

# ACCESS

1		<b>Making Sense of the Database Window in Access 2002</b>
2		<b>Finding Records in Your Access 2003 Tables</b>
5		<b>Filtering a Datasheet in Access 2003</b>
7		<b>Creating PivotCharts in Access 2003</b>
9		<b>Securing Your Access 2003 Database as an MDE File</b>
11		<b>Integrating Access 2003 with Other Microsoft Office Applications</b>
15		<b>Exporting Access 2003 Data to XML</b>

## [Making Sense of the Database Window in Access 2002](#)

Most of the time when you open a database, it appears in a window like Figure 1. This window gives you access to all the stuff in your database, provides tools to change displays or create new items, and generally helps you manage your database stuff. And it looks cool. Who can ask for more?

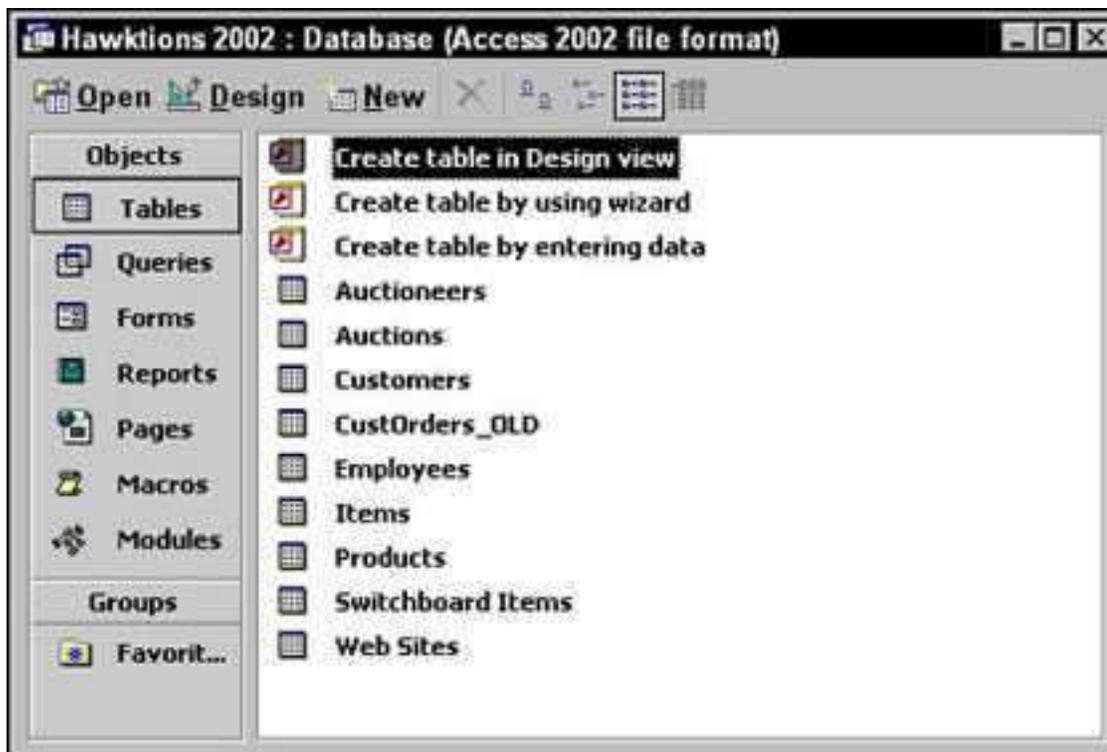


Figure 1: Well, this is another fine database that you've gotten yourself into.

The Objects bar buttons down the left side of the window switch between lists of the *objects* (tables, queries, reports, and so on) that make up the database. Four toolbar buttons sit at the top of the database window to help you work with the database's objects:

- Open displays the current object

- Design lets you change the object
- New creates a new object
- The X deletes the current object (kiss that table good-bye!)

The other buttons to the right of the X change how Access 2002 lists the objects that your database contains. Your choices run the gamut from colorful, friendly icons to detailed mini-dossiers. Feel free to try the settings yourself — you can't hurt anything! (Just don't accidentally click the X, okay?)

Your database *may* start up looking like Figure 2. Don't let the pretty face fool you, though — this window is just a fancier front hung onto Figure 1's database. Seeing something like this window is a clue that you're working with a formal Access 2002 application. Most likely, the form was created by one of your in-house nerds. This special form is called a *switchboard*.

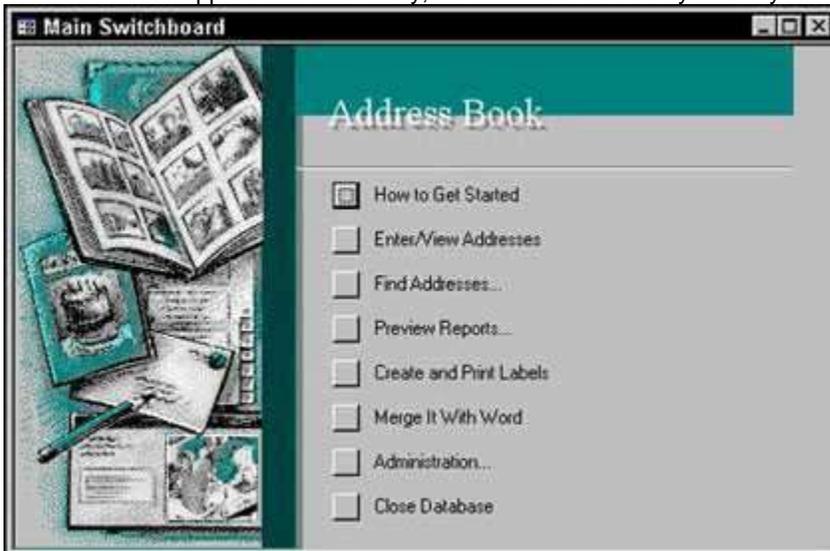


Figure 2: An example of a switchboard.

## [Finding Records in Your Access 2003 Tables](#)

When you want to track down a particular record *right now*, creating a query for the job is overkill. Fortunately, Access 2003 has a quick-and-dirty way to find one specific piece of data in your project's tables and forms: the Find command.

Find is available both on the toolbar and through the main menu. (Choose Edit --> Find. Keyboard-oriented folks out there can press Ctrl+F.) Access 2003 doesn't care which way you fire up the Find command — it works the same from either avenue.



Although the Find command is pretty easy to use on its own, knowing a few tricks makes it do its best work. After you know the Find basics (covered in the next section), you can fine-tune the Find command by using the tips in the section called, "Tuning a search for speed and accuracy," later in this article. That section shows you how to tweak the Find settings for more detailed search missions.

### **Finding first things first (and next things after that)**

Using the Find command is a straightforward task. Here's how it works:

1. **Open the table or form you want to search.**

Yes, Find works in both datasheet view and with Access forms.

## 2. Click the field that you want to search.



The Find command searches the current field in all the records of the table, so make sure that you click the right field before starting the Find process. Access 2003 doesn't care which record you click — as long as you hit the right field, Access is happy. (And it's important to keep your software happy!)

