

*EXCHANGING FILEMAKER DATA WITH  
MICROSOFT EXCEL:  
A MANUAL FOR FILEMAKER USERS.*



Karen Vagts  
TCC 4101: Technical Writing 1  
Northeastern University  
14 August 2003



\*Sample Chapter Provided

## Table of Contents

---

*Introduction	i-i
THE PURPOSE OF THIS MANUAL	i-i
What this Manual Covers	i-i
User Requirements	i-i
Review of Spreadsheets versus Databases	i-ii
Introduction of the Sample Data Set	iii

---

*Chapter 1: Overview of Data Exchange Methods	1-1
IMPORT/EXPORT FUNCTIONS	1-1
File Format Exchange Options	1-2
ODBC DATA EXCHANGE	1-3
XML	1-5

---

CHAPTER 2: Moving Excel Data into FileMaker	2-1
USING IMPORT / EXPORT FUNCTIONS	2-X
*USING ODBC CONNECTIONS	2-X
USING XML	2-X

---

CHAPTER 3: Moving FileMaker Data into Excel	3-1
---	-----

---

CHAPTER 4: Automating Data Exchange with FileMaker Scripts	4-1
--	-----

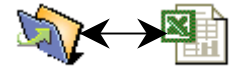
---

CHAPTER 5: Advanced Data Exchange Options: An Overview	5-1
--	-----

---

Appendices:

- A: Explanation of Sample Data Set
- B: Sample Excel Charts and Tables using FileMaker Data
- C: Resources



## Introduction

### The Purpose of this Manual

FileMaker™ database software and Microsoft Excel™ spreadsheet software are among the most popular and powerful business applications in use today.\* Used separately, each application lets you compile, analyze, and present data in multifold ways. Used together, the two applications let you take data analysis to higher levels while minimizing redundancy and input errors.

This manual describes the basic methods for exchanging data between FileMaker and Excel. It shows how, using built-in applications features, you can convert an Excel spreadsheet into a FileMaker database and then move your FileMaker data into Excel for charting, making pivot tables, and other features. By using these methods, you reuse your existing data without manually entering data in multiple places. You also protect the integrity of your original data while extending the functionality of your applications.

Most of the methods described here are standard procedures that FileMaker describes in its printed documentation and online help. Such information, however, is presented in a fragmented manner and with little guidance on selected the appropriate method for particular circumstances. The primary purpose of this manual therefore is to provide centralized, comprehensive information on this topic and to provide guidelines on making data exchange choices.

#### *WHAT THIS MANUAL COVERS*

This manual details the processes for data exchange between standard versions of FileMaker and Excel for Windows. The features described are embedded with the basic software installations and require no add-ins or special programming knowledge.

A special section of the manual presents an overview of other options for data exchange. The procedures for using them are not detailed as they require additional products or programming skills but are included here in order to expose you to these powerful and potentially exciting ways to enhance data sharing.

Appendices at the end of the manual contain supplemental information that you might find helpful as you pursue data exchange options. A portfolio of charting examples shows how you can display your FileMaker data in Excel-generated charts and graphs. The Resources section lists publications, websites, and other information sources about FileMaker and data exchange.

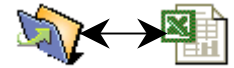
#### *USER REQUIREMENTS*

This manual assumes that you are an experienced and confident user of both Excel and FileMaker and that you have a clear sense of the features, benefits, and limits of both tools. You do not need to know other programming languages or have special plug-ins or add-ins.

The processes described focus on features available with FileMaker 6 and Microsoft Excel XP for the Windows platform; some features are available in earlier versions of these applications. If you use the latest Macintosh-based versions of the software, you use most of these features. Where that is the case, the processes are similar although platform-specific items such as menu layouts and keyboard strokes may differ.

---

\* FileMaker is a product of FileMaker Corporation ([www.filemaker.com](http://www.filemaker.com)); the current release version is 6.X. Microsoft Excel is a product of Microsoft Corporation ([www.microsoft.com/office/excel/](http://www.microsoft.com/office/excel/)) and included in the popular Microsoft Office software suite; the current release version is XP. Both are available through retail outlets and from their websites and are popular training tools in schools and universities.



## REVIEW OF SPREADSHEETS VERSUS DATABASES

Every new version of Excel and FileMaker introduces new features and “bells and whistles” that can obscure the software’s primary function and original purpose. In FileMaker, for example, you can create a tabular layout that looks exactly like a spreadsheet. In Excel, you can use “data” tools to apply database-like functions to a collection of cells. These similarities can cause you to select the wrong tool. Therefore, when planning for data exchange, keep in mind the appropriate uses for these applications.

A detailed comparison of spreadsheets and databases goes beyond the scope of this manual (see *Appendix C: Resources*) but their basic applications are summarized below:

### Use Spreadsheets for...

- Creating lists
- Building calculations, such you find in financial balance sheets
- Building “what-if” scenario testing
- Providing the base values for charts and graphs

### Use Databases for...

- Facilitating, controlling, and validating data entry
- Automating search or query building
- Presenting data in multiple formats and reports, from tables to forms to mailing labels
- Avoiding data duplication

Each application type has limits, but moving data among the two enables you to make the best of both worlds while minimizing redundant data entry and the possibility of data entry errors. In doing so, you also protect the integrity of your original data by keeping it in its original location while you manipulate it in other tools.

### CLARIFICATION OF TERMINOLOGY

Databases and spreadsheets apply slightly different terminology to data that you should keep in mind as you plan for data exchange:

A data set is a finite set of data components that present attributes of a topic. In the sample data used in this manual, the data set describes the spatial requirements for a group of departments in a company. In Excel, this set may be described as a table. In FileMaker, it may be described as a table, a database file, or simply a file.

In Excel, data is presented in a grid of cells. Each cell has its own unique address, format, and characteristics, and can operate independently of other cells. For Excel data to operate as a defined data set, the cells must be contiguous and organized in a logical matrix of columns and rows.

In FileMaker, Excel columns translate into *fields* and Excel rows translate into *records*. Every record contains the same set of fields and the fields’ contents are defined by the database: a number field always contains numbers, a calculation field always performs calculations, and an image container field only stores images.

When you exchange data between Excel and FileMaker, you *map* data between a spreadsheet *cell* and a database *field*. Proper mapping occurs when FileMaker recognizes an Excel row as a data record and an Excel column as a data field. The actual data exchange process is relatively easy and fast; the real work lies in organizing the data so that it can map properly from one location to the other.

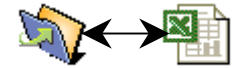


Figure i-i Comparison of spreadsheet and database structures

	Division <sup>A</sup>	Staff <sup>B</sup>	Category <sup>C</sup>	Location <sup>D</sup>
1	Accounting	Finance	10	Offices
2	Administrative	Operations	15	Offices
3	Financial	Finance	10	Offices

An Excel cell at the intersection of a column and a row is analogous to a data field. Here, Excel cell D2 maps to the Spaces field in the FileMaker record for the Accounting Department.

### INTRODUCTION OF THE SAMPLE DATA SET

The procedures described in this manual use a hypothetical data set for illustration. This data set describes the physical space location and square footage requirements for 11 departments in an imaginary company. This data helps the company's space planners and facilities managers analyze its space allocation among the departments and can be used in space planning.

Figure i-ii Presentation of the Sample Data used in this manual

This spreadsheet view of the sample data could originate from either Excel or FileMaker.

