

# Scripting Windows Installer Packages

*Use scripts to determine the contents of MSI setup files.*

by Bob Kelly  
November 2006



One of the many benefits touted in the long list of reasons Windows Installer is so great is that the setup (MSI) files are actually structured databases. In addition to providing a universal schema for setup authors to follow, the fact that you can reliably query a file for certain information adds another sometimes overlooked value. The best example I've seen that takes advantage of this among commercial tools is the advent of conflict checking features available in products like AdminStudio (Macrovision), Package Studio (Altiris) and MSI Studio (ScriptLogic). These tools query MSI packages to find potential conflicts between baseline snapshots and other packages in order to help you identify (and even resolve) potential conflicts between them. Without the full implementation of comparing contents with other packages (which you certainly could write in a script if you had some serious time on your hands) you can query the MSI file database to collect file information yourself as shown here:

```
MSIFile = "C:\Windows\System32\webfldrs.msi"
Set MSIobj = CreateObject("WindowsInstaller.Installer")
Set MSIdb = MSIobj.OpenDatabase(MSIFile, 0)

WQL = "SELECT FileName,Version,FileSize FROM File"
Set MSIView = MSIdb.OpenView(WQL)
MSIView.Execute()

On Error Resume Next
Set MSIrecords = MSIView.Fetch
Do While Err.number = 0
    WScript.Echo "Name: " & MSIrecords.StringData(1) & ",
Version: " & _
    MSIrecords.StringData(2) & ", Size: " &
MSIrecords.StringData(0)
    Set MSIrecords = MSIView.Fetch
Loop

MSIView.Close
```

This can be helpful for generating documentation of MSI packages. You can query any of the tables such as the registry table or property table in the same way. In the script below, we continue from that above (to avoid

repetition in establishing object references) to get all of the properties and their values:

```
WQL = "SELECT Property,Value FROM Property"
Set MSIView = MSIdb.OpenView(WQL)
MSIView.Execute()

Set MSIrecords = MSIView.Fetch
Do While Err.number = 0
    WScript.Echo MSIrecords.StringData(1) & " = " &
MSIrecords.StringData(2)
    Set MSIrecords = MSIView.Fetch
Loop
```

You can also obtain a specific property to just get what you want by modifying the query:

```
WQL = "SELECT Property,Value FROM Property WHERE
Property='ProductVersion'"
```

Using these same techniques, I have created two tools you may obtain freely along with the source code to see how it was done. The first is [MSI Viewer](#) which provides a graphical interface to show the results of queries just like those above. The second is [MSI Search](#), which enumerates the MSI files found below a specified root location where you may then search them for any string and have their results displayed to you. This can be helpful if you are having a problem with a certain files and want to know which application installations contain that file.

I hope seeing this capability and seeing some examples of how it may be used helps you to think of ways you can write a script to help you in our own environment. [M](#)

*Bob Kelly is president and co-founder of [AdminScriptEditor.com](#), home to an integrated suite of scripting tools and a shared library of scripts and language help. He has authored books on scripting and desktop administration and several white papers. Bob also owns and operates [AppDeploy.com](#), where he writes and produces videos on topics related to software deployment. You can contact Bob about "Scripting Windows Installer Packages" at [bkelly@adminscripteditor.com](mailto:bkelly@adminscripteditor.com).*