

Informant Systems, Inc.
11135-23A Avenue
Edmonton, AB T6J4W5 Canada
p: 780.908.6669 f: 780.434.8991
www.informant-systems.com

SNMP Informant™

Installation and Configuration Guide

Release 2010.1



"GET more out of Windows!"

**Windows SNMP support for
industry standard Network
Management Systems**

www.snmp-informant.com

Copyright

Copyright © 2004-2010 Informant Systems, Inc. All Rights Reserved.
Copyright © 1999-2004 Williams Technology Consulting Services

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the Informant Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from Informant Systems, Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of Informant Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, INFORMANT SYSTEMS DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Informant Systems may make changes to specifications and product descriptions at any time, without notice.

Trademarks or Service Marks

SNMP Informant is a registered trademark of Informant Systems, Inc. All other trademarks are the property of their respective companies.

Table of Contents

Introduction	1
About Informant Systems, Inc.	1
Statement of Limitations	2
NMS Compatibility	2
Warranty	2
SNMP Informant Overview	3
Product Description.....	4
Performance Providers	5
WMI Providers.....	6
Custom Providers	7
System Requirements	7
Installation and Configuration	8
Installing the Microsoft Windows SNMP service	8
Windows Vista/7	8
Windows XP/Windows Server 2000, Windows Server 2003	8
Windows Server 2008	8
Windows Server 2008 Core.....	9
Configuring the Microsoft Windows SNMP service	9
Windows Vista/7	9
Windows XP/Windows Server 2000, Windows Server 2003	9
Windows Server 2008	10
Windows Server 2008 Core.....	11
SNMP in General	11
Installing SNMP Informant.....	12
GUI Installation.....	12
Command Line Installation	21
Configuring SNMP Informant.....	23
Registry Settings and their Meanings	24
Using SNMP Informant	29
General Usage Notes	29
Using the PDH Providers.....	29
Using the WMI-Exchange Provider	34
Using the WMI-OS Provider	38
Using the MSCS Provider	40
Using the Custom Provider	40
An important note about SNMP Informant Helper Services.....	43
Common SNMP Informant OIDs	52
Troubleshooting SNMP Informant	54
Troubleshooting Table.....	54
Troubleshooting PDH Providers	55
Troubleshooting WMI Providers	56
Troubleshooting Custom Providers.....	58

Table of Figures

Figure 1 – SNMP Informant Functional Overview	4
Figure 2 – SNMP Informant Application Structure (Performance Providers)	5
Figure 3 – SNMP Informant Application Structure (WMI Providers)	6
Figure 4 – SNMP Informant Application Structure (WMI Providers with helper service)	7
Figure 5 – Anatomy of an SNMP Informant OID	30

Introduction

Thank you for downloading and using (or trying) SNMP Informant. We are sure you will like what you see, and recognize the value in our products. This document is intended to help you make the most of SNMP Informant. If you have any comments about this document (omissions, corrections, etc.), please contact product.support@informant-systems.com, and let us know.

We have *always* strived to provide excellent value for your money with SNMP Informant. If you are pleased with this product, please tell your colleagues and friends. If not, please tell *us*, so we can address your concerns as soon as possible.

- **Please note:** We're changing some terminology in this version. Where sometimes we referred to our products as "agents", they are in fact, data "providers", loaded as an "extension" to the base Microsoft SNMP agent (or a replacement SNMP stack, if compatible). In other terminology (outside of this document), SNMP extension agents are sometimes referred to as "sub-agents".

About Informant Systems, Inc.

Informant Systems has been developing and providing the network management community with cost-effective SNMP extension agents for Windows operating systems and server applications since 1999. Our flagship product, SNMP Informant™ is in use by small, medium and large organizations around the world, including Universities, financial institutions, Fortune 500 companies and large multi-national organizations.

Resellers or commercial product developers interested in bundling or reselling SNMP Informant are encouraged to contact product.info@informant-systems.com in order to find out more information.



Informant Systems, Inc.

11135 – 23A Avenue
Edmonton, AB T6J4W5 Canada
Phone: 780-908-6669
Fax: 780-434-8991
Web: <http://www.informant-systems.com>

Product Information: product.info@informant-systems.com
Product Support: product.support@informant-systems.com

Primary Contact: **Garth K. Williams – President and Managing Director**
garth.williams@informant-systems.com

Statement of Limitations

Although we have attempted to find and correct any bugs in the software, we will not be held responsible for any damage or losses (of ANY kind) caused by the use (or misuse) of this product. Names, icons, functionality, file format, etc. are subject to change in future versions of SNMP Informant without notice.

Also, while we are well aware that we cannot control who downloads and/or uses SNMP Informant, we would like to make it clear that:

UNDER NO CIRCUMSTANCES IS SNMP INFORMANT DESIGNED TO MANAGE, SUPERVISE CONTROL, MONITOR OR OTHERWISE INTERACT WITH INSTRUMENTS AND/OR EQUIPMENT THAT MIGHT POTENTIALLY AFFECT HUMAN LIFE.

For example:

SNMP Informant is not designed for, nor is it intended to be used to monitor or interact with computer systems that might be used to construct, operate or maintain any type of the following facilities (including but not limited to):

- hospitals
- nuclear power
- air traffic, rail and/or maritime control or navigation
- other commuter transport (bus, taxi, etc.)
- military (operations, control, etc.)

NMS Compatibility

The SNMP Informant MIBs are written to comply with RFC standards, and are compiled and tested on several different MIB compilers and applications in order to ensure maximum compatibility. Nonetheless, we make NO guarantees that they will compile on any SPECIFIC product. In the event that you have problems using SNMP Informant (i.e. compiling SNMP Informant MIBs) with your particular NMS, please consult the Product Support Forums.

Warranty

All versions of SNMP Informant are warranted to operate EXACTLY as described on the SNMP Informant web site (www.snmp-informant.com). If you have ANY questions about SNMP Informant's ability to gather certain performance metrics, please contact product.info@informant-systems.com, and we will be pleased to help you out.

- **Please note:** While we endeavour to ensure our software runs “worry-free”, we make no guarantees that it will be bug-free.

