Morales

"WMIDiag can help diagnose the underlying dependency that's causing a WMI issue."

Resolve WMI Problems Quickly with WMIDiag

The WMI Diagnosis utility gives you fast, powerful help in diagnosing WMI-related system problems

As a Windows administrator, you've probably encountered errors reported by Windows Management Instrumentation (WMI). For WMI problems that customers report to the Microsoft Global Escalation Services team, we've found that the root cause usually lies with an underlying dependency, such as DCOM settings, the registry, or the file system. A tool we often use in such situations is the WMI Diagnosis (WMIDiag) utility, which you, too, can use to help diagnose the underlying dependency that's causing the WMI issue and even suggest ways to fix the problem. You can download WMIDiag at go.microsoft.com/fwlink/?LinkId=62562.

WMI Architecture

To help you understand how to use WMIDiag, let's briefly review WMI's architecture. You can use WMI classes in scripts or applications to automate administrative tasks on remote computers, which is especially useful for managing a large number of systems. The WMI architecture contains three main components:

- WMI providers and managed objects: WMI providers are represented as COM objects and monitor objects such as logical or physical hard drives, OSs, processes, or services.
- WMI infrastructure: The infrastructure comprises the WMI service (winmgmt.exe) and the WMI repository, which is organized by namespaces, such as root\default or root\cimv2. The WMI service acts as the intermediary between WMI providers and the WMI repository. WMI obtains most data dynamically from the provider when a client requests it.
- WMI consumers (clients): A consumer can be a script or an enterprise application such as Microsoft System Center Operations Manager. Consumers can query WMI for system information, subscribe to events (e.g., when a policy changes on a system), or run management tasks remotely.

For more information about WMI, see msdn.microsoft.com/en-us/library/aa394582(VS.85).aspx and the articles in the Learning Path at www.windowsitpro.com, InstantDoc ID 100845.

What's WMIDiag?

WMIDiag is written in VBScript and can be run from the command line or by simply double-clicking the WMIDiag.exe file. When run without any command-line arguments, the tool verifies dozens of settings, registrations, service states, binary availability, suspicious shutdowns, and DCOM-related event log entries.

WMIDiag can diagnose problems such as these:

- Scripts fail to run or completely hang.
- Enterprise systems management applications such as Microsoft Systems Management Server (SMS), System Center Operations Manager, or HP OpenView fail to run routine tasks.
- Software and hardware inventory fails to collect some or all of the required information.
- Applications or service packs fail to install properly.
- Group Policy Objects (GPOs) fail to be deployed.
- Various DCOM error events logged in the Application event log indicate application failures.

A Real-Life WMI Problem

I recently worked with a customer who couldn't successfully run software inventory reports for several Windows XP SP2 clients. Software inventory reports are an important tool for ensuring your organization's software licensing and update compliance. Inventory-scanning applications such as SMS or System Center Configuration Manager 2007 connect to WMI and retrieve the instances of the Win32_Service class to determine what services are running on the system. So as our first step, we ran the built-in WMI tool WBEMTest on the customer's system (click Start, Run, and enter wbemtest); connected to the root\cimv2 namespace; and ran the following query:

```
Select * From Win32_Service
```

This query should have returned all the services running on that system. Instead, we received the error that Figure 1, page 12, shows.

As you can see, the error description, *Provider load failure,* is cryptic. As I mentioned earlier, providers are represented as COM objects. If COM objects aren't registered properly, they can't be loaded—thus resulting in the error in Figure 1. The question still remained, which provider wasn't registered properly? To help answer this question, our next step was to run WMIDiag.

Putting WMIDiag to Work

Running WMIDiag produces three files, by default in the %TEMP% directory:

- A .log file containing a verbose output of the WMIDiag tool activity.
- A .txt file containing a summarized report with warnings and errors worthy of investigation.

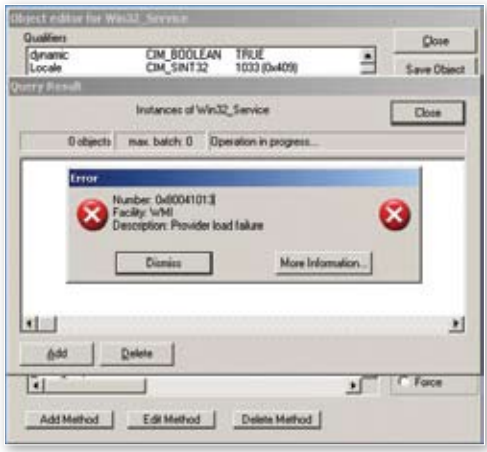

Figure 1: WMI query error

- A .csv file containing statistics that can be used to measure trends in WMI issues over time.