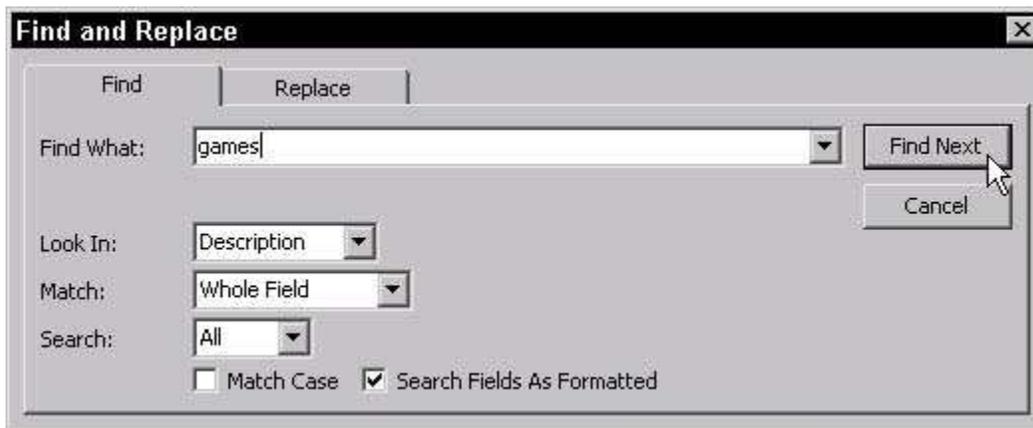
## 3. Start the Find command by clicking the Find toolbar button (the one with the binoculars on it) or choosing Edit --> Find.

The Find and Replace dialog box pops into action.

## 4. Type the text you're looking for into the Find What box, as shown in Figure 1.



Take a moment to check your spelling before starting the search. Access isn't bright enough to figure out that you actually mean hero when you type zero.



**Figure 1:** The Find and Replace dialog box gets ready to do its stuff.

## 5. Click Find Next to begin your search.

Before you can count to one by eighths, the Find command tracks down the record you want, moves the cursor there, and highlights the matching text. If Find doesn't locate anything, however, it laments its failure in a small dialog box.

## 6. If Find didn't find what you were looking for:

- a. Click OK to make the dialog box go away.
- b. Make sure that you clicked in the correct field and spelled everything correctly in the Find What box.
- You may also want to check the special Find options covered in the next section to see whether one of them is messing up your search.
- c. Click Find Next again.



What if the first record that Access finds isn't the one you're looking for? Suppose you want the second, third, or the fourteenth *John Smith* in the table? No problem — that's why the Find and Replace dialog box has a Find Next button. Keep clicking Find Next until Access works its way down to the record you want or tells you that it's giving up the search.

## Tuning a search for speed and accuracy

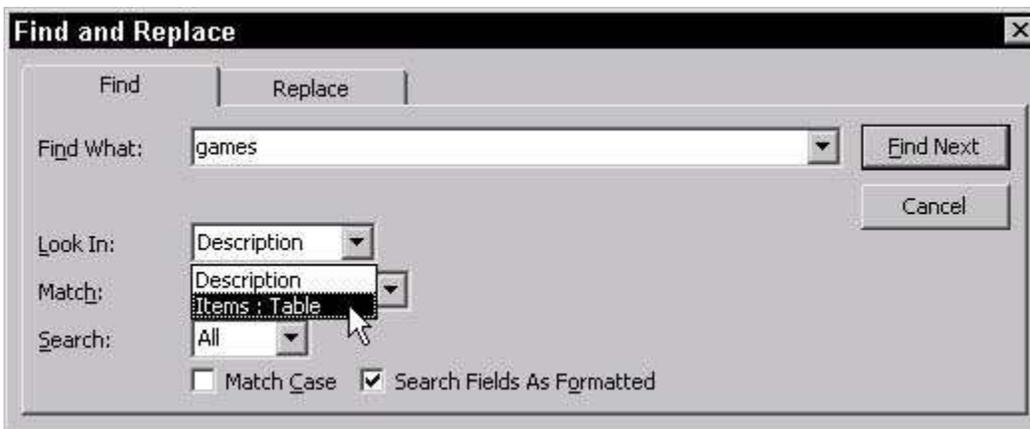
Sometimes, just providing the information in the Find What box isn't enough. Either you find too many records or the ones that match aren't the ones that you want. The best way to reduce the number of wrong matches is to add more details to your search.



Precise adjustment makes the pursuit faster, too.

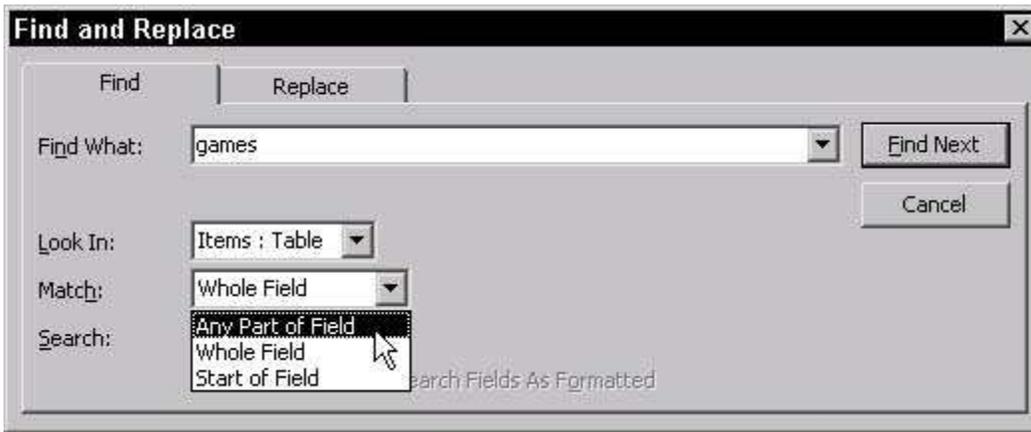
Access offers several tools for fine-tuning a Find. Open the Find and Replace dialog box by clicking the Find button on the toolbar or by choosing Edit --> Find. The following list describes how to use the various options:

- **Look In:** By default, Access looks for matches only in the *current* field — whichever field you clicked in before starting the Find command. To tell Access to search the entire table instead, change the Look In setting to the Table option, as shown in Figure 2.



**Figure 2:** To search the entire table, change the Look In setting.

- **Match:** Access makes a few silly assumptions, and this setting is a good example. By default, Match is set to Whole Field, which assumes that you want to find only fields that *completely match* your search text. The Whole Field setting means that searching for *Sam* doesn't find fields containing *Samuel*, *Samantha*, or *Mosam*. Not too bright for such an expensive program, is it? Change this behavior by adjusting the Match setting to Any Part of Field, which allows a match anywhere in a field (finding both *Samuel* and *new sample product*), or to Start of Field, which recognizes only a match that starts from the beginning of the field. To change this setting, click the down arrow next to the field (see Figure 3) and then make your choice from the drop-down menu that appears.



