase SQL Server Databases

You can use new or improved ODBC drivers to import and attach data from and export data to ORACLE Server and Sybase SQL Server databases. Detailed information on how to install and administer ODBC drivers and data sources is also now available.

Enhanced Importing of Fixed-Width Text Files

You can now import fixed-width text files whose records are not all the same length.

Importing of the Microsoft Excel Database Named Range

You can import the Database named range from Microsoft Excel spreadsheets. Also, when you export data from Microsoft Access to a Microsoft Excel spreadsheet, Microsoft Access automatically creates a Database named range containing this data.

Improved Connectivity to Btrieve Tables

In version 1.1, it is easier to import and attach data from and export data to Btrieve tables. For example, Microsoft Access no longer requires the INDEX.DDF file to read Btrieve data files. For more information on using Btrieve with Microsoft Access, see the text file BTRIEVE.TXT in your Microsoft Access directory.

Optional Login ID and Password Storage for SQL Databases

You can prevent users from storing login IDs and passwords for SQL databases locally or allow them to do so.

Ability to Ensure That Your Microsoft Access System Is Unique

You can ensure that your Microsoft Access system is unique. If you are using security, you can now prevent a user from creating another version of the same system and thereby gaining unauthorized access to it.

New Nordic Sort Orders

The single Nordic sort order in version 1.0 has been replaced by three country-specific Nordic sort orders—Swedish/Finnish, Norwegian/Danish, and Icelandic.

Miscellaneous Improvements

Microsoft Access version 1.1 includes additional improvements.

Exporting Microsoft Access Data to Word for Windows Mail Merge Data Files

You can use data from your Microsoft Access tables to create a data file and then use this data file with the mail merge feature in Microsoft Word for Windows to create form letters, mailing labels, and other merged documents. (In Word version 2.x for Windows, this feature is called "print merge.") You create a document (called the main document) in Word for Windows and then use the data file you created in Microsoft Access to merge the table data into the appropriate fields in the main document.

To export data to a mail merge data file

- 1 Open the database containing the table with the data you want to export, or switch to the Database window for the open database.
- 2 From the File menu, choose Export.
Microsoft Access displays the Export dialog box, which contains a list of data destinations.
- 3 In the Data Destination box, select Word for Windows Merge, and then choose the OK button.
Microsoft Access displays a list of tables in the current database.
- 4 Select the table with the data you want to export, and then choose the OK button.
Microsoft Access displays the Export to File dialog box.
- 5 Enter a file name for the data file you want to create, and then choose the OK button.
By default, this is a text (.TXT) file. If you want the data file to be in your Word for Windows directory, specify the directory as well.
Microsoft Access displays the Export Word Merge Options dialog box.
- 6 If necessary, specify text options for the data file fields. These options determine how dates, times, and numbers are formatted in the data file.
When you choose the OK button, Microsoft Access creates the data file containing all the data from your table.

Microsoft Access puts the field names from the table in the first record in the data file (the header record) and the data from the table in the succeeding records (the data records). When you use the mail merge feature, Word for Windows matches the fields in the main document with the field names in the header record and puts the appropriate data from the data records into these fields in the main document.

Because field names in Word for Windows mail merge data files can contain only letters, numbers, and underscores (_), when you export Microsoft Access data to a data file, Microsoft Access automatically performs field name conversions in the following order:

- Any spaces or illegal characters are converted to underscores.
- If the first character in the field name is a number or an underscore, Microsoft Access adds the prefix "m_." For example, 1Field becomes m_1Field, and _Field2 becomes m__Field2.
- Microsoft Access truncates any character after the twentieth character.
- If the conversion results in duplicate field names, Microsoft Access adds a number to the end of each duplicate name (replacing the final character, if the field name contains twenty characters). For example, the field names Order#, Order\$, and Order% become Order_, Order_1, and Order_2.

The field names in the header record in the data file must match the field names in the main document. If you aren't sure of the field names or errors occur when you try to merge, you can open the data file in Word for Windows and edit the field names, or you can edit the field names in the main document. Note that the data file Microsoft Access creates is a text (.TXT) file rather than a document (.DOC) file, which is the standard Word for Windows format, so you should look in your directories for .TXT files when you want to open and edit a data file created by Microsoft Access.

Tip When you export data from a Microsoft Access table to a data file, the data file contains all of the data in the table. If you want to export a subset of the data—for example, just the records for a particular range of dates or just certain fields

–you can specify these criteria in a make-table query. You can then export the data from the table created by this query.

For more information on using the Word for Windows mail merge feature, see your Word for Windows documentation.

Note In version 1.1, the TransferText action includes a new setting for the Transfer Type argument: Export Word for Windows Merge. You can use this setting to export Microsoft Access data to a Word for Windows mail merge data file using a macro.

TransferText Action

Access Basic

Imports or exports text between the current Microsoft Access database and a text file.

Action argument	Description
-----------------	-------------

Transfer Type	<p>The type of transfer you want to make. You can import data from or export data to delimited or fixed-width text files. You can also export data to a Microsoft Word for Windows mail merge data file, which you can then use with the Word for Windows mail merge feature to create merged documents such as form letters and mailing labels.</p> <p>Select Import Delimited, Import Fixed Width, Export Delimited, Export Fixed Width, or Export Word for Windows Merge from the drop-down list in the Action Arguments section of the Macro window. The default is Import Delimited.</p>
Specification Name	<p>The specification name for the set of options that determines how a text file is imported or exported. Required argument for fixed-width text files.</p> <p>You use the Imp/Exp Setup command on the File menu to create a specification for a particular type of text file, such as a delimited text file that uses tabs to separate fields and has an MDY format for dates. Select the desired options in the dialog boxes for the Imp/Exp Setup command and save the specification. You can then enter the specification name in this argument whenever you want to import or export the same type of text file.</p> <p>You can import or export delimited text or Word for Windows mail merge data files without entering a specification name for this argument. In this case, Microsoft Access uses the defaults from the dialog boxes of the Imp/Exp Setup command.</p>
Table Name	<p>The name of the Microsoft Access table to import text data to or export text data from. Required argument.</p> <p>If you select Import Delimited or Import Fixed Width for the Transfer Type argument, Microsoft Access appends the text data to this table if the table already exists. Otherwise, Microsoft Access creates a new table containing the text data.</p>
File Name	<p>The name of the text file to import from or export to. Include the full path name. Required argument.</p> <p>Microsoft Access creates a new text file when you export data from Microsoft Access. If the file name is the same as the name of an existing text file, Microsoft Access replaces the existing text file.</p>
Has Field Names	<p>Specifies whether the first row of the text file contains the names of the fields. If you select Yes, Microsoft Access uses the names in this row as field names in the Microsoft Access table when you import the text data. If you select No, Microsoft Access treats the first row as a normal row of data. The default is No.</p> <p>Microsoft Access ignores this argument for Word for Windows mail merge data files, for which the first row must contain the field names.</p> <p>When you export Microsoft Access table data to a delimited or fixed-width text file, Microsoft Access inserts the field names of your table into the first row of the text file if you have selected Yes for this argument.</p> <p>If you are importing a fixed-width text file and select Yes for this argument, the first row containing the field names must use the text delimiter set in the import/export specification to separate the field names. If you are exporting to a fixed-width text file and select Yes for this argument, Microsoft Access inserts the field names into the first</p>

row of the text file using this delimiter.

Remarks

Text data that you append to an existing Microsoft Access table must be compatible with the table's structure. Each field in the text must be of the same data type as the corresponding field in the table, and the fields must be in the same order (unless you set the Has Field Names argument to Yes, in which case the field names in the text must match the field names in the table).

This action is similar to choosing the Import or Export command from the File menu of the Database window. You can use these commands to select a data source, such as Microsoft Access or a type of database, spreadsheet, or text file. If you select a text file (delimited, fixed-width, or Word for Windows mail merge), a series of dialog boxes will appear in which you can select the name of the text file and other options. The arguments of the TransferText action reflect the options in these dialog boxes.

Tip An import/export specification stores the information Microsoft Access needs to import or export a text file. You can use stored specifications to import or export text data from or to similar text files. For example, you might receive weekly sales figures in a text file from a mainframe computer. You can create and save a specification for this type of data and then use the specification whenever you add this data to your Microsoft Access database.

TransferText Action – Access Basic

Syntax

DoCmd TransferText [*transfertype*] [, *specificationname*], *tablename*, *filename* [, *hasfieldnames*]

Argument	Description
<i>transfertype</i>	<p>One of the following <i>intrinsic</i> constants:</p> <p>A_IMPORTDELIM A_IMPORTFIXED A_EXPORTDELIM A_EXPORTFIXED A_EXPORTMERGE</p> <p>If you leave this argument blank, the default (A_IMPORTDELIM) is assumed.</p>
<i>specificationname</i>	<p>A string expression that is the name of an import or export specification you have created and saved in the current database. This argument is required for fixed-width text files. For delimited text files and Word for Windows mail merge data files, you can leave this argument blank to select the default import/export specifications.</p>
<i>tablename</i>	<p>A string expression that is the name for the Microsoft Access table you want to import text data to or export text data from.</p>
<i>filename</i>	<p>A string expression that is the full name, including the path name, of the text file you want to import from or export to.</p>
<i>hasfieldnames</i>	<p>Use the reserved word True (-1) to use the first row of the text file as field names when importing or exporting. Use the reserved word False (0) to treat the first row of the text file as normal data. If you leave this argument blank, the default (False) is assumed.</p> <p>This argument is ignored for Word for Windows mail merge data files, which must always contain the field names in the first row.</p>

Remarks

You can leave an optional argument blank in the middle of the syntax, but you must include the argument's comma. If you leave a trailing argument or arguments blank, don't use a comma following the last argument you specify.

Example

This example exports the data in the Microsoft Access table `External Report` to the delimited text file `APRIL.DOC` using the specification `Standard Output`.

```
DoCmd TransferText A_EXPORTDELIM, "Standard Output", "External Report", "C:\TXTFILES\APRIL.DOC"
```

Converting Version 1.0 Databases to Version 1.1 Format

In Microsoft Access version 1.1, a database can be as large as 1 gigabyte. The size limit for version 1.0 databases is 128 megabytes. You can work with version 1.0 databases in Microsoft Access version 1.1 with no problems, but you may want to take advantage of the increased size limit by converting your 1.0 databases to the new format.

To convert a version 1.0 database to version 1.1 format

- 1 In the Microsoft Access startup window, choose Compact Database from the File menu. Microsoft Access displays the Database to Compact From dialog box.
- 2 Select the database you want to convert to version 1.1 format, and then choose the OK button. Microsoft Access displays the Database to Compact Into dialog box.
- 3 In the List Files of Type box, select Access V1.1.
- 4 Specify a new name, drive, and directory for the converted database; or enter the same name in the File Name box; and then choose the OK button. Microsoft Access compacts the database and converts it to version 1.1 format. If you specify the same drive, directory, and file name, the new database replaces the old database.