Department	Division	Staff	Space	Category	Location	SF	SF / Staff
Accounting	Finance	10	Offices	Workspaces	Building A	2000	200
Administrative	Operations	15	Offices	Workspaces	Building B	3000	200
Financial	Finance	10	Offices	Workspaces	Building A	2500	250
Human Resources	Operations	8	Offices	Workspaces	Building B	800	100
Legal	Legal	10	Offices	Workspaces	Building A	2500	250
Marketing	Marketing	8	Offices	Workspaces	Building C	1400	175
Operations	Operations	50	Offices	Production	Building D	15000	300
Plant	Operations	15	Offices	Production	Building D	20000	1,333
Production	Operations	100	Factory	Production	Building E	15000	150
Public Relations	Marketing	5	Offices	Workspaces	Building C	875	175
R & D	Operations	75	Lab	Production	Building D	15000	200

As Figure i-ii shows, the data set consists of rows (or records) describing the departments that inhabit the buildings owned by this company. Each record contains eight columns (or fields) that describe key aspects of the department. See *Appendix A: Explanation of Sample Data Set* for a description of these fields' contents.

The FileMaker data file containing this data is called *Spaces.fp5*. The Excel spreadsheet is called *Spaces.xls*. General references to the data are simply referenced as the *Spaces* data or data set.



## Chapter 1: Overview of Data Exchange Methods

The basic versions of FileMaker 6 and Excel XP provide three built-in methods for data exchange:

- Import / Export
- Open Database Connectivity (ODBC)
- Extensible Markup Language (XML)

The method you select should depend on your objectives for the data exchange.

### Import/Export Functions

The easiest and most conventional data exchange method between FileMaker and Excel is through data export and import, methods that typically create a new file in the process. Data export and import is available in both FileMaker and Excel as well as in most other software applications.

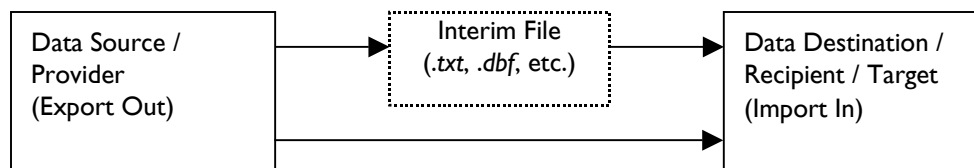
*Exporting* data copies data out of a file (data source) for use in other files, whether these files are the same file type or not.

*Importing* data copies data from an external data source into another file (data destination or target) you're working with. The data can come from a file created in the same application as the data source or from another application.

Many software applications now import (or read) files created in the proprietary file formats created by other applications. FileMaker, for example, now reads Excel *.xls* files directly; Adobe *InDesign* and *Quark Express* can read Word *.doc* files; and many CAD applications can read the AutoCAD *.dwg* file format. With some applications, however, you must export your data into an intermediary file using a common interchange file format, such as the ASCII-based text (*.txt*) file format, that the data destination can understand.

Figure 1-1 Diagram of the Export/Import data process

Depending on the application, a data recipient may be able to accept from another data source; otherwise, you must create an interim file in a data file exchange format.



The Export / Import method does not create dynamic links between the data source and destination. If data in the source changes and you want your data destination to reflect the changes, you must repeat the import process. In doing so, you risk overwriting changes you may have made subsequently in the target file. Moreover, because the data files are unlinked, they provide you with no alerts if changes are made

#### Export / Import Tip

In most software, you can determine the supported file formats for exporting and importing by opening the **File** menu and then, depending on the software, selecting the **Export...**, **Import...** and **Export...** or the **Open File** or **Save as...** options. In many graphics and page layout programs, the **File** menu > **Place...** command imports graphics or text files from other applications.



or if the data is unsynchronized.

Select the Export / Import method of data exchange if you are:

- Creating a new FileMaker database or Excel Spreadsheet.
- Merging multiple Excel data sets into a single FileMaker database.
- Working with a single (“flat-file”) FileMaker database rather than a relational database.
- Working with data that does not need to be synchronized with the data source.
- Creating quick “data dumps” from FileMaker for use in Excel.
- Moving Excel data into older FileMaker versions (FileMaker 4.x and earlier cannot read the Excel .xls file format directly).

**FILE FORMAT EXCHANGE OPTIONS**

Both FileMaker and Excel support a wide range of import and export formats. The following formats are suitable for data exchange between these two applications; they produce the least information loss and need for interim modifications.

**Table 1.1 Summary of Common File Exchange formats for FileMaker and Excel**

File Format	File Extension	Preserves Field Names	FileMaker...		Excel...		Comments
			Exports	Imports	Exports	Imports	
Excel	.xls	Yes	No	Yes	Yes	Yes	Only available for FileMaker versions 5.0 and later. Imports data directly from Excel worksheets, including named ranges. The most efficient method of importing Excel data into FileMaker.
Comma delimited (CSV)	.csv	No	Yes	Yes	Yes	Yes	A text file format. Exchanges data among BASIC programs and similar applications.
dBASE II, III, IV (DBF 2, 3, 4)	.dbf	Yes	Yes	Yes	Yes	Yes	An interchange format for databases. Retains field names and field formatting but is restrictive about field names, lengths, and other characteristics. Different versions support different columns of information for import; use DBF IV to be safe.
Data Interchange Format	.dif	Yes	Yes	Yes	Yes	Yes	An interchange format for spreadsheets. May not retain field type information.
Symbolic Link(SYLK)	.slk	No	Yes	Yes	Yes	Yes	An interchange format for spreadsheets. Imports field names into FileMaker as generic names and converts all fields to text format.
Tab-separated Text	.txt	No	Yes	Yes	Yes	Yes	Creates a text files that separates fields or cells by tabs. Generally preferable to the SYLK for most situations.

See *Appendix C: Resources* for more information on FileMaker and Import and Export functions.



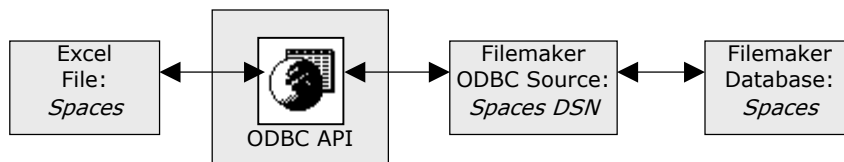
## ODBC Data Exchange

Open Database Connectivity (ODBC) is an application interface (API) that enables applications to connect to data sources that use Structured Query Language (SQL). With SQL, these applications can view data from other files, execute complex queries (or searches), and sort query results to create powerful analyses and presentations.

An ODBC connection can be reused, which enables you to maintain dynamic relations between multiple data sources. With a FileMaker and Excel ODBC link for the Spaces database, for example, weekly changes to staff counts made in Excel can be reflected in FileMaker through an automated ODBC routine. Therefore, ODBC is preferable to file import/ export for data sets that need to be synchronized on an ongoing basis.

Both FileMaker and Excel support ODBC and provide ODBC drivers to build connections. These drivers are installed in your Windows operating system and are accessed through the ODBC Administrator, a Microsoft utility. As Chapters 3 and 4 show, to build an ODBC connection, you select a driver, configure it to work with a specific data source, and then access the connection through Excel or FileMaker. You must create a separate ODBC connection (a Data Source Name or DSN) for each file you want to access; after you do that, however, you can access that data source from multiple applications.