**Figure 3:** Look inside all those little fields by using the Match option.

- **Search:** If you're finding too many matches, try limiting your search to one particular portion of the table with the Search option. Search tells the Find command to look either
  - At all the records in the table (the default setting)
  - Up or Down from the current record



Clicking a record halfway through the table and then telling Access to search Down from there confines your search to the bottom part of the table.

Tune your Search settings by clicking the down arrow next to the Search box and choosing the appropriate choice from the drop-down menu.

- **Match Case:** Match Case requires that the term you search for be exactly the same as the value stored in the database, including the case of the characters. In other words, if the database contains the oddly capitalized name SmltH, Match Case finds that value only if you search for the name using exactly the same odd capitalization (in this case, *SmltH*). Match Case works very well when searching your database for names.
- **Search Fields As Formatted:** This option instructs Access to look at the formatted version of the field instead of the actual data you typed. Limiting the search in this way is handy when searching dates, stock-keeping unit IDs, or any other field with quite a bit of specialized formatting. Turn on Search Fields As Formatted by clicking the check box next to it. This setting doesn't work with Match Case, so if Match Case is checked, Search Fields As Formatted appears dimmed. In that case, uncheck Match Case to bring back Search Fields As Formatted.



Most of the time, this option doesn't make much difference in your life. In fact, the only time you probably care about this Find option is when (or if) you search many highly formatted fields.

If your Find command isn't working the way you think it should, check the options in the preceding list. Odds are that one or more of these options aren't set quite right!

---

## [Filtering a Datasheet in Access 2003](#)

*Filtering* an Access 2003 datasheet is a way to focus on specific records, rather than all the records in a table. You can filter out records that aren't relevant to what you're trying to do at the moment.

When you filter data, you use criteria to tell Access what you want to see. A *criterion* is a test that the data passes in order to display after the filter applies. For example, you may ask Access to show you the records with an order date of 5/1/03. A more advanced criterion is orders with a date on or after 5/1/03.

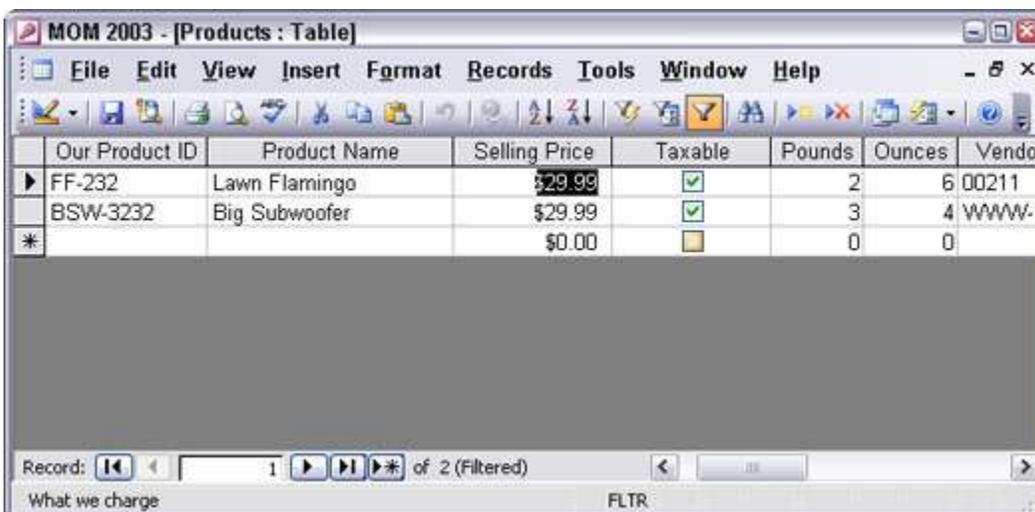
You can use five types of filters in a table: Filter by Selection, Filter by Form, and Advanced Filter/Sort. Table 1 shows you how to use each filter.

**Table 1: Types of Datasheet Filters**

<i>Type of Filter</i>	<i>When You Should Use It</i>
Filter by Selection	You have a record with a certain value in a field, and you want to find all the other records that have the same value in that field.
Filter by Form	You have more than one criterion; for instance, you want to find orders placed before 6/1/03 that were paid for by credit card.
Filter for Input	You want to type the value or values that you're looking for in a particular field, or you want to use an expression as your criterion.
Filter Excluding Selection	You can find a record with a certain value in a field, and you want to exclude all the records that have the same value in that field.
Advanced Filter/Sort	You want to do more than the other filters allow, such as sorting and applying criteria to multiple fields. Advanced Filter/Sort creates a query using only one table.

## Filtering basics

If you want to get a handle on the whole filtering concept, start out by taking a look at the parts of a datasheet that relate to filters. Figure 1 shows a datasheet with the filter buttons and indicators marked. This datasheet has a filter applied — you can tell because of the Filtered datasheet indicators at the bottom of the datasheet. Also, the Apply/Remove Filter button is highlighted — if it's clicked again, the filter is removed and all the records in the datasheet are displayed.



**Figure 1:** A filtered datasheet.

You can apply a filter to any datasheet — that includes a table, of course, but also subdatasheets and datasheets generated by queries. You can enter and edit data in a filtered datasheet. Just be aware that the filter is not applied to any new records until you reapply the filter (by choosing Records --> Apply Filter/Sort).

You may filter by using the Filter by Selection or Filter by Form buttons, or you can use the menu. Display the Records menu to see the filtering options, including those that don't have buttons (there are some on the Records menu, and some in the Filter sub-menu).

- To reapply the last filter you applied, choose Records --> Apply Filter/Sort.
- To remove a filter, click the Apply Filter button or choose Records --> Remove Filter/Sort.



A filter runs a simple query on one table — a good way to start analyzing your data. Filtering can help you warm up to creating more complex queries. If you're confused about queries, creating a filter can help you figure out how to write criteria for a query. When you create the filter, choose Records --> Filter --> Advanced Filter/Sort to see it in the query grid. Look at the Criteria row to see what the criteria look like. To close the query grid, click the Close button on the toolbar.



Filters appear in the Filter property of the Properties sheet. You can filter a table by entering an expression there, but almost no one does that because the filter stays applied, and some records may be filtered out the moment you open the table.

## Filtering by selection

Filtering by selection is the simplest kind of filter — it finds records with matching values in one field. To filter by selection, follow these steps:

### 1. Find a record with the value or text you want to match and then place your cursor in that cell to match the whole value.

- To find all products with the price of 29.99, place the cursor in a Price cell with the value 29.99.
- To match the beginning of the value, select the first character and as many thereafter as you want to match. To find all entries in the field that start with La, highlight the La in Lawn Flamingo before filtering.
- To match part of the value, select the characters in the middle of a value that you want to match. Select 99 to find all values that contain 99, such as 499.

### 2. Click the Filter by Selection button on the toolbar

Access filters the datasheet to display only records that have the same value in that field.

To see the entire table, click the Remove Filter button (which is the flip side of the Apply Filter button; the same button on the toolbar toggles between the Apply Filter button and the Remove Filter button).

## Filtering by exclusion

Filtering by exclusion is very similar to filtering by selection, except that rather than seeing only records that match your criteria, all the records that match are excluded from the datasheet.

