



Walt Disney World Swan and Dolphin Resort  
Orlando, Florida

## dbConnect for Facilities Management

Scott McFarlane - SHM Development, Inc.

**FM33-1** Facilities Management is a complex, multifaceted field. It is no surprise that commercial off-the-shelf software geared toward FM tends to be equally complex, and often demands steep learning curves. For many of us, however, AutoCAD's built-in dbConnect functionality provides a suitable alternative for meeting the needs of the typical facilities manager. This class provides an introduction to dbConnect, and shows how this technology can be used to accomplish specific FM tasks. We will explore the built-in capabilities of the dbConnect user interface, as well as demonstrate some simple, yet powerful custom macros that offer more advanced functionality.

### **About the Speaker:**

Scott is president of SHM Development, Inc., a software development consulting company specializing in custom database applications that use Autodesk software in the AEC and GIS industries. He has more than 20 years of programming experience, and has been integrating databases with AutoCAD software ever since it was possible. He is the author of AutoCAD Database Connectivity from Autodesk Press, as well as several articles on databases and AutoCAD. Scott has served two two-year terms on the AUGI Board of Directors.

**[scott@shmdevelopment.com](mailto:scott@shmdevelopment.com)**



## Introduction

Facilities Management, in the broad sense, is the practice of integrating people, process (the work they do) and space. As facilities managers, we are often seen a support function, and therefore less critical to the primary mission of the organization. Facilities are looked at by management as fixed assets that appear on the balance sheet – company overhead – a business expense that must be minimized.

The fact is that facilities management is first and foremost about organizational effectiveness. The decisions made about facilities are business decisions. Many organizations are now realizing that effective and proactive management of their facilities is critical to the success of the business. The role that the facilities manager plays with respect to the overall management of a company is becoming more important as companies realize this.

The more complex the job of the facilities manager is, the more he or she must rely on technology as a tool to perform many tasks. So where can dbConnect help in the day-to-day business of facilities management? A small but important part of the job of the facilities manager is *asset management* – keeping an inventory of the physical environment. In addition to physical assets, such as desks, chairs and computers, the inventory often includes people, departments and space. Having an accurate inventory of people, space, and the objects within the space will help the facilities manager with other important tasks such as organizational management, facility planning, space allocation and management, operations, maintenance and repair, and workspace installation, relocation and alteration.

This session focuses specifically on how dbConnect can be used to assist the facilities manager with *asset management* tasks.

## Tools and Expertise

AutoCAD and dbConnect, along with a well designed database, can provide the adequate tools for most asset management tasks. What is also critical, however, is that the facilities manager develops the necessary expertise with these tools to use them effectively.

This technical expertise includes:

- Operating System – File management
- AutoCAD – Layers, Blocks/Attributes, dbConnect
- Database – Database design, SQL

## The Importance of Good Database Design

While the software tools and expertise are important, I cannot emphasize how critical a well-designed database is to the success of an FM solution. The database must be able to support the kinds of decision-making questions that the facilities manager is asked by its management. All the software tools in the world are useless if the database does not meet the needs of the application.

Good design and organizational discipline are just as important in the graphic database as they are in the non-graphic database. The facilities manager must master good organizational techniques within AutoCAD, such as correct use of layers, colors, blocks and attributes. Also, correct use of object types, such as blocks for count-based items, polylines for length based items, and closed polylines for space and area-based items. AutoCAD is a very powerful and flexible graphic environment. While this is clearly its biggest strength, at times it can also be a weakness – it shifts much of the responsibility to us, the users, to exercise discipline and care in maintaining an accurate and organized representation of the facilities we are managing.

## dbConnect User Interface

The dbConnect user interface is made up of the following primary components:

- The Data Source Configuration utility
- The dbConnect Manager
- The Data View window
- The Query Editor

## Invoking dbConnect Commands

There are three basic methods you can use to invoke the various commands associated with dbConnect: the toolbars and shortcut menus within dbConnect Manager, the pull-down menus and, of course, the command line. As you become familiar with the dbConnect user interface, you will probably find the toolbars and menus to be the most efficient methods. However, there are times when a repetitive task can be accomplished more quickly using the command line.

## The dbConnect Manager

The dbConnect Manager is the primary interface to AutoCAD's database connectivity features. It is a dockable window that has a tree view and a tool bar. The tree view has one root node for each open drawing, and a single root node that expands into the list of available *Data Sources* (described below). Once a connection to a data source has been established, its node can be expanded to display the list of the tables in the data source. Many dbConnect commands can be invoked from the tree view by using shortcut menus. Shortcut menus are available by clicking your right mouse button on items in the tree view.