SNMP Informant Overview

SNMP Informant products are advanced [Simple Network Management Protocol](#) (SNMP) extension agents that provide the capability to access Microsoft Windows Operating System and Application Server Performance Counters, WMI classes and other (i.e. registry and custom script) information through the SNMP protocol

SNMP Informant agent information can be accessed natively using SNMPv1, SNMPv2 or SNMPv3 (see below) protocols from an SNMP Network Management System (NMS). *

* NMS applications include (but are not limited to):

- Sciencelogic EM7
- HP Network Node Manager
- Paessler IPCheck
- Groundwork
- Netmon,
- IPMonitor
- OpenNMS
- Zenoss

... and others

... you will be amazed at how much Windows information SNMP Informant can expose to your NMS!

- IANA Private Enterprise Number 9600 is registered to WTCS (Informant Systems, Inc.). All our OIDS start with .1.3.6.1.4.1.9600.

SNMPv3

Since SNMP Informant is an SNMP Extension Agent (sometimes called sub-agents), it does not in and of itself support SNMPv3. It is the job of the SNMP service "stack" to support SNMPv3. The native Windows 2000, XP and 2003 SNMP service only supports SNMPv1 and SNMPv2. However, there are some Windows SNMP service replacements in the market today that claim to be 100% compatible with extension agents like SNMP Informant. Here are a couple of alternatives:

One such product is **NuDesign Technologies** "Agent Service for MS Windows". You can find out more about this product below:

- <http://www.ndt-inc.com/SNMP/AgentService.html>
 - NuDesign Technologies is an Informant Systems Partner

Another option is **Logisoft AR**, who have developed a product called SNMP Agent Defender, an SNMP v3 replacement stack. Find out more below:

- <http://www.logisoftar.com/ProductsSnmpInquisitor.htm>
- **Please note:** It is up to you (the customer), to do your due diligence when deciding on an SNMP stack replacement.

SNMP Traps

At present, SNMP Informant does not *generate* SNMP traps. We are looking at this functionality as a future enhancement.

Product Description

SNMP Informant agents are DLL ([Dynamic Link Libraries](#)) extensions to the Microsoft Windows SNMP service. *The Windows SNMP Service must be installed properly, configured and running before the SNMP Informant agent is accessible.*

SNMP Informant providers are designed to be installed on Windows Server software, although they **can** be installed on Windows XP, Vista and Windows 7. They are categorized as follows:

- **Performance Providers** – These providers connect to the [Performance Counter](#) sub-system, and allow performance counter information to be collected via SNMP.
- **WMI Providers** – also called State and Configuration providers, they connect to the Windows [WMI](#) sub-system, and allow WMI information to be collected via SNMP.
- **Custom Providers** – providers of this type allow you to collect either Performance or State/Configuration information from server applications that might not otherwise provide this information to either the Performance or WMI sub-systems.

Regardless of the provider type, taking advantage of the information SNMP Informant provides is very simple:

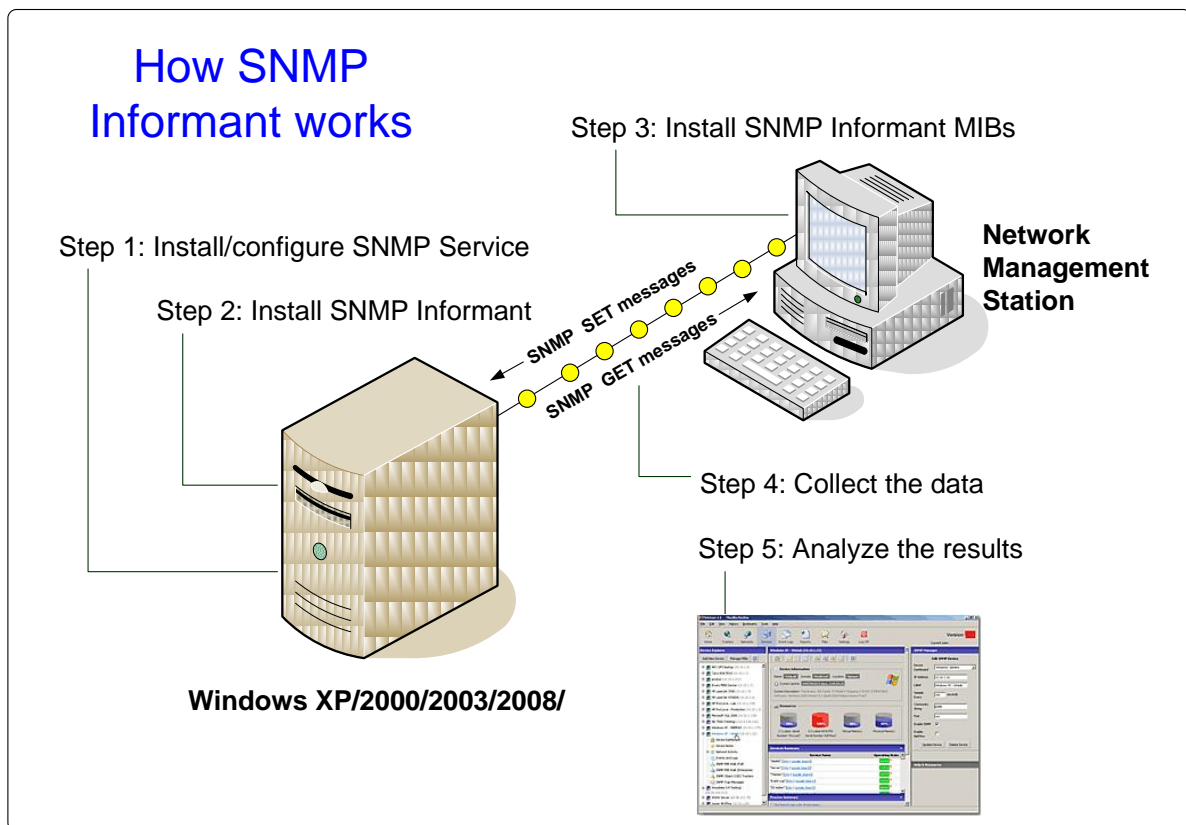


Figure 1 – SNMP Informant Functional Overview

Installing SNMP Informant is very easy. Once the SNMP service is correctly installed and configured on the designated system, the SNMP informant installer can be run (see the [Installation and Configuration](#) section). Depending on the validation key you enter, different providers will be enabled for installation.

