

Bonus Chapter 3: Using Custom Programs

In This Chapter

- ✓ **Identifying an application file**
- ✓ **Loading and unloading an application**
- ✓ **Loading an application file automatically**
- ✓ **Running a program from an application file**

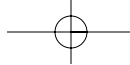
Before doing any custom programming, you need to understand a little more about how custom programs that you create or download from the Internet can be used in AutoCAD. AutoCAD itself has a variety of custom programs that extend the core functionality of the application. And AutoCAD 2009 also ships with an add-on called Express Tools, which is a collection of custom programs that are not part of the core application. See Book IX, Chapter 4 for more information on Express Tools.

So how do custom programs that you've created or downloaded get loaded into AutoCAD? The approach varies based on the type of custom program. Before you even get a program loaded, however, you need to understand how different custom programs are represented on disk through Windows Explorer.









Identifying Application Files

Custom programs are stored in application files, which can be identified by their file extensions or their designated icons in Windows Explorer. The programming interface in which you choose to write your custom programs determines the application file's extension. At some point, you'll most likely download programs from the Internet or perhaps coworkers will share some custom programs that they've collected over the years. So you'll find yourself working with different types of application files.

Table BC3-1 lists the various application files you might encounter in AutoCAD.

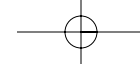


BC24 Loading and Unloading Applications

Table BC3-1 Application Files		
<i>Icon</i>	<i>File Extension</i>	<i>Description</i>
	LSP	AutoLISP application source file
	MNL	AutoLISP menu source file
	FAS	Fast-load AutoLISP file
	VLX	Visual LISP executable file
	DVB	AutoCAD VBA source file
	ARX	ObjectARX file
	DBX	ObjectDBX file
	DLL	.NET/application extension file

Loading and Unloading Applications

Most of the custom programs, or applications, that you can use in AutoCAD can be loaded or unloaded through a single dialog box. But, like everything else in AutoCAD, you can load or unload custom programs in a number of ways. The Load/Unload Applications dialog box provides the basics for loading and unloading applications. But if you're sharing custom programs with other users, you can load application files in much more efficient ways.

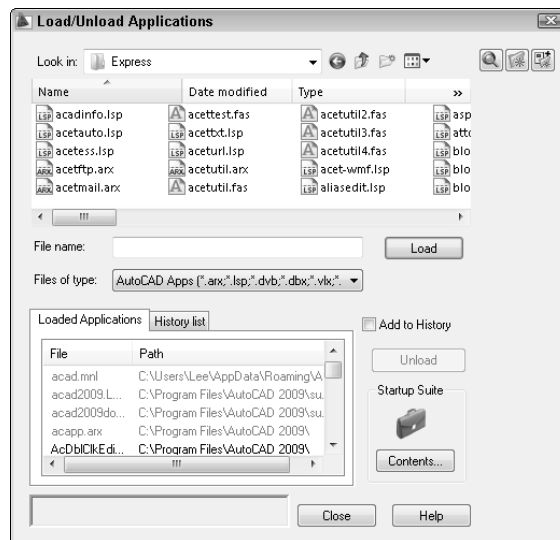


Loading and Unloading Applications **BC25**

The Load/Unload Applications dialog box

The Load/Unload Applications dialog box (see Figure BC3-1) allows you to perform a variety of tasks that deal with loading and unloading application files. You can load application files created with AutoLISP (LSP, FAS, and VLX), VBA (DVB), ObjectARX (ARX), or ObjectDBX (DBX). .NET application files can't be loaded by using the Load/Unload Applications dialog box. Instead, they must be loaded with the NETLOAD command, which I cover in the "Loading a .NET file" section later in this chapter. The APPLOAD command launches the Load/Unload Applications dialog box.

Figure BC3-1:
You can load a custom program or application with the Load/Unload Applications dialog box.



Bonus
Chapter 3

Using Custom
Programs

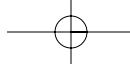
Launching the Load/Unload Application dialog box

To launch the Load/Unload Applications dialog box, do one of the following:

- ◆ On the ribbon, click Tools tab → Applications panel → Load Application.
- ◆ On the menu browser or menu bar, click Tools menu → Load Application.
- ◆ At the command prompt, type **APPLOAD** or **AP** and press Enter.