The dbConnect Manager window can be toggled on and off using any one of the following methods:

- Select "dbConnect" from the "Tools" menu
- Type CTRL + 6
- Type "dbconnect" or "dbcclose" on the command line

## Establishing a Database Connection

Before you can do anything with databases in AutoCAD (or any application for that matter) you must establish a connection with the database. Most database systems employ some level of security, and by establishing a connection you are, in essence, logging into the database system as a specific user. You must define a *Data Source* for any database to which you are connecting from within AutoCAD.

## Data Links and Data Sources

A Data Source contains all the information needed to establish a connection with a particular database system. The amount of information needed to create a data source depends on the type of database system you are connecting to.

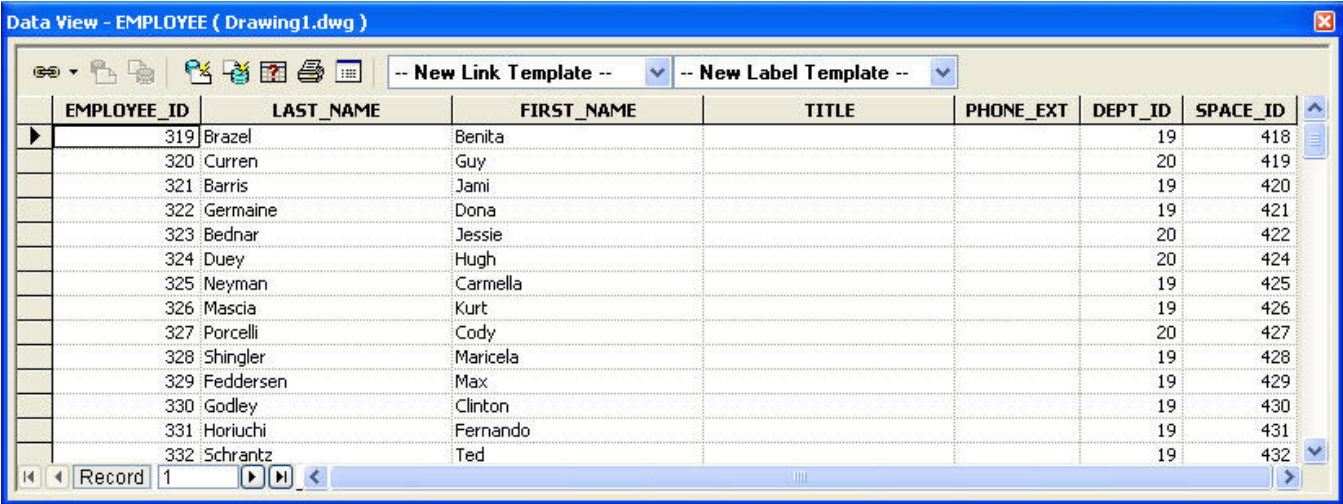
Since AutoCAD is using the OLE DB technology, the data source configuration is handled by a utility that is a part of Microsoft's data access components. OLE DB uses the term *Data Link* to describe a specific database connection, while AutoCAD uses the term *Data Source*.

The following steps demonstrate how you would create a data source to an Access database called office.mdb.

1. Start AutoCAD.
2. From the **Tools** menu, choose **dbConnect**. This displays the **dbConnect Manager**. A **dbConnect** pull-down menu is also added to the AutoCAD menu bar.  
*Tip: You can also toggle the display of the dbConnect Manager by typing CTRL+6.*
3. Right-click the **Data Sources** branch in the **dbConnect Manager** and choose **Configure Data Source...** from the shortcut menu. The **Configure a Data Source** dialog box is displayed.
4. Type "office" as the data source name, and click **OK** or press **ENTER**. This launches the **Data Link Properties** dialog box.
5. On the **Provider** tab, select **Microsoft Jet 4.0 OLE DB Provider** in the list of OLE DB Providers, and click **Next >>** to switch to the **Connection** tab.
6. Click the ellipsis icon located just to the right of the file box and browse for the **office.mdb** file.
7. Click **Test Connection** to verify that a connection can be made.
8. Click **OK**.
9. Double-click the **office** data source in the **dbConnect Manager** window to establish a connection.

### The Data View Window

dbConnect provides a built-in spreadsheet-like interface for viewing and editing table data called the **Data View** window. The Data View window functions much like the table-viewing window in Microsoft Access. You can move and resize columns, freeze columns, sort and search and replace data. The Data View window is the primary interface to working with tables that are linked to AutoCAD objects.



