

# Deploying applications

# 9

## IN THIS CHAPTER

- **ActiveX registration**
- **MapObjects file requirements**
- **Using the Visual Basic Package and Deployment Wizard**
- **Using the SETUP.LST file**
- **Testing the setup program**
- **Distributing data with your application**
- **Sharing MapObjects components**

Once you have completed your MapObjects application, you may want to distribute it to others.

To distribute an application, you can create a *setup program*. A setup program copies all the necessary files needed by an application and registers all the pertinent software objects to a target machine; it replaces the need for you to manually copy and register each file that is needed to deploy an application.

Depending on the programming environment you are using, a tool, such as a wizard, may be available to create a setup program. There is also software specifically for creating setup programs. The language or application you use to create these programs is entirely up to you.

This chapter shows how to use the Visual Basic 6.0 Package and Deployment Wizard, formerly the Visual Basic 5.0 Setup Wizard, to create a setup program. Although the focus is on Visual Basic, the concepts presented are relevant for any MapObjects deployment.

The information in this chapter focuses on installation requirements for MapObjects and does not address the requirements of additional components or run-time libraries that your application may also need. You should use your programming environment documentation in conjunction with this chapter to properly distribute your applications.

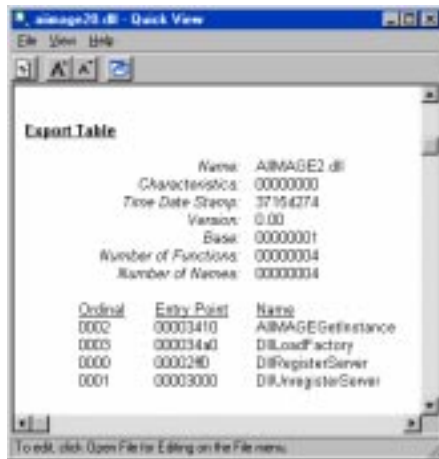
## ActiveX registration

When deploying an application, some MapObjects ActiveX components will need to be registered in the target computer's system registry. This section discusses how to determine if a file is self-registering and if so, its dependencies. Also discussed in this section are tools you can use to register ActiveX components.

### Determining if a file is self-registering

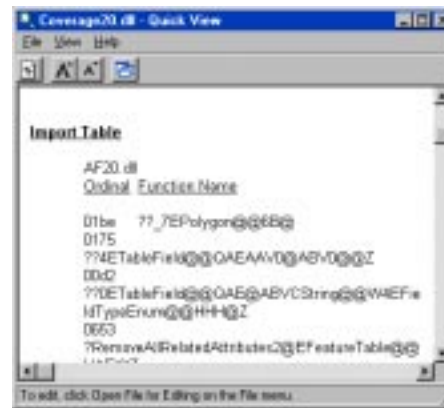
You can determine if a file is self-registering by browsing the file with Microsoft's QuickView utility to see if it contains the two functions, DllRegisterServer and DllUnregisterServer. Components are self-registering if they have an entry point for the "DllRegisterServer" function.

The QuickView utility is installed during the installation of your operating system. Using QuickView, look for the DllRegisterServer and DllUnregisterServer functions listed in the Name column of the Export Table section.



### Determining a file's dependencies

Before a self-registering file can be registered, you must make sure that all its dependencies are installed and accessible. Using QuickView to view the file, any files listed under Import Table are dependencies of the file. Be sure to scroll down to view all the dependencies.



In the illustration above, which depicts the MapObjects version 2.0 component Coverage20.dll, the file AF20.dll is listed under Import Table; that file must be accessible before attempting to register Coverage20.dll. In order for AF20.dll to be accessible, it must be copied to the same folder that Coverage20.dll is in; AF20.dll, Coverage20.dll, and the other MapObjects components should be installed into the target computer's CommonFiles folder. See 'MapObjects file requirements' section for more details on the location of installed components.

In addition to viewing a file with the QuickView utility, you can also use the Dependency Walker utility to determine a file's list of dependencies. You can learn more about Dependency Walker and



download the latest version from [msdn.microsoft.com/library/techart/redistribvc6.htm](http://msdn.microsoft.com/library/techart/redistribvc6.htm).