Loading an application file

The following steps initiate the APPLOAD command and explain how to load an existing application file:



BC26 Loading and Unloading Applications

1. Launch the Load/Unload Applications dialog box by using one of the methods just mentioned.

AutoCAD launches the Load/Unload Applications dialog box.

2. In the Files of Type drop-down list, specify the type of application file you want to use to filter the list box.

The selected option controls the type of application files that are displayed in the list box. The default selection is AutoCAD Apps, which lists all available application files. The drop-down list box allows you to filter on AutoLISP, VBA, ObjectARX, and ObjectDBX files.

3. In the Look In drop-down list, specify the location for the application file.

The Look In drop-down list allows you to select a folder or named location to start in, such as My Documents.

a. If the application file is located in a folder below the location you selected in the Look In drop-down list, double-click the folder in the list box below the Look In drop-down list.

The list box displays any folders or files located under the folder selected in the Look In drop-down list. The types of files displayed in the list box depend on the file filter selected in the Files of Type drop-down list.

b. Keep double-clicking folders until you navigate to the folder that contains the application file that you want to load. Select the application file from the list box.

The application file should be in the list box and its name is added to the File Name text box.

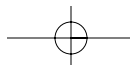
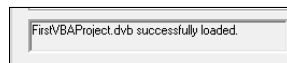
4. Click Load.

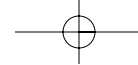
The selected application file is loaded into AutoCAD. You can tell that the file was loaded by checking for the Successfully Loaded status message in the lower-left corner (see Figure BC3-2) of the dialog box.

5. Click Close.

Clicking the Close button exits the Applications dialog box.

Figure BC3-2:
The load status of an application file.





Loading and Unloading Applications BC27



If you're going to use a custom program fairly often, you can select the Add to History check box below the Load button. This adds the application file to the History List tab. The next time that you want to load the application file, just select it from the History List tab and click the Load button instead of browsing to it again. To remove the application file from the History List tab, select it from the list and click Remove.



AutoLISP files are different from VBA, ObjectARX, ObjectDBX, and .NET application files because they're accessible only in the drawing that they were loaded into. So if you use the Load/Unload Applications dialog box to load an AutoLISP file, it's available only in the current drawing. The exceptions to this rule are AutoLISP menu source (MNL) files and any AutoLISP files that are loaded by other application files or that use a special naming convention that AutoCAD looks for. These specially named files are covered in the "Automatically loading AutoLISP files" section later in this chapter.

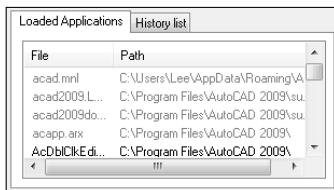


Because AutoCAD is a Windows-based program, it supports many standard Windows features. One of the supported features is the capability to use drag and drop. You can drag and drop application files such as AutoLISP (LSP), VBA (DVB), and ObjectARX (ARX) files directly over the drawing window to load them. This is a fast and easy alternative to the APPLOAD command when you're testing custom programs.

Determining which application files are loaded

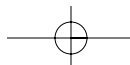
You can tell which application files have already been loaded into AutoCAD by checking the items listed in the Loaded Applications tab (see Figure BC3-3) at the bottom of the Load/Unload Applications dialog box. The list displays all application files loaded into AutoCAD, including AutoLISP, VBA, ObjectARX, ObjectDBX, and .NET files.

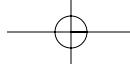
Figure BC3-3:
Loaded application files.



Unloading an application file

Unloading an application file is the reverse of loading an application file. It seems obvious, but AutoCAD doesn't allow you to unload everything that can be loaded because after certain files are loaded, AutoCAD becomes dependent on them while a drawing is open. If you take a look at the Loaded Applications tab, you can see that some items are in black text, whereas a majority of them are in gray text.





BC28 Loading and Unloading Applications

The items in black text can be unloaded without any effect on an open drawing or AutoCAD in general. Unloading a file is rarely required; from time to time, however, you might need to unload an application file. The following steps explain how to unload an application from AutoCAD:

1. On the menu browser or menu bar, click Tools menu⇨Load Applications.

AutoCAD launches the Load/Unload Applications dialog box.

2. In the lower half of the Load/Unload Applications dialog box, click the Loaded Applications tab.

The applications that are currently loaded in AutoCAD are displayed.