To filter by exclusion, select a value to exclude in the same way that you select a value to match when filtering by selection:

- Place the cursor anywhere in the cell to exclude values that match the whole value. For instance, place the cursor in a cell with the value ME to exclude all addresses in the state of Maine.
- Select the beginning of the value and as many characters thereafter as you want to match to exclude all records with matching beginning values.
- Select the characters in the middle of a value to exclude all records containing the selection anywhere in the field.

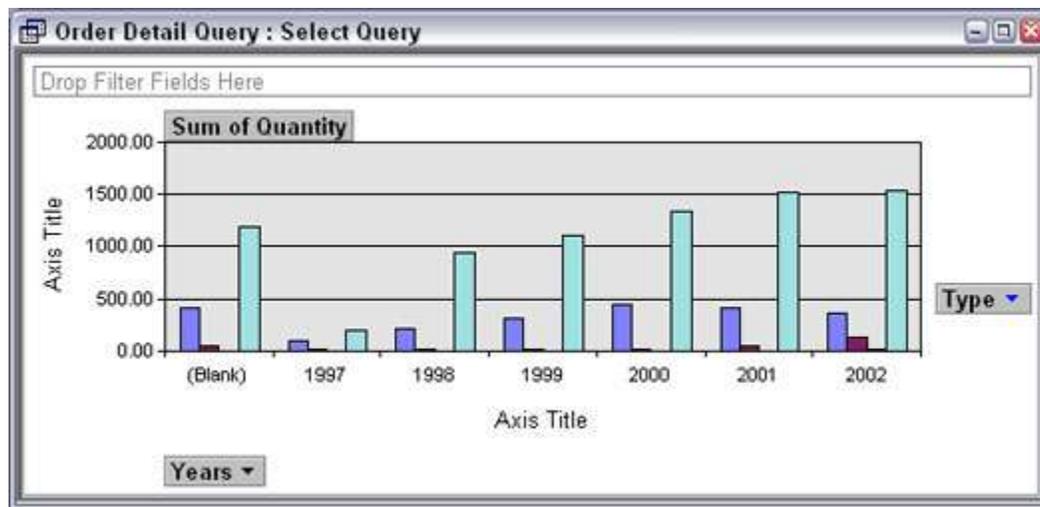
To filter by exclusion after you select the values you want to exclude, choose Records --> Filter --> Filter Excluding Selection. (This type of filter isn't on the toolbar.)

---

## Creating PivotCharts in Access 2003

You can display the information from any Access 2003 table, query, or PivotTable as a PivotChart. When you create a PivotChart, you actually create a special view of a form, and the PivotChart is stored in the form. The easiest way is by using the PivotChart AutoForm, which was new in Access 2002.

To create a PivotChart based on a PivotTable, open the PivotTable and choose View --> PivotChart View from the main menu. Access displays the same information contained in the PivotTable as a PivotChart. (See Figure 1.)



**Figure 1:** A PivotChart contains the same data as on a PivotTable.

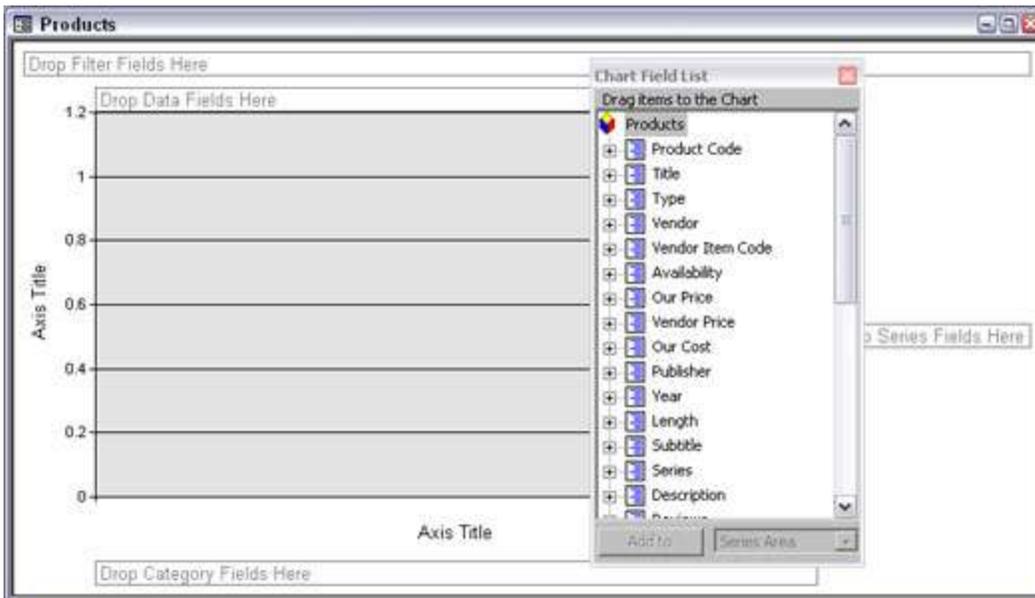
To create a PivotChart from any table or query, follow these steps:

- 1. Click the Tables or Queries button in the Objects list of the Database window and then select the table or query that contains the information you want to analyze.**

If the Database window isn't visible, press F11. You don't need to open the table or query — just select it.

- 2. Choose Insert --> Form from the main menu and, in the New Form dialog box that appears, select the AutoForm PivotChart option and then click OK.**

Access creates a new, blank form and a control containing the PivotChart. The Chart Field list also appears, showing the fields from the table or query on which the PivotChart is based, as shown in Figure 2. A new toolbar also appears: the PivotChart toolbar.



**Figure 2:** You create a PivotChart by dragging fields to the blank chart.

### 3. Specify what's on your chart by dragging fields from the Chart Field list to the PivotChart drop areas.

If the Chart Field list doesn't appear, click the Field List button on the PivotChart toolbar. The Chart Field list is like the Field list that appears when you edit a form: It lists the available fields to drag to the PivotChart.

The PivotChart drop areas are composed of

- Filter fields: Fields you want to use to filter the data shown in the chart. To filter the data, click the drop-down list and click to remove check marks — only checked data is included in the chart. This drop area works like the Criteria row in an Access query.
- Data fields: Fields containing the data you want to chart (for example, the numbers that are represented by the heights of the bars of a bar chart). The values of these fields are measured by the numbers on the Y-axis.
- Category fields: Fields that contain values that you want to run along the bottom edge (X-axis) of the chart.
- Series fields: Different values in these fields are represented by different lines in a line chart, different bars in a bar chart, or different colored graph elements. To display legends for the series, click the Show Legend button the PivotChart toolbar or choose PivotChart --> Show Legend from the main menu. If you want a stacked or clustered bar chart, a line graph with more than one line, or a multi-ring doughnut chart, drag more than one field to the Series drop area.

### 4. Make changes to the type of chart, which fields are graphed, and which values of each field are included.

### 5. To see the chart better, close the Chart Field list by clicking its X button.

You can always open it again if you want to add more fields: Right-click in the PivotChart and choose the Field List option from the shortcut menu that appears.