Both the Import Table of a file and the Dependency Walker utility do not list delayed loading dll's if they are present. You can learn more about delayed loading dll's from MSDN.

### Tools for registering files

You can use either RegSvr32.exe or the Register In Menu.reg tool to register files

#### Using RegSvr32.exe

If your setup program generator cannot self-register files, you can add lines to your setup program that call RegSvr32.exe, a freely distributable program provided by Microsoft for registering ActiveX components. RegSvr32.exe is installed with the MapObjects installation. If you accepted the defaults during installation, you can find it in the C:\Program Files\ESRI\MapObjects2 folder.

You can add a call to RegSvr32.exe in your setup program and pass the name of the component file to be registered. You can also use a /s at the beginning of the command line to make the process silent; otherwise, RegSvr32.exe will display a success or failure dialog after the self-registration attempt.

```
RegSvr32.exe /s C:\Program Files\Common Files\MO20.ocx
```

Refer to your setup program generator documentation for information on how to call external programs.

You can also perform the registration of individual files by creating a shortcut to RegSvr32.exe on your desktop and dragging a file onto the RegSvr32.exe icon. Using a shortcut can avoid the typing errors that may occur if you use RegSvr32.exe on the command line.

### Registration tool from MapObjects

The MapObjects installation also includes a utility for registering files; this utility allows you to register or unregister files from your Windows Explorer by right-clicking on a file.

You can find this utility at  
C:\Program Files\ESRI\MapObjects2\Register In Menu.reg.

## MapObjects file requirements

The files listed here under “Core MapObjects components” and “Core Microsoft components” are required to deploy any MapObjects application. Depending on the functionality included with your MapObjects application, you will need to deploy some or all of the files listed in the other sections. For example, if your application does not display images, you will not need to include any of the components for image display.

You may under no circumstances distribute the file named MO20.LIC. Distribution of this file constitutes a violation of your license agreement with ESRI.

Files marked with an asterisk (\*) in the sections below require that they be self-registered on the target system; the other files need only be copied. See the “ActiveX registration” section for details on self-registering files.

Keep in mind that some of the files listed here are dependencies for other files during self-registration. In general, it is a good idea to copy all files before your setup program begins self-registration to make sure all dependent files have been copied and are accessible.

### Core MapObjects components

All applications using MapObjects must be redistributed with *at least* the following core component .DLL files:

MO20.ocx \*  
AF20.dll  
Shape20.dll \*  
PE.dll  
SG.dll  
Mtch.dll

### Core Microsoft components

All applications must be redistributed with *at least* the following core Microsoft component .DLL files:

Mfc42.dll \*  
Msvcp60.dll  
Msvcrt.dll  
Msvcrt40.dll  
OLEAUT32.dll \*  
OLEPRO32.dll \*  
stdole2.tlb

### ARC/INFO coverage components

If your application has ARC/INFO coverages, you must install the following ARC/INFO component:

Coverage20.dll \*

### SDE components

If your application has SDE data, you must install the following SDE components:

SDE30.dll  
MoSDE302.dll \*  
MoSDE302Table.dll \*

### Image components

If your application has image layers (images or Grids), you must install the following image components:

AiImage20.dll \*  
Mobase.dll  
Mofeat.dll



Mogridio.dll  
dforrt.dll  
Libtiff.dll  
Elib.dll  
Efileio.dll  
Ergxlib.dll  
Ethrlib.dll  
ISO8211.dll

Also, if your application requires ERDAS IMAGINE® support, you must install all files from the MapObjects2 folder called Mo\_image, to a folder below the Elib.dll. For example, if the MapObjects components are installed in the Common Files folder, the ERDAS IMAGINE support would be copied to C:\Program Files\Common Files\ESRI\Mo\_image.