3. In the list, select one of the application files to unload.

Although you can select any of the application files in the list, only the application files that are displayed in black text can be unloaded.

4. Click the Unload button, which is located above the Startup Suite section.

Clicking Unload removes the application file from AutoCAD's memory and disables any command that might have been available when the file was loaded.

5. Click Close.

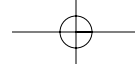
Clicking the Close button exits the Load/Unload Applications dialog box.

Loading an AutoLISP file

Other than the Load/Unload Applications dialog box, AutoCAD has no built-in command that you can use to load an AutoLISP file. However, you can use an AutoLISP function called Load, which allows you to load an AutoLISP file. The syntax for the Load function is

```
(load AutoLISP_filename [Error_message])
```

- ◆ **AutoLISP_filename** is the full path and name to an AutoLISP file. Normally you use a backslash (\) between folders in Windows Explorer, but AutoLISP interprets that as something else, so you need to use double backslashes (\\) or a single forward slash (/). For example, C:\Program Files\AutoCAD 2009\Support\3d.lsp would be C:/Program Files/AutoCAD 2009/Support/3d.lsp.
- ◆ **Error_message** is optional. The value entered in this parameter is echoed at the command prompt if the AutoLISP file is not loaded.



Loading and Unloading Applications **BC29**

An example of the Load function is

```
(load "C:/Program Files/AutoCAD 2009/Support/3d.lsp" "3d.lsp
missing")
```



If Express Tools is installed, a command called LSP is available. The LSP command allows you to load an AutoLISP file as well as see which commands and variables have been defined through AutoLISP. See Book IX, Chapter 4 for more information on Express Tools.

Loading and unloading a VBA project file

VBA project files can be loaded by using the command prompt, a dialog box, or even AutoLISP functions. Only a few of the different methods for loading and unloading a VBA project file are explained in this chapter; the others are covered in Bonus Chapter 5.

Using VBALOAD and VBAUNLOAD with a VBA file

Working with VBA project files in AutoCAD is straightforward. To load a VBA project file, you use the VBALOAD command; to unload it, you use the VBAUNLOAD command. It really can't get too much simpler than that.

To load a VBA application file, do one of the following:

- ◆ On the ribbon, click Tools tab ⇨ Applications panel's title bar ⇨ Load Project.
- ◆ On the menu browser or menu bar, click Tools menu ⇨ Macros ⇨ Load Project.
- ◆ At the command prompt, type **VBALOAD** and press Enter.

Loading a VBA file

The following steps initiate the VBALOAD command and explain how to load an existing VBA project file:

- 1. Display the Open VBA Project dialog box by using either of the two methods just listed.**

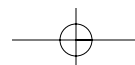
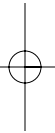
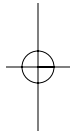
AutoCAD launches the Open VBA Project dialog box, which allows you to browse for an existing VBA application file.

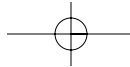
- 2. In the Look In drop-down list, specify the location for the VBA application file.**

The Look In drop-down list allows you to select a folder or a named location such as My Documents to start in.

Bonus
Chapter 3

Using Custom
Programs





BC30 Loading and Unloading Applications

- a. **If the VBA project file is located in a folder below the location that you selected in the Look In drop-down list, double-click the folder in the list box below the Look In drop-down list.**

The list box displays any folders or files located under the folder selected in the Look In drop-down list. The types of files that are displayed in the list box depend on the file filter selected in the Files of Type drop-down list.

- b. **Keep double-clicking folders until you navigate to the folder that contains the project file you want to load. Select the project file from the list box.**

The VBA project file is highlighted in the list box and its name added to the File Name text box.

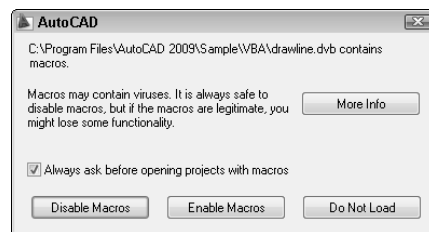
3. Click Open.

The Open VBA Project dialog box closes and AutoCAD starts loading the VBA project file.