Figure 1-2 Diagram showing an ODBC link between the Excel and FileMaker Spaces data sets as described in Chapters 3 and 4

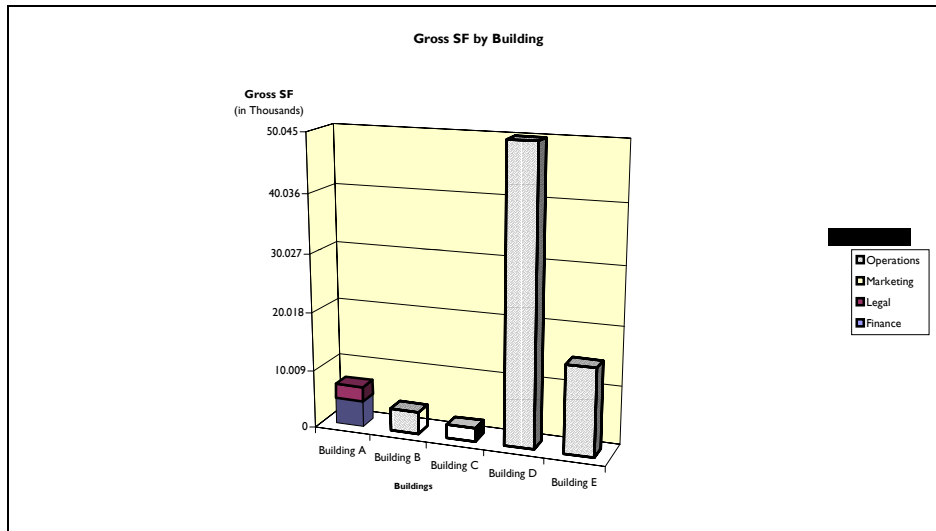


ODBC takes data connections and importing further than the standard file export / import functions described earlier in this chapter. SQL queries let you not only latch onto another file's data but to screen it for particular values and to sort the found results. With the Sample Data set, for example, you could build an SQL Query that finds only those departments with a staff count of 50 or more which are located in Buildings A or C and then sort the results by department name. The SQL query language resembles FileMaker **Find** function, relying on similar search operators and logic and enabling equally complex search criteria. These queries also let you draw data from multiple data sets and reflect relationships among these data sources.



Figure 1-3 Example of FileMaker data placed in an Excel chart using ODBC

Through an ODBC connection, Excel references the FileMaker *Spaces* data, summarized it by gross square footage cross-sectioned by division, and presented it in a 3D bar chart to emphasize that Building D contains by far the greatest amount of square feet, occupied entirely by the Operations division.



Select the ODBC method of data exchange if you are:

- Maintaining an ongoing connection between multiple data sets that are updated continuously.
- Using data in subsets or that must be filtered and sorted in a specific manner in the destination application.
- Merging data from multiple data sources or from related databases where the relational integrity of the data must be maintained.
- Exchanging data with a data “warehouse” application server software such as Microsoft SQL Server.
- Presenting data from both FileMaker and Excel with other applications, such as HTML, Visual Basic, or report writer such as Crystal reports.
- Presenting FileMaker data in Excel charts, graphs, and pivot tables.

See *Appendix C: Resources* for more information on FileMaker and ODBC.

**Important**

Although ODBC capabilities are embedded in the FileMaker and Excel applications, ODBC connections are configured *outside* these applications through the ODBC Data Source Administrator. Your ability to use ODBC properly on your computer system therefore depends on factors apart from the FileMaker and Excel software. The installed ODBC drivers must be compatible with your operating system configuration and your system memory resources. If you are configuring ODBC connections for use with other computers, external databases, or over a network, you should consult your network administrator.



## XML

FileMaker 6 and Excel XP can read and write documents using the Extensible Markup Language (XML) file format. This provides great opportunities for data exchange among multiple applications and deployment of the data in multiple formats, from webpages to printed documents to tiny applets.

Contrary to popular impression, XML is not an application but rather a structured markup language intended for universal data interchange. Although it specifies a structure for data, this structure is not rigid in the manner of most database software; rather, it can be modified or extended by user groups applying a common *schema* using XML markup tags. It also can be used on any platform, with any XML-friendly application, and any device. It can populate websites, CD-ROM files, and documents intended for printing.

Because it encases file contents in tags, XML appears similar to its sibling language, Hypertext Markup Language (HTML), but while HTML focuses primarily on the appearance of webpage contents, XML is a meta-data file format concerned with the structure, order, content, and validation of data. To do this, XML applies XML markup tags to contents based on a selected schema. Figure 1-4 shows how a FileMaker record from the Spaces database could be described in XML.

