

Let WMI do the heavy lifting of determining system information

devblogs.microsoft.com/oldnewthing/20041021-00

October 21, 2004



Raymond Chen

Windows Management Instrumentation is a scriptable interface to configuration information. This saves you the trouble of having to figure it out yourself.

For example, here's a little program that enumerates all the CPUs in your system and prints some basic information about them.

```
var locator = WScript.CreateObject("WbemScripting.SwbemLocator");
var services = locator.ConnectServer();
var cpus = new Enumerator(services.ExecQuery("SELECT * FROM Win32_Processor"));
while (!cpus.atEnd()) {
    var cpu = cpus.item();
    WScript.Stdout.WriteLine("cpu.ProcessorType=" + cpu.ProcessorType);
    WScript.Stdout.WriteLine("cpu.CurrentClockSpeed=" + cpu.CurrentClockSpeed);
    WScript.Stdout.WriteLine("cpu.MaxClockSpeed=" + cpu.MaxClockSpeed);
    WScript.Stdout.WriteLine("cpu.Manufacturer=" + cpu.Manufacturer);
    WScript.Stdout.WriteLine();
    cpus.moveNext();
}
```

Save this program as `cpus.js` and run it via `cscript cpus.js`.

There's a whole lot of other information kept inside WMI. You can get lost amidst all the classes that exist. The Scripting Guys have their own tool called WMI Scriptomatic which lets you cruise around the WMI namespace. (The Scripting Guys also wrote Tweakomatic which comes with hilarious documentation.)

Added 11am: It appears that people have misunderstood the point of this entry. The point here is not to show how to print the results to the screen. (I just did that to prove it actually worked.) The point is that you can let WMI do the hard work of actually digging up the information *instead of having to hunt it down yourself*. Want BIOS information? Try Win32_BIOS. Change the query to "SELECT * FROM Win32_BIOS" and you can get the manufacturer from the Manufacturer property. Plenty more examples in MSDN.

Raymond Chen

Follow