You can use this same procedure to convert a version 1.1 database to version 1.0 format—just select Access V1.0 in the List Files of Type box in step 3.

You can also specify version 1.1 or version 1.0 format when you create a new database using the New Database command on the File menu. You should specify version 1.1 format when you create a new database or compact an existing database unless the new or compacted database will be used by someone with version 1.0 of Microsoft Access. (You can't use version 1.1 databases with Microsoft Access version 1.0.)

Notes

- The only way to convert a version 1.0 database to version 1.1 format is to compact it. Opening a version 1.0 database in Microsoft Access version 1.1, changing data or database objects, and saving the changes will not convert the database to version 1.1 format.
- If you have a version 1.0 database that uses the Nordic sort order, you should convert the database to version 1.1 format. This enables you to take advantage of the new country-specific Nordic sort orders and prevents unexpected sorting results that can occur if you work with a version 1.0 database that uses the old Nordic sort order in version 1.1 of Microsoft Access.

Using Microsoft FoxPro Data with Microsoft Access

In Microsoft Access version 1.1, you can import and attach data from and export data to Microsoft FoxPro version 2.0 and 2.5 databases.

To import or attach a FoxPro file

- 1 Open the database where you want to import or attach the FoxPro data, or switch to the Database window for the open database.
- 2 From the File menu, choose Import or Attach Table.
Microsoft Access displays a list of data sources.
- 3 In the Data Source box, select FoxPro 2.0 or FoxPro 2.5, and then choose the OK button.
Microsoft Access displays the Select File dialog box.
- 4 Select the .DBF file you want to import or attach, and then choose the Import button or the Attach button.
 - If you're importing, Microsoft Access creates a new table named after the FoxPro file you selected and imports the data from this file.
 - If you're attaching, Microsoft Access displays a dialog box you can use to associate FoxPro index (.IDX or .CDX) files with the attached file.
Select the FoxPro index files you want to associate one at a time, and choose the Select button. When you have finished associating indexes, choose the Close button.
Microsoft Access adds an attached table icon to the list of tables in the Database window. This icon represents the link to the FoxPro file.
- 5 If you want to import or attach another FoxPro file, repeat step 4. When you have finished importing or attaching, choose the Close button.

Important You can associate one or more FoxPro index files with an attached FoxPro table to improve performance when you access data in the attached table. Microsoft Access keeps track of the indexes in a special information (.INF) file. Do not delete or move the index files or the information (.INF) file. If you do, Microsoft Access won't be able to open the attached table. In addition, if you use FoxPro to update data in a .DBF file that you have attached, you must update any associated indexes as well.

To export a Microsoft Access table to a FoxPro file

- 1 Open the database containing the table you want to export, or switch to the Database window for the open database.
- 2 From the File menu, choose Export.
Microsoft Access displays the Export dialog box, which contains a list of data destinations.
- 3 In the Data Destination box, select FoxPro 2.0 or FoxPro 2.5, and then choose the OK button.
Microsoft Access displays a list of tables in the current database.
- 4 Select the table you want to export, and then choose the OK button.
Microsoft Access displays the Export to File dialog box.
- 5 Specify a name, drive, and directory for the new FoxPro file, and then choose the OK button.
Microsoft Access creates the FoxPro file containing all the data from your table.

Using ORACLE Server and Sybase SQL Server Data with Microsoft Access

In Microsoft Access version 1.1, you can import and attach data from and export data to ORACLE Server and Sybase SQL Server databases, in addition to Microsoft SQL Server databases.

Note For each ORACLE Server or Sybase SQL Server database that you want to use for importing, exporting, or attaching, you must install the appropriate ODBC driver and set up a data source for the data before you can import, export, or attach to this database. Each ODBC driver has specific requirements for installing the driver and setting up data sources. For more information, see ODBC Drivers.

To import or attach an SQL database table

- 1 Open the database where you want to import or attach the SQL database table, or switch to the Database window for the open database.
- 2 From the File menu, choose Import or Attach Table.
Microsoft Access displays a list of data sources.
- 3 In the Data Source box, select <SQL Database>, and then choose the OK button.
Microsoft Access displays the SQL Data Sources dialog box. Any data sources you have set up for ORACLE Server or SQL Server databases appear in this dialog box.

Note This dialog box lists the data sources for any ODBC drivers you have installed. This list may include drivers that have not been tested and verified for use with Microsoft Access. Contact the driver vendor if you want to know whether a particular ODBC driver has been tested and verified for use with Microsoft Access. (For information on drivers that Microsoft Access supports, see ODBC Drivers and Built-in Drivers.)

- 4 Select the SQL data source that contains the table you want to import or attach, and then choose the OK button.
Microsoft Access displays the login dialog box for the SQL data source you selected.
- 5 Enter your login ID and password for the SQL database server, and then choose the OK button.
You may need to enter additional information in this dialog box, depending on the particular server you are connecting to. For detailed information on connecting to a server, see SQL Server Driver and ORACLE Server Driver.
Microsoft Access connects to the SQL database server and displays the list of tables in the SQL database that you can import or attach.

- 6 Select the table you want, and choose the Import button or the Attach button.
 - If you're attaching a table, select the check box if you want to save the login ID and password locally. If you leave the check box cleared, all users must enter the login ID and password every time they open the table in a new session with Microsoft Access.
Note Your SQL database administrator can choose to disable this feature, requiring all users to enter the login ID and password each time they connect to the SQL database.
Microsoft Access adds an attached table icon to the list of tables in the Database window. The icon represents the link to the SQL database table.
 - If you're importing, Microsoft Access creates a new table named after the table you selected and imports the data from the SQL database.

- 7 If you want to import or attach another table from the same data source, repeat step 6. When you have finished importing or attaching, choose the Close button.

Note To edit an attached SQL database table, the table must contain a unique index. If you want to update the data in an attached table that doesn't have an index (for example, a table exported from Microsoft Access to an SQL database), you will need to add an index to the table in the SQL database and then reattach the table.

To export a Microsoft Access table to an SQL database

As with importing or attaching, you must install an ODBC driver and set up a data source for each SQL database you want to export data to.

- 1 Open the database containing the table you want to export, or switch to the Database window for the open database.

2 From the File menu, choose Export.
Microsoft Access displays the Export dialog box, which contains a list of data destinations.

3 In the Data Destination box, select <SQL Database> and then choose the OK button.
Microsoft Access displays a list of tables in the current database.

4 Select the table you want to export, and then choose the OK button.
Microsoft Access displays the Export dialog box.

5 Enter a name for the new SQL table, and then choose the OK button.
Microsoft Access displays the SQL Data Sources dialog box. Any data sources you have set up for ORACLE Server or SQL Server databases appear in this dialog box.

Note This dialog box lists the data sources for any ODBC drivers you have installed. This list may include drivers that have not been tested and verified for use with Microsoft Access. Contact the driver vendor if you want to know whether a particular ODBC driver has been tested and verified for use with Microsoft Access. (For information on drivers that Microsoft Access supports, see [ODBC Drivers and Built-in Drivers](#).)

6 Select the SQL data source you want to export to, and then choose the OK button.
Microsoft Access displays the login dialog box for the SQL data source you selected.

7 Enter your login ID and password for the SQL database server, and then choose the OK button.

You may need to enter additional information in this dialog box, depending on the particular server you are connecting to. For detailed information on connecting to a server, see [SQL Server Driver](#) and [ORACLE Server Driver](#).

Microsoft Access connects to the SQL database server and creates the new table.

Importing the Microsoft Excel Database Named Range

In Microsoft Access version 1.1, you can import the special named range Database that you have defined in a Microsoft Excel spreadsheet. This is a rectangular range of Microsoft Excel spreadsheet cells defined as the Database range; the first row of the range contains the field names.

To import the Database named range

- 1 Open the database where you want to import the Database range, or switch to the Database window for the open database.
- 2 From the File menu, choose Import.
Microsoft Access displays a list of data sources.
- 3 In the Data Source box, select Microsoft Excel, and then choose the OK button.
Microsoft Access displays the Select File dialog box.
- 4 Select the spreadsheet file containing the Database range you want to import, and then choose the Import button.
Microsoft Access displays the Import Spreadsheet Options dialog box.
- 5 Select the First Row Contains Field Names check box so that Microsoft Access will use the first row of the Database range as the field names.
- 6 To create a new table using the spreadsheet data, choose the Create New Table option. To append the data to an existing table, choose the Append to Existing Table option, and select a table in the list.

If you append the data to an existing table, the field names in the Database range must match the field names in the Microsoft Access table, and each field in the Database range must have the same (or a compatible) data type as the corresponding field in the Microsoft Access table.

- 7 In the Spreadsheet Range box, type **Database**. This name is reserved by Microsoft Access for this special named range.
- 8 Choose the OK button.
Microsoft Access creates a table named after the spreadsheet file you selected (unless you are appending the data to an existing table) and imports the data from the Database range in the spreadsheet.
- 9 Choose the Close button.

For more information on importing spreadsheet data, including information on import errors, see Chapter 4, "Importing, Exporting, and Attaching," in the *Microsoft Access User's Guide*. For more information on using the Database named range in Microsoft Excel spreadsheets, see your Microsoft Excel documentation.

Note When you export data from Microsoft Access to a Microsoft Excel spreadsheet, Microsoft Access automatically creates a Database named range containing this data.

Importing Fixed-Width Text Files

In Microsoft Access version 1.0, you can import data from fixed-width text files to Microsoft Access as long as each field (column) in the fixed-width text file is the same width for each record (row), and all records are the same length. Version 1.1 places fewer restrictions on the fixed-width files you can import:

- Each field must start at a particular location in the record and have a maximum width.
- If fields at the end of the record contain no data, these fields can be absent (you don't have to pad these fields with blanks).
- The last field with data in the record can be less than the maximum width.
- If all the records in the file are the same length, there can be an embedded row separator (such as a carriage return and linefeed) in the middle of a record. If the records aren't all the same length, embedded row separators should not be used because Microsoft Access will treat the embedded row separator as the end of the record.

To import a fixed-width text file

Before you import a fixed-width text file, you must create an import/export specification that specifies the starting location and maximum width for each field and other attributes of the file.