**Figure 1-4** Data from the FileMaker Spaces.fp5 file presented in a simple XML document

Data for the Administrative Department. Each field of data is enclosed between XML tags named to match the source FileMaker data fields. Other XML-compliant applications with analogous tags or fields can interpret this tagged content.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <FMPDSORESULT xmlns="http://www.filemaker.com/fmpdsoreresult">
<ERRORCODE>
  0
</ERRORCODE>
<DATABASE>
  Spaces
</DATABASE>
<LAYOUT></LAYOUT>
<ROW MODID="7" RECORDID="2">
  <Department>Administrative</Department>
  <Division>Operations</Division>
  <Staff>25</Staff>
  <Space>Offices</Space>
  <Category>Workspaces</Category>
  <Location>Building C</Location>
  <GrossSF> 3000.0 </GrossSF>
  <SfperStaff> 120 </SfperStaff>
  <Address>6 Paradise Drive</Address>
  <Buildingsf>75000.0</Buildingsf>
  <Year>1988.0</Year>
  <Floors> 3.0 </Floors>
</ROW>
```

As mentioned above, XML itself is not an application but works with, or can be embedded in, applications when used in conjunction with tools like Extensible Stylesheet Language Templates (XLT), XPath, Cascading Style Sheets (CSS), Document Type Definitions (DTDs), Dynamic HTML (DHTML), Javascript, and programming languages such as Visual Basic.

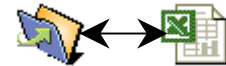
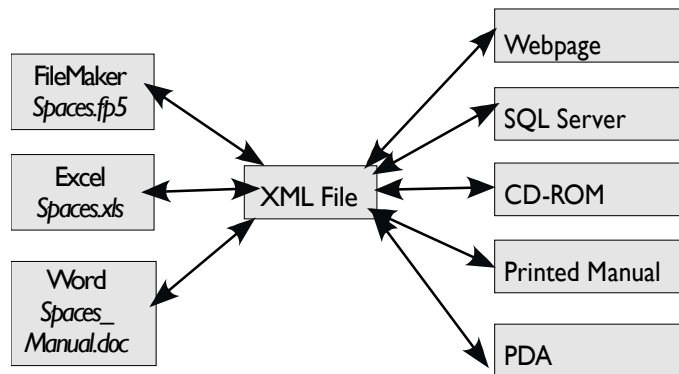


Figure 1-5 Schematic diagram showing how an XML file can connect various data sources to other data sources and output types.

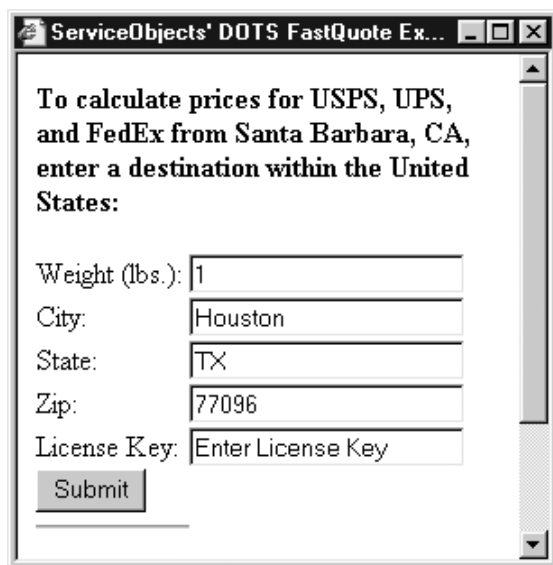


XML provides new opportunities for exchanging FileMaker data with other data resources. Examples include dynamic webpages that support user input, Visual Basic applets, and Web objects. XML can extend and automate the functionality provided by the export-import and ODBC data exchange methods and probably will become the default data exchange method in future versions of FileMaker.

Figure 1-6 ServiceObjects DOTS Fastquote Express for Shipping

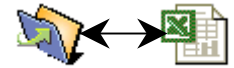
Service Object DOTS are XML-SOAP web services. This application, which consolidates quotes for shipping from the major vendors, demonstrates XML-extensibility using FileMaker.

@Copyright 2002 by Service Objects, Inc., Santa Barbara, CA ([www.serviceobjects.com](http://www.serviceobjects.com))



Select the XML method of data exchange if you are:

- Combining Excel and FileMaker data with data from other applications.
- Seeking control over the structure, format, and contents of the data.
- Building high-powered data-driven tools using applications that support XML.



- Building static HTML-based webpages that draw on both FileMaker and Excel data.
- Deploying data to multiple formats, from printed materials to webpages, wireless devices, and PDAs.

See *Appendix C: Resources* for more information on FileMaker and XML.



## Chapter 2: Moving Excel Data into FileMaker

[The following is an excerpt from Chapter 2. It discusses how to use ODBC to import in Excel data in FileMaker].

### *USING ODBC CONNECTIONS*

The ODBC exchange method connects existing Excel Spreadsheets to existing FileMaker databases. Using the **ODBC Data Source Administrator** utility, you define your Excel file as a data source for SQL queries. This allows you to grab build SQL queries on that Excel file from within your FileMaker database.

Building an ODBC connection takes three sets of steps:

1. Preparing your Excel file to serve as a data source
2. Using an Excel ODBC driver to create an ODBC data name
3. Running an SQL query in your FileMaker database to connect to the ODBC data name

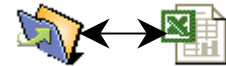
### PREPARING YOUR EXCEL DATA FILE TO SERVE AS A DATA SOURCE

The contents of your Excel spreadsheet must be organized so that your ODBC data source and SQL queries can recognize it as a data set (rather than a random collection of unrelated cells). Organize your Excel data according to the following specifications:

- All cells in the data set are contiguous.
- The data is organized so that each column represents a field and each row represents a record (see Figure i.ii).
- The first row contains column headers, which can be interpreted as field names. The Excel ODBC driver assumes that the data in the first row represents field names.
- The column headers comply with SQL standards for field names. Do not use SQL keywords and use  (underscore) rather than spaces to separate words.
- All Excel worksheet within an Excel workbook (file) and all named ranges within a worksheet contain distinct, SQL-compliant name. These names help you distinguish among multiple data sets if you build an SQL query that references multiple data sets. See *Section 2: Using Import / Export Functions* for additional information on Excel Named Ranges.

### Note

You can restrict access to your Excel data source by assigning a password to it. Select **File** menu > **Save as...** and then the **Options** pull-down menu > **General Options**. If you do apply passwords, you must use them to access the file when you run Excel queries. Make sure to store your password in a safe location; you cannot access a password-protected Excel file without a password.

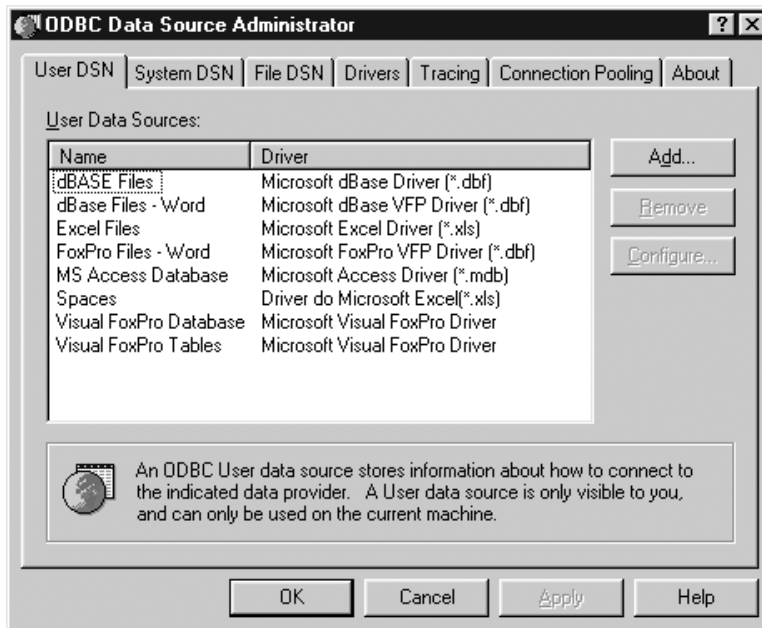


### CREATING YOUR ODBC DATA SOURCE USING AN EXCEL DRIVER

To create your ODBC data source, you must launch the **ODBC Data Source Administrator**. The Administrator is a Microsoft Windows System utility that enables you to create, modify, and store all your ODBC data sources in one place. It provides access to all the data drivers installed on your system.

1. On the Windows desktop, select **Start menu > Control Panel > ODBC data sources**. This launches the **ODBC Data Source Administrator** dialog box (Figure 2.1). The dialog displays a list of existing Data Source (DSN) files. You will be adding a DSN to this list.