4. In the AutoCAD message box, click Enable Macros to finish loading the VBA project file.

The AutoCAD message box (see Figure BC3-4) is displayed when macro virus protection is enabled. This option protects you and your drawings from macro viruses that can become embedded in a drawing file. This option and others for VBA are explained in Bonus Chapter 5. After you click Enable Macros, the VBA project file is added to the Projects list box of the VBA Manager dialog box.

Figure BC3-4:
The virus protection warning displayed when loading a VBA project.

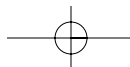


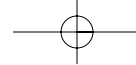
5. Click Close.

Clicking the Close button exits the VBA Manager dialog box.



The VBAMAN command can be used to display a dialog box that allows you to access a majority of the VBA environment options. The VBAMAN command is explained in Bonus Chapter 5.





Loading and Unloading Applications BC31

Loading a VBA project file by using AutoLISP

You can load a VBA project file by using the AutoLISP function `vl-vbaload`. The `vl-vbaload` function works just like the AutoLISP `Load` function discussed in the section “Loading an AutoLISP file” earlier in this chapter. The syntax for the `vl-vbaload` function is

```
(vl-vbaload VBA_filename)
```

`VBA_filename` should be the full path and name to a VBA project file. Normally you use a backslash (\) between folders in Windows Explorer. But with AutoLISP, the backslash is interpreted as something else, so you need to use double backslashes (\\) or a single forward slash (/).

An example of the `vl-vbaload` function is

```
(vl-vbaload "C:/Program Files/AutoCAD 2009/Sample/VBA/  
drawline.dvb")
```

If the project file is loaded, the value returned by the function is the path to the file, and if the project file is not loaded, the value `nil` is returned.

Unloading a VBA project file

The `VBAUNLOAD` command is used to unload VBA project files and is accessible only from the command prompt. The following procedure explains how to unload a VBA project file.

1. Type `VBAUNLOAD` at the command prompt and press Enter.

AutoCAD displays the command prompt:

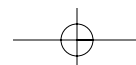
```
Unload VBA Project:
```

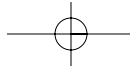
2. At the command prompt, enter the full path and name of the VBA project file.

For example, to unload a VBA project file called `Drawline.DVB` located in the directory `C:\Program Files\AutoCAD 2009\Sample\VBA`, you enter **`C:\Program Files\AutoCAD 2009\Sample\VBA\Drawline.DVB`**.

Loading and unloading an ObjectARX file

Unlike with AutoLISP files, AutoCAD offers a built-in command that allows you to load an ObjectARX or ObjectDBX file. This command is called `ARX`, which makes sense, considering the type of application files that it can load or unload. There are also two AutoLISP functions that can be used to load and unload an ObjectARX or ObjectDBX file.





BC32 Loading and Unloading Applications

Using the ARX command

The ARX command offers a number of different options, but the two most important are Load and Unload. The command prompt for the ARX command looks like this:

Enter an option [?/Load/Unload/Commands/Options]:

The command options for the ARX command are described as follows:

- ◆ ? lists all the ObjectARX and ObjectDBX application files that are currently loaded in AutoCAD.
- ◆ Load displays the Select ARX/DBX File dialog box to specify an ObjectARX or ObjectDBX application file to load.



It's possible to load an ObjectARX or ObjectDBX application file through a command macro or a script file by setting the FILEDIA system variable to a value of 0. When FILEDIA is set to 0, the following command prompt is displayed for the ARX command:

Enter ARX/DBX file name to load:

- ◆ Unload allows you to unload an ObjectARX or ObjectDBX application file that is currently loaded and displays the command prompt:

Enter ARX/DBX file name to unload:

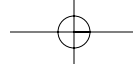
- ◆ Commands lists all the commands that are exposed in the loaded ObjectARX application files. ObjectDBX application files usually do not contain user interface elements and commands.
- ◆ Options contains three suboptions that are used mainly for developing an ObjectARX application. The three options are Group, Classes, and Services. If you want to know more about these options, look up the ARX command in the AutoCAD Online Help system under the Command Reference Guide.