- 1 Open the database into which you want to import the fixed-width text file, or switch to the Database window for the open database.
- 2 From the File menu, choose Import.
Microsoft Access displays a list of data sources you can import.
- 3 In the Data Source box, select Text (Fixed Width), and then choose the OK button.
Microsoft Access displays the Select File dialog box.
- 4 Select the text file you want, and then choose the Import button.
Microsoft Access displays the Import Text Options dialog box.
- 5 To create a new table using the text data, choose the Create New Table option. To append the text data to an existing table, choose the Append to Existing Table option, and select a table in the list.
- 6 In the Specification Name box, select a fixed-width import/export specification.
- 7 Choose the OK button.
Microsoft Access creates a table named after the file you selected (unless you chose to append data to an existing table) and imports the data from the text file.
- 8 If you want to import another text file, repeat steps 4 through 7. When you've finished importing, choose the Close button.

For more information on importing text files, including information on import errors, see Chapter 4, "Importing, Exporting, and Attaching," in the *Microsoft Access User's Guide*.

Note When you export data from a Microsoft Access table to a fixed-width text file, Microsoft Access uses the import/export specification you select and creates a fixed-width text file whose records are all the same length.

Creating and Editing Import/Export Specifications

Before you import or export a fixed-width text file, you must specify field widths and other attributes of the file in an import/export specification. When you import or export a text file, any specifications you have created in the current database will be available in the Specification Name box in the Import Text Options dialog box or the Export Text Options dialog box.

To create or edit an import/export specification

- 1 Open the database where you want to store the import/export specification, or switch to the Database window for the open database.
- 2 From the File menu, choose Imp/Exp Setup.
Microsoft Access displays the Import/Export Setup dialog box with default settings for text file attributes.
- 3 If you want to edit an existing import/export specification (instead of creating a new one), select the specification you want in the Specification Name box.
- 4 Fill out the columns in the Field Information section for each field in the text file.
Microsoft Access ignores the settings in the Text Delimiter and Field Separator boxes for fixed-width text files.
- 5 Specify other attributes for the text file, such as date and time formats.
- 6 If you're creating a new specification, choose the Save As button, type a name for the specification, and then choose OK.
Microsoft Access saves the new specification in the database so that you can use it to import text files.
- 7 Choose the OK button.

Using the New Nordic Sort Orders

In Microsoft Access version 1.1, the Nordic sort order available in version 1.0 has been replaced by three country-specific sort orders—Swedish/Finnish, Norwegian/Danish, and Icelandic.

To select one of the new Nordic sort orders for a new database

- 1 Before you create the database, open any existing database.
- 2 From the View menu, choose Options.
Microsoft Access displays the Options dialog box. The General category is selected.
- 3 In the Items box, click the New Database Sort Order box, and then click the down arrow to view the list of available sort orders.
- 4 Select one of the new Nordic sort orders, and then choose the OK button.

When you create a new database, Microsoft Access uses this sort order for the database. This sort order is used for all databases you subsequently create or compact (see the following procedure), until you repeat this procedure and select a different sort order.

To change the sort order for an existing database

- 1 Repeat the preceding procedure, selecting the sort order you want to use for the existing database.
- 2 Close the open database.
- 3 In the Microsoft Access startup window, choose Compact Database from the File menu.
- 4 In the Database to Compact From dialog box, select the database whose sort order you want to change to the sort order you just selected, and then choose the OK button.
- 5 In the Database to Compact Into dialog box, select this same database, and then choose the OK button.

Microsoft Access compacts the existing database and sets its sort order to the one you selected.

Note If you have a version 1.0 database, follow this procedure and select Access v1.1 in the List Files of Type box. This converts your 1.0 database to version 1.1 format. It also prevents unexpected sorting results that can occur if you work with a version 1.0 database that uses the old Nordic sort order in version 1.1 of Microsoft Access.

New Database Sort Order - Swedish/Finnish

This setting tells Microsoft Access to use the Swedish/Finnish language sort order when you create a new database.

The following table shows the character and the ANSI code for the Swedish/Finnish sort sequence.

Char	ANSI code	Char	ANSI code	Char	ANSI code	Char	ANSI code
!	33	i	161	Ç	199	P	80
"	34	ç	162	ç	231	p	112
„	132	£	163	D	68	Q	81
“	147	¤	164	d	100	q	113
”	148	¥	165	Ð	208	R	82
«	171	¦	166	ð	240	r	114
»	187	§	167	E	69	S	83
#	35	¨	168	e	101	s	115
\$	36	©	169	É	201	Š	138
%	37	ª	170	é	233	š	154
&	38	¬	172	È	200	T	84
'	39	®	174	è	232	t	116
,	130	-	175	Ê	202	U	85
<	139	°	176	ê	234	u	117
'	145	±	177	Ë	203	Ú	218
'	146	´	180	ë	235	ú	250
>	155	µ	181	F	70	Û	217
(40	¶	182	f	102	ù	249
)	41	·	183	G	71	Û	219
*	42	,	184	g	103	û	251
+	43	º	186	H	72	V	86
,	44	¼	188	h	104	v	118
-	45	½	189	I	73	W	87
-	150	¾	190	i	105	w	119
—	151	¿	191	Í	205	X	88
—	173	×	215	í	237	x	120
.	46	÷	247	ì	204	Y	89
/	47	0	48	ì	236	y	121
:	58	1	49	î	206	Ý	221
;	59	¹	185	î	238	ý	253
<	60	2	50	ï	207	ÿ	159
=	61	²	178	ï	239	ÿ	255
>	62	3	51	J	74	Ü	220
?	63	³	179	j	106	ü	252
@	64	4	52	K	75	Z	90
[91	5	53	k	107	z	122
\	92	6	54	L	76	Å	197
]	93	7	55	l	108	å	229
^	94	8	56	M	77	Ä	196
_	95	9	57	m	109	ä	228

`	96	A	65	N	78	Ö	214
{	123	a	97	n	110	ÿ	246
	124	Á	193	Ñ	209	Ø	216
}	125	á	225	ñ	241	ø	248
~	126	À	192	O	79	Œ	140
f	131	à	224	o	111	œ	156
...	133	Â	194	Ó	211	ß	223
†	134	â	226	ó	243	Ɔ	222
‡	135	Ã	195	Ô	210	Ɔ	254
^	136	ã	227	ò	242	Æ	198
‰	137	B	66	Ò	212	æ	230
•	149	b	98	ô	244		
~	152	C	67	Õ	213		
™	153	c	99	ö	245		

Note A black rectangle indicates that the character is available only in TrueType fonts. To see how the character will appear, consult Appendix A, "ANSI Character Set," in the *Microsoft Access Language Reference*.

New Database Sort Order - Norwegian/Danish

This setting tells Microsoft Access to use the Norwegian/Danish language sort order when you create a new database.

The following table shows the character and the ANSI code for the Norwegian/Danish sort sequence.

Char	ANSI code	Char	ANSI code	Char	ANSI code	Char	ANSI code
!	33	ı	161	Ç	199	P	80
"	34	ç	162	ç	231	p	112
„	132	£	163	D	68	Q	81
“	147	¤	164	d	100	q	113
”	148	¥	165	Ð	208	R	82
«	171	ı	166	ð	240	r	114
»	187	§	167	E	69	S	83
#	35	¨	168	e	101	s	115
\$	36	©	169	É	201	Š	138
%	37	ª	170	é	233	š	154
&	38	¬	172	È	200	T	84
'	39	®	174	è	232	t	116
,	130	-	175	Ê	202	U	85
<	139	°	176	ê	234	u	117
'	145	±	177	Ë	203	Ú	218
'	146	´	180	ë	235	ú	250
>	155	µ	181	F	70	Ù	217
(40	¶	182	f	102	ù	249
)	41	·	183	G	71	Û	219
*	42	¸	184	g	103	û	251
+	43	¸	186	H	72	Ü	220
,	44	¼	188	h	104	ü	252
-	45	½	189	I	73	V	86
-	150	¾	190	i	105	v	118
—	151	¿	191	Í	205	W	87
—	173	×	215	í	237	w	119
.	46	÷	247	ì	204	X	88
/	47	0	48	ì	236	x	120
:	58	1	49	î	206	Y	89
;	59	¹	185	î	238	y	121
<	60	²	50	ï	207	Ý	221
=	61	²	178	ï	239	ý	253
>	62	3	51	J	74	ÿ	159
?	63	³	179	j	106	ÿ	255
@	64	4	52	K	75	Z	90
[91	5	53	k	107	z	122
\	92	6	54	L	76	þ	222
]	93	7	55	l	108	þ	254
^	94	8	56	M	77	Æ	198
_	95	9	57	m	109	æ	230

`	96	A	65	N	78	Ä	196
{	123	a	97	n	110	ä	228
	124	Á	193	Ñ	209	Ø	216
}	125	á	225	ñ	241	ø	248
~	126	À	192	O	79	Ö	214
f	131	à	224	o	111	ö	246
...	133	Â	194	Ó	211	Å	197
†	134	â	226	ó	243	å	229
‡	135	Ã	195	Ô	210	AA	65, 65
^	136	ã	227	ò	242	aa	97, 97
‰	137	B	66	Ò	212	ß	223
•	149	b	98	ô	244	Œ	140
~	152	C	67	Õ	213	œ	156
™	153	c	99	ö	245		

Note A black rectangle indicates that the character is available only in TrueType fonts. To see how the character will appear, consult Appendix A, "ANSI Character Set," in the *Microsoft Access Language Reference*.

New Database Sort Order - Icelandic

This setting tells Microsoft Access to use the Icelandic language sort order when you create a new database.

The following table shows the character and the ANSI code for the Icelandic sort sequence.

Char	ANSI code	Char	ANSI code	Char	ANSI code	Char	ANSI code
!	33	í	161	B	66	Ó	213
"	34	ð	162	b	98	ö	245
„	132	£	163	C	67	Ó	211
“	147	¤	164	c	99	ó	243
”	148	¥	165	Ç	199	P	80
«	171	¦	166	ç	231	p	112
»	187	§	167	D	68	Q	81
#	35	¨	168	d	100	q	113
\$	36	©	169	Ð	208	R	82
%	37	ª	170	ð	240	r	114
&	38	¬	172	E	69	S	83
'	39	®	174	e	101	s	115
,	130	-	175	È	200	Š	138
<	139	°	176	è	232	š	154
'	145	±	177	Ê	202	T	84
'	146	´	180	ê	234	t	116
>	155	µ	181	Ë	203	U	85
(40	¶	182	ë	235	u	117
)	41	·	183	É	201	Ù	217
*	42	¸	184	é	233	ù	249
+	43	¸	186	F	70	Û	219
,	44	¼	188	f	102	û	251
-	45	½	189	G	71	Ü	220
-	150	¾	190	g	103	ü	252
—	151	¿	191	H	72	Ú	218
.	46	×	215	h	104	ú	250
/	47	÷	247	I	73	V	86
:	58	0	48	i	105	v	118
:	58	1	49	ì	204	W	87
;	59	1	185	ì	236	w	119
<	60	2	50	î	206	X	88
=	61	2	178	î	238	x	120
>	62	3	51	ï	207	Y	89
?	63	3	179	ï	239	y	121
@	64	4	52	í	205	ÿ	159
[91	5	53	í	237	ÿ	255
\	92	6	54	J	74	Ý	221
]	93	7	55	j	106	ý	253
^	94	8	56	K	75	Z	90
~	95	9	57	k	107	z	122
˘	96	A	65	L	76	þ	222

{	123	a	97	l	108	þ	254
	124	À	192	M	77	Æ	198
}	125	à	224	m	109	æ	230
~	126	Â	194	N	78	Ö	214
f	131	â	226	n	110	ö	246
...	133	Ä	196	Ñ	209	Ø	216
†	134	ä	228	ñ	241	ø	248
‡	135	Å	195	O	79	Œ	140
^	136	å	227	o	111	œ	156
‰	137	Å	197	Ò	210	ß	223
•	149	å	229	ò	242		
~	152	Á	193	Ô	212		
™	153	á	225	ô	244		

Note A black rectangle indicates that the character is available only in TrueType fonts. To see how the character will appear, consult Appendix A, "ANSI Character Set," in the *Microsoft Access Language Reference*.

