

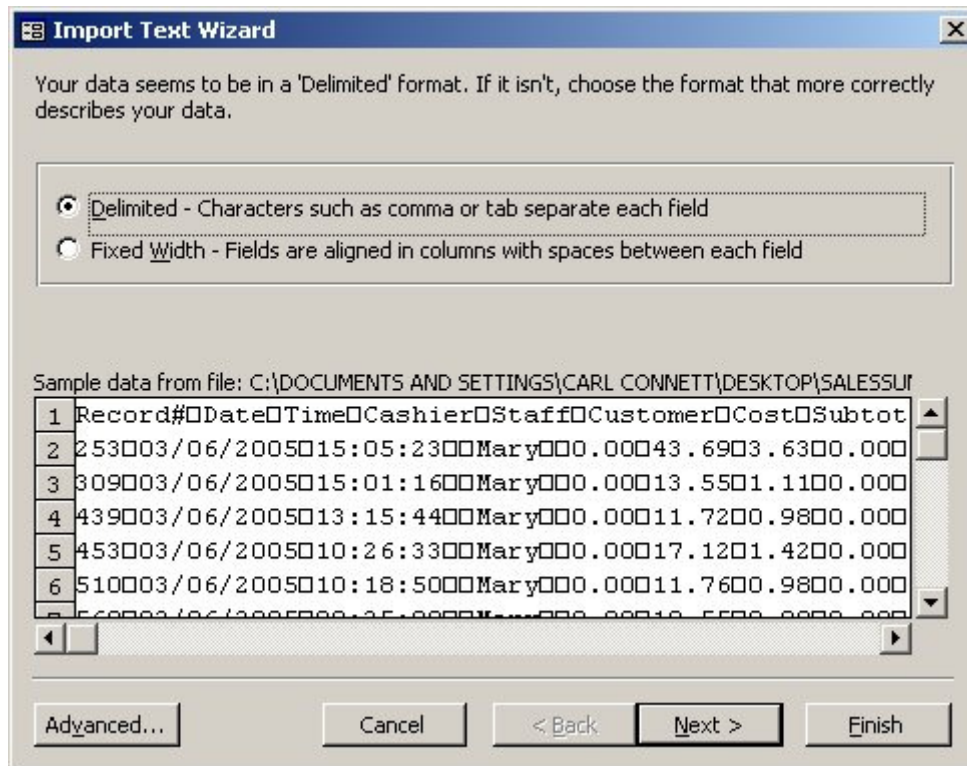
Chapter 7: Acquiring Data

Importing Data

Doing a Manual Import

Not only can you import Access tables, as we discussed in Chapter 5, but you can also import other kinds of data. Text files, dBASE, Excel, and HTML are just a few of the data formats that can be imported.

To bring non-Microsoft Access data into your current database, use the File, Import option from the menu bar. When the browser box appears, use the “Files of type” combo box at the bottom to select the data type that you wish to import. After you have selected the data file, click Import.