Performance Providers

SNMP Informant Performance Providers connect through the Windows Performance Data Handler library to access server performance counters. A diagram illustrating this concept is shown below. As you can see, an [SNMP GET request message](#) is passed through the SNMP Informant DLL to the Windows performance sub-system and back again to the requesting entity.

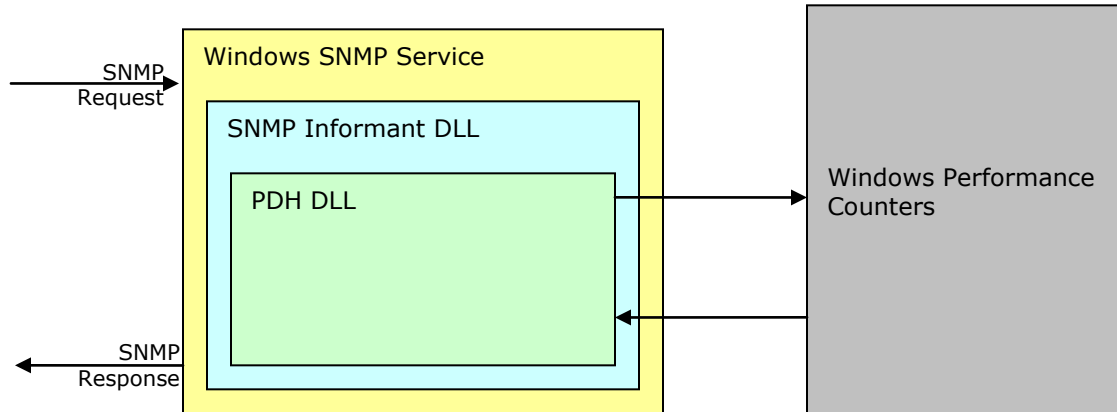


Figure 2 – SNMP Informant Application Structure (Performance Providers)

Standard and Advanced Performance Provider Notes

Windows 2000 does not come "out of the box" with logical disk performance counters enabled*. Unless activated, the only Windows 2000 disk counters accessible by SNMP Informant are the physical disk performance counters. In order to activate logical disk performance counters on Windows 2000, do this:

1. Open an OS prompt
2. Type "diskperf -y" (omit the "")
3. Close the OS Prompt
4. reboot the system

* Windows XP, Windows 2003, Windows Vista and Windows 7 dynamically activate logical disk counters as needed.

Application Server Performance Provider Notes

SNMP Informant Server Performance Provider Agents extend SNMP Informant functionality by allowing SNMP to be used to query application specific performance counters using SNMP. Such applications at present include:

- **ISA Server** – supports ISA 2000/2004/2006
- **BizTalk Server** – supports BizTalk 2000/2004/2006
- **SQL Server** - supports SQL 2000/2005/2008 (including MSDE and Express)
- **Exchange Server** – supports Exchange 2000/2003/2007
- **Forefront Server** (supports Forefront TMG)

WMI Providers

Similar to the PDH Providers, SNMP Informant WMI Providers make data requests to the local WMI sub-system on the system where SNMP Informant is installed. See the figure below for a diagram representing the data flow for this type of agent.

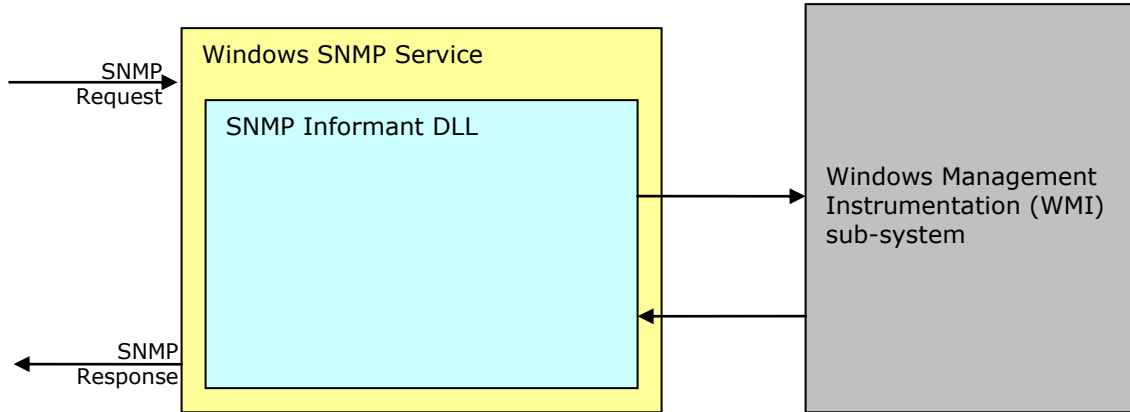


Figure 3 – SNMP Informant Application Structure (WMI Providers)

WMI Provider Notes

SNMP Informant WMI Providers extend SNMP Informant functionality by allowing SNMP to be used to server specific WMI information using SNMP. Such applications at present include:

- **Virtual Server** – allows SNMP to be used to access Virtual Server information
- **Hypervisor (Hyper-V)** – allows SNMP to be used to access Hyper-V information
- **Citrix** - SNMP to be used to access Citrix Presentation Server
- **OS** - allows SNMP to be used to access OS specific WMI information
- **HW** – allows SNMP to be used to access hardware specific WMI information
 - *Note: This agents' functionality will differ depending on motherboard hardware manufacturers' support for the Windows 2000, 2003 and 2008 WMI system. If the motherboard manufacturer does not provide information to the WMI system, then SNMP Informant will not be able to access it.*

WMI Provider Helper Services

The SNMP Informant WMI-OS and WMI-Exchange providers incorporate the use of a "helper service". This helper service sits between the extension agent DLL and the WMI sub-system.

SNMP requests are received (by the SNMP service) from the NMS for OIDs that are derived from a WMI class, and passed from the SNMP Informant extension agent DLL to the helper service. Then, the helper service passes (proxies) that request to the WMI sub-system, and waits for a response.