Figure 2-1 The ODBC Data Source Administrator Dialog Box



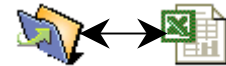
2. Click one of the three DSN tabs

- Select the **User DSN** tab to create a data source that only you or those working under your system password can use. This is only available on your local Windows profile on your machine. This selection is recommended if you are doing an initial testing of an ODBC link or are the only user of the data source.
- Select the **System DSN** tab to create a data source that is available to any user or network service that can access your system. This makes your system a data source for other users.
- Select the **File DSN** tab to create a data source provided by an existing data provider. This can be located on another computer or on a file server.

3. In the DSN tab you selected, click **Add...** to set up a new data source.

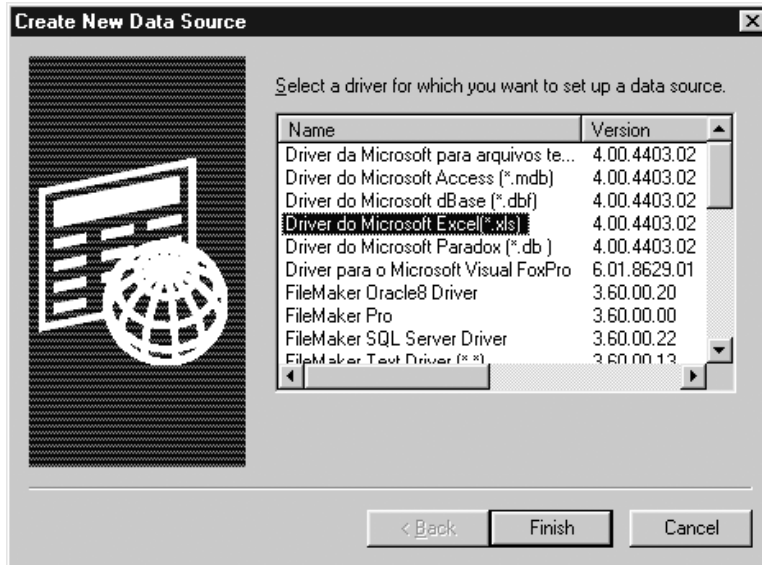
#### ODBC Tip

Each application that supports ODBC provides custom drivers that require different setup information. A single application may provide multiple drivers. Moreover, every application querying a data source uses a different methodology and terminology for the process. The general procedure described in this section, however, works for nearly every ODBC-compliant application.



4. In the ODBC **Create New Data Source** dialog box (Figure 2.2), scroll through the list of available drivers installed on your system, until you see the Microsoft Excel driver. Select it and click **Finish**.

Figure 2-2 The ODBC Create New Data Source Dialog Box

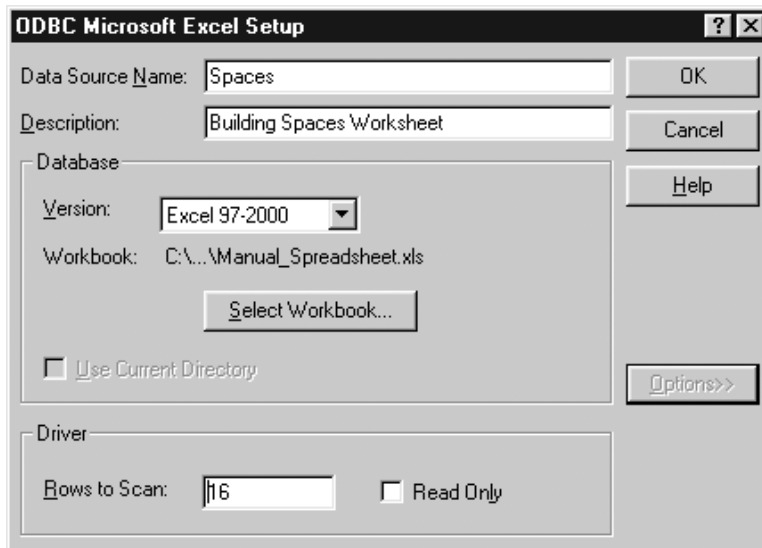


**Note**

If no Excel driver appears in the drivers list, you may need to modify your Excel installation. If your computer is part of a network system, you may need to obtain a driver from your IT department.

5. In the ODBC **Microsoft Excel Setup** dialog box (Figure 2.3), enter data into the fields for Data Source name and Building Spaces Worksheet.

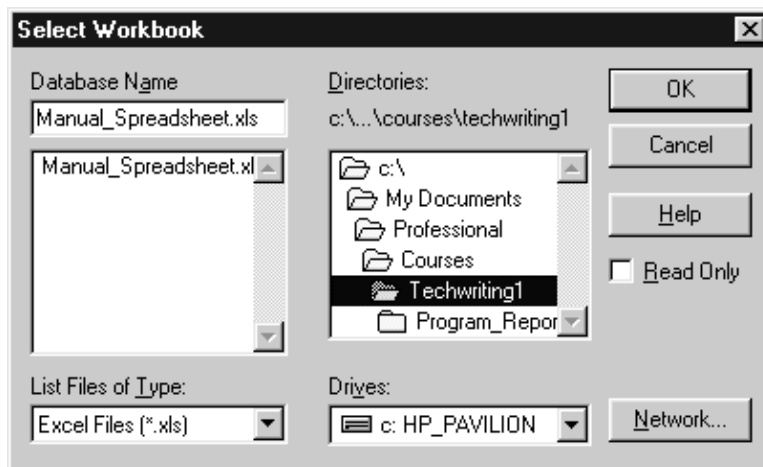
Figure 2-3 The ODBC Microsoft Excel Setup Dialog Box





- i In the **Data Source Name** field, enter a short but meaningful and unique name. This name then appears in any list of DNSs.
- ii In the **Description** field, enter a concise yet meaningful description of the Excel file's contents.
- iii In the **Version** pull-down menu, select the Excel version that file is stored as.
- iv Click **Select Workbook** to specify the source Excel file (Figure 2-4). Scroll through your file directories until you locate the file. Select the **Read Only** option to prevent users from modify data in the Excel file (this is irrelevant for FileMaker but can apply in other applications). Click **OK** to return to the ODBC Microsoft Excel Setup dialog box.

Figure 2-4 The ODBC Select Workbook Dialog Box



- v Back in the **ODBC Microsoft Excel Setup** dialog box (Figure 2-3), in the **Rows to Scan** field, modify the default number of rows (optional). The number of rows determines how many times the SQL Query process must return to Excel to retrieve data; more times reduces these retrievals but increases to the time necessary to process them.
  - vi Select or de-select the **Read Only** box to limit the ability of others to change your source data. This is the same entry as the **Read Only** box in the **Select Workbook** dialog box.
  - vii Click **OK** to save your entries. Click **Cancel** to exit the Setup process. In the **ODBC Data Source Administrator** dialog box, your newly created Excel driver appears in the list of existing drivers.
6. Click the **OK** button to close the application.

**Note**

Each ODBC data source applies a specific application driver to a particular file. You must create separate ODBC sources for each file you want to make a data source. When created, each source is available for any other application source that is ODBC-compliant and works with SQL queries. Therefore, the Excel data source you create for data exchange can also be used with another Excel file, with a Word or Access file, or with other SQL-enabled files.



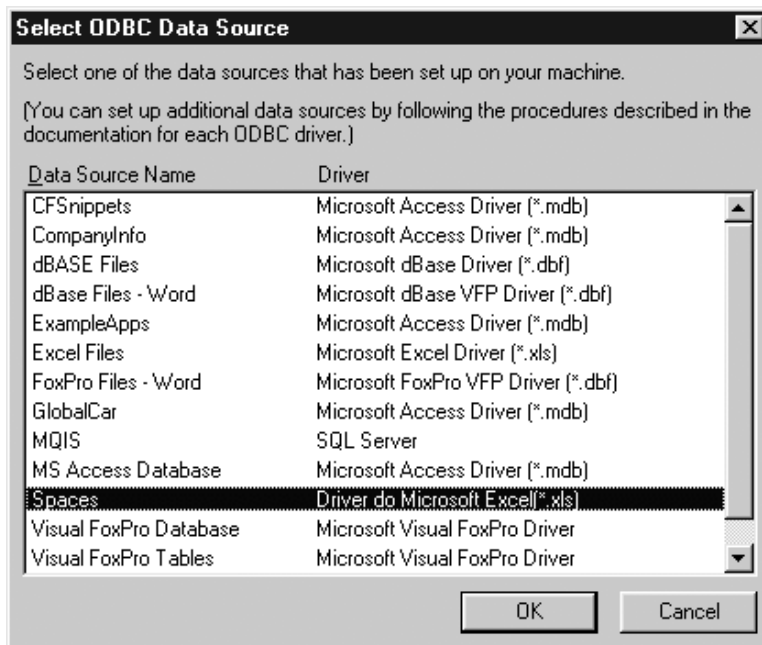
### RUNNING SQL QUERIES IN FILEMAKER

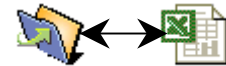
After you've configured your ODBC data source, you then can query that data from within your FileMaker database. You will do this by building an SQL query that looks into the your data source, extracts the fields you want to use, and imports it into your FileMaker database.

1. Open your FileMaker database and make sure the current FileMaker found record set contains all the records that you want the ODBC process to update. The SQL Queries only work on the current found set of records.
  - Select **Records** menu > **Show All Records** to display all the records in the database.
  - Perform a find (**View** menu > **Find Mode**) to select a subset of records. This is recommended if you want to protect certain records from being updated.
  - Select **Records** menu > **Show All Records** and then **Records** menu > **Show Omitted** to hide all the records in the database. This is recommended if you want to add records to the database and not update any existing ones.
2. Select **File** Menu > **Import Records** > **ODBC Source...**
3. In the **Select ODBC Data Source** dialog box (Figure 2-5), select your Excel data source and click **OK**.

Figure 2-5 The Select ODBC Data Source Dialog Box

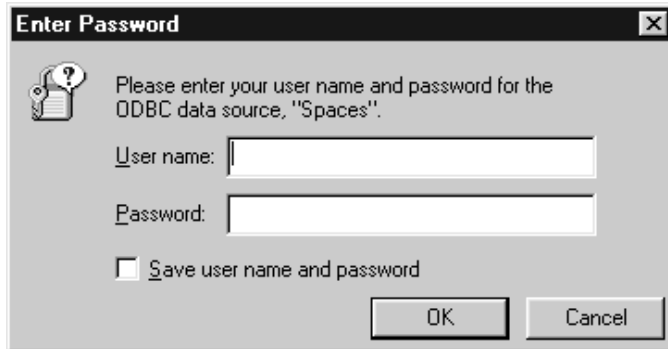
The Spaces DSN appears in the list of data sources that FileMaker can access.





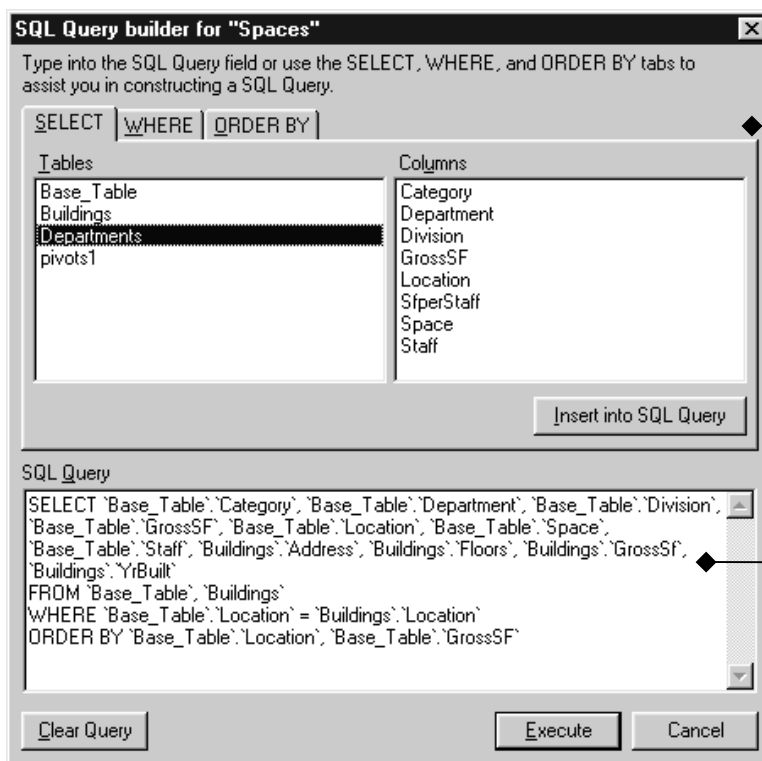
4. In the **Enter Password** dialog box (Figure 2-6), enter your User name and Password if you assigned a password to your Excel file. Otherwise, leave the fields blank. Click **OK**. The **SQL Query Builder** dialog box (Figure 2.7) appears.

Figure 2-6 The Enter Password Dialog Box