### MrSID components

If your application uses MrSID™ format images, you must install the following components:

MoMrSID.dll \*  
MrSIDD.dll

### VPF components

If your application has VPF files, you must install the following VPF component:

vpf20.dll \*

### CAD components

If your application has CAD files, you must install the following CAD components:

Cad20.dll \*  
Edat.dll  
Panio.dll

### INFO Tables

If you use INFO tables in your application, you will need to install the following component:

InfoTable20.dll \*

### Microsoft Data Access Objects (DAO) Tables

Microsoft DAO can be used to connect to Microsoft Access databases and to many other standard databases through JET Indexed Sequential Access Method (ISAM) extensions. If you use this functionality in your MapObjects application, you will need to install the following components:

JetTable20.dll \*  
DAO35.dll  
MSJET35.DLL  
MSJINT35.DLL  
MSJTER35.DLL  
MSRD2X35.DLL  
MSRDO20.DLL  
MSREPL35.DLL  
DAO350.DLL

Consult the Microsoft documentation for details on the redistribution and licensing of DAO. If your application uses



DAO, Visual Basic's PDW automatically adds DAO to your installation.

If you prefer to install DAO 3.5 yourself, a DAO 3.5 installation program is included with MapObjects Version 2.0. It is advised that you consult the Microsoft Developer Network (MSDN) to learn how to correctly install and register DAO components. The DAO components must be installed in the proper locations with version checking to ensure that newer versions of the files are not overwritten by your installation.

If you accepted the defaults when installing MapObjects, you can find the DAO 3.5 installation program in the folder C:\Program Files\ESRI\MapObjects2\DAO.

### **Install location for MapObjects components**

The MapObjects components are installed in the target computer's Common Files\ESRI folder. You can determine the path to the Common Files folder by checking the following registry value:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\CommonFilesDir

When you are deploying your MapObjects application, even though the component files can be installed and registered in any folder, the convention recommended by ESRI is to install and register the components in the Common Files\ESRI folder. Because MapObjects shares components with other ESRI software such as ArcView GIS, installing the component files in the Common Files\ESRI folder avoids conflicts with other ESRI software during installation and uninstallation of a MapObjects application.

# Using the Visual Basic Package and Deployment Wizard

The steps outlined here show how to use the Visual Basic Package and Deployment Wizard (PDW) to generate a setup program.

When the Package button is clicked on the first dialog, the wizard will pause to collect the files that are associated with the application; these files are read from a dependency file.

The wizard will determine which files should be included in the package from which files are checked ON as references and components when your application was created. The wizard will include all components and references that are checked ON, regardless of whether or not they are being used in the application. During the creation of the setup program, you will want to remove files that are checked ON but not being used.

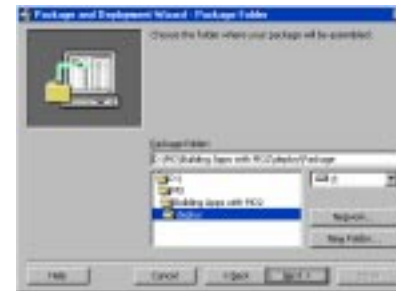
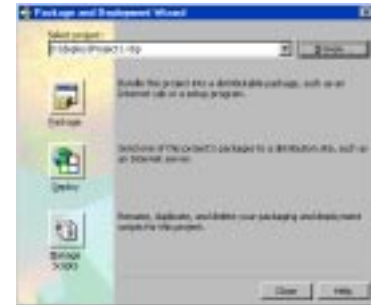
### Tip

#### Getting help

You can get help with the Package and Deployment Wizard from MSDN or [support.microsoft.com/support/vbasic/pdw.asp](http://support.microsoft.com/support/vbasic/pdw.asp).

DEPLOYING APPLICATIONS

1. From your Start Menu, open the Package and Deployment Wizard.
2. Click Browse to navigate to the project you want to package.
3. Click the Package button.
4. Click Standard Setup Package as the Package type and click Next.
5. Click the folder where the package will be located and click Next.
6. If you are using DAO 3.5 in your application, click the drivers in the Available drivers list that you would like to include. Use the arrows to move the drivers to the Included drivers list.



The Package and Deployment Wizard is a *guide* for the deployment process; it is your responsibility as the application developer to ensure that all applicable components and support files are copied and registered in the target computer's registry.

In Step 7, you should verify that all the files needed for your application (as listed in the "MapObjects file requirements" section earlier in this chapter) are included in the list on the Included files dialog.