When a response is received, the helper service passes it back to the extension DLL, and the extension DLL passes it back to the SNMP service. Figure 4 below illustrates this data flow.

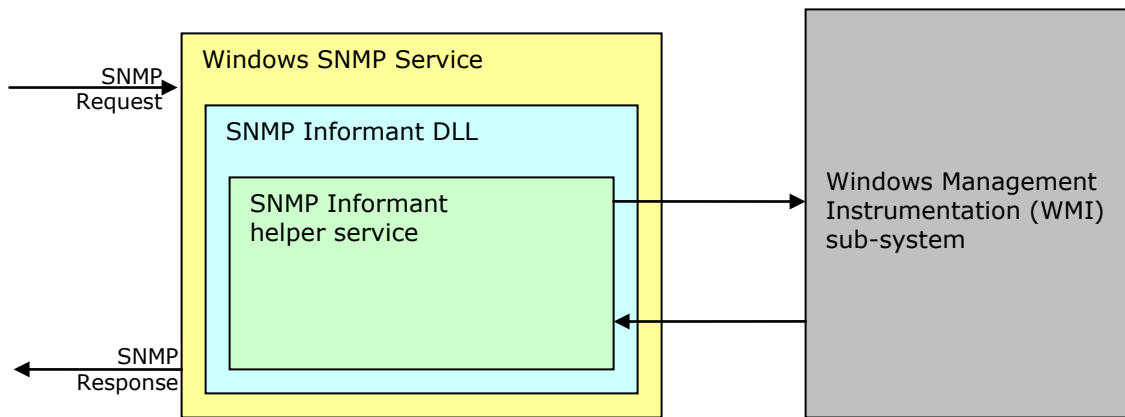


Figure 4 – SNMP Informant Application Structure (WMI Providers with helper service)

Custom Providers

SNMP Informant 2010.1 includes custom SNMP support for Microsoft Cluster Services and Citrix Presentation Server through APIs published for those products. This “bridge” allows information to be collected from those applications. This software must be installed on the server where SNMP Informant is installed.

In addition, SNMP Informant now (as of version 2010.1) supports the ability to define your own OIDs for custom performance counters, registry entries, and remotely spawned processes (i.e. scripts) that collect data from other means.

System Requirements

The SNMP Informant Agent executes on the following operating systems. It does not run on Microsoft Windows 95, 98, ME, or NT.

Operating Systems Supported:

- Microsoft Windows 2000 Professional
- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Datacenter Server
- Microsoft Windows XP Home/Pro(x86/x64)
- Microsoft Windows Vista Home/Business/Premium (x86/x64)
- Microsoft Windows 7 Home/Business/Ultimate (x86/x64)
- Microsoft Windows 2003, Standard Edition(x86/x64/ia64)
- Microsoft Windows 2003, Enterprise Edition (x86/x64/ia64)
- Microsoft Windows 2003, Datacenter Edition (x86/x64/ia64)
- Microsoft Windows 2003, Web Edition (x86/x64/ia64)
- Microsoft Windows 2003, Small Business Server (x86/x64/ia64)
- Microsoft Window 2008, Standard Edition (x86/x64/ia64)
- Microsoft Window 2008, Enterprise Edition (x86/x64/ia64)

Processor Requirement: minimum of a Pentium II class

Memory Requirement: 32 MB

Disk Space Requirement: 30MB

Installation and Configuration

The SNMP Service is not installed by default on the Microsoft Windows operating systems and is not configured by default on the Microsoft Windows 2003/2008 operating systems. **The SNMP Service must be installed and configured prior to installing the SNMP Informant agent.** If the SNMP Service is already installed and configured, then skip to the Installing SNMP Informant section.

Installing the Microsoft Windows SNMP service

Since the Microsoft Windows operating systems vary slightly, the steps to install the SNMP Service may be deviate a little from this guide.

Windows Vista/7

To install SNMP on Windows 7, open Control Panel and then click on Programs and Features. Then, click on "Turn Windows features on or off" link in the left pane. If UAC prompted, then click on Yes. Then, in the Windows Features window, scroll down and select "Simple Network Management Protocol (SNMP)" check box and click on OK. Then, wait for some time to install SNMP.

Windows XP/Windows Server 2000, Windows Server 2003

You may also refer to the Microsoft Windows Help (Start\Help) under "SNMP Service (installing)" for more information on installing the SNMP Service.

You must be logged on as an administrator or a member of the Administrators group to complete this procedure. If your computer is connected to a network, network policy settings may also prevent you from completing this procedure.

1. Click Start, point to Settings, click Control Panel, double-click Add or Remove Programs, and then click Add/Remove Windows Components.
2. In Components, click **Management and Monitoring Tools** (but do not select or clear its check box), and then click **Details**.
3. Select the Simple Network Management Protocol check box, and click OK.
4. Click **Next**.
5. Insert the respective CD or specify the complete path of the location at which the files are stored.
6. SNMP starts automatically after installation.

Windows Server 2008

You can add the SNMP service through the Add Features wizard. SNMP is a feature.

1. Start **Control Panel** then **Programs and Features**
2. Select **Turn Windows Features on or off**
3. Select **Features**
4. Select **Add Features**
5. Select **SNMP**
6. Choose **Install**

Windows Server 2008 Core

Use the following command within the core command prompt to install the SNMP service.

1. **Start /w ocsetup SNMP-SC**

Configuring the Microsoft Windows SNMP service

The Microsoft Windows SNMP Service must be configured before it can be accessed by any SNMP Manager software. Since the Microsoft Windows operating systems vary slightly, the steps to configure the SNMP Service may deviate a little from this guide. You may also refer to the Microsoft Windows Help (Start\Help) under "SNMP Service (security, configuring)" for more information on configuring the SNMP Service.

Windows Vista/7

After installing SNMP, click on Start Orb and then type Services.msc in the Search box and hit Enter. Then, scroll down in the right pane and right click on SNMP Services and select Properties. Then, click on Traps tab. Now, in the Community Name box, type the community name to which your computer will send trap messages and then click on "Add to list" button. Then, click on Apply and then OK.

