

Knowledge Base

INFO: Useful Tools for Package and Deployment Issues

PSS ID Number: 198038

Article Last Modified on 11/19/2003

The information in this article applies to:

- Microsoft Visual Basic Learning Edition for Windows 6.0
- Microsoft Visual Basic Professional Edition for Windows 6.0
- Microsoft Visual Basic Enterprise Edition for Windows 6.0

This article was previously published under Q198038

SUMMARY

Microsoft Visual Basic ships with the Package and Deployment Wizard (PDW), a tool designed to aid developers in packaging and deploying their Microsoft Visual Basic applications. Packages created with the PDW use compressed cabinet files (.cab files) for distribution. This article discusses several utilities that may be used to modify, view or manipulate cabinet files. As well, during the distribution process, problems may arise when deploying applications to various environments or when testing. This article also discusses some tools that may be useful for troubleshooting application deployment issues.

MORE INFORMATION

This article discusses the following utilities:

Utility	Description
Extract.exe	Extract files from a CAB file.
Makecab.exe	Builds CAB Files.
Cabview	View/Manipulate the contents of a CAB file.
Cabarc.exe	Build and extract files from a CAB file.
Regsvr32.exe	Registers ActiveX Components.
Regocx32.exe	Registers ActiveX Controls (.OCX Files).
Regit.exe	Registers in process ActiveX servers (.OCX/.DLL Files).
Depends.exe	View file dependencies.
Depends.exe	Profiling Options.
Filemon.exe	Monitors file access activity.
Regmon.exe	Monitors registry access activity.
Clireg32.exe	Registers Distributed Component Object Model Components on Client computers.
Vbrun60.exe	Installs core Visual Basic files.
Setup Toolkit	Used to customize Visual Basic application Installations.

Extract.exe

Extract.exe is a utility that allows you to extract all files or specific files contained within a cabinet (.cab) file.

The Extract.exe utility is found in your Windows, Windows\System32 or Windows\Command folder on Windows NT, Windows 2000, Windows Me, Windows 98, or Windows 95. It may also be found on your Windows installation CD-ROM.

To use the Extract.exe utility perform the following steps:

1. Open an MS-DOS Command prompt.
2. Navigate to a folder that contains a cabinet file from which to extract the files.
3. Execute the following command from the MS-DOS Command prompt:

```
Extract.exe Project1.cab /E
```

NOTE: Project1.cab is the name of the cabinet file.

Executing the preceding command extracts all of the files contained within the Project1.cab into the current directory.

Extract.exe may be used to extract individual files or may be used to extract files from multiple cabs at one time. For more information about the options available with the Extract.exe file, execute the following command from a MS-DOS command prompt:

```
EXTRACT.EXE /?
```

For additional information, click the article number below to view the article in the Microsoft Knowledge Base:

[191212](#) HOWTO: Modify and Rebuild .CAB Files Built with PDW

Makecab.exe

Makecab.exe is a utility that allows you to take existing files and package them into a cabinet (.cab) file. This may be useful if you have extracted files from a .cab using the Extract.exe utility and then would like to repackage the files into the cabinet.

The Makecab.exe utility is installed by Microsoft Visual Basic and may be found in the following folder by default:

```
C:\Program Files\Microsoft Visual Studio\VB98\Wizards\PDWizrd
```

To use the Makecab.exe utility, perform the following steps:

1. Open an MS-DOS Command prompt.
2. Navigate to the folder that contains the files to package into a cabinet.
3. Run the following command from the Command prompt:

```
MAKECAB.EXE FILENAME CABFILE
```

NOTE: The filename is the file to compress and cabfile is the cabinet (.cab) file.

Makecab.exe packages the file listed into a .cab for distribution. If the .cab file does not exist it will be created. When used with the preceding syntax, files must be added to a .cab file one at a time. With the use of a .ddf file, archive options and multiple files can be

added to a .cab file at the same time.

For additional information on the Makecab.exe command line parameters, execute the following command from an MS-DOS Command line:

```
MAKECAB.EXE /?
```

For additional information please see the following article in the Microsoft Knowledge Base:

[191212](#) HOWTO: Modify and Rebuild .CAB Files Built with PDW

Cabview

Cabview is a Windows 95/Windows 98 PowerToy utility that allows you to graphically view and manipulate the contents of a .cab file. Windows 98 has Cabview installed natively as part of the operating system.