Storing SQL Database Login IDs and Passwords Locally

In Microsoft Access version 1.0, if you attach an SQL database table, you can choose whether you want Microsoft Access to store your login ID and password locally. If you don't, Microsoft Access prompts for your login ID and password each time you connect to the SQL database containing the table. If you want Microsoft Access to store the connection information in your Microsoft Access database so you won't have to type it each time, you can select the Save Login ID and Password Locally check box in the Attach Tables dialog box when you attach the SQL database table.

This feature is also present in Microsoft Access version 1.1. For Microsoft SQL Server, Sybase SQL Server, and ORACLE Server databases, however, your SQL database administrator can now choose to disable this feature, requiring all users to enter their login IDs and passwords each time they connect to the SQL database.

To disable the ability to store login IDs and passwords locally, your SQL database administrator must create a table called MSysConf in the SQL database. When a user connects to the SQL database, Microsoft Access looks for this table in the database, and if it finds it, queries the table. If the values in the table correctly specify that local storing of login IDs and passwords should be disabled, Microsoft Access does so regardless of whether the Save Login ID and Password Locally check box is selected. If the table isn't present or doesn't specify disabling of the feature, users can store login IDs and passwords locally.

The Structure of the MSysConf Table

The SQL database table MSysConf should have the following structure.

Column name	Data type	Allows Null?
Config	A data type that corresponds to a 2-byte integer	No
chValue	VARCHAR(255)	Yes
nValue	A data type that corresponds to a 4-byte integer	Yes
Comment	VARCHAR(255)	Yes

Note If the data source you're working with is case-sensitive, use the table and column names exactly as shown.

All users must have permission to use the SELECT statement on this table, and only the system administrator may have permission to use the DELETE statement on this table.

For the purpose of disabling password and login ID storage, the table needs to contain only one row, defined as follows.

Column name	Value	Explanation
Config	101	This is the only valid value for Microsoft Access version 1.1.
chValue	NULL	This is reserved for future use.
nValue	0 or 1	Use 0 to prevent the password and login ID from being stored; use 1 to permit password and login ID storage as in version 1.0. The default is 1.
Comment	Allow storage of passwords	

and login IDs in
Microsoft
Access.

Miscellaneous Improvements

The following improvements have been made in Microsoft Access version 1.1.

Importing and Attaching Data on Read-Only Drives

You can import and attach data on read-only drives (for example, a database on a CD-ROM drive) for FoxPro, dBASE, and Btrieve databases.

For FoxPro or dBASE databases, you must add the following entry to your MSACCESS.INI file in the [dBase ISAM] section:

INFPATH = *path*

In this entry, *path* is a valid path where the corresponding information (.INF) file can be written. Microsoft Access uses this file to keep track of information about each FoxPro or dBASE file.

For Btrieve tables, you don't have to add any information to your MSACCESS.INI file.

You cannot import or attach Paradox version 3.x data on a read-only drive, because Paradox must write the Paradox lock (.LCK) files to the same location as the database.

Importing and Attaching External Tables in Use

You can import or attach external tables in another database (for example, a Paradox, FoxPro, dBASE, or Btrieve database) while that database is in use. (The database doesn't have to be opened exclusively by Microsoft Access.)

Faster Importing of Data to Databases Opened Nonexclusively

Importing spreadsheet or text data to a Microsoft Access database that has been opened nonexclusively (so it can be used by more than one user at a time on a network) has been speeded up considerably in version 1.1. Importing to a database opened nonexclusively should be comparable in speed to importing to a database opened exclusively. (To open a database nonexclusively, clear the Exclusive check box in the Open Database dialog box.)

Easier Creation of Custom AccessWizards

If you are developing your own application, it is much easier in version 1.1 to create and install custom AccessWizards. For more information, call Microsoft End-User Sales and Service at (800) 426-9400. If you are located outside the United States, please contact your local Microsoft representative.

Ensuring That Your Microsoft Access System Is Unique

Under some circumstances, it's possible for users to gain unauthorized access to a secure system by creating another version of the same system. Microsoft Access identifies a system by combining the serial number of your Setup disks with the user and company name you specify when you create the system database (SYSTEM.MDA) during Setup. The system database defines a Microsoft Access system, which is sometimes also referred to as a work group.

A user may be able to recreate a system database by:

- Installing Microsoft Access using the same Setup disks and then specifying the same user and company name during Setup.
- Installing Microsoft Access by running Setup from the same network or CD-ROM drive and then specifying the same user and company name during Setup.
- Creating a new system database using the Microsoft Access Change Workgroup utility and then specifying the same disk serial number or personal ID number (PIN) and the same user and company name.

To ensure that your Microsoft Access system database is unique, in Microsoft Access version 1.1 you can use the Change Workgroup utility to create a system database that can be recreated only by using a PIN that you specify.

Important After recreating a system database, a user can log on as a member of the Admins group on that system and have full access to any databases or objects created on that system. Any member of a system's Admins group is always able to change permissions for the databases and objects created in that system.

To create a unique Microsoft Access system database

- 1 In the Microsoft Access group, double-click the Microsoft Access Change Workgroup icon. Microsoft Access displays the Change Workgroup dialog box.

Note If your Change Workgroup icon has been deleted, you can create your own Program Manager item by selecting New from the File menu in the Program Manager and selecting the Program Item option. In the Command Line box of the Program Item Properties dialog box, type *path* **stfsetup /w** where *path* is the path to the Setup program STFSETUP.EXE. For example, if you installed Microsoft Access in the \ACCESS directory on drive C, you'd type **c:\access\stfsetup /w** in this box.

- 2 Choose the New button.

Microsoft Access displays the Create System Database dialog box, where you can enter the information that Microsoft Access uses to identify the system database.

- 3 Type your name and company name in the dialog box.
- 4 Type a unique PIN. You can use any combination of up to 20 numbers or letters.

As long as you don't distribute the PIN, users won't be able to recreate your system, even if they specify the same name and company name during Setup or using the Change Workgroup utility.

Important Be sure to write down your PIN and keep it in a secure place. If you ever need to recreate the system database, you must supply the exact same PIN, name, and company name. If you forget or lose the PIN, you can't recover it.

- 5 In the Path box, type the path to the directory where you want to store the SYSTEM.MDA system database file.

Note Make sure that a version of SYSTEM.MDA doesn't already exist in the directory you specify. Microsoft Access won't overwrite an existing version of SYSTEM.MDA. If you want to keep the existing SYSTEM.MDA, specify a different directory in the Path box. If you want the new SYSTEM.MDA to replace the existing version, delete the existing SYSTEM.MDA before creating the new version.

- 6 Choose the OK button.

Microsoft Access creates a new SYSTEM.MDA file and modifies your MSACCESS.INI file so that the next time you start Microsoft Access, it uses this system database, including its security and option settings.

Answers to Common Questions About Microsoft Access

The following are the questions people most commonly ask about Microsoft Access. Please scan this list before you call Microsoft Product Support Services—maybe you can save yourself the phone call!

To see the answer to a particular question, click the underlined word "Answer" following the question.

Performance

1. How can I improve the performance of Microsoft Access on a computer with 4 megabytes of RAM?
[Answer](#)

Setup

2. How can I prevent earlier versions of shared dynamic-link libraries (DLLs) from causing problems with Microsoft Access?
[Answer](#)

Tables and Table Design

3. Why am I unable to define a relationship between two tables that have a common Counter field?
[Answer](#)
4. How can I change the starting value of a Counter field to a number other than 1?
[Answer](#)
5. How can I create calculated fields in tables?
[Answer](#)
6. Is data stored in a sorted order in a table? How can I view my data in a sorted order?
[Answer](#)

Queries

7. Why does closing and reopening a query sometimes change the order of the columns in the query and its dynaset?
[Answer](#)
8. How can I check for null fields using a query?
[Answer](#)
9. How can I find a subset of records that don't have matching values in two tables?
[Answer](#)

Forms

10. Do form rules override table rules?
[Answer](#)
11. When are validation rules on a form evaluated?
[Answer](#)
12. In a form or report, how can I display fields from a table or query other than the one to which my form or report is bound?
[Answer](#)
13. How can I display a record based on the value I select in a combo box?
[Answer](#)
14. How can I create my own navigation system on a form without using the navigation buttons Microsoft Access provides?
[Answer](#)

Reports

15. In a report, how can I sort the data on a field from a table other than the one on which the report is based?
[Answer](#)

16. When I print my report, every other page is blank. What causes this problem?

Answer

Expressions

17. When is it appropriate to use the ! (exclamation point) operator versus the .(dot) operator to identify objects and properties in an expression such as **Forms!**
Form1.Visible?

Answer

18. How can I refer to controls on a subform or subreport?

Answer

Macros

19. How can I prevent the AutoExec macro from running when I open a database?

Answer

Security

20. How should I plan database security?

Answer

General

21. How can I change the behavior of the insertion point when I tab into a field?

Answer

22 How can I use Microsoft Access as a dynamic data exchange (DDE) server?

Answer

Improving Performance

Q. How can I improve the performance of Microsoft Access on a computer with 4 megabytes of RAM?

If you run Microsoft Access on a computer with 4 MB of RAM, you can improve performance by altering your computer's memory configuration. This topic includes tips on altering your computer's memory configuration and suggests other ways you can enhance the relative performance of Microsoft Access on computers with 4 MB of RAM. (Using these tips may also improve performance even if you have 6 MB of RAM or more.)