Loading an ObjectARX file by using AutoLISP

You can load an ObjectARX or ObjectDBX application file by using the AutoLISP function Arxload. Arxload works just like the AutoLISP function Load, which I discuss in the section "Loading an AutoLISP file" earlier in this chapter. The syntax of the Arxload function is

```
(arxload ObjectARX/ObjectDBX_filename [Error_message])
```

- ◆ ObjectARX/ObjectDBX_filename is the full path and name to the ObjectARX or ObjectDBX file. Normally you use a backslash (\) between folders in Windows Explorer, but AutoLISP interprets the backslash as something else, so you need to use double backslashes (\\) or a single forward slash (/).



Loading and Unloading Applications **BC33**

- ◆ `Error_message` is optional. The value entered in this parameter is echoed at the command prompt if the ObjectARX or ObjectDBX file is not loaded.

An example of the `Arxload` function is

```
(arxload "C:/Program Files/AutoCAD 2009/geom3d.arx"
"geom3d.arx missing")
```

If the application file is loaded, the return value of the function is the file name. If the application file is not loaded, the error message is returned.

Unloading an ObjectARX file by using AutoLISP

You can unload an ObjectARX or ObjectDBX file by using the AutoLISP function `Arxunload`. The syntax of the `Arxunload` function is

```
(arxunload ObjectARX/ObjectDBX_filename [Error_message])
```

- ◆ `ObjectARX/ObjectDBX_filename` is the file name of the ObjectARX or ObjectDBX file to unload.
- ◆ `Error_message` is optional. The value entered in this parameter is echoed at the command prompt if the ObjectARX/ObjectDBX file is not unloaded.

An example of the `Arxunload` function is

```
(arxunload "geom3d.arx" "geom3d.arx cannot be unloaded")
```

If the application file is unloaded, the return value of the function is the file name. If the application file is not unloaded, the error message is returned.

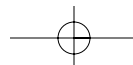
Loading a .NET file

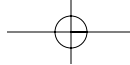
Unlike the other types of programming files that I discuss earlier in this chapter, .NET files can't be loaded into AutoCAD with the `APPLOAD` command. It might seem like .NET files were left out of the loop, but you can still load them with the `NETLOAD` command. By default, this command displays the Choose .NET Assembly dialog box, which allows you to browse for a .NET application file and load it. After it is loaded, the .NET application file can't be unloaded from AutoCAD unless AutoCAD is restarted.



You can load a .NET application through a command macro or a script file by setting the `FILEDIA` system variable to a value of 0 before using the `NETLOAD` command. When `FILEDIA` is set to 0 and the `NETLOAD` command is used, the following command prompt is displayed:

Assembly file name:





BC34 Automatically Loading Application Files



As a workaround to not being able to load .NET application files by using the Startup Suite in the Load/Unload Applications dialog box, you can create an AutoLISP file that uses the NETLOAD command. For information on the Startup Suite, see the next section.

An example of loading a .NET application file with AutoLISP is as follows:

```
(setvar "FILEDIA" 0)
(command ".NETLOAD" "C:/DotNet/Samples/DockingPalette.dll")
(setvar "FILEDIA" 1)
```

Automatically Loading Application Files

Just being able to load custom programs is a huge step toward increasing your productivity, but having to load application files again and again can be counterproductive. You can have AutoCAD do the work of loading custom programs for you in a couple of different ways. One is by using the Load/Unload Applications dialog box, and the other is through specially named application files that AutoCAD loads automatically.

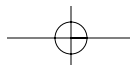
Using the Startup Suite

The Startup Suite (see Figure BC3-5) area is located in the lower-right corner of the Load/Unload Applications dialog box. The Startup Suite controls which application files are loaded at the startup of AutoCAD and with each drawing that is opened. The Startup Suite has the same limitations as loading application files through the Load/Unload Applications dialog box. The Startup Suite is limited to AutoLISP (LSP, VLX, and FAS), VBA (DVB), ObjectARX (ARX), and ObjectDBX (DBX) application files, so you can't directly load a .NET application file by using the Startup Suite.

Figure BC3-5:
The Startup Suite helps to automatically load application files.



To add an application file to the Startup Suite, follow these steps:



Automatically Loading Application Files **BC35**

1. On the menu browser or menu bar, click Tools menu⇨Load Applications.

AutoCAD launches the Load/Unload Applications dialog box.

2. In the Startup Suite area located in the lower-right corner of the Load/Unload Applications dialog box, click the Contents button.

The Startup Suite dialog box (see Figure BC3-6) is displayed.