At the top of the Data View window is a toolbar that gives you quick access to the most frequently used functions. Just below the toolbar is the table data in a "grid" format. Each column in the grid has a column header that displays the column name as it is stored in the table. Selecting columns is done using standard Windows conventions. To select a column, simply click on the column header. To select multiple columns, drag the mouse across the headers you want to select, or hold

down the shift key to select adjacent columns, or hold down the control key to select non-adjacent columns.

Using your mouse, you can double click on a column header to quickly sort the table data on that column. Double click again on the same column header to reverse the sort order. You can resize the display width of a column by dragging the dividing line between that column and the column immediately to its right. You can hide a column by simply dragging its width to the left until the column header disappears. You can also rearrange the columns by positioning the mouse pointer in the middle of the column header and dragging the column to the desired position.

To view the column popup menu, position the mouse pointer in the middle of any column header and right-click. Choose **Sort** on the popup menu to display the Sort dialog. This dialog is useful if you need to sort the table data by more than one column. Also on the popup menu is a **Hide** option which temporarily hides any selected columns. To redisplay any hidden columns, choose **Unhide All** from the popup menu.

From this popup menu, you can also **Freeze** one or more columns. Freezing columns locks them in place at the left side of the Data View grid so that they always stay visible, even when you scroll horizontally to the right.

## Linking to a Database

The ability to link objects to specific rows in an external database table is the essence of AutoCAD's database connectivity feature. For the facilities manager, this is how a floor plan of an office is linked to a database of employees, space, fixed assets, etc. In order to link an object to a row in the table, you must be able to uniquely identify that row from all other rows in the table. This can be accomplished using a single ID column in your table. It may be necessary to use multiple columns to uniquely identify a row, but this is not recommended. This column (or columns) is generally referred to as the *key* or *primary key* column.

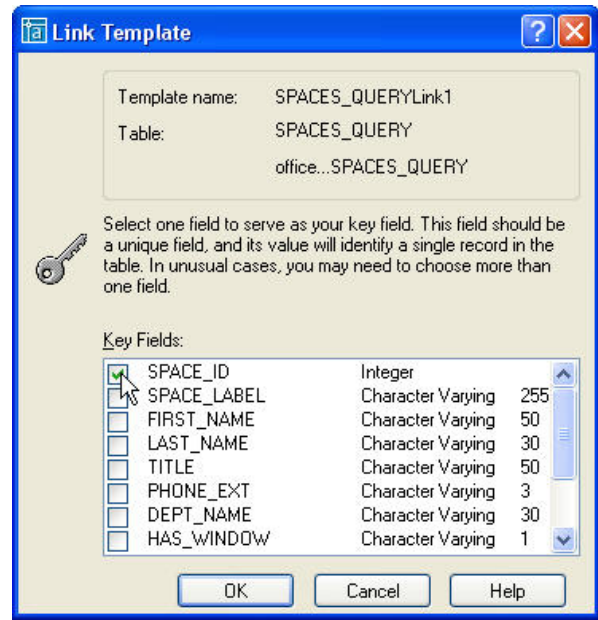
A link is simply a piece of information stored on an AutoCAD object that can uniquely identify a row in an external table. Any type of AutoCAD object can have one or more links attached to it. The information needed to establish a link between an object and a row in the table is as follows:

- *Data Source* – The database management system used
- *Catalog* – The database. A data source may contain one or more catalogs (databases)
- *Schema* – The subset of the database that is available to a particular user
- *Table* – The table you are linking to
- *Key column(s)* – The names of the columns that uniquely identify the row in the table
- *Key value(s)* – The specific values that are used to find the row that is being linked

Notice that this list represents a hierarchy: A data source contains multiple catalogs, a catalog contains multiple schemas, a schema contains multiple tables, and a table contains multiple columns. This hierarchy comes from the SQL standard upon which dbConnect is based.

## The Link Template

Typically, you may have hundreds or even thousands of objects linked to different rows in the *same* table. This means that for each object, the data source, catalog, schema, table, and key column are the same. The only thing different on each object is the key value. As you can imagine, storing all of this data on every object would result in a tremendous amount of unnecessary duplication. To eliminate this duplication, dbConnect uses a concept called the *Link Template*. A Link Template stores everything AutoCAD needs to know about a particular link except the link value. Then for each object you want to link, you simply specify the link template, and then the specific key column values for that object. Another advantage to this is that it gives you the flexibility to change certain properties of the Link Template, without having to re-link the objects.



You must create a Link Template before you can start creating links on entities. Since link templates are associated with a particular table, you must be connected to a data source before they can be created. There are a number of ways to create a link template within the dbConnect user interface. In the dbConnect Manager, you can right-click on a table name choose "New Link Template" from the shortcut menu, or use the toolbar button. You can also use the dbConnect pull-down menu.