Other files that you will want to add to your package can be included by clicking the Add button and selecting the appropriate files.

Files that you may want to include are any data files that your application uses and any ESRI TrueType font files (.ttf) that your application is using. The TrueType font files will be installed into your operating system's Fonts folder, e.g., C:\WINNT\Fonts.

7. Uncheck files to exclude them from the package. Click Add to include those not already listed.

See the side panel on this page for more information regarding which files should be included here.

8. Depending on the size of the cab file you are creating, you can click Single cab or Multiple cab. Click Next to continue.

9. Enter a banner title that you would like to appear when your installation program runs.

10. Click the start menu groups and items that should be created and click Next.



7



8



9



10



In Step 11, all of the MapObjects components should have an install location of \$(CommonFiles)\ESRI. See the “MapObjects file requirements” section for more details on the install location of the MapObjects components.

Marking a file as *shared* is necessary if the target computer is running other applications that use the MapObjects components. See the “Sharing MapObjects components” section later in this chapter for more details.

When you click Finish, the wizard will generate the setup program in the specified directory.

A resulting text file, SETUP.LST, will also be generated in the setup directory. This file contains the instructions that will be followed by the setup program during installation of your application. See the next section, “Using the SETUP.LST file,” for more details on how to use this file.

#### Tip

#### Getting help

*In addition to the instructions in this chapter, refer to MSDN or your programming environment’s online Help for topics related to deploying an application.*

#### DEPLOYING APPLICATIONS

11. Use this panel to change the install locations of files.

The possible install locations include:

\$(CommonFiles)

\$(Shared)

\$(AppPath)

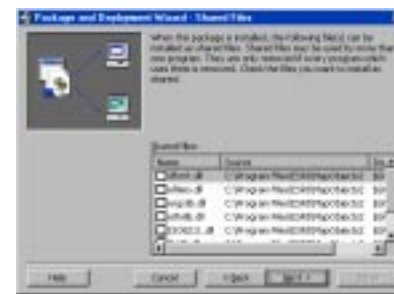
\$(WinSysPath)

Click Next to continue.

12. Use this panel to indicate whether or not a file is shared.
13. Click Finish to generate the setup program.



11



12



13

## Using the SETUP.LST file

The SETUP.LST file contains all the files and associated instructions that will be followed by the setup program during installation of your application. After using the Visual Basic PDW, you can find your application's SETUP.LST file in the folder where the package was assembled. You should use the SETUP.LST file as a tool to understand what has been included in your application's package. You can read it as a kind of report as to which files were included in the package, where they will be installed, and if they are self-registering.

### Taking a closer look at the SETUP.LST file

Each section in the SETUP.LST file begins with bracketed text such as [Bootstrap]. Of particular importance to you are the [Bootstrap Files] and [Setup1 Files] sections that list the files that have been included in the package.

Each line in the [Bootstrap Files] and [Setup1 Files] sections pertains to a different file included in the package. Each line contains the following information:

- The name of the file
- The location where the file will be copied on the target computer
- Whether or not the file is self-registering
- Whether or not the file is shared
- The file's date and time
- The file's size
- The file's version number

The location where files can be installed are

- \$(WinSysPath): the target computer's Windows directory.
- \$(AppPath): the folder where the application is installed on the target computer.

- \$(CommonFiles): the target computer's Common Files folder. The path to the Common Files folder can be determined by checking the following registry value:  
HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\CommonFilesDir
- \$(Font): the target computer's Fonts directory.

If a file is shared, the line will have \$(Shared); this indicates that the component can be shared between multiple programs and that DLL counting should be used when installing and uninstalling. Refer to the "Sharing MapObjects components" section later in this chapter for more details about shared components.

If a file is self-registering, the line will have \$(DLLSelfRegister) and \$(EXESelfRegister); this tells the setup program to register these components in the target system's registry.

### Editing the SETUP.LST file

The SETUP.LST file is editable; however, edits should be made with caution.

The section of the file that is typically and safely edited is the [Setup] section. This section includes the information on where the application will be installed on the target machine. You may want to change the directory where the application will be installed by editing the line that begins with: DefaultDir=.