The **SQL Query Builder** dialog box is where you build your SQL query referencing the Excel DSN. The Query requires you to enter specifications in three tabbed windows titled **Select**, **Where**, and **Order By**.

Figure 2-7 The SQL Query Builder Dialog Box



The upper half of the **SQL Query Builder** shows the **Select**, **Where**, and **Order By** tabs. Enter your query specifications here.

The lower half of the **SQL Query Builder** displays the SQL Query specifications you have inserted.

The upper portion of each tab is different. The low portion of each tab is identical and tracks the SQL statements you have inserted into your query.

- Move back and forth among the tabs to view and modify your entries.
- Click **Clear Query** to erase all entries made and to start building a query from scratch.

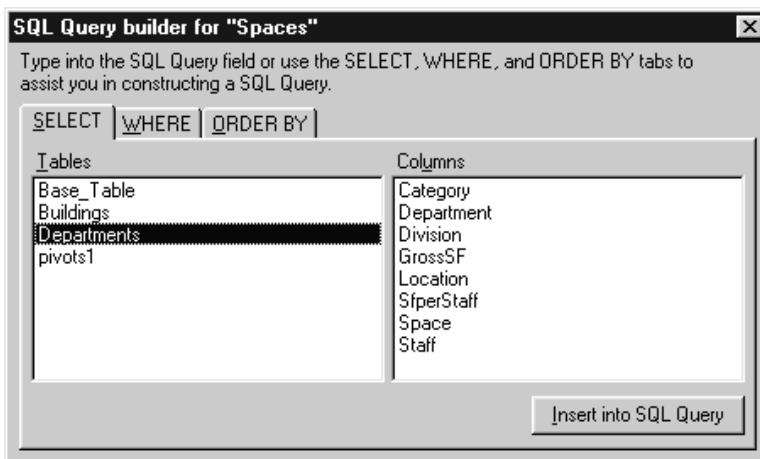


- Click **Execute** to perform your query. If your query is incomplete or improperly formed, you will receive an ODBC alert message and be returned to the SQL Query Builder.
- Click **Cancel** or the Windows **Close Window** icon (X) to exit from the ODBC import process and to return to the FileMaker database.

**Note**

As an alternative to using the tabs, you can create and edit the query directly in the SQL Query window. Place your cursor anywhere within the text in the window and create and edit the text as you would with a simple text editor. You can also paste in text from the Windows clipboard. This approach is only recommended if you are familiar with SQL syntax.

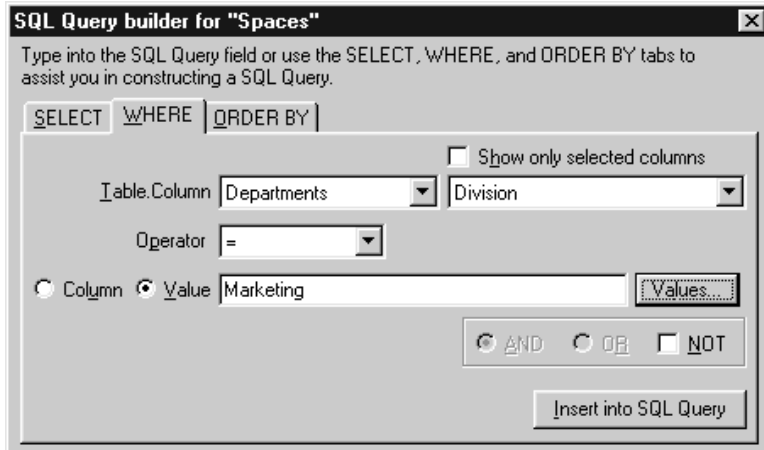
Figure 2-8 The Select Tab in the SQL Query Builder Dialog Box



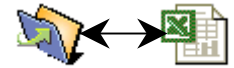
5. In the **Select** tab (Figure 2-8), select your data source table and the columns (fields) you want to insert into your SQL query. This information is required for the query to execute.
  - i In the **Tables** field, select the first table (i.e. the Excel worksheet or named data range) that you want to use. A list of data fields appears in the Columns field. Select the first field that you want to link to and click **Insert into SQL Query**.
  - ii Repeat this process for all the columns you want include.
  - iii To include data from other tables in the Excel worksheet, select those names and repeat the process of insert columns from those tables in your SQL query.



**Figure 2-9** The Where Tab in the SQL Query Builder Dialog Box  
In this example, the Query is directed to screen the Departments table for records pertaining to the Marketing Division.



6. In the **Where** tab (Figure 2-9), specify data filters so that you only retrieve records which meet certain criteria. For example, you might only need data for departments located in Building B or for certain divisions. If you want to use all the records, ignore this screen and proceed to the **Order By** tab. If you want to filter data, use this dialog box to build your instructions:
  - i Select the **Show only selected columns** option to display only the columns that you selected in the Select tab. This minimizes your selection options. Otherwise, all the columns in the table appear.
  - ii In the **Table Column fields**, select the Excel table you want from the pull-down list and then the appropriate table column.
  - iii In the **Operator** pull-down list, select an operator. For example, select “=” to indicate an exact match, “>” to indicate a value greater than. These modifiers resemble FileMaker Find modifiers.
  - iv Select the **Column** option to indicate that the table column value selected in Step 1 should match values in another data column. For example, you might want to look for records in the Department field that have the same value in the Division field. You can also link two related tables with this option; see the section later in this Chapter on *Linking Multiple Tables with ODBC Joins*.
  - v Select the **Value** option to look for records that match a particular value in the same column. Then click **Values...** Select a value from the list of data values contained in the records. For example, if you selected the Building column, select the Building B value to screen for departments located in Building B.
  - vi Click the **Insert into SQL Query** button to add this to your SQL query.
  - vii Repeat the above steps if you want to add filters. After you have create an initial Where specification, the AND, OR, or NOT selection modifiers activate.
    - The **AND** option specifies that the data must match criteria in two or more columns. For example, you can use the AND option to filter data for departments located in Location B *and* occupying over 1,000 square feet of space.
    - The **OR** option looks for records that match specified values in one or more of columns but not necessarily both. For example, you can use the OR option to filter for departments located in Building A *or* Building B.



- The **NOT** option omits (filter out) records you don't want. For example, you can tell the Query *not* to include departments in Building C.
- 

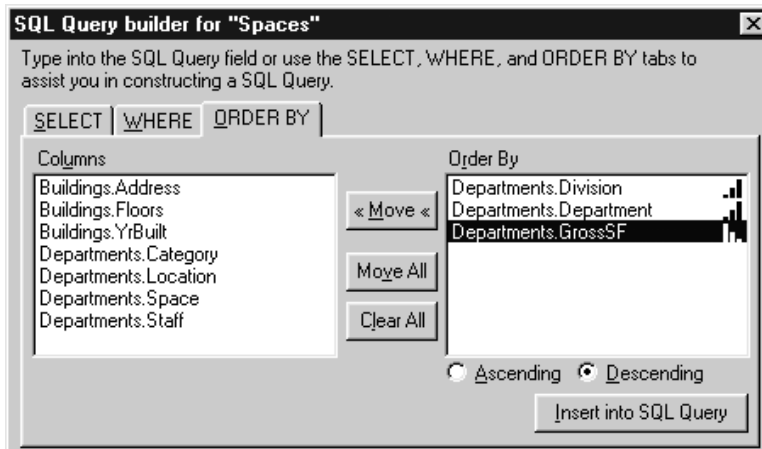
Repeat the Query Building process as often as needed to make your query more specific.

#### Notes

- The more Where statements you insert, the fewer records you are likely to retrieve. To avoid omitting records you need, keep the query broad. After the data is imported into FileMaker, you can filter the data further.
- The order of your Where statements determine the query results. A NOT statement placed first, for example, can eliminate more records than if it appears as a subset of AND and OR statements.
- If your Excel data set is relatively small, you can dispense with Where statements in the SQL query and opt to bring in the entire data set into FileMaker. Then within FileMaker, you can perform finds to filter the data.



Figure 2-10 The Order By Tab in the SQL Query Builder Dialog Box