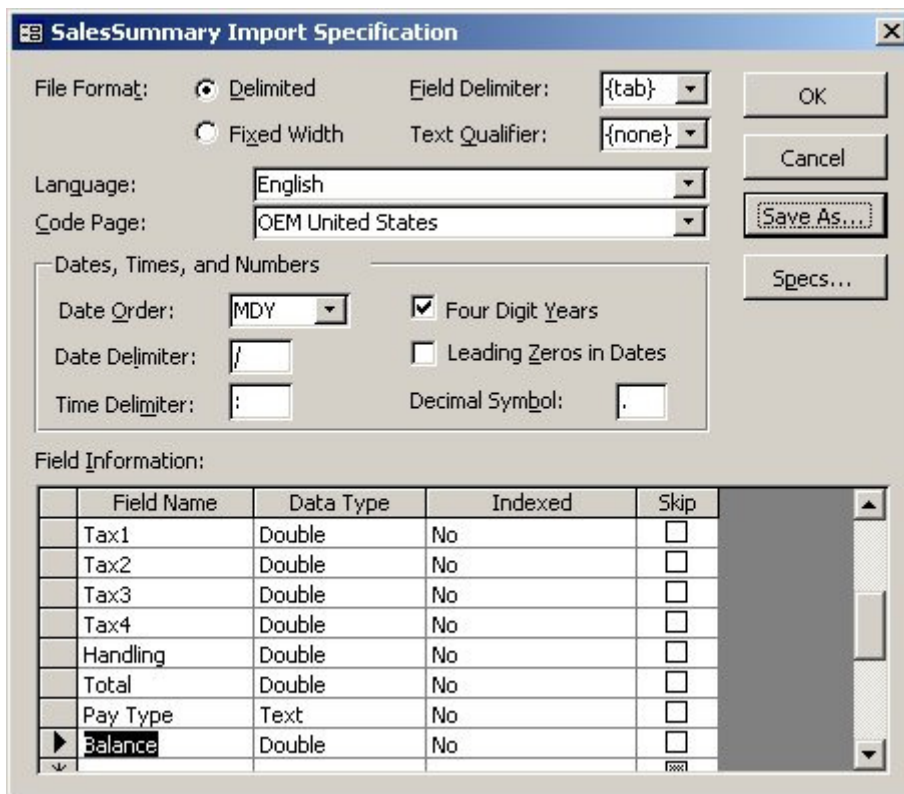


In the example above, a text file is being imported. It is delimited by tabs and has a header row. As you click Next, you will be asked for the kind of delimiter, whether you would like the imported data to go to a new table or an existing table, and for field types, names, and indexes. When you click Finish, the wizard will use the information that you have entered so far to complete the import.

The wizard options change for different data types. dBASE files import directly with no wizard. Excel files ask for if there is a header row, whether you would like the imported data to go to a new table or an existing table, for field names and for indexes.

Import (and Export) Specifications

For text and HTML files, there is an Advanced button on the lower left of the wizard window. Before you click Finish, you may click the Advanced button and then use the Save As button to save the import specification, which is basically a template that you created during the wizard steps that can be used again in a macro or in code. The same process can be used to save a specification for data that you are exporting from the database.



Using Code to Import Data

If your solution will be periodically importing consistent kinds of data, you can write a two-step procedure that automates the import process. Typically, the first step is to ask the user for a file name. To provide a browser box, Microsoft Access has some built-in browser capability, you may use the ActiveX Common Dialog control (if you have Visual Basic installed), or you may use the CommonDlg class provided with the Access 2002 Desktop Developer's Handbook.

The second step is to write code that imports the data. Here are some methods that can be used:

TransferDatabase	This method allows you to import, export, or link dBASE, Paradox, Microsoft Access, etc. tables.
TransferSpreadsheet	This method allows you to import, export, or link Excel and Lotus worksheets.
TransferText	This method allows you to import, export, or link text and HTML files.

In the following example to import a dBASE file, varFile is the full path and name of the file, DB3PWImport is the name of the dBASE table, and ImportData is the name of the table in the current database that will receive the data.


```
DoCmd.TransferDatabase acImport, "dBase 5.0", varFile, acTable, _  
    "DB3PWImport", "ImportData"
```

The order of the arguments is slightly different to transfer a spreadsheet. In the example below, ImportData is the name of the table in the current database that will receive the data, varFile is the full path and name of the file, and the last argument shown tells the method that the first row of the spreadsheet has field names.

```
DoCmd.TransferSpreadsheet acImport, acSpreadsheetTypeExcel9, _  
    "ImportData", varFile, True
```

You can make use of an import specification when you import a text file. In the following example, the file is a delimited text file, the import specification name is SalesSummary Import Specification, ImportData is the name of the table in the current database that will receive the data, varFile is the full path and name of the file, and the last argument shown tells the method that the first row of the text file does not contain field names.

```
DoCmd.TransferText acImportDelim, _  
    "SalesSummary Import Specification", "ImportData", _  
    varFile, False
```

 *Build a procedure to export the Sales table and run the code. Create an import specification to import the text file into the database and then import the file.*

Linking Tables

Along with importing and exporting data, Microsoft Access allows you to link to data files. You may link to Microsoft Access tables in other databases, link to other kind of database files, and even link to Excel and text files. The linking process stores a path and connection information about the linked data source in a hidden system table. The linked table looks and acts, in many ways, like a native Microsoft Access table. In most cases, you may edit the records, add new records, delete records, and build queries against the data.

Creating a Manual Link

To link data into your current database, use the File, Link Tables option from the menu bar. When the browser box appears, use the “Files of type” combo box at the bottom to select the data type that you wish to link. After you have selected the data file, click Link.

As with importing, there is a Link Wizard that is designed for each kind of data that you may link. For Microsoft Access tables, after you select a database to import from, the wizard will show you a list of tables in the database that may be selected for linking. Microsoft Excel files will allow you to choose the worksheet in the workbook that you would like to link. dBase files link directly with no wizard. Text files are linked through a wizard very similar to the importing wizard, also allowing you to save a specification template for later use.