Making edits to other sections of this file is not recommended.

### Example SETUP.LST file

An example of a SETUP.LST file is shown on the following pages. Because each application has its own set of functionality, your SETUP.LST file will not be exactly the same as the example shown in this chapter; your application may use different files and different version numbers for the .DLL files.



```
[Bootstrap]
SetupTitle=Install
SetupText=Copying Files, please stand by.
CabFile=test.CAB
Spawn=Setup1.exe
Uninstall=st6unst.exe
TmpDir=msftqws.pdw
Cabs=1
```

```
[Bootstrap Files]
File1=@MSVCRT40.DLL,$(WinSysPathSysFile),,5/31/98 1:00:00 AM,326656,4.21.0.0
File2=@vb6stkit.dll,$(WinSysPathSysFile),,2/1/99 4:36:38 PM,101888,6.0.83.97
File3=@COMCAT.DLL,$(WinSysPathSysFile),$(DLLSelfRegister),,5/31/98 1:00:00 AM,22288,4.71.1460.1
File4=@STDOLE2.TLB,$(WinSysPathSysFile),$(TLBRegister),,12/18/98 2:27:10 PM,17920,2.40.4268.1
File5=@ASYCFILT.DLL,$(WinSysPathSysFile),,12/18/98 2:27:44 PM,147728,2.40.4268.1
File6=@OLEPRO32.DLL,$(WinSysPathSysFile),$(DLLSelfRegister),,12/18/98 3:17:36 PM,164112,5.0.4268.1
File7=@OLEAUT32.DLL,$(WinSysPathSysFile),$(DLLSelfRegister),,12/18/98 2:27:46 PM,598288,2.40.4268.1
File8=@MSVBVM60.DLL,$(WinSysPathSysFile),$(DLLSelfRegister),,9/25/98 1:00:00 AM,1409024,6.0.82.68
```

```
[IconGroups]
Group0=Test
PrivateGroup0=True
Parent0=$(Programs)
```

```
[Test]
Icon1="test.exe"
Title1=Test Application
StartIn1=$(AppPath)
```

[Setup]

Title=Rob's Test 1.1f

DefaultDir=\$(ProgramFiles)\Project1

AppExe=test.exe

AppToUninstall=test.exe

[Setup1 Files]

File1=@WINGDNG2.TTF,\$(Font),,\$(Shared),11/21/96 9:37:00 AM,59696,0.0.0.0

File2=@ROADS.DBF,\$(AppPath),,,12/18/97 5:42:36 PM,791491,0.0.0.0

File3=@WASH.BMP,\$(AppPath),,,12/18/97 5:42:36 PM,6320014,0.0.0.0

File4=@ROADS.SHX,\$(AppPath),,,12/18/97 5:42:36 PM,52396,0.0.0.0

File5=@ROADS.SHP,\$(AppPath),,,12/18/97 5:42:36 PM,668876,0.0.0.0

File6=@wash.bmpw,\$(AppPath),,,12/18/97 5:42:36 PM,222,0.0.0.0

File7=@Mfc42.dll,\$(WinSysPath),\$(DLLSelfRegister),\$(Shared),9/26/98 1:00:00 AM,995383,6.0.8267.0

File8=@MSVCRT20.DLL,\$(WinSysPathSysFile),,,5/31/98 1:00:00 AM,253952,2.11.0.0

File9=@msvcrt10.dll,\$(WinSysPath),,,12/19/97 2:42:36 AM,210944,0.0.0.0

File10=@msvcrt.dll,\$(WinSysPath),,\$(Shared),1/5/99 1:00:00 AM,266293,6.0.8337.0

File11=@MSVCP60.DLL,\$(WinSysPath),,\$(Shared),6/17/98 12:00:00 AM,401462,6.0.8168.0

File12=@Shape20.dll,\$(CommonFiles)\ESRI,\$(DLLSelfRegister),\$(Shared),4/15/99 12:50:48 PM,86016,2.0.0.11