Configuring Your Computer's Memory

- Don't use any of your RAM for a RAM disk.
- Use a maximum of 512 kilobytes for SmartDrive or other disk caches.
- Set your network installation to use less than 200K of RAM, if possible.
- Use a permanent Windows swap file with 32-bit access if your hard disk supports it. (This feature isn't available if you are running Microsoft Windows in standard mode, but you can use it if you are running Windows version 3.1 in 386 enhanced mode.) To set swap file options, double-click the 386 Enhanced icon in the Control Panel window, choose the Virtual Memory button, choose the Change button, and then enter the desired options in the Virtual Memory dialog box. For a description of swap files, see your Windows documentation.
- If you are using a disk-compression utility, set your temp directory and swap files to the uncompressed portion of the disk.

Running Applications

- Run Windows in standard mode. (To do this, start Windows by typing **win /s** at the command prompt.)
- Don't run several other applications that require large amounts of memory at the same time. Running even one other large application (such as a spreadsheet, word processor, or alternate desktop manager like Norton Desktop for Windows) can significantly degrade the performance of Microsoft Access.
- Minimize the number of small applications or Windows accessories (such as WordArt or Equation Editor) that you have running in the background.
- If you have a full-screen background bitmap (or "wallpaper") on your Windows desktop, replace it with a smaller bitmap or no bitmap at all. For a standard VGA display, this frees about 256K of RAM. For a 1024 x 768 256-color display, this can free about 750K. (Your actual memory savings may vary, depending on the display.)

Microsoft Access Settings

- When you open a Microsoft Access database, check the Exclusive and Read Only check boxes in the Open Database dialog box, if possible.
- Set the MaxBufferSize entry in the [ISAM] section of the MSACCESS.INI file to a value less than 512. In low-memory cases this will help, even though Microsoft Access will use less memory for buffering disk operations.

Note Casual users should avoid changing the installed settings in the [ISAM] section of MSACCESS.INI. This can produce unexpected results. However, advanced users can tune Microsoft Access performance by adding or changing entries in this section. These entries are described in the text file PERFORM.TXT, which the Microsoft Access Setup program puts in the directory where you installed Microsoft Access.

- If you don't want to use Microsoft Access Wizards, you can disable them and free over 300K of RAM by removing the following line from the [Libraries] section of your MSACCESS.INI file:
wizard.mda=ro

Background Information

The 4-MB RAM requirement for Microsoft Access is based on a 386/20 computer using the following hardware and software: MS-DOS version 5.0, unmodified Windows version 3.1, a VGA display, a mouse, and network workstation software. (This has been verified by benchmarks with Microsoft Access fully installed.) However, if your system's configuration

is more sophisticated, you will require more memory for peak Microsoft Access performance.

Keep in mind that CD-ROM device drivers, sound board drivers, screen savers, MIDI drivers, multimedia support drivers, and other drivers take extra memory. If you need to have several drivers running under Windows, Microsoft Access will require more than 4 MB of memory to run efficiently.

Shared Dynamic-Link Libraries (DLLs)

Q. How can I prevent earlier versions of shared DLLs from causing problems with Microsoft Access?

If your system has shared DLLs—DLLs used by several Microsoft Windows-based applications

—that are older than those supplied with Windows version 3.1, they can cause unexpected errors with Microsoft Access. Microsoft Access supplies the same DLL versions as Windows 3.1, and it requires these (or later) versions for correct operation; earlier versions can cause errors.

To avoid problems, make sure that the following shared DLLs are dated no earlier than 3/10/92 and are all located in your Windows System subdirectory (see "To search for an installed DLL" below).

- COMMDLG.DLL
- CTL3D.DLL
- OLECLI.DLL
- OLESRV.DLL
- DDEML.DLL
- SHELL.DLL
- VER.DLL
- DBNMP3.DLL
- W3DBLIB.DLL
- NETAPI.DLL

Additional Information

Microsoft Access and Windows copy shared DLLs into the Windows System subdirectory, and Microsoft Access looks for the DLLs there first. If it doesn't find one or more of the shared DLLs in this subdirectory, it looks in the directory where Microsoft Access is installed, and then in the current directory.

Often, Microsoft Access finds outdated or incorrect DLLs because third-party software with older or foreign DLLs has been installed or reinstalled, overwriting the correct DLLs. Some third-party software packages copy DLLs into directories other than the Windows System directory, creating multiple copies and making the problem harder to track and resolve.

The Microsoft Access Setup program won't install DLLs previously loaded by another software package; it installs only shared DLLs not already loaded. Consequently, if you run the Setup program and another application is running and using an outdated DLL, the Setup program won't install the newer version of the DLL.

If you start Microsoft Access and it fails to locate a shared DLL, or if it detects an earlier version of one, it displays a message such as "Couldn't open file <DLL file name>" or "Outdated <DLL file name> file. Please reinstall Microsoft Access." You must locate and rename or delete any earlier version of the DLL and then install the new version.

To search for an installed DLL

This procedure searches for COMMDLG.DLL. You can use the same procedure for any shared DLL.

- 1 Exit Windows.
- 2 Type the following commands at the command prompt to locate all copies of the file on your default drive. You need to do this for each drive specified in the **path** command in your AUTOEXEC.BAT file, the drive where your Windows directory is located, and the drive where Microsoft Access is installed.

```
cd \
```

```
dir COMMDLG.DLL /s
```

Note You must have MS-DOS version 5.0 or later to use the **/s** option of the **dir** command.

Only one copy of COMMDLG.DLL should be on your disk, and it should be in the Windows System directory. If other copies have been installed elsewhere by other applications, you

should rename or delete them.

If you have deleted all duplicates of COMMDLG.DLL and Microsoft Access still displays an error message, you can install a new COMMDLG.DLL file.

To install a new DLL

This procedure installs a new COMMDLG.DLL file. You can use the same procedure for any shared DLL.

- 1 Copy COMMDLG.DL_ from Disk 1 of the Microsoft Access package to your hard disk. Do not copy COMMDLG.DL\$ from the Microsoft Access disks. This file can be decompressed only by using the Microsoft Access Setup program.
- 2 Copy EXPAND.EXE from Disk 3 of your Windows version 3.1 package to your hard disk. COMMDLG.DL_ can be expanded using this expand utility.
- 3 Close all running applications, and exit Windows.
- 4 At the command prompt, type the following line:

c:\path1\EXPAND path2\COMMDLG.DL_ c:\windows\system\COMMDLG.DLL

where *path1* is the path to the EXPAND.EXE file on your hard disk (the directory into which you copied the file) and *path2* is the path to the COMMDLG.DL_ file on your hard disk.

Important The shared DLL can't be in use by an application when you install the new version. If you are on a multiuser network, arrange a time to install the shared DLL when it isn't being used by applications on the network.

Displaying Fields from Another Table or Query

Q. In a form or report, how can I display fields from a table or query other than the one to which my form or report is bound?

Frequently, you'll want to display information in a form or report from a table other than the one to which your form or report is bound. Or you might want to update the value of one or more controls based on a combo box or list box selection. You can accomplish this in several ways:

- With forms, you can take advantage of dynamic lookup.
- With forms and reports, you can use the Column property of a combo box or list box.
- With forms and reports, you can display values using the **DLookup** function.

Using Dynamic Lookup

Examples

Dynamic lookup is a very powerful feature of Microsoft Access that allows you to change a foreign key value and have information about the key updated automatically. The dynamic lookup technique works in a query or in a form based on a query.

To use dynamic lookup, you must be working with tables that have a one-to-many relationship or a one-to-one relationship. Most often, you'll take advantage of dynamic lookup with tables in a one-to-many relationship. (For a description of using dynamic lookup with tables that have a one-to-one relationship, see the note below).

For example, the Categories and Products tables in the Northwind Traders sample database have a one-to-many relationship. Each category can contain many products. The Products table contains the field Category ID, which is the foreign key that identifies the category for a product.

When the value of a foreign key (from the table on the "many" side of the relationship) is changed in a record in a multiple-table query, Microsoft Access automatically performs dynamic lookup, finding and displaying the associated values from the table on the "one" side of the relationship. Any fields in the record from the table on the "one" side of the relationship now display values associated with the changed key value.

For example, if you change the category ID from the Products table in a record containing values from the Products and Categories tables, this record now displays the appropriate information for the new category.

If no matching information is found, Microsoft Access displays an error message when the focus leaves the record. You must enter a valid value for the foreign key.

When dynamic lookup updates data, Microsoft Access automatically recalculates any totals or expressions that are dependent on the updated data.

To see examples of using dynamic lookup in a query and in a form, click "Examples" at the beginning of this topic.

For another example, see the Products and Suppliers form and the underlying Products and Suppliers query in the Northwind database. This form and query are described in Chapter 10, "Creating Forms Based on More than One Table," in the *Microsoft Access User's Guide*.

Note To take advantage of dynamic lookup with tables that have a one-to-one relationship, make the join between the tables a left outer join with the updatable table on the left. For more information on what is and what is not updatable, search Microsoft Access Help for "updating underlying tables."

Close

Copy

Print

Dynamic Lookup Examples

Example 1: Create a query that uses dynamic lookup

- 1 Start Microsoft Access and open NWIND.MDB.
- 2 Create a new query, and add the Products and Categories tables.
- 3 Add all the fields from the Products table.
- 4 Add all the fields except Category ID from the Categories table.
- 5 Save your query as "Dynamic Lookup Example".
- 6 Run the query.

To verify that the query will perform dynamic lookup, move to the end of the query's dynaset and start a new record. Type a value in the Category ID field that already exists in the Categories table (for example, BEVR). When you type a valid value for Category ID, Microsoft Access fills in the rest of the information about the category (information from the Categories table such as the category name and a description) accordingly.

Example 2: Create a form that uses dynamic lookup

- 1 Create a new form based on the Dynamic Lookup Example query (created in the first example) using the Single-Column FormWizard.
- 2 When the FormWizard asks which fields you want on the form, select the fields from the Categories table (Category Name, Description, Picture) first, and then select the fields from the Products table.

The FormWizard will place the controls for the Categories fields together at the top of the form and the controls for the Products fields below them. This layout will help you see the results of the dynamic lookup easily.

- 3 When the FormWizard has finished designing the form, switch to Design view.
- 4 Delete the text box control for the Category ID, and replace it with a new combo box bound to the Category ID field using the following property settings:

ControlSource Products.Category ID
RowSource Categories
ColumnCount 1

Because the Category ID field is the first column in the Categories table, setting the ColumnCount property to 1 will make the combo box display the category ID.

- 5 Switch to Form view.

Now when you enter new records and select a category from the Category ID combo box, the Category Name, Description, and Picture information is automatically displayed.

Note If you included the primary key (Category ID) from the table on the "one" side of the relationship (Categories) in the Dynamic Lookup Example query, you won't be able to add new records to the query or change the key value. Information displayed from the "one" side of this relationship is not updatable.