For Windows 95, CABVIEW is available at the following location:

<http://www.microsoft.com/windows95/downloads/contents/wutoys/w95pwrtosyset/>

Refer to the download page for the Cabview utility for instructions on how to install the utility.

Cabview provides a graphical view of the contents of a .cab file similar to Windows Explorer. This allows easy file manipulation through drag-and-drop of files to and from the .cab file.

To use Cabview, perform the following steps:

1. Navigate through Windows Explorer to the .cab file you wish to view.
2. Right-click the file, and select the View option. CABVIEW presents the files contained with the .cab file. You can drag files into the CABVIEW window to add files, and drag files from the CABVIEW window to extract them onto the system.

Cabarc.exe

Cabarc.exe is another CAB file manipulation utility. With Cabarc.exe you can create new .cab files, extract files and list the contents of a .cab file. Cabarc.exe does not allow adding additional files to an already existing .cab file, however.

Cabarc.exe is available with the CAB Software Development Kit (SDK).

Cabarc.exe's best feature is the ability to list information about the files contained within a CAB file in an orderly way (better than Extract.exe).

To view the contents of a .cab file with Cabarc.exe, perform the following steps:

1. Open an MS-DOS Command Prompt.
2. Navigate to the folder containing the .cab file that has the contents you want to view.
3. Run the following command from the Command prompt:

```
CABARC.EXE L CABINET.CAB
```

NOTE: Cabinet.cab is the CAB file that has the content you want to browse.

For additional information about the Cabarc.exe file, refer to the documentation that ships with the CAB SDK.

Regsvr32.exe

Regsvr32.exe allows you to register/un-register DLL and OCX files that are self-registerable. Regsvr32.exe provides rudimentary error codes as well as a silent install option.

Regsvr32.exe is installed with Microsoft Visual Basic and can be found in the System (or System32) directory.

When used, Regsvr32.exe attempts to load the component and call its DLLSelfRegister function. If successful Regsvr32.exe displays a dialog indicating success. If the registration fails, Regsvr32.exe returns a basic error code.

To use the Regsvr32.exe utility to register a component perform the following steps:

1. Open an MS-DOS Command prompt or from the Windows Start menu, select Run.
2. Execute the following command line:

```
REGSVR32.EXE C:\WINDOWS\SYSTEM\COMPONENT.DLL
```

Component.dll is the DLL/OCX to register onto the system. The path to the component must be included in the command line.

Components can be unregistered using the /u command line parameter.

For additional information on Regsvr32.exe command line parameters, execute the following command from an MS-DOS command line:

```
REGSVR32.EXE /?
```

Regocx32.exe

Regocx32.exe is used to register ActiveX controls. This utility is designed to work with setup programs and does not offer any user interface. This is helpful when using an installation program/script that needs to shell out to register ActiveX controls.

Regocx32.exe ships with Microsoft Visual Basic and is located on the installation CD at the following location:

```
\Common\Tools\VB\Regutils
```

Copy the file into the System (or System32) folder.

To use the Regocx32.exe utility perform the following steps:

1. Open an MS-DOS Command prompt or from the Windows Start menu, choose Run.
2. Execute the following command line:

```
REGOCX32.EXE C:\WINDOWS\SYSTEM\CONTROL.OCX
```

Control.ocx is the ActiveX control you want to register. The path to the component must be included in the command line.

Note that Regocx32.exe does not raise any dialogs or return any error codes to determine if the ActiveX control successfully registers.

Regit.exe

Regit.exe allows you to register DLL and OCX files that are self-registerable. Regit.exe accepts multiple filenames and wildcards, allowing numerous components to be registered at the same time. Regit.exe also provides basic error codes if a component will not register.

Regit.exe ships with Microsoft Visual Basic and is located on the installation CD at the following location:

\\Common\Tools\VB\Regutils

Copy the file into the System (or System32) folder.