File13=@Sg.dll,\$(CommonFiles)\ESRI,\$(Shared),3/29/99 9:26:12 PM,248320,3.0.2.1

File14=@Pe.dll,\$(CommonFiles)\ESRI,\$(Shared),3/29/99 9:26:12 PM,524288,4.0.0.0

File15=@mtch.dll,\$(CommonFiles)\ESRI,\$(Shared),3/29/99 9:26:10 PM,294912,3.2.3.0

File16=@mogridio.dll,\$(CommonFiles)\ESRI,\$(Shared),3/29/99 9:26:10 PM,502784,2.0.0.8

File17=@mofeat.dll,\$(CommonFiles)\ESRI,\$(Shared),3/29/99 9:26:10 PM,926208,2.0.0.8

File18=@mobase.dll,\$(CommonFiles)\ESRI,\$(Shared),3/29/99 9:26:08 PM,475648,2.0.0.8

File19=@libtiff.dll,\$(CommonFiles)\ESRI,,,4/14/99 2:50:48 PM,352256,0.0.0.0

File20=@ISO8211.dll,\$(CommonFiles)\ESRI,,,4/14/99 2:49:18 PM,118784,0.0.0.0

File21=@ethrlib.dll,\$(CommonFiles)\ESRI,,,3/29/99 9:26:08 PM,8704,0.0.0.0

File22=@ergxlib.dll,\$(CommonFiles)\ESRI,,,3/29/99 9:26:08 PM,9728,0.0.0.0

File23=@elib.dll,\$(CommonFiles)\ESRI,,\$(Shared),3/29/99 9:26:08 PM,3432448,8.3.1.0  
File24=@efileio.dll,\$(CommonFiles)\ESRI,,,3/29/99 9:26:04 PM,6144,0.0.0.0  
File25=@dforrt.dll,\$(CommonFiles)\ESRI,,,3/29/99 9:26:02 PM,410624,0.0.0.0  
File26=@aimage20.dll,\$(CommonFiles)\ESRI,\$(DLLSelfRegister),\$(Shared),4/15/99 12:50:14  
PM,339968,2.0.0.11  
File27=@AF20.dll,\$(CommonFiles)\ESRI,,\$(Shared),4/15/99 12:49:58 PM,675840,2.0.0.11  
File28=@Mo20.ocx,\$(CommonFiles)\ESRI,\$(DLLSelfRegister),\$(Shared),4/15/99 12:51:12 PM,892928,2.0.0.11  
File29=@test.exe,\$(AppPath),,,5/21/99 2:25:40 PM,24576,1.0.0.0

## Testing the setup program

### Where to test the setup program

In order to properly test your setup program, ideally you should deploy your application to a computer that does not have MapObjects or Visual Basic installed.

It is not an adequate test of the setup program if you deploy on the same computer you used to develop the application because all of the necessary files have already been installed and registered.

### Using the ST6UNST.LOG file

After your application has been deployed to a target machine, an ST6UNST.LOG file is generated; this file is a record of what happened during the installation. Both this log file and the SETUP.LST file are useful diagnostic tools for confirming that all of the necessary files have been included in your setup package.

Here is an excerpt from the log file:

```
ACTION: SystemFile: "C:\WINDOWS\SYSTEM\MSVCRT40.DLL"  
(File currently on disk was already up to date)
```

```
ACTION: SharedFile: "C:\Program Files\Common Files\ESRI\Mo20.ocx"  
(File was not found or was an older version -- new file copied)
```

### Useful topics about deploying applicatons

As mentioned already, MSDN is an invaluable resource for learning more about setup programs and the process of deploying applicatons. Some topics of particular interest can be found at:

[msdn.microsoft.com/library/sdkdoc/buildapp/setup\\_3ks4.htm](http://msdn.microsoft.com/library/sdkdoc/buildapp/setup_3ks4.htm)

[msdn.microsoft.com/library/devprods/vs6/vb/html/vbcondeploymentfeatures.htm](http://msdn.microsoft.com/library/devprods/vs6/vb/html/vbcondeploymentfeatures.htm)

[msdn.microsoft.com/library/devprods/vs6/vb/html/vbconsetupwizardfeatures.htm](http://msdn.microsoft.com/library/devprods/vs6/vb/html/vbconsetupwizardfeatures.htm)

[support.microsoft.com/support/kb/articles/q192/4/61.asp](http://support.microsoft.com/support/kb/articles/q192/4/61.asp)

[support.microsoft.com/support/kb/articles/Q140/5/70.asp](http://support.microsoft.com/support/kb/articles/Q140/5/70.asp)

## Distributing data with your application

In many cases you will be supplying data with your application. The most significant issue regarding this aspect of deploying your application is how to accommodate where users install the data. Users may choose to install the application in a location other than the default, so you may want to provide a means for your installation to locate the data.