## Saving and viewing your PivotChart

Like PivotTables, a PivotChart is a special view of an Access object (tables and queries have pivot views available to them, too). When you close a PivotChart, Access asks you to name the form that contains the PivotChart and then stores the form with your other forms. When you close a PivotChart view of a table or query, Access just saves the information as part of the table or query. You can save your changes while editing a PivotChart by choosing File --> Save or pressing Ctrl+S.



To avoid confusion, consider including the word "PivotChart," "Pivot," or "PC" in the names of forms that contain PivotCharts.

To open a PivotChart again, click the Forms button in the Objects list of the Database window and, from the list that appears in the right pane of the window, double-click the name of the form that contains the PivotChart. Access opens the form in PivotChart view. The title bar of the PivotChart View window shows the name of the table or query that provides the record source for the chart.

You can switch to other views by clicking the View button on the toolbar or choosing View --> Datasheet View, View --> Form View, View --> PivotTable View, or View --> Design View. Switching to PivotTable view shows the same information as rows and columns of text. Switching to Datasheet, Form, or Design views is usually pointless, though — you see only the datasheet or AutoForm of the underlying table or query.

---

## [Securing Your Access 2003 Database as an MDE File](#)

If you make an Access 2003 database for other people — especially people who may be a teeny bit clueless about Access — you may want to lock up your database to prevent other users from making changes that may break it. You can add security in the form of user names and passwords, but a simpler option is to turn your database from an MDB file to an MDE file.

An *MDE file* is the same as a regular Access MDB database file, with the following changes:

- All VBA procedures are *compiled* — converted from human-readable code (more or less readable, anyway) to a format that only the computer understands. This change prevents a database user from reading or changing your VBA code.
- No one can create forms or reports or modify the existing ones (you can't even open them in Design view). You can't import any, either.



Be sure to keep a copy of your original MDB file! If you need to make changes to your VBA code, forms, or reports (or create new ones), you need to use the MDB file, not the MDE file. MDE files are most commonly used for the front-end database when you split an application into two databases (front end and back end).

### **Creating an MDE file**

Saving your MDB file as an MDE file is easy. Follow these steps:

**1. Make sure your database is in Access 2002/2003 file format by opening the database.**

Take a look at the title bar of the Database window. (Press F11 if it's not visible.) If the title bar says "(Access 2000 file format)" then you need to convert it to the latest file format.

**2. Choose Tools --> Database Utilities --> Make MDE File.**

Access closes the database to do the conversion. Then you see the Save MDE As dialog box.

**3. Specify the folder and file name for the file and click the Save button.**

Access creates the new MDE file while leaving the original MDB file untouched. Then the new MDE file opens.

If Access runs into a problem while making the MDE file, a message appears with a Show Help button. Click the button to find out what's wrong.

## Making updates later

Sooner or later, you are going to want to make a new report or fix an annoying typo in a form. You have to go back to your MDB file to make these kinds of changes, because you can't make changes in an MDE file.

If the MDE file is a front-end file, with no data stored in it, you can just make your changes to the original MDB file and resave it as an MDE file. Because all your data lives in the back-end database, you're all set.

However, if your MDE file contains tables full of valuable information, you can't just abandon it. If you use the MDE file to do data entry and editing, that file contains your up-to-date tables. The original MDB file has editable forms, reports, and VBA code, but doesn't have the latest version of the data stored in your tables. Not a problem.

Follow these steps to update your MDE file:

### 1. Rename your MDE file as a backup file.

For example, add today's date to the end of the file name (right before the .mde part). You're about to create a new MDE file, but you don't want to lose the data in this file.

### 2. Open the original MDB file and make any changes to contain the forms, reports, and VBA code that you want.

If you plan to make drastic changes, make a backup copy of the MDB first.

### 3. Choose Tools --> Database Utilities --> Make MDE File and save it as an MDE file with the name that your MDE file originally had.

Now you have an updated MDE file with new, improved forms, reports, and VBA procedures, but with old data. You also have an updated MDB file with your new, improved forms, reports, and VBA code (but out-of-date tables).

### 4. Delete all the tables from this new MDE file.

In the Database window, click the Tables button in the Objects list, click each table in the list that appears in the right pane and then press the Delete key for each table. You need to confirm each deletion by clicking the Yes button. Deleting tables sounds dangerous, but remember you have all these tables stored safely in your old MDE file.

### 5. Import the tables from the old MDE file to the new one.

Choose File --> Get External Data --> Import and choose the name you gave your old MDE file in Step 1.

You see the Import Objects dialog box, with tabs for Tables, Queries, Forms, Reports, and other objects.

### 6. Click the Select All button with the Tables tab selected and then click OK.

Access imports your tables from the original MDE to the new MDE files, replacing the older data in the tables.

### 7. Import any queries or macros in the old MDE database that you created or changed.

Repeat Steps 5 and 6, but use the Queries and Macros tabs on the Import Objects dialog box to import whatever has changed.

If you are going to do this often, consider splitting your table into a front end and a back end. With a split database, you don't have to re-import your updated tables: You can just leave them in the unchanged back-end database.

---

## [Integrating Access 2003 with Other Microsoft Office Applications](#)

Visual Basic for Applications (VBA) isn't a programming language for Microsoft Access 2003 alone. It's a programming language for all application programs that support Automation. *Automation* (with a capital *A*) refers to the ability of a program to be manipulated programmatically, or controlled automatically using a programming language such as VBA. All the major applications in Microsoft Office, including Microsoft Access, Microsoft Excel, Microsoft Outlook, Microsoft PowerPoint, and Microsoft Word support Automation. You can write code to automatically share data among them.



The terms *Component Object Model (COM)* and *OLE automation* are older terms for (but are basically synonymous with) what's currently called Automation.

### **Before you start writing code**

Before you exchange data between Access and other programs in the Microsoft Office Suite, be aware that writing code to do the job is rarely required. You can do plenty of importing and exporting data between Microsoft Office applications without writing any code at all. For example, you can perform the following actions:

- Import and export data by using options on the Access File menu.
- E-mail Access objects, such as reports, by choosing Send To --> Mail Recipient.
- Use the OfficeLinks feature to send objects to other programs.
- Use basic Windows cut-and-paste techniques and OLE (Object Linking and Embedding) to copy and link data between programs.
- Merge data from Access tables to Microsoft Word letters, labels, envelopes, or other reports, using the Word Mail Merge feature. (Search the Word Help system for *merge*.)

If you're just looking to get data from Access to another program (or vice versa), writing code is probably not the easiest approach. Any of the previous approaches are easier than writing custom VBA code to do the job.

Of course, once in a proverbial blue moon, you come across a situation where writing VBA code is the only, or perhaps just the best, way to get the job done. This article shows you some basic concepts that you'll want to understand.

### **Loading the appropriate object models**

The Access object model provides a means of referring to objects by name so that you can manipulate those objects by using VBA code. Every Office application program that exposes itself to VBA has an object model, just as Access does. After an application program's object library is available, you can use VBA to control that application.



An *object model* defines the names and organization of objects in the application. An object library is the actual file that's stored on your hard disk and defines that conceptual object model.