To use the Regit.exe utility perform the following steps:

1. Open an MS-DOS Command prompt or from the Windows Start menu, choose Run.
2. Execute the following command line:

```
REGIT.EXE C:\WINDOWS\SYSTEM\COMPONENT.DLL
```

-or-

```
REGIT.EXE C:\WINDOWS\SYSTEM\*.OCX *.DLL
```

Component.dll is the DLL to register onto the system. The path to the component(s) must be included in the command line.

Depends.exe

The Dependency Walker, Depends.exe, is used to determine the explicit dependencies a file has. A dependency is a secondary file, which is necessary for a file to load/run properly. For example, a Microsoft Visual Basic application is dependent upon the Microsoft Visual Basic run-time files.

Depends.exe also presents plenty of valuable information about dependent components, such as their export table, internal version number, and so forth.

Depends.exe ships with Microsoft Visual Basic and is located on the installation CD's at the following location:

\\Common\Tools\VB\Unsupprt\Depend

Copy all the files from the Depend folder into the System (or System32) folder.

To use depends.exe, perform the following steps:

1. From the Windows Start menu, choose Run.
2. Type in the path and name to the Depends.exe file. You can also run the Depends.exe file by double-clicking the file through the Windows Explorer. Once Depends.exe executes at least once on your system, you may right-click any .exe, .dll, .ocx or other file and select View Dependencies from the shortcut menu to launch the Depends.exe file.
3. When Depends.exe launches, select File and then choose Open.
4. Browse in the Open dialog box for an .exe, .dll, or .ocx file and then click Open.
5. The explicit dependencies for that particular component will appear.

For more information on the Depends.exe file, please refer to the Utilities On-line help.

Depends.exe Profiling Options

To download a later version of the Dependency Walker utility, visit the following Dependency Walker Web site:

<http://www.dependencywalker.com>

Microsoft provides third-party contact information to help you find technical support. This contact information may change without notice. Microsoft does not guarantee the accuracy of this third-party contact information. The later versions provide a profiling option that allows you to monitor the loading and the running of an application.

To profile an application, follow these steps:

1. Click **Start**, and then click Run.
2. In the **Run** box, type the path to the Depends.exe file and type the name of the Depends.exe file.

Alternately, you can use Windows Explorer to run the Depends.exe file.

3. When Depends.exe launches, click **File**, and then click **Open**.
4. Find the .exe file in the **Open** dialog box, and then click **Open**.
5. On the **Profile** menu, click **Start Profiling**.
6. Configure any of the options on the **Profile** module dialog box that are displayed, and then click **OK**.

Files are loaded and calls are made in the log window.

Note For more information about the Depends.exe file, see the Utilities On-line help.

FileMon.exe

You can use the File Monitor utility (FileMon.exe) to monitor file access on a system in real-time. FileMon.exe can help you determine what application is modifying a file or is narrowing down errors such as "Access Denied." For more information about FileMon.exe, visit the following Sysinternals Freeware Web site:

<http://www.sysinternals.com>

Microsoft provides third-party contact information to help you find technical support. This contact information may change without notice. Microsoft does not guarantee the accuracy of this third-party contact information. To use the File Monitor utility, follow these steps:

1. Click **Start**, and then click Run.
2. In the **Run** box, type the path to the FileMon.exe file and type the name of the FileMon.exe file.

Alternately, you can use Windows Explorer to run the FileMon.exe file.

When FileMon.exe starts, it immediately begins to monitor any file access that occurs on the computer. You can use the Filter Option by pressing the Ctrl+L key combination to limit the monitoring to a particular file or to a particular directory.

For more information about the FileMon.exe file, see the Utilities On-line help.

RegMon.exe

You can use the Registry Monitor utility (RegMon.exe) to monitor registry access on a system in real-time. The RegMon.exe tool can determine what applications are reading and what applications are modifying the registry. To download this product, visit the following Sysinternals Web site:

<http://www.sysinternals.com>

Microsoft provides third-party contact information to help you find technical support. This contact information may change without notice. Microsoft does not guarantee the accuracy of this third-party contact information. To use the Registry Monitor utility, follow these steps:

1. Click **Start**, and then click **Run**.
2. In the **Run** box, type the path to the RegMon.exe file and type the name of the RegMon.exe file.