Figure BC3-6:
The Startup Suite dialog box.



3. Click the Add button.

The Add File to Startup Suite dialog box is displayed.

4. In the Files of Type drop-down list, specify the type of application file to use for filtering the list box.

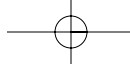
The selected option controls the type of application files that are displayed in the list box. The default selection is AutoCAD Apps, which lists all available application files. The drop-down list box allows you to filter on AutoLISP, VBA, ObjectARX, and ObjectDBX files.

5. In the Look In drop-down list, specify the location for the application file.

The Look In drop-down list allows you to select a folder or a named location such as My Documents to start in.

a. If the application file is located in a folder below the location that you selected in the Look In drop-down list, double-click the folder in the list box below the Look In drop-down list.

The list box displays any folders or files located under the folder selected in the Look In drop-down list. The types of files displayed in the list box depend on the file filter selected in the Files of Type drop-down list.



BC36 Automatically Loading Application Files

- b. Keep double-clicking folders until you navigate to the folder that contains the application file you want to load. Select the application file from the list box.**

The application file is highlighted in the list box and its name added to the File Name text box.

- 6. Click Add.**

The application file is added to the Startup Suite dialog box and is loaded into AutoCAD. You return to the Startup Suite dialog box.

- 7. Click Close.**

Click Close to exit the Startup Suite dialog box and return to the Load/Unload Applications dialog box.

- 8. Click Close again.**

Clicking the Close button exits the Load/Unload Applications dialog box.

If you no longer want a file to be automatically loaded through the Startup Suite, click the Contents button in the Startup Suite section of the Load/Unload Applications dialog box. Then from the Startup Suite dialog box, select the application file to remove and click Remove. Click Close twice to exit the open dialog boxes.

Getting AutoCAD to do some of the work

The Startup Suite is great for automatically loading application files for a small group of AutoCAD users, but isn't very efficient for large groups. This is why AutoCAD is designed to look for application files with specific types and names when it starts and when a drawing is created or opened.

Automatically loading AutoLISP files

AutoCAD looks for several different AutoLISP application files when it starts, when a new drawing is created or opened, and when a Customize User Interface (CUI) file is loaded. These specially named files are loaded in a specific order to allow you the option of overriding standard AutoCAD commands and defining your own custom commands. These specially named files can be great for loading AutoLISP, VBA, ObjectARX, ObjectDBX, and .NET application files. As explained in previous sections, you can use different AutoLISP functions to load many of the different types of application files that AutoCAD supports.

Table BC3-2 lists the specially named AutoLISP files that AutoCAD looks for when it starts and when a new drawing is created or opened. The table is arranged in the order that the files are loaded, from first to last.

Automatically Loading Application Files **BC37**

Table BC3-2	Load Order for Specially Named AutoLISP Files
<i>File name</i>	<i>Description</i>
ACAD2009.LSP	The file is installed with AutoCAD and is loaded only once per AutoCAD session.
ACAD.LSP	The file is not installed with AutoCAD but rather needs to be created manually with a text editor such as Notepad. Like ACAD2009.LSP, this file is loaded only once per AutoCAD session, unless the system variable ACADLSPASDOC is set to a value of 1.
ACADDOC2009.LSP	The file is installed with AutoCAD and is loaded with each drawing that is created or opened.
ACADDOC.LSP	The file is not installed with AutoCAD, but rather needs to be created manually with a text editor like Notepad. Like ACADDOC2009.LSP, this file is loaded with each drawing that is created or opened.
ACAD.MNL	The file is installed with AutoCAD and is loaded when the customization file ACAD.CUI is loaded. ACAD.CUI is the default customization file that is loaded with AutoCAD.
< cui_filename >.MNL	When a partial or enterprise customization file is being loaded, AutoCAD looks for an .MNL file that has the same name as the CUI file and loads it.



If you don't know a lot about AutoLISP, but you happen to have been given some AutoLISP files to use, you can use the Customize User Interface (CUI) Editor to associate AutoLISP files to a CUI file under the LISP Files node. After a CUI file is loaded into AutoCAD, any associated AutoLISP files are loaded automatically. You can use this as an alternative to using the Startup Suite in the Load/Unload Applications dialog box or using the AutoLISP Load function to load an AutoLISP file in combination with the < cui_filename >.MNL file. To find out more about customization files, see Book IX, Chapter 3.