Windows XP/Windows Server 2000, Windows Server 2003

1. Click **Start**, point to **Settings**, and then click **Control Panel**. Double-click **Administrative Tools** and then double-click **Computer Management**.
2. In the console tree, click **Services and Applications** and then click **Services**.
3. In the details pane, scroll down and click **SNMP Service**.
4. On the **Action** menu, click **Properties**.
5. On the **Security** tab, select **Send authentication trap** if you want a trap message to be sent whenever authentication fails.
6. Under Accepted community names, click **Add**.
7. Under **Community Rights**, select a permission level for this host to process SNMP requests from the selected community.
8. In **Community Name**, type a case-sensitive community name, and then click **Add**.
9. Specify whether or not to accept SNMP packets from a host:
10. To accept SNMP requests from any host on the network, regardless of identity, click **Accept SNMP packets from any host**.
11. To limit acceptance of SNMP packets, click **Accept SNMP packets from these hosts**, click **Add**, type the appropriate host name and IP or IPX address, and then click **Add** again.
12. Click **Apply** to apply the changes.

Windows Server 2008

After installing the SNMP service, you need to configure agent properties, which contains general information such as who is responsible for managing the agent host and the types of services with which the agent will interact on the computer.

Right-click the SNMP service in the Services console and choose Properties to open the properties for the SNMP Service, or select the service and choose Action and then Properties to display the service's property sheet. The General, Log On, Recovery, and Dependencies pages are the same as for other services.

Click the Agent tab to configure the following agent properties:

1. **Contact:** Specify the name of the person responsible for managing the host computer.
2. **Location:** Specify the physical location of the computer or the contact's location or other information (phone number, extension, and so on).
3. **Physical:** Select this option if the agent host manages physical hardware such as hard disk partitions.
4. **Applications:** Select this option if the agent uses any applications that transmit data using the TCP/IP protocol.
5. **Datalink and Subnetwork:** Select this option if the agent host manages a bridge.
6. **Internet:** Select this option if the agent host is an Internet gateway.
7. **End-to-End:** Select this option if the host uses IP. This option should always be selected.

Configuring traps

Use the Traps tab of the SNMP service to configure computers to which the SNMP service sends traps. From the Community Name drop-down list, select the community for which you want to assign a trap destination. If you have no communities set yet, type the community name in the combo box and click Add to List. Then, click Add to display a simple dialog box in which you can specify the host name, IP address, or IPX address of the remote computer to receive the trap notification. Repeat the process to add other trap destinations as needed.

Configuring security

Use the Security tab of the SNMP Service's properties to configure the communities in which the agent participates and optionally a list of hosts from which the agent accepts SNMP packets. By default, the agent accepts packets from all hosts. This presents a security risk, however, so take care to configure security settings to permit SNMP traffic only from authorized hosts. The Security page includes the following options:

1. **Send Authentication Trap:** Select this option to have the agent send a message to all trap destinations if the agent receives an SNMP request from a host or community not listed in the "Accepted community names" list or the "Accept SNMP packets from these hosts" list. The message is sent to all hosts in the trap destination list on the Traps property page to indicate that a remote management system failed authentication (potentially indicating an unauthorized access attempt).
2. **Accepted Community Names:** Use this list and the related buttons to alter the list of communities in which the agent participates and the community rights for each. You can select from the following rights:

- a. None: This option prevents the agent host from processing any SNMP requests from the specified community. For instance, you may configure None for the Public community for enhanced security.
 - b. Notify: Select this option to permit the agent host to send traps only to the selected community.
 - c. Read Only: Use this option to permit remote consoles to view data in the local MIB but not change it. This option prevents the agent from processing SNMP SET requests.
 - d. Read Write: Use this option to permit remote consoles to make changes on the managed system. This option allows the agent to process SNMP SET requests.
 - e. Read Create: Use this option to permit the agent to create new entries in the SNMP tables.
3. **Accept SNMP Packets from Any Host:** Select this option to permit the agent to process requests from all hosts in the "Accepted community names" list.
 4. **Accept SNMP PACKETS from These Hosts:** Select this option to define a specific list of hosts from which the agent will process SNMP requests.

Windows Server 2008 Core

There are two different methods to configure the SNMP service.

Method 1

Configure SNMP service on a Windows 2008 server that has the full GUI using the directions above. Using regedit, export the following registry key to a file:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP

Copy the .reg file to the Windows 2008 core machine. On the core machine, run regedit.exe and import .reg file into the HKEY_LOCAL_MACHINE hive.

Method 2

Run the Computer Management on a Windows 2008 server that has the full GUI. Right click on the **Computer Management** from the tree root and select **Connect to another computer...** Enter the name of the Windows 2008 core machine you wish to configure. Configure the SNMP service using the directions above.

SNMP in General

Since we have some space on this page, here are some handy links to help you learn more about SNMP.

Description	Hyperlink
Lessons about SNMP	http://www.et.put.poznan.pl/snmp/main/mainmenu.html
SNMP: Simple? Network Management Protocol	http://www.rane.com/note161.html
Microsoft SNMP implementation	http://msdn.microsoft.com/en-us/library/aa379100(VS.85).aspx

Installing SNMP Informant

SNMP Informant Agent installation programs provide two methods to install:

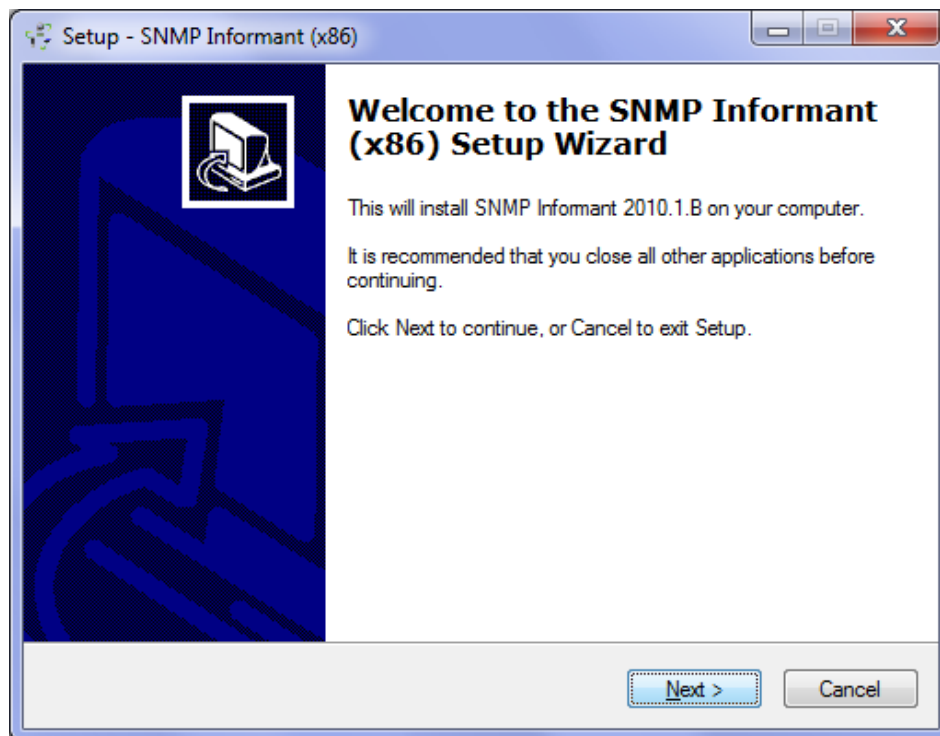
- Graphic user interface (GUI) – A graphics wizard based installation requiring input from the user either with the mouse and keyboard.
- Command line interface – An interface where you can install the software without any intervention from the user. Also known as an unattended or silent install.

This section will cover the installation of the SNMP Informant-Premium product, which includes all SNMP Informant providers. Depending on the product(s) purchased, Individual installs of various SNMP Informant agents will differ somewhat, and specific requirements will be noted.

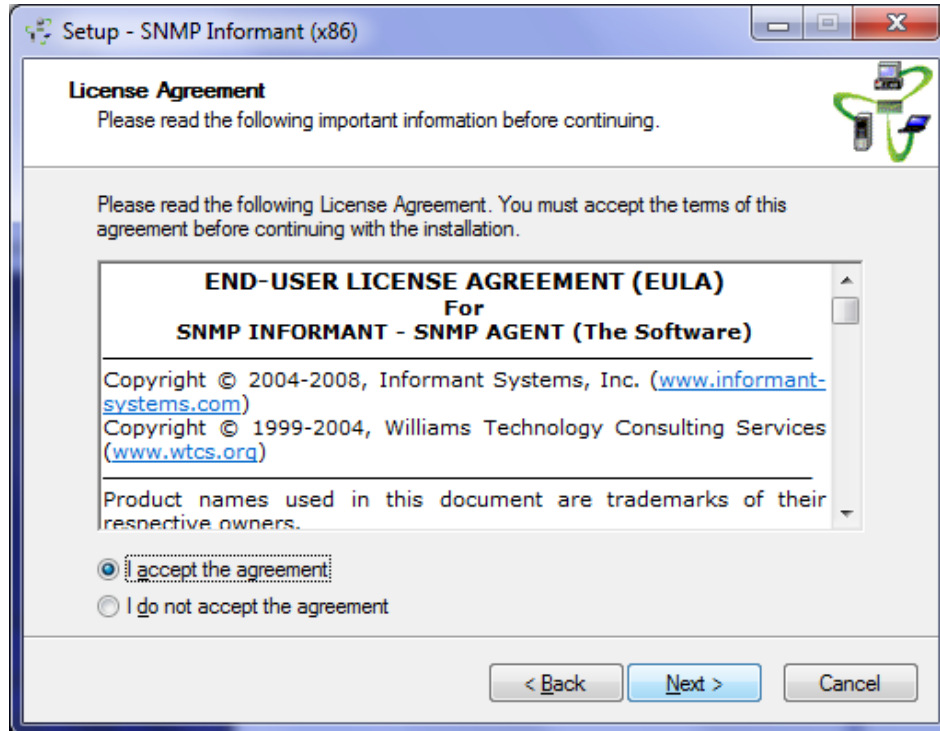
GUI Installation

Start the informant executable.

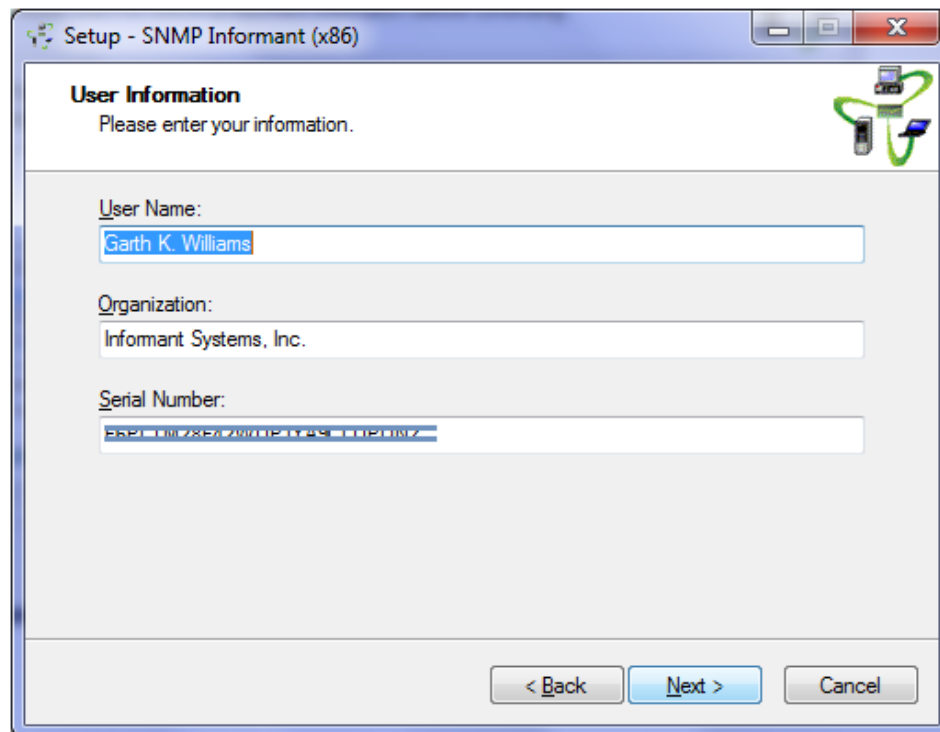
Click the Next button in the welcome screen. The installer will detect whether the host OS is x86, x64, or ia64 bit version.