Reasons to Link

Here are a few reasons why you may wish to link tables into your database:

1. Your solution requires data from other Microsoft Access databases and the data is updated outside of the control of your solution. For example, your solution needs to link to another department’s data that your solution is not allowed to edit.
2. Your solution requires data from other solutions that store data in a different format and the data is updated outside of the control of your solution. This is similar to the first reason and likely because many solutions are created in Excel but the data can still be used by your solution.
3. As the developer, you would like to be able to make changes to the forms and reports for the solution without having to take the solution down while you make the changes. The data is held in a backend database that the users can continue to edit while you are making improvements in another version that is also linked.

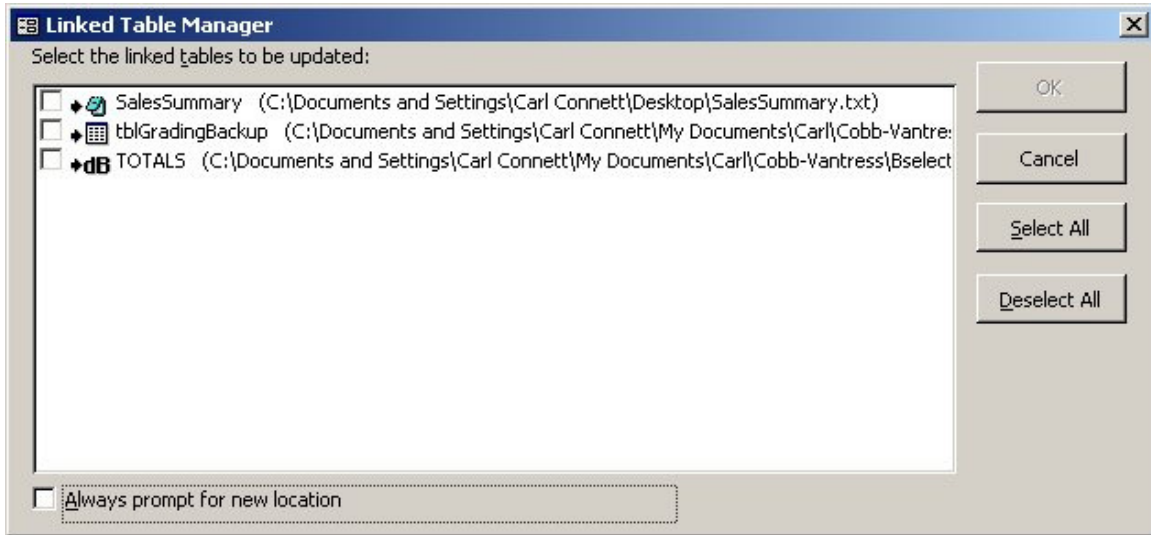
Maintaining Links After the Solution is Delivered

As mentioned earlier, links are comprised of file names and connection information, stored in a system table in the database. The file names are always absolute, not relative, so if a file path changes (it gets moved or has a different path for the user), the link will break. Microsoft Access offers the Linked Table Manager to manually update links, or you may write code to update the links.

The Linked Table Manager

Use the Tools, Database Utilities, Linked Table Manager option to update links manually. The window that appears will show a complete list of the linked tables, along with their path. If you select a linked table, then select “Always prompt for new location” before clicking OK, a browser box will appear to let you select a new file path for the linked table. If you do not select “Always prompt”, the table link will just be refreshed, not changed. If all the linked tables are Microsoft Access and all of them come

from the same database, use the Select All and “Always prompt”. The browser box will just appear once since all the tables are in the same database.



Linking With Code

There are several ways to use VBA to link and maintain links, ranging from using ADOX to change the connection properties of the linked tables to just deleting the existing link and then replacing it. The simplest possible example below assumes that the linked table source is in the same folder as the current database, which may be the case for a single-user solution.

```
Dim strFile As String


' Build the file name of the database that has the
' table to link, using the same folder that this
' database is in.
strFile = CurrentProject.Path & "\" & "BSelect.mdb"


' Delete the existing link to tblGradingBackup.
DoCmd.DeleteObject acTable, "tblGradingBackup"

' Relink the table.
DoCmd.TransferDatabase acLink, "Microsoft Access", _
    strFile, acTable, "tblGradingBackup", _
    "tblGradingBackup"
```

In this example, the source and destination table names were the same. The example can be modified by first checking to see if the link is broken (if it is not broken, there may be no need to fix it, saving time as the solution starts up) and by asking the user for a file name for the linked file. The last modification would be necessary for a multi-user solution where the backend is on the network.

MICROSOFT ACCESS AND VBA: Building a Complete Solution

 Read “Using the Office FileDialog Object”, p. 788 and “Using the CommonDlg Class, p. 1061.

 Build a procedure that uses the `Application.FileDialog` object to link a text file to the database; i.e. use the Sales table file that was exported earlier.