Microsoft KB Archive/286705

< Microsoft KB Archive
Jump to:navigation, search
Knowledge Base

Contents

HOW TO: Script Compatibility Layers in Windows XP

Article ID: 286705

Article Last Modified on 1/25/2006

APPLIES TO

- Microsoft Windows XP Home Edition
- Microsoft Windows XP Professional

This article was previously published under Q286705

IN THIS TASK

SUMMARY

Windows XP includes technology to dynamically apply compatibility fixes for programs that are not designed to run on Windows XP. The compatibility fixes are provided to assist an earlier version of a program in avoiding any issues that may arise because of the changes in this new operating system. These compatibility fixes may be used individually or in ready-to-use groups called compatibility layers. These layers have been defined for several commonly encountered configurations.

An example is a layer that provides the compatibility fixes necessary for a program that was written specifically for Microsoft Windows 95. This article describes how you can script compatibility layers in Windows XP.

back to the top

Applying Compatibility Layers

Microsoft has created a series of tools to make the use of the compatibility technologies accessible both through graphical user interface (GUI) tools and from the command prompt. Normally, the GUI tools are the preferred method of applying compatibility fixes or layers, but occasionally you may encounter cases where the layers must be applied through a script or batch file. This article explains how to apply a compatibility fix through a script or batch file.

The following command syntax should be used when you apply compatibility layers in batch files or scripts:

```
set __COMPAT_LAYER=[!]layer_name1 [layer_name2 ...]
```

NOTE: There is a space followed by two underscores (_) between "set" and "COMPAT" in this command syntax. The command does not work without both underscores.

Layer_name: The short name of the compatibility layer being applied.

[!]Layers should not be added to existing compatibility fixes in the database files.

NOTE: Compatibility layer names must be separated by a space.

For example:

```
set __COMPAT_LAYER=Win95 DisableThemes
```

```
set __COMPAT_LAYER=!Win95 DisableThemes
```

To disable the compatibility layer after it has been applied in this fashion, use the same command without specifying a compatibility layer.

For example:

```
set __COMPAT_LAYER=
```

Each of the most common compatibility layers has a short version of its name that can be used with the set __COMPAT_LAYER command in a script.

The list of available compatibility layers includes the following short name and its full name:

DisableThemes - Disable Visual Themes
ProfilesSetup - Profile Setup Support
256Color - 256 Color
640x480 - 640 x 480 Screen Resolution
Win95 - Windows 95
Win98 - Windows 98 / Windows Me
Win2000 - Windows 2000
NT4SP5 - Windows NT 4.0 SP 5

When you apply compatibility layers in this way, you also ensure that the compatibility layer is passed on recursively to any other process that might be spawned by the commands in the batch file. After you reach a point in the script or batch file where the compatibility layer is no longer needed, cancel the layer with the set __COMPAT_LAYER command without specifying a compatibility layer. Any processes that were started while the compatibility layer was in effect will still run under the compatibility layer until they are terminated.

back to the top

Troubleshooting

As an example of how you might implement this command, consider the following scenario: You have a batch file that performs several vital configuration tasks before starting a program. The program requires a series of compatibility fixes that are included in the Windows 95 compatibility layer. If the compatibility layer needs to be applied only to the Setup.exe program, and not to any processes that Setup.exe may spawn, the GUI tools are preferable. You then right-click the .bat or .cmd file and apply the compatibility layer to the batch file.

The following sample illustrates a portion of a batch file that might be used in this type of scenario:

```
Md C:\Program Files\MyCompany\MyApp
Copy D:\i386\*.* C:\Program Files\MyCompany\MyApp
C:\Program Files\MyCompany\MyApp\Setup.exe
```

If the Setup.exe program requires the compatibility fixes provided by the Windows 95 compatibility layer, and they must be passed on recursively to any process spawned by Setup.exe, the GUI tools are not appropriate. You can resolve this issue by including the following command in the batch file before you call the program that requires the compatibility layer:

```
Md C:\Program Files\MyCompany\MyApp
Copy D:\i386\*.* C:\Program Files\MyCompany\MyApp
set __COMPAT_LAYER=Win95
C:\Program Files\MyCompany\MyApp\Setup.exe
```

This effectively applies the Windows 95 compatibility layer to the Setup.exe program and recursively to any process spawned by Setup.exe.

This technique can be used in batch files, as shown in the previous example, and it can also be applied successfully in logon scripts. This provides network administrators with a solution to issues raised by any program that requires compatibility layers and that must be called through a batch file where the program may spawn other processes.

back to the top

REFERENCES

back to the top

Additional query words: "app compat" act "program compatibility"

Keywords: kbhowto kbhowtomaster KB286705

Send feedback to Microsoft