## Creating Links on Objects

Once you have created a Link Template for a specific table in your database, you can start linking objects to that table. Depending on how large your dataset is, this may prove to be the most time consuming part of setting up your dbConnect environment.

With the Data View window open, you can select a row in your table, click the Link toolbar button, and select the object that is associated with that row. This process is then repeated for every link you wish to create.

Unfortunately, AutoCAD does not include any tools to assist you in linking large amounts of data to a drawing. And, in many cases, your drawing will not contain enough information to "automate" the linking process anyway. If your drawing *does* contain some inherent intelligence, such as blocks with attributes or text objects within closed polygons, you could write a program using Visual Lisp or VBA to generate the links for you. If you use Autodesk Map, you do have some automated linking tools that can take advantage of some of this intelligence.

## Linking Schemes

The AutoCAD dbConnect technology imposes no restrictions on the number of rows that can be linked to an object nor does it restrict the number entities that can be linked to a single row. As a result, you have the flexibility to choose a linking scheme that fits best with the specific needs of your application. There are four possible ways to establish links between AutoCAD objects and external database tables:

- Many-to-one (several objects are linked to a single row in the table)
- One-to-many (each object has links to multiple rows in the table)
- Many-to-many (multiple objects are linked to multiple rows)

- One-to-one (each object is linked to a single unique row in the table)

### **Many-to-One or One-to-Many**

The biggest disadvantage to these linking schemes is that some aspects of the database structure are not known to the database. For example, in a many-to-one scheme, it is impossible to determine from the database how many AutoCAD objects are linked to a particular row.

### **Many-to-Many**

This particular scheme is mentioned only because it is possible – not because it is recommended. If a many-to-many relationship is necessary, then it should not be implemented through the link. Instead, use a one-to-one linking scheme and create the many-to-many relationship within your database.

### **One-to-One**

The most desirable configuration is a one-to-one relationship between the objects and the table. In this scenario, each object has a single link to a unique row in a table. Ideally, for every row in the table there is a graphical object and for every object there is a row in the table.

In the AutoCAD drawing, you should exercise the same kind of data orderliness as the database imposes on your data. For example, objects that are linked to a common table should all be the same type of object and reside on their own layer. This keeps the graphic information organized and distinguishes the objects that are linked from the ones that are not.

For example, consider a space management database model in which there is a table containing a row for every space in an office building. A comprehensive spatial database consists of a drawing with every space represented as an AutoCAD object, typically a closed polylines, and each space object is linked to a unique row in the space table. You should be able to gather information about each space and modify non-graphic information about a space, as well as count the number of space that exist in your system, without having to consult the AutoCAD drawing. But as soon as you need to add, delete or modify a space, you must do this from within AutoCAD.

To maintain the integrity of a true one-to-one scheme, you must reduce the possibility of having orphan rows (rows in your table that do not have an object linked to them) or orphan entities (entities linked to nonexistent rows). Completely eliminating the potentiality of orphans is impossible as long as the graphic and non-graphic data are managed by two independent systems. There is no built-in mechanism that would prevent a user from deleting a row from a database table without also deleting the object to which it is linked. Similarly, there is also no built-in mechanism that would prevent a user from deleting an object from an AutoCAD drawing without also deleting the associated row in the table.

Maintaining the integrity of the links between the graphics and the database then becomes the responsibility of the user. This is an ideal situation where a simple custom application could be written to help the user with this issue.

## **Performing Queries**

One of the most important responsibilities of the facilities manager is space planning. But in order to predict the future, you must be able to understand the past and present. A well-designed database and an accurate graphic interface (your linked AutoCAD drawing) are the tools you need to start looking at your data in many different ways. Once you have your drawing linked, it can be used as a powerful query and reporting tool. dbConnect includes a feature called the Query Editor for this purpose. The Query Editor allows you to perform SQL queries on your data, and then highlight the AutoCAD objects that are linked to the resulting rows.

The Query Editor gives you four different ways to create queries:

- Quick Query for very simple queries
- Range Query to query based on a range of values
- Query Builder for more complex queries
- SQL Query for advanced users who just want to type the necessary SQL statements to create the query.

Queries can be saved within the drawing for later use. Queries stored in the drawing appear as child nodes of the drawing node in the dbConnect Manager. You can also save a drawing's query set to an external file. Query sets saved to a file have a .dbq extension.

When you execute a query using the dbConnect Query Editor, the resulting rows can be displayed in the Data View window. You also have the option to highlight the objects in the drawing that are linked to the resulting rows. The objects are added to the "pick-first" selection set, and can be operated on by any AutoCAD command that uses a selection set of objects.