Using the Column Property

You can use the Column property of a multiple-column combo box or list box in an expression to display one column from the current combo box or list box selection in a text box. Microsoft Access automatically updates the text box as the focus changes from row to row in the combo box or list box.

To display the value in a column of a multiple-column combo box or list box in a text box

- 1 Create a form or report based on the desired table or query.
- 2 In Design view, create a multiple-column combo box or list box that retrieves information from more than one field.

For example, you might use the following SQL statement as the RowSource setting for a combo box to display information from several columns in the Categories table (from the Northwind Traders sample database):

```
SELECT [Category ID], [Category Name], [Description] FROM Categories ORDER BY [Category Name];
```

- 3 Set the ColumnCount property to the number of columns in the combo box or list box (in this example, 3), and set the ColumnWidths property to appropriate sizes for the columns in this combo box or list box.

When you switch to Form view, the combo box or list box will display the data from the selected fields.

- 4 Create a text box control, and make it a calculated control by setting its ControlSource property to an expression like this:

```
=[controlname].Column(1)
```

In this expression, *controlname* is the name of the combo box or list box as set in that control's ControlName property. Column(1) refers to the second column in this combo box or list box. (Use Column(0) to refer to the first column, Column(1) to refer to the second column, and so on.) In the example, Column(1) would refer to [Category Name] –the second column in the combo box.

- 5 Switch to Form view, and make a selection in the combo box or list box.

The text box displays the category name for the item you selected. (Note that because the text box is a calculated control, it is read-only in Form view.)

Tip You can also use the Column property with the SetValue macro action to set the value of a bound control to the value of a column in the combo box or list box. Use an expression like that in the example above in the Expression argument of the SetValue action. Attach the macro to the AfterUpdate property of the combo box or list box.

Using the DLookup Function

You can use the **DLookup** function in a form or report to display values from a table or query other than the one to which a form or report is bound.

The **DLookup** function uses the following syntax:

DLookup(*expression, domain [, criteria]*)

The **DLookup** function returns the value in a field (specified by the *expression* argument) from a particular set of records called the domain (specified by the table or query identified in the *domain* argument or by an SQL statement). You can specify *criteria* for the domain. To make the domain dependent on one or more values contained in controls on a form or report, refer to the controls in the *criteria* argument.

The following example looks in the Employees table (the domain) and returns the last name of the employee whose employee ID matches the value in the Employee Number control on the Order Information form.

```
=DLookup("[Last Name]", "Employees", "[Employee ID] = Forms![Order Information]![Employee Number]")
```

Note If the control in this case, Employee Number

is on the current form or report, the **Forms!***formname* or **Reports!***reportname* identifier isn't required. However, it is often a good idea to use the fully qualified name for a control in a **DLookup** function to make sure the *criteria* argument specifies the correct control or field, especially if there is a chance that the form and the table containing the value you are looking up have controls or fields with the same name.

You can use an expression containing the **DLookup** function in the ControlSource property of a control to display the result of **DLookup** in this control. For example, suppose you have an Order Information form based on a query that contains an Employee Number field but not a Last Name field for the employee. On this form you could create a control called Salesperson's Last Name and set its ControlSource property to the expression above so that in Form view this control would display the employee's last name, based on the value in the Employee Number control.

DLookup returns one value from a single field even if more than one record satisfies the *criteria* argument. If several records satisfy *criteria*, **DLookup** returns the value from the first record that satisfies *criteria*. If no record satisfies *criteria* or if the domain contains no records, **DLookup** returns **Null**.

Column Order in Queries

Q. Why does closing and reopening a query sometimes change the order of the columns in the query and its dynaset?

The order of columns in a query is determined by the stored query definition, as defined by Microsoft Access. For example, Sort fields are moved to the leftmost columns in the QBE grid. When you reopen a query in Design view or look at the query's dynaset, you may notice that the columns aren't in the same order in which you put them in the QBE grid.

Microsoft Access places columns in the following order in the QBE grid and in the dynaset, moving from left to right:

1. Columns used for sorting
2. Columns shown in the dynaset
3. Columns not shown in the dynaset but used as criteria for the dynaset (these appear only in the QBE grid)

If you want to save the order of the columns in the dynaset, create a form based on the query and arrange the columns in the datasheet for this form in the order you want. From the File menu, choose Save Form. When you open the form in Datasheet view, the columns will be in the order you specified.

Displaying a Record Based On the Value Selected in a Combo Box

[Example](#)

Q. How can I display a record based on the value I select in a combo box?

You can create a macro containing the FindRecord action and attach it to the AfterUpdate property of the combo box. Set the Find What argument of the FindRecord action to an expression like this:

=controlname

where *controlname* is the name of the combo box control.

The Suppliers form in the Northwind Traders sample database shows an example of how to do this. On this form, the Find Company macro attached to the AfterUpdate property of the Company Pick List combo box (at the bottom of the form) finds the record for the company name you select in the combo box.

For a complete description of this macro and its use with the Suppliers form, see Chapter 22, "Using Macros with Forms," in the *Microsoft Access User's Guide*.

You could also use an Access Basic function to display the record in the Suppliers form. To see how, click "Example" at the beginning of this topic.

Close

Copy

Print

Example: Using Access Basic to Display a Record Based On the Value Selected in a Combo Box

To display the Suppliers form record for the company selected in the Company Pick List combo box, you could attach the following function to the AfterUpdate property of the combo box (replacing the Find Company macro). To do this, type **=FindCompany()** in the AfterUpdate property box, and paste the code below into a module in the Northwind database.

Using the function instead of a macro means that the form doesn't have to contain two separate controls with the company name (the pick list and the Company Name control). You can remove the Company Name control and replace it with the unbound Company Pick List control (renaming the pick list to Company Name if you want).

Function FindCompany () As Integer

```
' Description:  When called from the AfterUpdate property of a
' combo box containing company names, this function tries to find
' the matching company's record.  If the matching record is found,
' the function goes to it.  If the record isn't found, the focus
' stays on the current record.
' Accepts:      (nothing)
' Returns:      True  - The company was found.
'              False - The company wasn't found.

Dim Criteria As String      ' This is the argument to the FindFirst
method.
Dim MyDyna As Dynaset      ' Dynaset used to search.
Dim CompanyName As String  ' The name of the company to search for.

Set MyDyna = Screen.ActiveForm.Dynaset

' Build the criteria.
CompanyName = Chr$(34) & Screen.ActiveControl & Chr$(34)
Criteria = "[Company Name]=" & CompanyName

' Perform the search.
MyDyna.FindFirst Criteria

If MyDyna.NoMatch Then
    MsgBox "Could not find the company: " & CompanyName
    FindCompany = False
Else
    ' Synchronize the form's record to the dynaset's record.
    Screen.ActiveForm.Bookmark = MyDyna.Bookmark
    FindCompany = True
End If

MyDyna.Close

End Function
```

Creating a Custom Navigation System

Q. How can I create my own navigation system on a form without using the navigation buttons Microsoft Access provides?

Using Access Basic, you can replace the navigation buttons (sometimes called "VCR buttons") in the lower-left corner of the window with custom navigation buttons. You might want to do this if you turned off the display of the horizontal scroll bar for the form (using the ScrollBars property), or if you want to create navigation buttons with different shapes or containing text.

To create custom First, Last, Next, and Previous buttons on a form to replace the navigation buttons

These command buttons can be used in the same way as the First, Last, Next, and Previous navigation buttons.

- 1 Create four command buttons, and place them on the form in an appropriate place.
- 2 Make the following changes to the command button properties.

Command button 1

ControlName	First
Caption	First
OnPush	=GoToRecord("First")

Command button 2

ControlName	Last
Caption	Last
OnPush	=GoToRecord("Last")

Command button 3

ControlName	Next
Caption	Next
OnPush	=GoToRecord("Next")

Command button 4

ControlName	Previous
Caption	Previous
OnPush	=GoToRecord("Previous")

- 3 Create a new module or open an existing module, and copy or type the GoToRecord() function in the module.
- 4 From the File menu, choose Save to save the function in the module.
- 5 Switch back to the form in Form view, and try pushing each button.

Close

Copy

Print

Example: Creating a Custom Navigation System Using an Access Basic Function

This function uses the ActiveForm and FormName properties together (Screen.ActiveForm.FormName) to return the name of the active form. This means that the function isn't form-specific, so you can use it and the four buttons on any form. For the function to work correctly, the control names of the buttons must be identical to the names specified in this example.

```
Function GoToRecord (ByVal Direction As String) As Variant

' Description: This function moves the focus of a [form] record to
' the specified record.
' Accepts:      Direction - String that is the record you want to
' go to: "First", "Last", "Previous", or "Next"
' Returns:      Null      - Direction is invalid.
'              Number > 0 - Direction is valid.

Const CASEINSENSITIVE = 1
Dim X As Integer, MyControlName As String

X = (InStr(1, "First Last PreviousNext ", Direction,
CASEINSENSITIVE) + 7) \ 8
If X <> 0 Then
    X = Choose(X, A_FIRST, A_LAST, A_PREVIOUS, A_NEXT)
    On Error Resume Next
    DoCmd GoToRecord A_FORM, Screen.ActiveForm.FormName, X
    On Error GoTo 0
    GoToRecord = X
Else
    MsgBox "Invalid direction argument.", , "GoToRecord"
    GoToRecord = Null
End If

End Function
```

Referring to Controls on a Subform or Subreport

Q. How can I refer to controls on a subform or subreport?

After you create a subform or subreport, you can refer to the values of the subform or subreport's controls in an expression. For example, you can attach a macro to a button on a form that uses an expression to set the value of a control on a subform. (Note that when you refer to a subform from the main form, you are actually referring to the subform control ■ a subform is a type of control on the main form.)

To refer to the value of a control on a subform or subreport in an expression, type the identifier for the Form or Report property that represents the subform or subreport, followed by the ! operator and the control identifier. For example, you can use the following expression in the Expression argument of a SetValue action to create a macro that increases the Unit Price value on the Orders Subform by 10 percent:

Forms![Orders]![Orders Subform].Form![Unit Price]*1.1

If the macro is attached to a button on the Orders form, you can use this expression:

[Orders Subform].Form![Unit Price]*1.1

If you are referring to a control on a subform or subreport from another control on the same subform or subreport, you don't have to enter the Form or Report property identifier. For example, to refer to the Unit Price value on the Orders Subform in a macro attached to a button on the Orders Subform, you can enter:

[Unit Price]

You can use the Form or Report property to display on the subform's or subreport's parent (main) form or report a value calculated in a hidden control on the subform or subreport. For example, enter the following expression as the ControlSource property for the Subtotal control on the Orders form to display a value calculated in the hidden Order Subtotal control on the Orders Subform:

=[Orders Subform].Form![Order Subtotal]

To refer to the value of a control on the parent (main) form or report from a control on a subform or subreport, use the Parent property. For example, use the following expression to refer to the Customer ID field of a form in a control on a subform of this form:

=Parent![Customer ID]

To refer to a control on a nested subform (a subform on a subform) or subreport, use the following syntax:

Forms!*[name of main form]*!*[controlname of subform]*.Form!*[controlname of nested subform]*.Form!*[name of control on nested subform]*

Note You can use the preceding syntax to refer to controls on subforms or subreports in expressions in forms or reports or in macros. However, in Access Basic, you can use this syntax to refer only to controls on subforms, not to controls on subreports.