Before you write code to control an external application from Access, you need to load the appropriate object library into the References dialog box in Access. Follow these steps:

#### **1. Make sure that you're in the Visual Basic Editor.**

Press Alt+F11 if you're in the Access program window.

## 2. Choose Tools --> References from the Visual Basic Editor menu.

The References dialog box opens.

## 3. Scroll through the Available References list and select the object libraries for the programs that you want to control.

In Figure 1, the selected object libraries are for Access, Excel, Outlook, PowerPoint, and Word (among others).

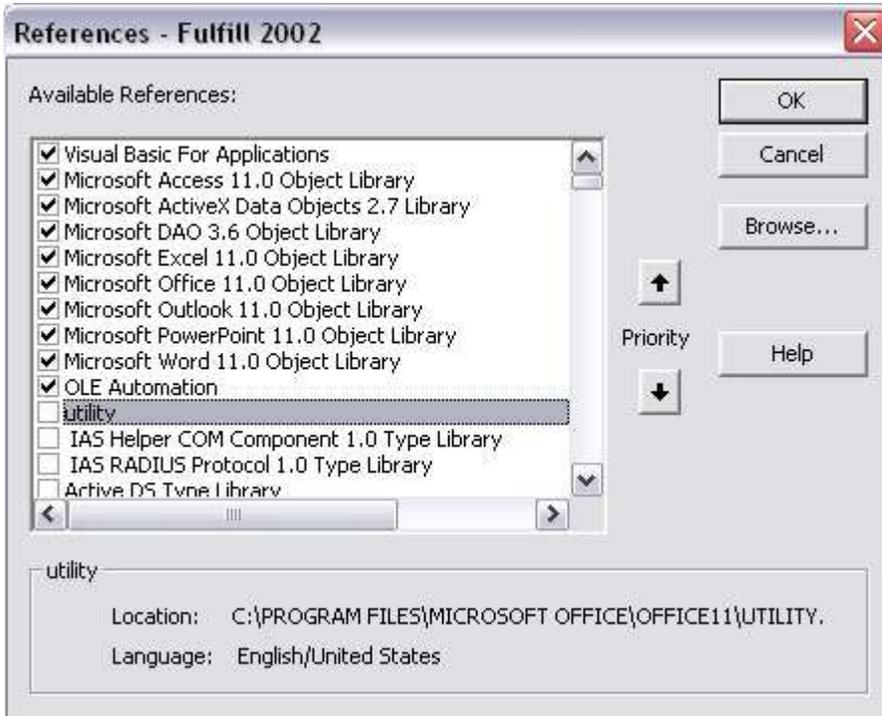


Figure 1: Several object libraries selected in the References dialog box.

## 4. Click OK.

All the selected object libraries will be opened, and you have access to all their object models from this point on.

### Unloading object libraries

Loading more object libraries than necessary is wasteful because VBA programs run slower. In real life, you rarely need to select all the object libraries available (see Figure 1) unless you really intended to interact with all those programs from the current database.

You can unload libraries as easily as you load them. Open the References dialog box and clear the check mark next to any object library that you don't really intend to use.

### Exploring a program's object model

The Object Browser in the Visual Basic Editor provides access to all the object models currently selected in the References dialog box. Each loaded object model contains many objects, classes, properties, and such. But for Automation, you mainly want to look at each program's Application Object. For example, the Access Application Object exposes Access to other programs that support Automation. The Excel Application Object (contained within the Excel object library) exposes Excel to other Automation programs and so on.



To open the Object Browser in the VBA Editor, choose View --> Object Browser or press F2. To get help with an item in the Object Browser, click its name, and then click the Help (?) button in the Object Browser toolbar.

When you choose <All Libraries> from the Project/Library list in the Object Browser and scroll down the Classes list in the left column, you see several Application objects. When you click one of the Application objects, the name of the application appears down near the bottom of the dialog box. Members of that application object appear in the pane to the right, as shown in Figure 2.

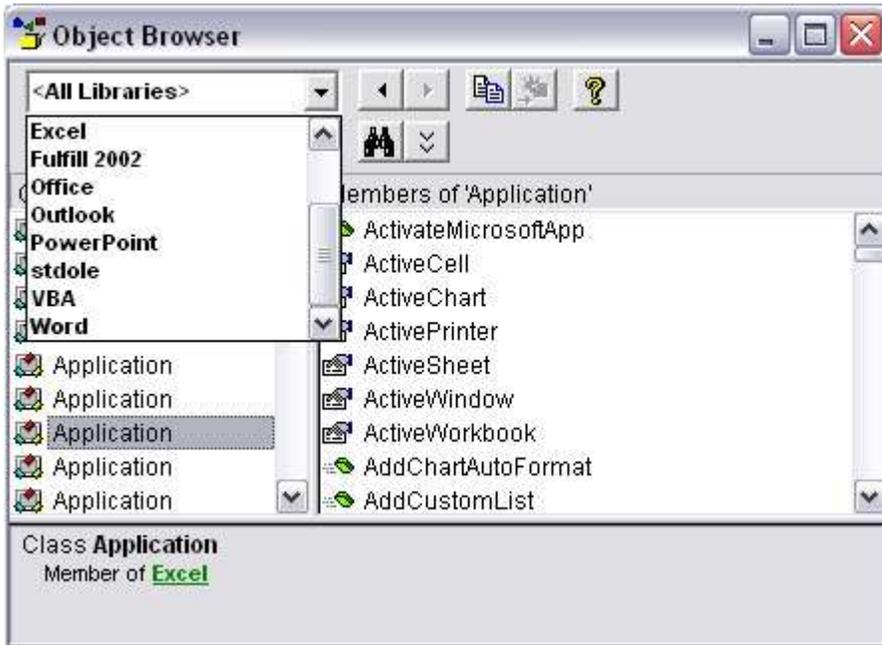


Figure 2: The Object Browser after adding multiple object libraries.

## Setting references to other programs

With object models loaded, you're ready to start setting up references to available programs in VBA code. The first step is to use a Dim statement to create an object variable that refers to the application to which you want to connect, using the following syntax:

```
Dim anyName As [New] program.Application
```

In the syntax, *anyName* is any name you like, to be used within your code to refer to the application. The *program* argument is the official name of the Automation program. The New keyword is optional. If included, the New keyword opens a copy of the application in the background (not necessarily visible on-screen) before the code runs.

Some examples of Dim statements are the following:

```
Dim appExcel As New Excel.Application  
Dim appOutlook As New Outlook.Application  
Dim appPowerPoint As New PowerPoint.Application  
Dim appWord As New Word.Application
```



A Dim statement is valid only if the appropriate object library is loaded. For example, the Dim appWord As New Word.Application statement fails if the object library for Microsoft Word isn't selected in the References dialog box.

The Dim statements merely create object variables that refer to the application from within your code. To actually make the connection to an application's Application object, you need to set the object variable name to the application. The syntax is

```
Set objectVariable As CreateObject("appName.Application")
```

where *objectVariable* is the same as the name you specified in the Dim statement, and *appName* is the name of the application program. Referring to the earlier Dim statements, the Set statements that you use for each defined object variable are the following:

```
Set appExcel = CreateObject("Excel.Application")
Set appOutlook = CreateObject("Outlook.Application")
Set appPowerPoint = CreateObject("PowerPoint.Application")
Set appWord = CreateObject("Word.Application")
```

Each application in the Office suite has its own object model with its own objects and properties, but the basic ideas are the same regardless of which program and object model you use.