7. In the **Order By** tab (Figure 2-10), specify a sort order for the data you are querying. Sorting the data is optional but useful if you are querying large quantities of records; then the results in FileMaker may appear more logical.
  - i Select a column in the **Columns** window.
  - ii Click **>>Move>>** to move the selected column to the **Order By** window. The selected column appears in the right window. Repeat this process for as many columns as you need.

---

    - Click **Move All** to move all the columns over.
    - Click **Clear All** to remove all the columns that appear in the **Order By** window.
    - Click **<<Move<<** to remove a selected column from the **Order By** window. This option only activates when you select a column in the **Order By** window.

---
  - iii Move the columns left and right between the **Columns** and **Order By** windows until you are satisfied with your selection.
  - iv For each column in the **Order By** window, specify whether the sort is to be ascending or descending. Select a column and then select the Ascending or Descending options under the window. The default sort order is ascending.

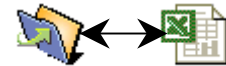
---

    - For text fields, ascending sorts in alphabetical order; descending in reverse alphabetical order.
    - For numeric fields, ascending starts with the smallest amount; descending with the largest.
    - For date fields, ascending starts with the earliest date; descending starts with the most recent.

---
  - v Select a column and drag it up or down to change the sort order for the query.

Notes

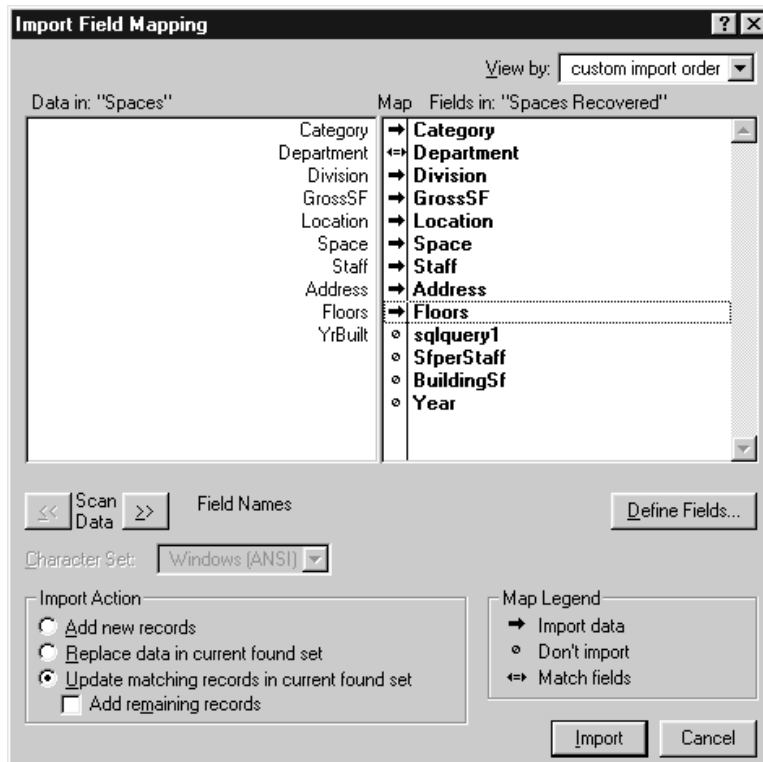
- You cannot sort columns unless you select them for inclusion in the your SQL Query.
- Although you can include all the columns in a sort, the more you select the longer the query takes to sort. If this taxes your system memory, the query may fail. Usually two or three sort columns are adequate. You can perform additional sorts after the data is moved into FileMaker.
- The sort order is hierarchical. Therefore, select the order carefully. In Figure 2.10, for example, the data sorts first by building location, then by division within the building, then by gross square feet for the department. This produces different results than if the first sort column was division or



square feet. Make sure the sort order makes sense in the context of what you want your data to show.

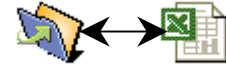
8. When you are satisfied with your SQL Query, click **OK**.
9. In the **Import Field Mapping** dialog box (Figure 2-11), review the proposed mapping of the queried Excel data (left column) to FileMaker fields (right column). If necessary, modify the mapping to fine-tune the data moves.

Figure 2-11 The FileMaker Import Field Mapping Dialog Box



- i In the **View by** menu, select the order of the fields in the FileMaker Field list. Adjusting the order can help you analyze the field mapping for large numbers of fields.

- **Creation Order** sorts by the order the fields were created in the FileMaker database.
- **Custom Order** orders according to how you drag field names up or down the field list.
- **Field Names** sorts by field name. This is helpful when you map large quantities of fields.
- **Field Types** sorts by field type (text, date, number, etc.)
- **Last Order** sorts by a previous import process into this file.



- ii In the **Map Column**, click on the existing symbol to change how the data is imported. As the Map Legend in the bottom of the dialog box indicates, you have three choices:

- 
- **Import data** (➔) replaces existing data in the FileMaker field.
  - **Don't import** (⊖) ignores the data from the source file and leaves existing data in the field unchanged.
  - **Match fields** (↔) specifies a match field for the update. It does not change the data in the field but tells the exchange process to update data in other fields when the values in these fields match. With the Spaces data, for example, a logical match field would be departments. When the department names in both data sets match, FileMaker knows into which record to put the other Excel data into.
- 

- iii Select an **Import Action**.

- 
- **Add new records** creates new records in addition to what you already have. Existing records remain unchanged.
  - **Replace data in current found set** overrides existing data in the current found set of records. If the found set contains more records than there are rows in the source data, the excess records are unchanged.
  - **Update matching records in the current found set** updates records in mapped fields based on the match field you specify.
  - **Add remaining records** creates new FileMaker records if the number of Excel rows exceeds the number of FileMaker records in the current found set.
- 

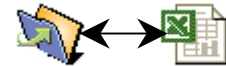
- iv Click **Scan Data** to preview your mapped data using actual data from both data sets.

- v Click **Define Fields** to create FileMaker fields for Excel data that has no analogous FileMaker field. After you've created the field, it appears in the Field Mapping dialog box and you can map it to the Excel source.

10. Click **Import** to perform the final step of your ODBC import. Click **Cancel** to stop the SQL Query import altogether.

#### Notes

- Match fields should be used to update existing data; otherwise, you will override your existing data.
- ODBC import requires that selected match field or fields provide a unique identification for each record. In the Spaces data, for example, the Department field is a suitable match field because each Department name is unique. If the name weren't unique, you need to add another match field such as Department number. If you are importing data on employees, an employee ID or Social Security number would be a logical choice. If the math values are not unique, FileMaker can't determine which record to import the data into.