Bypassing the AutoExec Macro

Q. How can I prevent the AutoExec macro from running when I open a database?

Hold down the Shift key when you open the database, and Microsoft Access won't execute the AutoExec macro.

Defining Relationships Between Counter Fields

Q. Why am I unable to define a relationship between two tables that have a common Counter field?

Only a one-to-one relationship can be defined between two Counter fields fields. To create a one-to-one relationship between two tables, choose Relationships from the Edit menu, select the two tables, and then choose One in the Type group.

If you want to define a one-to-many relationship, the matching field on the "many" side should be defined as Long Integer if the matching field on the "one" side is a Counter. To define a field in a table as Long Integer, select the Number data type for the field in the table's Design view, and select Long Integer for the Field Size field property.

Database Security

Q. How should I plan database security?

There are two things to remember when dealing with security:

- Plan your security scheme from the beginning, as you're designing your database. Planning a good security scheme from the beginning will ensure you a minimum of work in the future.
- Assign permissions to groups, NOT to users. This is very important. If you create appropriate, well-thought-out groups, a user will have permissions to particular objects based on all the groups the user is a member of. (For example, if a user is a member of Group A, which does not have permissions to an object, but also is a member of Group B, which does have the permissions to the object, the user will have Group B's permissions to the object.)

You will find that maintenance work will decrease significantly if you plan your groups carefully. A typical database setup should have a maximum of four to five groups. This isn't a maximum number of groups limitation for Microsoft Access, merely a guideline for easier security management.

To give a group or user permissions to more than one object at a time, you must select each object individually and then assign permissions.

User Name Case Sensitivity

User and group names aren't case-sensitive when you are prompted to type them to start a Microsoft Access session. However, if you are recreating a user account or group, you must match the case of each letter in the name exactly because of the way the system stores the account information.

Note Passwords are case-sensitive at all times.

For more information on security, see Chapter 25, "Administering a Database System," in the *Microsoft Access User's Guide*.

Creating a Subtract Query

Q. How can I find a subset of records that don't have matching values in two tables?

You can create a "subtract" query, which compares two tables and returns a dynaset that includes only those records from the first table that do not have matching records in the second table.

For example, suppose you have two tables joined on a common field (Customer ID). You want to find the records in the first table that do not have a matching Customer ID value in the second table. To return a dynaset containing only the records that don't match on the joined field, you must create a query to join the two tables and subtract the matching records.

To create a subtract query

- 1 In Design view, create a new query, and add the two tables.
For example, to find out which customers in the Northwind Traders database haven't placed orders, you would add the Customers and Orders tables.
- 2 If necessary, join the two tables on the appropriate field.
In the example, Microsoft Access joins the tables automatically on the Customer ID field because there is an underlying relationship between them.
- 3 Double-click the join line.
Microsoft Access displays the Join Properties dialog box.
- 4 Depending on the type of join you want, choose option 2 or option 3.
Use the join information in the dialog box to decide which type of join you want. For this example, you want all the records from the first table (Customers) and only those records that match from the second table (Orders), so you would choose option 2.
- 5 Choose the OK button.
- 6 In the Query window, drag the joined field from the first table to the QBE grid, type **Is Null** in the Criteria row, and clear the check box in the Show row (so that Microsoft Access won't display this field in the query's dynaset).
In the example, you would drag the Customer ID field from the Orders table.
- 7 From the first table, drag any other fields you need, such as the primary key field, to the QBE grid.
To see that the values don't match, drag the Customer ID fields from both tables to the QBE grid.
- 8 Run the query.
The Is Null criterion ensures that the dynaset contains only those records from the first table (the Customers table) that don't have a match on the joined field (Customer ID) in the second table (the Orders table).

The following is an example SQL statement for the same query described above: Using the Orders and Customers tables from the Northwind database, it returns all the customers who have not placed an order.

```
SELECT DISTINCTROW Customers.[Company Name]
FROM Orders, Customers,
Customers LEFT JOIN Orders
ON Customers.[Customer ID] = Orders.[Customer ID]
WHERE ((Orders.[Customer ID] Is Null));
```

Note If you want to paste this SQL statement into the SQL dialog box, use the layout exactly as shown.

Using Microsoft Access As a DDE Server

Q. How can I use Microsoft Access as a DDE Server?

Microsoft Access supports dynamic data exchange (DDE) as both a destination (client) application and a source (server) application. As a DDE server, Microsoft Access supports the following topics:

- The System topic
- The name of a database
- The name of a table
- The name of a query
- A Microsoft Access SQL statement

Each DDE conversation is established on a particular topic, usually a data file, with the conversation thereafter limited to the data items associated with that topic. For example, if you are running Microsoft Word for Windows and want to insert data from a particular Microsoft Access database into a Word for Windows document, you need to initiate a DDE conversation with Microsoft Access by opening a channel and specifying the database name as the topic. You can then use that channel to obtain data from the database.

Once you have established a DDE conversation, you can use the **DDEExecute** statement to send a command from the client to the server application. When used as a DDE server, Microsoft Access recognizes any of the following as a valid command:

- The name of a macro in the currently open database.
- Any action that you can execute in Access Basic using the **DoCmd** statement.
- The OpenDatabase and CloseDatabase actions, which are used only for DDE operations. (For an example of how to use these actions, see the example below.)

Note When you specify an action as a **DDEExecute** command, the action and any arguments follow the **DoCmd** syntax and must be enclosed in brackets ([]). However, applications that support DDE don't recognize intrinsic constants in DDE operations. Also, string arguments must be enclosed in quotation marks if the string contains a comma. Otherwise, quotation marks aren't required.

The following example shows how you can create a Word for Windows WordBasic macro that uses Microsoft Access as a DDE server. (For this example to work, Microsoft Access must either be running or be listed in the **path** command in your AUTOEXEC.BAT file.)

```
Sub MAIN
' Using the System topic, open the NWIND.MDB database.
' The database must be open before you can use other DDE topics.
Chan1 = DDEInitiate("MSAccess", "System")
DDEExecute Chan1, "[OpenDatabase C:\ACCESS\NWIND.MDB]"

' Get all the data from the Orders and Products query.
Chan2 = DDEInitiate("MSAccess", "NWIND;QUERY Orders and Products")
MyData$ = DDERequest$(Chan2, "All")
DDETerminate Chan2

' Close the database.
DDEExecute Chan1, "[CloseDatabase]"
DDETerminate Chan1

' Insert the data into a text file.
Open "MYDATA.TXT" For Append As #1
Print #1, MyData$
Close #1
End Sub
```

For information on using Microsoft Access as a DDE client, see Chapter 13, "Using Pictures, Graphs, and Other Objects," in the *Microsoft Access User's Guide* or Chapter 9, "Dynamic Data Exchange," in *Introduction to Programming*.

The System Topic

The System topic is a standard topic for all Windows-based applications. It returns information about the topics supported by the application.

The System topic supports the following Microsoft Access data items.

Item	Returns
SysItems	A list of items supported by the System topic in Microsoft Access.
Formats	A list of the formats Microsoft Access can copy onto the Clipboard.
Status	"Busy" or "Ready".
Topics	A list of all open databases.

Example

```
' In a WordBasic macro, initiate a DDE conversation with Microsoft Access.
Chan1 = DDEInitiate("MSAccess", "System")
' Request a list of topics supported by the System topic.
Res$ = DDERequest$(Chan1, "SysItems")
' Run the OpenDatabase action to open NWIND.MDB.
DDEExecute Chan1, "[OpenDatabase C:\ACCESS\NWIND.MDB]"
```

The *database* Topic

The *database* topic is the file name of an existing database. You can type just the base name (NWIND) or its full path and .MDB extension (C:\ACCESS\NWIND.MDB). After you initiate a DDE conversation with the database, you can request a list of the objects in that database.

Note You can't use DDE to query the system database (SYSTEM.MDA).

The *database* topic supports the following items.

Item	Returns
TableList	A list of tables.
QueryList	A list of queries.
FormList	A list of forms.
ReportList	A list of reports.
MacroList	A list of macros.
ModuleList	A list of modules.

Example

```
' In a WordBasic macro, initiate a DDE conversation with NWIND.MDB.
' Make sure the database is open.
Chan2 = DDEInitiate("MSAccess", "Nwind")
' Request a list of forms in NWIND.MDB.
Res$ = DDERequest$(Chan2, "FormList")
' Run the OpenForm action and arguments to open the Employees form.
DDEExecute Chan2, "[OpenForm Employees,0,,1,0]"
```

The TABLE *tablename*, QUERY *queryname*, and SQL *sqlstring* Topics

These topics use the following syntax:

*database*name; TABLE *tablename*

*database*name; QUERY *queryname*

*database*name; SQL [*sqlstring*]

Argument	Description
<i>database</i> name	The name of the database that the table or query belongs to or that the SQL statement applies to, followed by a semicolon (;). The database name can be just the base name (NWIND) or its full path and .MDB extension (C:\ACCESS\NWIND.MDB).

tablename The name of an existing table.

queryname The name of an existing query.

sqlstring A valid SQL statement up to 255 characters long, ending with a semicolon. To exchange more than 255 characters, omit this argument and instead use successive **DDEPoke** statements to build an SQL statement.

For example, the following WordBasic code uses **DDEPoke** to build an SQL statement and then request the results of the query:

```
Chan1 = DDEInitiate("MSAccess", "NWIND;SQL")
DDEPoke Chan1, "SQLText", "SELECT *"
DDEPoke Chan1, "SQLText", " FROM Orders"
DDEPoke Chan1, "SQLText", " WHERE [Order Amount] > 1000;"
Res$ = DDERequest$(Chan1, "NextRow")
DDETerminate Chan1
```

The following table lists the valid items for the TABLE *tablename*, QUERY *queryname*, and SQL *sqlstring* topics.