---

## [Exporting Access 2003 Data to XML](#)

For those of you who are already familiar with XML in Access 2002, note that the enhanced XML support in Access 2003 enables you to specify XSL (Extensible Stylesheet Language) transform files when importing data from, or exporting data to, XML. When importing, the transform is applied to data as soon as importation starts, before a new table is created or an existing table is appended to.

When exporting data to XML, you can include any predefined filters or sort order when exporting data to XML. You can export just the data, just the schema, or both. Also, if a table contains lookup values stored in a separate database, you can include that data in the exportation.

You can easily export any table, query, form, or report to XML files from a regular Access database (.mdb file) or an Access project and Microsoft SQL Server database (.adp file). When you export a form or report, you actually export the data behind the form or report (you export the data from the form or report's underlying table or query). Regardless of which type of object you export, though, the procedure is the same.

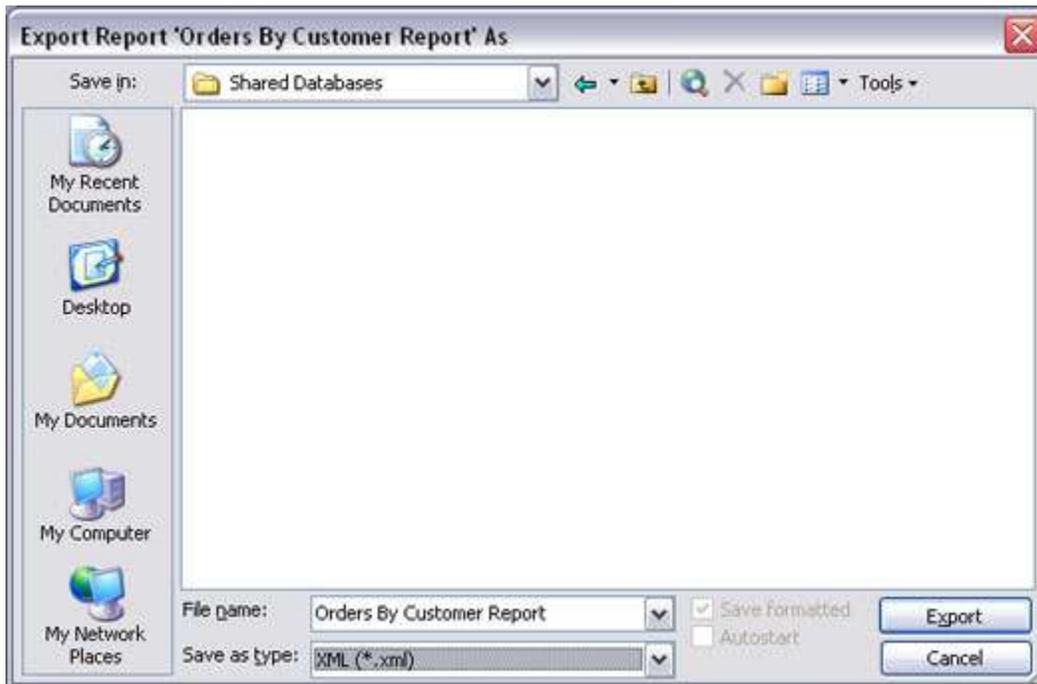
The first step is to click the Tables, Queries, Forms, or Reports button in the Objects list in the Database window and select the item you want to export from the list that appears in the right pane of the window. If you want to export all the data that the object stores, you can just right-click the object's name and choose the Export option from the shortcut menu. Then skip the next paragraph.

If you don't want to export a complete table or query, you need to export data from a table or query. Open the table or query, and then do any of the following:

- If you want to export a single record only, select that record.
- If you want to filter records for export, apply a filter to the records now.
- If you want to specify a sort order for the records, arrange the records into that order now.

After you select the records, choose File --> Export from the Access menu.

The Export dialog box opens. Navigate to the folder in which you want to store the exported data. Choose the XML (\*.xml) option from the Save As Type drop-down menu, as shown in Figure 1.



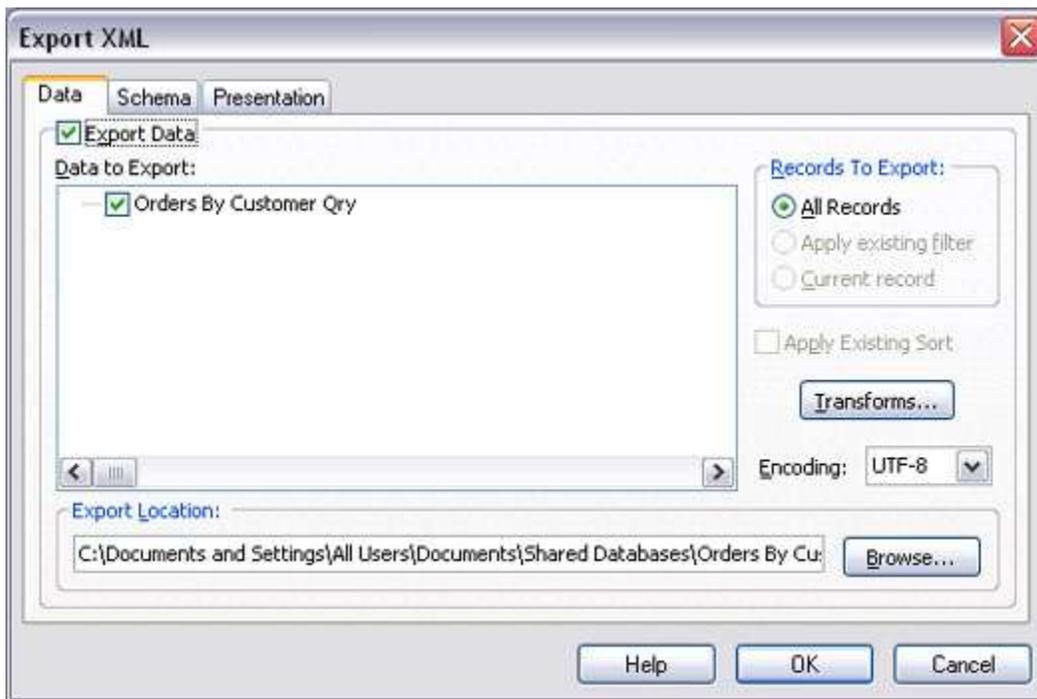
**Figure 1:** Setting up the Export dialog box to export a report to XML.

Click the Export (or Export All) button in the dialog box, and the Export XML dialog box opens, as shown in Figure 2. You have a choice of exporting data only to an XML file, exporting a schema of your data to an XSD (*XML Schema Definition*) file, exporting the presentation of your data to an XSL file, or any combination thereof.