Most of the MapObjects sample projects, installed by default at C:\Program Files\ESRI\MapObjects2\Samples, include a module that has a function to return the data path of a folder containing the required data. You may want to incorporate similar functionality into your application. The following sample code assumes the data is in a folder named Data that is at the same level as the folder that contains the application.

```
' Function name: ReturnDataPath
' Return type:   String
' Dependencies:  Uses InStrL function
' Description:  Returns the full path to the specified data directory.
' Args: dataDir - Directory to search for
```

```
Public Function ReturnDataPath(dataDir As String) As String
    Dim sPath As String
    Dim iLastPos As Integer
    sPath = App.Path
    iLastPos = InStrL(sPath, "\")
    ReturnDataPath = Left(sPath, iLastPos - 1) + "..\Data\" + dataDir
End Function
```



```
' Function name: InStrL
' Return type: Integer
' Dependencies: None
' Description: Returns the character position of the last occurrence
'               of srchString in inString. If there is no match, 0 is
'               returned. This provides the same functionality that VB's
'               InStr function provides, except that it searches backwards.
' Args: inString - String to examine
'       srchString - String to search for
```

```
Public Function InStrL(inString As String, srchString As String) As Integer
    Dim iLastPos As Integer 'Set to 0 on initialization
    ' Check srchString - a 0-length string will match every time
    If Len(srchString) Then
        ' Set iLastPos to the last matching position
        Dim iCurPos As Integer
        Do
            iLastPos = iCurPos
            iCurPos = InStr(iCurPos + 1, inString, srchString, vbTextCompare)
        Loop Until iCurPos = 0
    End If
    InStrL = iLastPos
End Function
```





## Sharing MapObjects components

When you distribute a MapObjects application it is important to understand that many applications, perhaps yours and another company's, may be using the same MapObjects components. To prevent conflicts with an uninstallation program removing components that are needed by another application, your setup and uninstallation program should install and uninstall MapObjects components as *shared components*. A shared component is copied just as any normal file would be except it is handled differently by the registry during installation and uninstallation.

Also, per usual DLL standards, MapObjects components should be copied with *version checking* to ensure that a newer version of the file is not overwritten by your installation.

### Installing shared components

In order for a file to be installed as a shared component, you must indicate to the setup program generator that the file is to be shared. Most setup program generators, including the Visual Basic Package and Deployment Wizard, can determine if a file is to be installed as shared. As a note, Microsoft system files should never be installed as shared components.

Each time a shared component file is installed, the file is first copied, then a registry key value under `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SharedDLLs` is created or incremented. The name of the key is the fully qualified installed location of the file, and the value is set to the number of times the component is installed; for example, if it is the second time the component is installed, the value is 2.

### Uninstalling shared components

Each time a shared component file is uninstalled, the same value that was created or incremented when it was installed is decremented.

When a DLL-counted file is uninstalled and the decremented value reaches zero, the uninstallation program prompts the user to remove the component file. Some programs may ask if you want to remove all shared component files that have reached a DLL-count of zero. If the file was registered when it was installed, the component should be unregistered by calling the file's `DllUnregisterServer` function before it is removed.

When a DLL-counted file is uninstalled and the value is decremented, but the DLL-count is *not zero*, the uninstall program does not prompt for anything, and the component is left there. (Installing an application more than once on a target machine increments the count several times, thereby causing a file to have a DLL-count greater than zero when it is uninstalled.) It is for this reason that some uninstallation programs leave files after the software has been uninstalled.

By ensuring that shared files with a DLL-count greater than zero are not deleted or unregistered, other applications that use MapObjects components will continue to work after an uninstallation of your application.