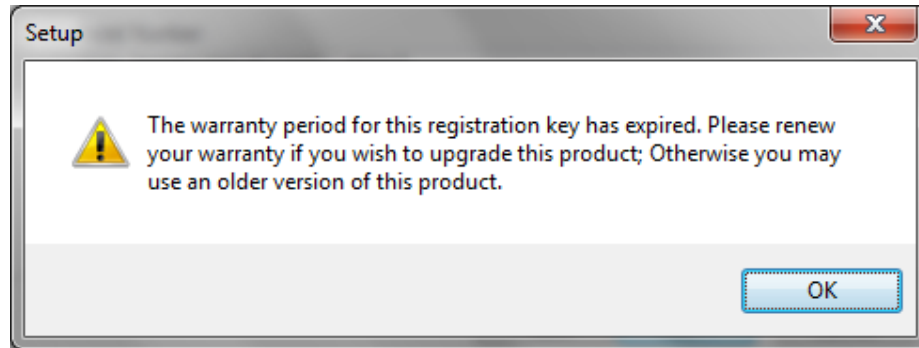
Read the License Agreement and click the "I accept the agreement" radio button if you agree with the license. Click the Next button.



Enter your registration information and the Serial Number/Validation Key supplied with your product. Click the Next button after entering the correct Serial Number/Validation Key.

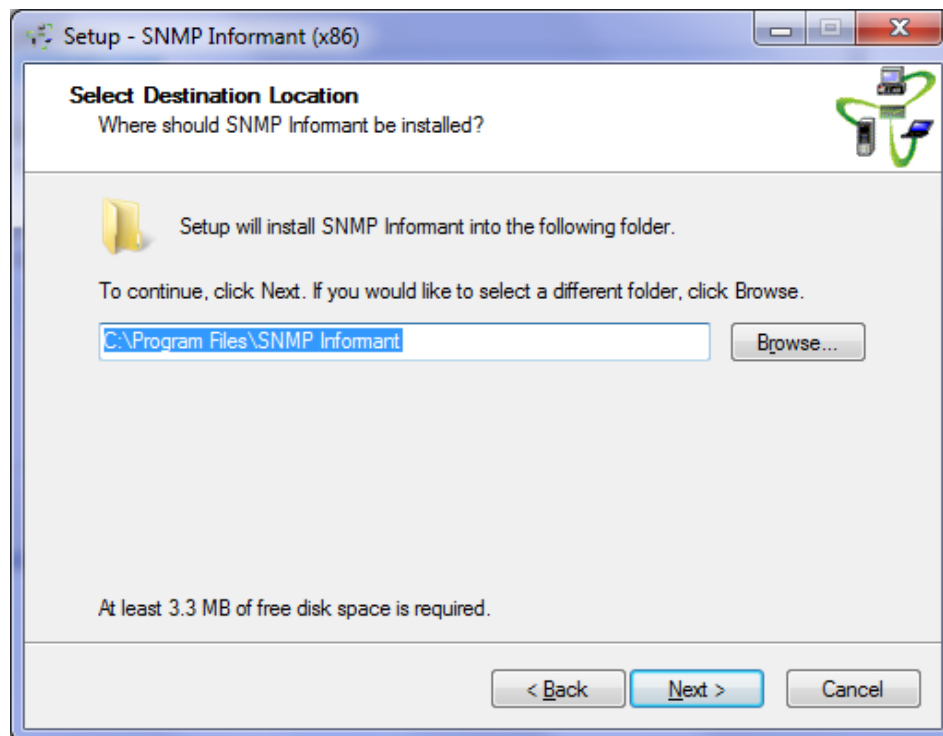


If you are using a validation key that has expired, and are trying to install a newer version of SNMP Informant, you will be presented with a dialog box that tells you so, like this:

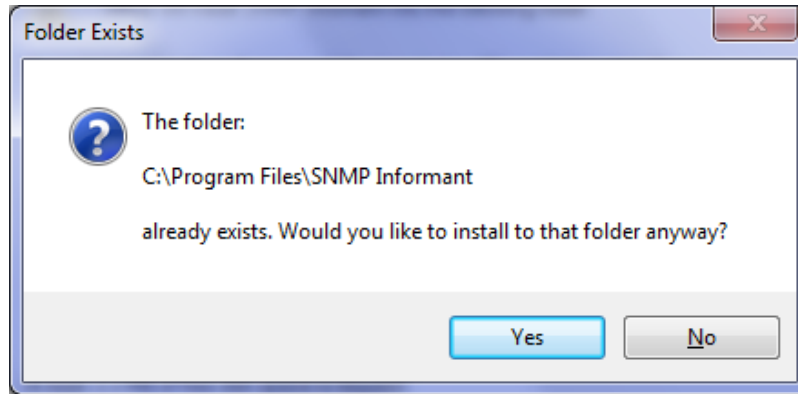


As the dialog box suggests, you can still use an earlier (preceding) version of SNMP Informant, or you can request an evaluation key to get you by for the next 30 days while you upgrade your maintenance. Contact product.support@informant-systems.com for more information.