Item	Returns
All	All the 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	A two-row list of field names (first row) and their data types (second row). These are the values returned and the data types they represent: 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 (integral date, fractional time) 9 Binary data, 255 bytes maximum 10 <u>ANSI</u> text, not case-sensitive, 255 bytes 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 for PrevRow fails.
FirstRow	The data in the first row of the table or query.
LastRow	The data in the last row of the table or query.
FieldCount	The number of fields in the table or query.
SQLText	An SQL statement representing the table or query. For tables, this item returns an SQL statement in the form "SELECT * FROM <i>table</i> ;"
SQLText;n	An SQL statement, in <i>n</i> -character chunks, representing the table or query, where <i>n</i> is an integer up to 255. For example, suppose a query is represented by the following SQL statement:

```
"SELECT * FROM Orders;"
```

The item "SQLText;7" would return the following tab-delimited chunks:

```
"SELECT "  
"* FROM "  
"Orders;"
```

Example

```
Sub MAIN  
  ' In a WordBasic macro, get data from the Categories table,  
  ' Catalog query, and Orders table in NWIND.MDB.  
  ' Make sure the database is open.  
  Chan1 = DDEInitiate("MSAccess", "NWIND;TABLE Categories")  
  Chan2 = DDEInitiate("MSAccess", "NWIND;QUERY Catalog")  
  Chan3 = DDEInitiate("MSAccess", "NWIND;SQL SELECT * FROM Orders")  
  
  Res1$ = DDERequest$(Chan1, "All")  
  Res2$ = DDERequest$(Chan2, "FieldNames;T")  
  Res3$ = DDERequest$(Chan3, "FieldNames;T")  
  
  DDETerminate Chan1  
  DDETerminate Chan2  
  DDETerminate Chan3  
  
  ' Insert the data into a text file.  
  Open "MYDATA.TXT" For Append As #1  
  Print #1, Res1$  
  Print #1, Res2$  
  Print #1, Res3$  
  Close #1  
End Sub
```

ANSI (American National Standards Institute)

The ANSI character set is an 8-bit set used by Microsoft Windows that enables you to represent up to 256 characters (0-255) using your keyboard. The *ASCII character set* is a subset of the ANSI set.

data source

An *SQL database*, such as a Microsoft or Sybase SQL Server or ORACLE Server database, and the information needed to get to that database. For example, a SQL Server data source is the SQL Server database, the server on which it resides, and the network used to access that server. You create a data source for each SQL database whose data you want to import or attach to or export from Microsoft Access.

ODBC driver

A dynamic-link library (DLL) that Microsoft Access uses to connect to an *SQL database*, such as a Microsoft or Sybase SQL Server or ORACLE Server database. Each type of SQL database requires a different ODBC driver.

data file

A file containing records that Microsoft Word for Windows uses to create form letters, mailing labels, and other merged documents. The data file (also called a data source) contains field names in the first record (the *header record*) and the corresponding data for these fields in the succeeding records (the *data records*). You create a *main document* in Word for Windows. The fields in this document must match the fields in the data file's header record. When you use the Word for Windows mail merge feature (also called "print merge"), the data from the data records in the data file is merged into the corresponding fields in the main document.

main document

A document created in Microsoft Word for Windows that contains fields where you want to insert (merge) data from a source such as a database table. You create a *data file* containing the data you want to merge and then use the Word for Windows mail merge feature (also called "print merge") to merge this data into the main document's fields.

header record

The first record in a *data file*. This record contains the names of the fields in the file. When you use the Microsoft Word for Windows mail merge feature (also called "print merge"), the fields in the *main document* are matched with the field names in the header record. This is also known as the header row.

data records

The records in a *data file* containing the data. You use the Microsoft Word for Windows mail merge feature (also called "print merge") to merge this data into the fields in the *main document*.

SQL database

A database for which Microsoft Access supplies an *ODBC driver* that you can use for importing, attaching, or exporting data. Currently, Microsoft Access supplies ODBC drivers for Microsoft and Sybase SQL Server and ORACLE Server databases. Other vendors may supply ODBC drivers for other types of databases. You should contact the driver vendor to see if a particular ODBC driver has been tested and verified for use with Microsoft Access.

foreign key

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

group

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

work group

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

permissions

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

exclusive

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

user account

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

one-to-one relationship

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

one-to-many relationship

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

left (outer) join

A join that includes all the records from the first (left-hand) table, even if there are no matching values for records in the second (right-hand) table.

Counter field

A field that assigns the next consecutive number for each successive record added in the same field. A Counter field always takes the Counter data type.

database objects

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

secure system

A Microsoft Access system in which users log on with a user name and password and in which access to database objects is restricted according to permissions specified for user accounts and groups.

append query

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

calculated field

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

case-sensitive

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

dynaset

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

expression

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

insertion point

The place on screen where text is inserted when you type. Its location is indicated on-screen by a blinking vertical line.

null field

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

QBE (Query by Example) grid

The grid that appears in the lower portion of the Query window. You use it to define a query.

Field:	LastName	FirstName	Order Amount
Table:	Employees	Employees	Orders
Total:	Group By	Group By	Sum
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:			>1000
or:			

Sorting and Grouping box

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

Sorting and Grouping	
Field/Expression	Sort Order
Country	Ascending
Salesperson	Ascending
Shipped Date	Ascending
Order ID	Ascending
Group Properties	
Group Header	Yes
Group Footer	Yes
Group On	Each Value
Group Interval	1

validation rule

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

SQL string/statement

An expression that defines an SQL command, such as SELECT, UPDATE, or DELETE, and includes clauses such as WHERE and ORDER BY. SQL strings/statements are typically used in queries and in aggregate functions.

calculated control

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

join

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

- An *equi-join*, which combines records from two tables whenever there are matching values in the joined fields.
- An *outer join*, which includes all the records from one table and only those records from the second table in which the joined fields are equal. In the Join Properties dialog box, you can specify which table will contribute all of its records.
- A *self-join*, which is a table joined to itself. A self-join is useful when you ask hierarchical questions, such as a query that shows all the managers and the people who report to them.

RAM disk

A portion of memory that is used as if it were a physical disk. RAM disks are much faster than physical disks because your computer can read information from memory faster than from a physical disk. However, information on a RAM disk is lost when you turn off or reboot your computer. Also known as RAM drive or virtual drive.

SQL (Structured Query Language)

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

PIN (personal identification number)

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

primary key

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

Changing the Starting Value of a Counter Field

Q. How can I change the starting value of a Counter field to a number other than 1?

There are two ways to change the starting value of a Counter field:

- One way is to create a new, temporary table with just one field, a Number field, that has the same name as the Counter field in the original table. Enter a value in the temporary table. This value should be 1 less than the desired starting value for the Counter field in the original table. Create an append query to append the record from the temporary table to the original table. After you append the record, delete the temporary table, and then delete the dummy record from the original table.

Note Don't compact the database before adding the first record to the original table; if you do, the Counter value will be reset to start at 1.

- For the second method to work, there must be at least one record in the table you want to set the Counter value for. This can be a temporary record. Create an append query that will append the original table to itself. Add the Counter field to the QBE grid in this query and enter an expression that contains a number in the Field cell of the counter column that is 1 less than the desired starting value. Run the query, and then close it without saving it. Delete the temporary record from the table if needed.

Creating Calculated Fields in Tables

Q. How can I create calculated fields in tables?

In general, it helps to think of queries as tables. You can use a query whenever you can use a table. To add a calculated field to a query, open the query in Design view and enter an expression in the Field row in the QBE grid. If you're familiar with Microsoft SQL Server terminology, using a query in this way is very similar to creating a VIEW. Unlike most implementations of views, however, Microsoft Access views are generally updatable even if they involve joins from disparate data sources.

To learn more about when queries are updatable, see Chapter 6, "Designing Select Queries," in the *Microsoft Access User's Guide*.

Viewing Table Data in Sorted Order

Q. Is data stored in a sorted order in a table? How can I view my data in a sorted order?

No. Data is stored in the order in which it is entered. To view data in sorted order, create a query, add the field you want to sort on to the QBE grid, and then select Ascending or Descending in the Sort row for that field.

For more information, see Chapter 5, "Query Basics," in the *Microsoft Access User's Guide*.

Checking for Null Fields

Q. How can I check for null fields using a query?

In the QBE grid, place the expression Is Null in the Criteria row under the field you want to check for null values.

Form Rules and Table Rules

Q. Do form rules override table rules?

Yes. Note, however, that when you create a control on a form by dragging a field from the field list, the control inherits the field's `ValidationRule` property setting, which is defined in the table's Design view.

For information on setting field properties, see Chapter 3, "Changing and Customizing Tables," in the *Microsoft Access User's Guide*. For information on setting control properties, see Chapter 9, "Designing Forms."

Form Validation Rules

Q. When are validation rules on a form evaluated?

Microsoft Access validation rules operate at the field level. A validation rule is evaluated only when data is entered or edited in a field and the focus moves to another field or record. If the field is left unchanged, Microsoft Access doesn't evaluate the validation rule. Microsoft Access also validates a field on a form when you switch views or close the form.

If you want to perform record-level validation, you can attach a macro to the form's BeforeUpdate property. For more information, see Chapter 22, "Using Macros with Forms," in the *Microsoft Access User's Guide*.

Using Identifier Operators

Q. When is it appropriate to use the ! (exclamation point) operator versus the . (dot) operator to identify objects and properties in an expression such as Forms! Form1.Visible?

A good rule of thumb is to use the ! operator to the left of the name of anything you create (such as a form or a control on the form). Use the .(dot) operator to the left of the name of anything defined by Microsoft Access (such as a property).

For more information on expression syntax, see Appendix C, "Expressions," in the *Microsoft Access User's Guide*.

Changing the Behavior of the Insertion Point

Q. How can I change the behavior of the insertion point when I tab into a field?

By default, if you use the arrow keys to move to the next field, the entire field is selected. To have the insertion point move to the next character instead of the next field, choose Options from the View menu, select Keyboard, and change the setting for Arrow Key Behavior from Next Field to Next Character.

Sorting Data in a Report

Q. In a report, how can I sort the data on a field from a table other than the one on which the report is based?

Create a query that includes the field from the other table, and then base the report on the query. If you have already created the report, verify that the query includes all the fields used in the report, and then change the report's ControlSource property to the name of the query. In the report's Design view, open the Sorting and Grouping box, and add the field to sort the report on (the field from the other table).

For more information on creating queries, see Chapter 5, "Query Basics," and Chapter 6, "Designing Select Queries," in the *Microsoft Access User's Guide*. For more information on basing your report on a query, see Chapter 17, "Report Basics," and Chapter 18, "Designing Reports." For more information on sorting data in reports, see Chapter 19, "Sorting and Grouping Data."

Eliminating Blank Pages in Reports

Q. When I print my report, every other page is blank. What causes this problem?

The width of your report plus the left and right margins should not exceed the width of the page (usually 8.5 in.).