You'll want to look at the summarized .txt file. In the report file for our customer's WMI issue, we saw the error that Figure 2 shows, which identifies the failing provider. Notice in Figure 2 that WMIDiag also gives suggestions for resolving the issue. In this case, the resolution was to reregister the provider by issuing the command suggested by WMIDiag (c:\regsvr32.exe tscfgwmi.dll). Note that when a query is made, all providers for that class and derived classes will be loaded. In our case, the base class being queried was Win32_Service, and one of the derived classes is the Win32_TerminalService class, which means that the Terminal Services WMI provider will be loaded as well. Reregistering the provider identified by WMIDiag resolved the issue for our customer and enabled the customer to collect a full software inventory report.

Another issue I recently worked on involved applying GPOs. Domain controllers (DCs) in our customer's environment were completely failing to process GPOs. As a result, the following Userenv errors were logged in the Application event log every five minutes: *Windows cannot bind to xxxx .com domain. (Timeout). Group Policy processing aborted.*

This sort of error can be especially frustrating because it's so generic and offers no helpful hints about where to start investigating the problem. I've found that the best place to start troubleshooting is with the DCOM settings because DCOM affects connectivity and permissions to the entire system. WMIDiag will help diagnose incorrect DCOM settings and give you the exact commands that you can run to fix the configuration setting.

After I ran WMIDiag on the customer's system to help troubleshoot the GPO errors, I got the output that Figure 3 shows. As in the previous troubleshooting instance, WMIDiag not only revealed what was wrong but also provided options for resolving the issue. The GUI option, running dcomnfg .exe, could be suitable to fix a small number of systems, whereas the command-line option would be appropriate for scripting a solution to fix a large number of systems. Running dcomnfg.exe fixed the problem on the customer's two DCs after making the changes recommended by WMIDiag and rebooting the server.

Note that you can use WMIDiag to analyze the health of WMI on hundreds of servers simultaneously. To do so, run WMIDiag with the SMS parameter, which suppresses message boxes that are normally displayed when run interactively. If you want to designate a remote file share to store all the files created by WMIDiag, run WMIDiag with the LogFilePath parameter and specify the file path.

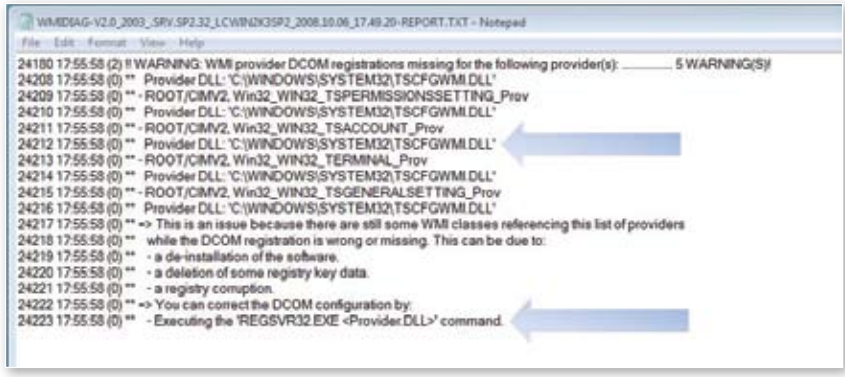

Figure 2: WMIDiag output showing specific provider that failed

Microsoft Support Offers WMIDiag Help

Would you like to have Microsoft Support diagnose your WMIDiag output? Although we can't guarantee that we'll respond to every question, you can send your WMIDiag output to wmidiag@microsoft.com, and our tech support team may be able to help diagnose your issue. Please briefly describe the issue in your email. And send me your suggestions or questions about the use of WMIDiag, or visit www.microsoft.com/technet/scriptcenter/topics/help/wmidiag.mspx.

InstantDoc ID 100845

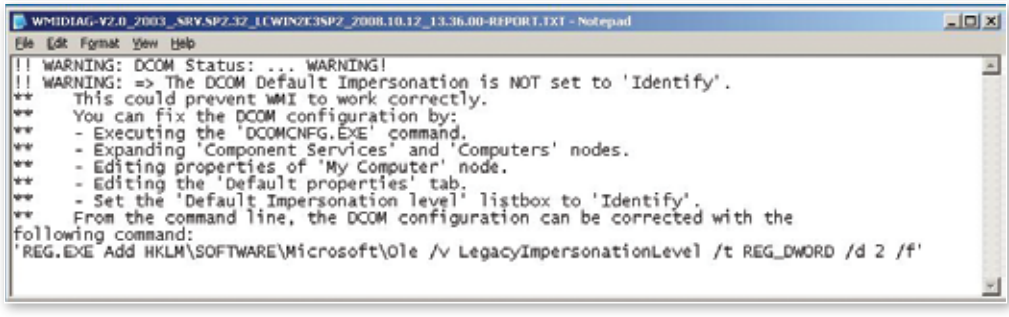

Figure 3: WMIDiag showing DCOM error

MICHAEL MORALES
(morales@microsoft.com) is a senior escalation engineer for Microsoft's Global Escalation Services team. He specializes in advanced Windows debugging and performance-related issues. For information about Windows debugging, visit blogs.msdn.com/ntdebugging.

Copyright of Windows IT Pro is the property of Penton Publishing and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.