## Managing Change

Organizations, especially larger ones, tend to be very dynamic in order to stay competitive. Changes in organizational structure, staff, technology, and space needs are happening all the time. It is the job of facilities manager to not only play a role in driving these changes, but to also keep the asset inventory up to date with them.

One important goal of any AutoCAD/database solution is to minimize the effort involved in keeping up with change. Below are some things that will make your life much easier when it comes to data maintenance tasks.

- If you are designing your own database, make sure you understand the basics of data normalization – don't try to pack everything into a single table. Use the rules of normalization to reduce redundancy in your data.
- Pick up a book on SQL. Your database will likely contain several tables that you will need to perform queries on. The more comfortable you are with SQL, the more comfortable you will be with normalizing your data into multiple tables.
- Implement database linkages using a one-to-one scheme. Each object should only contain one link key value that uniquely identifies a single row in the linked table. Your goal should be to minimize, if not eliminate, the need to add, change, or delete link key values on an object. Once the object is linked, the object and its associated data become a single autonomous object. In other words, the linked row of data is an extension of the object, and the object is an extension of the row of data. Maintaining a pure one-to-one linking scheme also means that there should be exactly the same number of linked objects as there are rows in the linked table.
- The column you use to link to AutoCAD should exist solely for the purposes of linking. Link columns should be used only to link either to graphic objects or to other tables. The Link column should have no other meaningful purpose in the application. If it does, there is always a chance that data in that column might change. Subsequently, the link key value on the object would also need to change. Ideally, users would never even see the link column and would never have the need, let alone the opportunity, to modify it.
- A table should be linked only to a single type of object. A table should only contain information that is related to a single type of object. The rules of relational database normalization should apply to the graphic database as well. For example, if you are linking a table of spaces to closed polyline objects in AutoCAD, then only link rows to closed

polylines, and do not link lines, arcs, or text. In addition, it is also good practice to separate the linked objects onto their own layer. This makes it easier to write applications that check for data integrity. It would be very easy, for example, to determine if all the polylines on the space layer have links on them and that the linked rows exist in the space table.

- The table and the object to which it is linked should represent the same type of real-world object. For example, if you are tracking office spaces and the employees that work in each space, you should not link rows in the employee table to the objects that represent office spaces. An office and an employee are logically two different types of objects. Instead, you should maintain a table of spaces and a table of employees and establish the relationship between the two tables inside the database. Then link the spaces table to the objects that represent those spaces. Then you can move employees from space to space by modifying the links that are internal to the database – and never have to touch the drawing.
- Understand data integrity pitfalls. When external databases are used to link information to an AutoCAD drawing, you always have at least two files to manage. As long as the graphics and data are in two separate systems, there is the potential for them to become out of sync. For example, if you need to maintain an exact one-to-one relationship between the objects and the database, your application must ensure that when an object in the drawing is deleted, the corresponding row in the linked table is also deleted, or at least moved to another table for archiving purposes. Likewise, when additions or deletions are made to the database, those changes need to be reflected in the drawing. AutoCAD will not deal with this issue by itself.
- Understand that AutoCAD and a database have a very different transaction model. In computer lingo, a transaction starts when you begin modifying a file or database and ends when the changes are written (permanently) to disk. In a typical session of AutoCAD, you work with an in-memory copy of the drawing file. Changes you make as you work are not written to disk until you explicitly save the file. Document management becomes an important aspect of your workflow. It is easy for users to make copies of a drawing file to work on. There are times when you might work on a copy of the drawing for hours or even days before you decide whether or not to commit the changes you have made. AutoCAD by itself has no control over how many copies of a drawing file exist, nor does it know which copy is the "real" one. This transaction model is vastly different from the way most database systems work. In a database, changes you make are immediately written to the database. Some database systems support longer transactions, but in the database world, long could mean just a few seconds.
- Finally, if you are using vanilla AutoCAD, consider upgrading to Autodesk Map. The name of this product is misleading – it implies that it can only be useful to the GIS industry. The fact is that when it comes to the fundamental capabilities of a software tool that connects graphics to a database, there is very little difference between FM and GIS. Autodesk Map includes features that can be very useful to the facilities manager. These include automated linking tools, structured object data, advanced query tools, thematic display, automatic labeling, and polygon overlay analysis. Here is an example: Suppose you are installing new carpet in part of your office space, and you need to charge-back the cost to the departments based on the area of new carpet in each department. If you have polygons that represent departmental space, and polygons that represent the new carpet area, you can use Autodesk Map's polygon overlay analysis tools to determine the breakdown of new carpeted area by department.