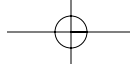
Automatically loading VBA files

When AutoCAD starts, it looks for a VBA application file named acad.dvb and loads it automatically. If a public procedure named AcadStartup is defined in one of the standard code modules, it runs automatically after the file has been loaded.

If you're using VBA for your custom programs, using the AcadStatup procedure is a great way to initialize some variables and do anything else you need to do when AutoCAD starts. The AcadStartup procedure is run only when AutoCAD starts, so if you need to perform a task every time a new drawing is created or opened, you need to handle that through events. I explain more about using VBA in Bonus Chapter 5.

**Bonus
Chapter 3**

**Using Custom
Programs**



BC38 Running a Program in an Application File

Automatically loading ObjectARX files

Similar to how AutoCAD looks for specially named AutoLISP or VBA files, AutoCAD looks for a file to load ObjectARX application files when it starts. The file it looks for is named acad.rx. If AutoCAD finds an acad.rx file, it steps through the file and loads each of the ObjectARX files listed in it. The acad.rx is a file that must be created manually with a text editor such as Notepad. Following is an example of what an acad.rx file might look like:

```
C:\ObjectARX\Utilities\customInsert16.arx  
geom3d.arx
```

It is possible to use the full path to the ObjectARX file or just the file name. If you use only the file name to load the ObjectARX file, the file must be found in one of the support paths listed under Support File Search Path on the Files tab in the Options dialog box.

Running a Program in an Application File

You usually load an application file because you want to run one of the custom programs defined in the file. Custom programs are usually accessible in two ways: as a command or as a function. A custom command is run by typing the command name at the command prompt and pressing Enter to execute it. A custom command can be created by using AutoLISP, ObjectARX, or .NET, but not VBA.

A custom function, if it is exposed through AutoLISP or ObjectARX, must be enclosed in parentheses to be executed. If the function is defined with VBA, it is known as a procedure and can be executed with the VBAMAN command. The VBAMAN command, along with other methods used to execute a procedure defined in a VBA project file, are covered in greater depth in Bonus Chapter 5.

To execute a procedure in a VBA project file, follow these steps:

- 1. On the menu browser or menu bar, click Tools menu⇨Macros⇨VBA Manager.**

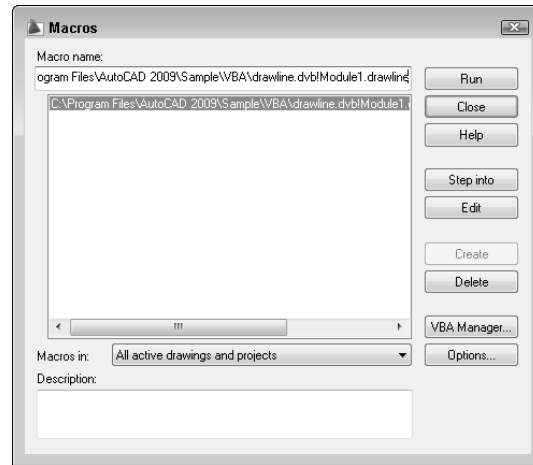
AutoCAD launches the VBA Manager dialog box.

- 2. On the right side of VBA Manager, click the Macros button .**

The Macros dialog box (see Figure BC3-7) is displayed and shows all the available procedures in the loaded VBA project files.

Running a Program in an Application File **BC39**

**Figure
BC3-7:**
Putting a
procedure
(or macro)
to work.



3. In the list box in the middle of the dialog box, select the procedure that you want to run.

The selected procedure is highlighted in the list box. You can resize the Macros dialog box by clicking one of the corners of the dialog box and dragging outward. This allows you to see more of the names of the procedures in the list box.

4. Click Run.

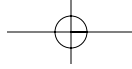
The procedure starts executing. Follow any on-screen prompts that might be displayed while the procedure is being executed.



You can use the `-VBARUN` command to execute a procedure in a VBA project file from a command macro or a script file. I explain the `-VBARUN` command in Bonus Chapter 5.

Bonus
Chapter 3

Using Custom
Programs



BC40 AutoCAD 2009 & AutoCAD LT 2009 All-in-One Desk Reference