- Match fields in FileMaker can be any field type except container, summary, global, or unstored calculation fields.
- When you complete the process, the FileMaker screen displays only those records that changed by the SQL query. Select **Records** menu > **Show All Records** to display all the records in the database.



### Linking Multiple Tables with ODBC Joins

As described above, your SQL Query can bring data from multiple Excel data sets at one time. The data in these sets can be completely unrelated to each other. But they also can be related, in which case you can reflect that relationship through an ODBC “join.”

In the sample Spaces data, for example, the Excel file, *Spaces.xls*, contains a related table called *Buildings*. This table stores additional information about the buildings listed in the Location column of the first table. By separating this information into a separate table, you minimize redundant data entry.

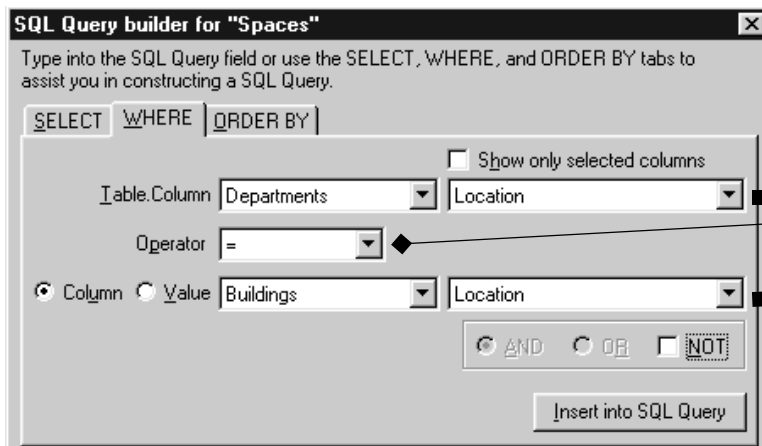
Figure 2-12 Diagram showing how the location column in the Excel Spaces table is joined to a related Building table

Spaces Table	Buildings Table				
Location	Location	Address	GrossSf	Floors	YrBuilt
Building B	Building A	4 Paradise Drive	100,000	5	1985
	<b>Building B</b>	5 Paradise Drive	75,000	3	1988
	Building C	6 Paradise Drive	60,000	3	1992
	Building D	7 Paradise Drive	120,000	6	1995
	Building E	8 Paradise Drive	100,000	5	1999

To include this information in your SQL query, modify your Query as follows:

- In the Query **Select** tab (Figure 2.8), include columns from the Buildings table as well as the Spaces table. Do NOT include the location column from the related (Buildings) table as you will be instructing the Query to import redundant data into FileMaker.
- In the Query **Where** tab (Figure 2.13), build a statement to indicate that the value of the Spaces Location column should match (=) the value of the Building Location column. This instructs the query to pull in information from the Building table when it encounters a match on the location value. When it does not, the query ignores the column contents.

Figure 2-13 The SQL Where Builder Select Tab for an ODBC join



The column specified in the Table Column field is the primary match key.

The Operator field tells the Query to match exactly the values in the Buildings Table Location column with the field below.

The column specified in the Buildings Table Location field is the secondary match key.

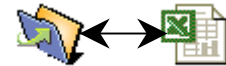


Figure 2-14 Sample FileMaker report showing data retrieved from Excel in an ODBC exchange

<b>Weekly Summary on Staff and SF Data</b>			
Organized by Division within Location			
	Staff	Gross Square Feet (SF)	SF per Staff
<b>Building B</b>			
Operations			
Administrative	15.0	3,000	200
Human Resources	8.0	800	100
SF Subtotal for Building B		3,800	
<b>Building D</b>			
Operations			
Operations	50.0	15,000	300
Plant	15.0	20,000	1,333
R & D	75.0	15,000	200
SF Subtotal for Building D		50,000	
<b>Building E</b>			
Operations			
Production	100.0	15,000	150
SF Subtotal for Building E		15,000	
GrandTotal SF		68,800	

Notes

- An SQL query can contain multiple table joins.
- While SQL queries can capture data from multiple external tables, they only can import into a single FileMaker database. To query data for multiple FileMaker files, you must repeat the query process from within each FileMaker database.
- If your FileMaker database consists of related files, the query can update the relationships. In the above example, you can have a FileMaker database analogous to the Excel Buildings table. A FileMaker relationship can match the location fields in both the Spaces and Buildings Excel tables. When data in both FileMaker files is updated by an SQL query, the updated data is viewed through the related files. For additional information on relational databases in FileMaker, see the *FileMaker Pro 6 User's Guide*, Chapter 8.



### Automating ODBC Data Exchange and SQL Queries

If you want to repeat a successful SQL query, you can automate the process, (and limit the likelihood of errors) by capturing them and reusing them. You can do this by several methods:

- Copy the SQL query out of the SQL Query window and paste it into a text file. When you rerun the Query, paste the text back into the Query window.
- Create a FileMaker field to store your query. You can access the query text by clipping it out of the field and pasting it into the Query window. You can also access the field through the FileMaker script step **Execute SQL**, specifying that the script uses the field contents to execute the query.
- Write a FileMaker script to both store the query and launch it. You can automate the entire process of querying within FileMaker, including specifying the ODBC source, any passwords, and import order. If you want to run multiple queries in succession, you can build a script that accesses sub-scripts for each query. You also can combine the ODBC exchange process with other FileMaker functions, such as telling FileMaker to print a report that reflects updated data. For more information on FileMaker scripting and ODBC, see Chapter 4: *Automating Data Exchange with FileMaker Scripts* of this manual as well as the Chapter 15 of the *FileMaker Pro 6 User's Guide*.
- Use a plug-in to automate the process. You can purchase a third-party plug-in for FileMaker or create your own if you program in C or C++. With plug-ins, you can automate processes that occur not only in FileMaker but also in other applications. For example, you could automate a two-way ODBC exchange between FileMaker and Excel. For more information on FileMaker scripting and ODBC, see Chapter 5: *Advanced Data Exchange* and *Appendix C: Resources* in this manual.