Alternately, you can use Windows Explorer to run the RegMon.exe file.

When RegMon.exe starts, it automatically monitors any registry access that occurs on the computer. You can use the Filter option by clicking the Ctrl+L key combination to limit the monitoring to a particular application.

For more information about the RegMon.exe file, see the Utilities On-line help.

Clireg32.exe

Clireg32.exe is a utility used to register remote server components onto client computers. This allows a client program to instantiate an object that resides on a remote computer through the Distributed Component Object Model (DCOM) or Remote Automation.

Clireg32.exe is installed by Microsoft Visual Studio, and can be found in the following folder by default:

\Program Files\Microsoft Visual Studio\Common\Tools\

Clireg32.exe uses an ActiveX EXE/DLL's VBR file (Remote Automation Registration File) to register a remote component onto a clients computer.

To use Clireg32.exe to register a remote server, perform the following steps:

1. Open an MS-DOS Command prompt or from the Windows Start menu, choose Run.
2. Execute the following command:

```
CLIREG32.EXE <PATH>\PROJECT.VBR
```

Project.vbr is the VBR file to register, and <PATH> is the path to that file. For the preceding command line to work, the path to the CLIREG32.EXE file must be part of the environment Path variable.

3. When Clireg32.exe launches, it presents a title dialog box. Click the OK button, and the Options dialog box appears. You can then set the properties as desired to let the client computer know about the remote server.

For additional information on the Clireg32.exe file command line parameters, execute the following command from an MS-DOS command line:

```
CLIREG32.EXE /?
```

Vbrun60.exe

The Vbrun60.exe file contains the necessary run-time files needed to run an application written in Microsoft Visual Basic 6.0.

The core run-time files for Microsoft Visual Basic include the following:

```
Msvbvm60.dll  
Oleaut32.dll  
Olepro32.dll  
Stdole2.tlb  
Asycfilt.dll  
Comcat.dll
```

The Vbrun60.exe file ships with Microsoft Visual Basic 6.0, and is located (in CAB format) on the installations CD-ROM at the following location:

\Common\Tools\VB\Cabinets

From time to time Microsoft releases a Service Pack for Microsoft Visual Basic, which contains updates to the run-time files. You can locate the most current version of the run-time in the following Microsoft Knowledge Base article:

[192461](#) FILE: VBRUN60.EXE Installs Visual Basic 6.0 Run-Time Files

To install the Microsoft Visual Basic 6.0 run-time, perform the following steps:

1. Open an MS-DOS Command prompt or from the Windows Start menu, choose Run.
2. Run the Vbrun60.exe file. You can also run the Vbrun60.exe file by double-clicking the file through the Windows Explorer.

Vbrun60.exe contains a silent install switch (/q). This installs the run-time files without any dialog boxes appearing.

NOTE: Installing the run-time files may require you to reboot the computer.

For more information about the Vbrun60.exe file, see to the preceding Knowledge Base article.

Setup Toolkit

Microsoft Visual Basic ships with the source code to the setup program used to install application distributions. This source code is usually located in the \Wizards\PDWizard\Setup1 directory where Microsoft Visual Basic is installed. You may modify this code to customize the setup package created by the PDW.

NOTE: Microsoft Technical SupF194621port does not support the modification of the setup process or any of the setup files. Support is

provided for the Package and Deployment Wizard on an "as is" basis only.

REFERENCES

For additional information, please see the following article in the Microsoft Knowledge Base:

[194022](#) INFO: Visual Studio 6.0 Service Packs, What, Where, Why

Additional query words: Extract Makecab Cabview Cabarc Regsvr32 Regocx32 Regit Depends Clireg32 Vbrun60 Setup Toolkit

Keywords: kbAppSetup kbcode kbDeployment kbDownload kbinfo kbwizard KB198038

Technology: kbAudDeveloper kbVB600 kbVB600Search kbVBSearch kbZNotKeyword2 kbZNotKeyword6

[Send feedback to Microsoft](#)

[© 2004 Microsoft Corporation. All rights reserved.](#)