**Figure 2:** The Export XML dialog box.

Optionally, you can further refine your selections by clicking the More Options button. Clicking the More Options button expands the Export XML dialog box to . . . show more options! Note the three tabs across the top of the dialog box: Data (shown in Figure 3), Schema, and Presentation.



**Figure 3:** The Data tab on the expanded Export XML dialog box.

### Choosing XML data options

On the Data tab, you can choose to export all records, filtered records, or the current record. Also, you can opt to apply the sort order that's currently applied to the object. (However, those options are dimmed if you export something to which those options don't apply, such as an entire report.)

If you previously defined relationships between tables in the Relationships window in Access or the Database Diagrams in SQL Server, you can choose whether or not to export data from related tables as well. Click the + sign next to any table name to see the name of any related table. Select the check mark that appears next to any table name to include that table's data in your export.

The Transforms button allows you to choose from any custom XML transforms that you may have written or acquired. If you export data that is imported into some esoteric database program, the owner of that database may supply you with a transform file. You can then click the Transforms button and choose that transform file. Likewise, if you need to use a special encoding for that esoteric database, you can choose one from the Encoding drop-down list.

### Choosing XML Schema options

The Schema tab, shown in Figure 4, allows you to choose options for exporting a schema file for your object. You can choose to include or ignore primary key and index information. You can also choose whether you want the schema information to be embedded in the XML data document, or stored as a separate file. How you choose options here depends on the program to which the exported data is later imported.

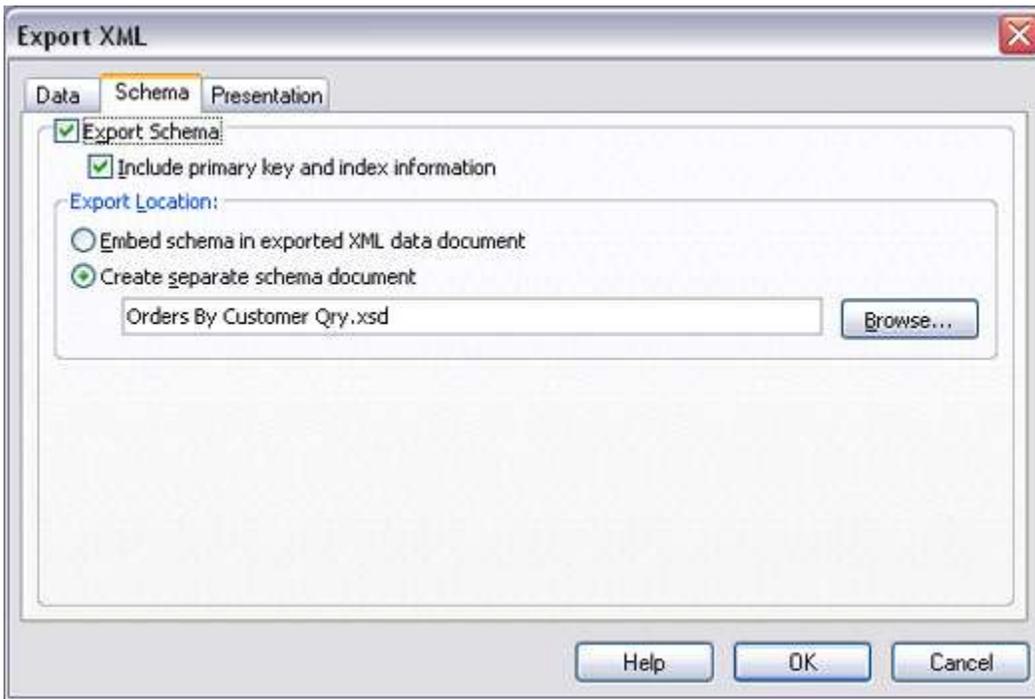


Figure 4: The Schema tab on the expanded Export XML dialog box.

### Choosing XML presentation options

The Presentation tab, shown in Figure 5, provides options for defining an XSL file. You can choose to export to Client (HTML), which can be read by any Web browser. If you plan to put the XSL file on a Web server that supports Active Server Pages, you can choose the Server (ASP) option instead.

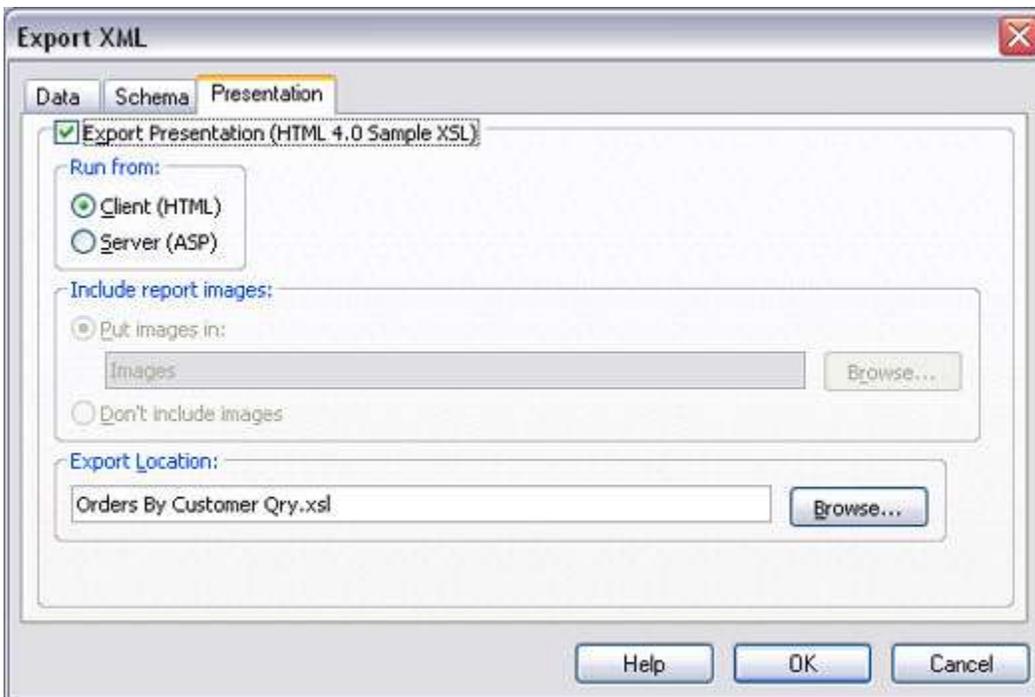


Figure 5: The Presentation tab on the expanded Export XML dialog box.

If the data you export includes pictures, you can choose to export those pictures or just ignore them. If you opt to include images, you can specify the name of the folder in which the images are placed.

Choosing to export a presentation file actually generates two files. One is an .xsl file, which includes all the XSLT code needed to define how the data is presented. The second file is an .htm file — a relatively plain and simple Web page. That Web page is a "snapshot" of the data at the moment of exportation, not live data that's connected to the database. However, the .htm file opens in Internet Explorer to reveal the actual data from the table or query, not just a bunch of XML tags and raw data.

When you finish making your selections, click the OK button. The exportation may only take a few seconds, depending on the amount of data you export. When done, you return to Access. You won't see any changes in your Access database, but the exported files are in whatever folder you specified during the exportation process.