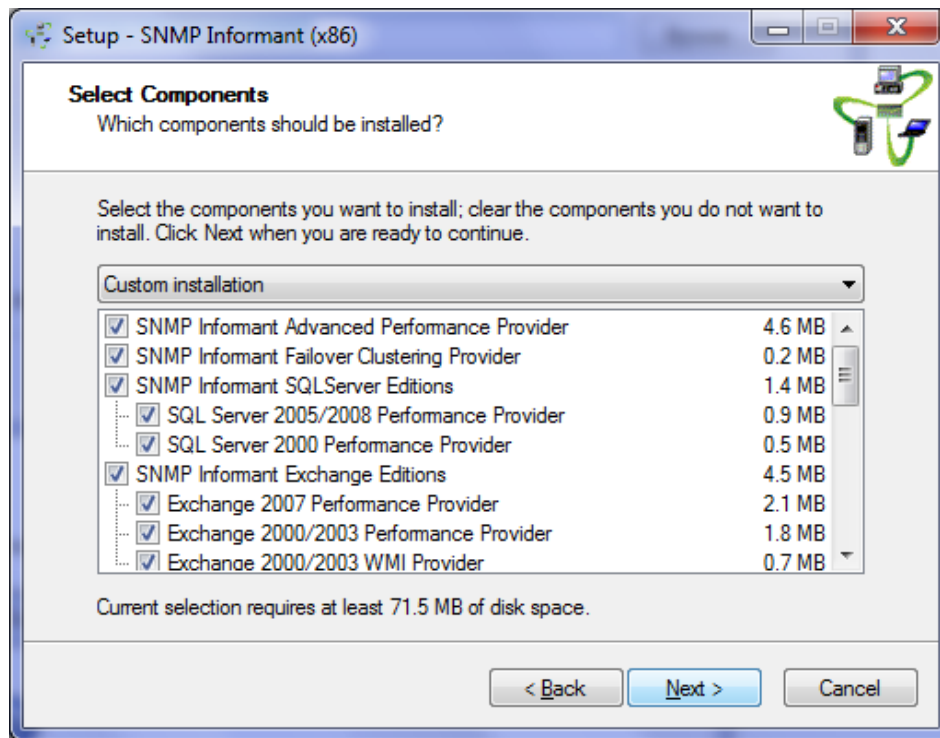
Enter where you would like to install the SNMP Informant Agent. Click the Next button.

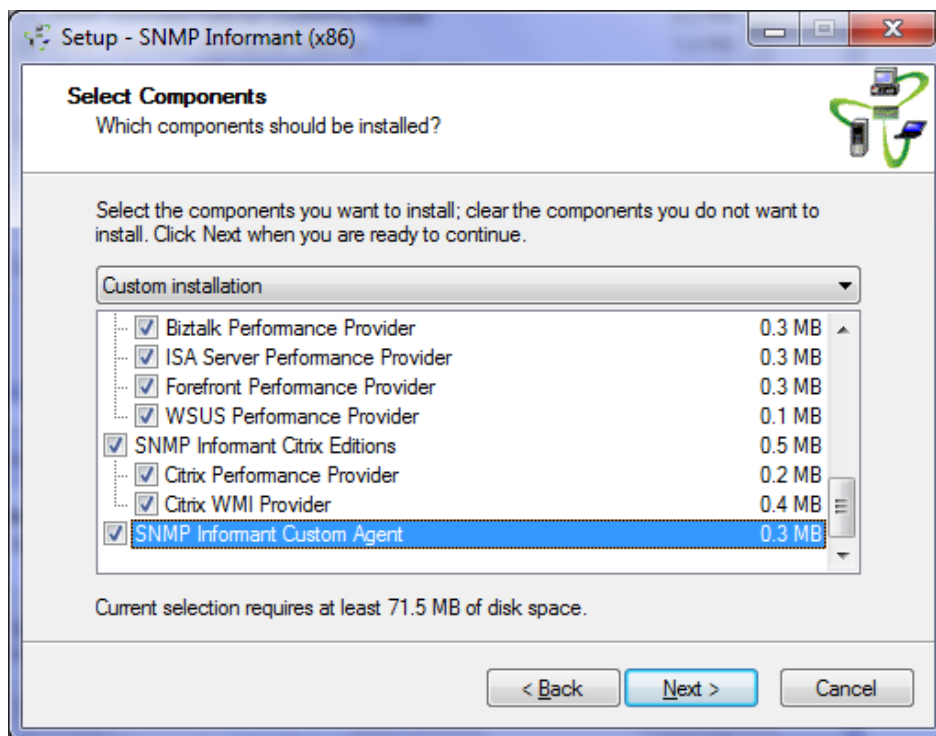
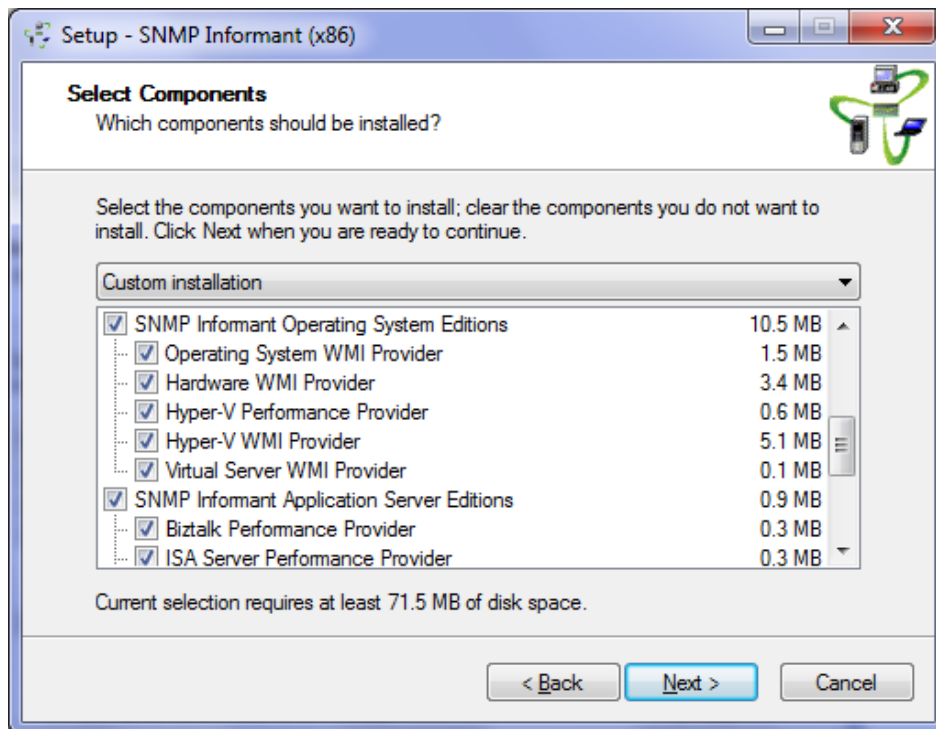


If you already HAVE installed SNMP Informant, you will see this message.

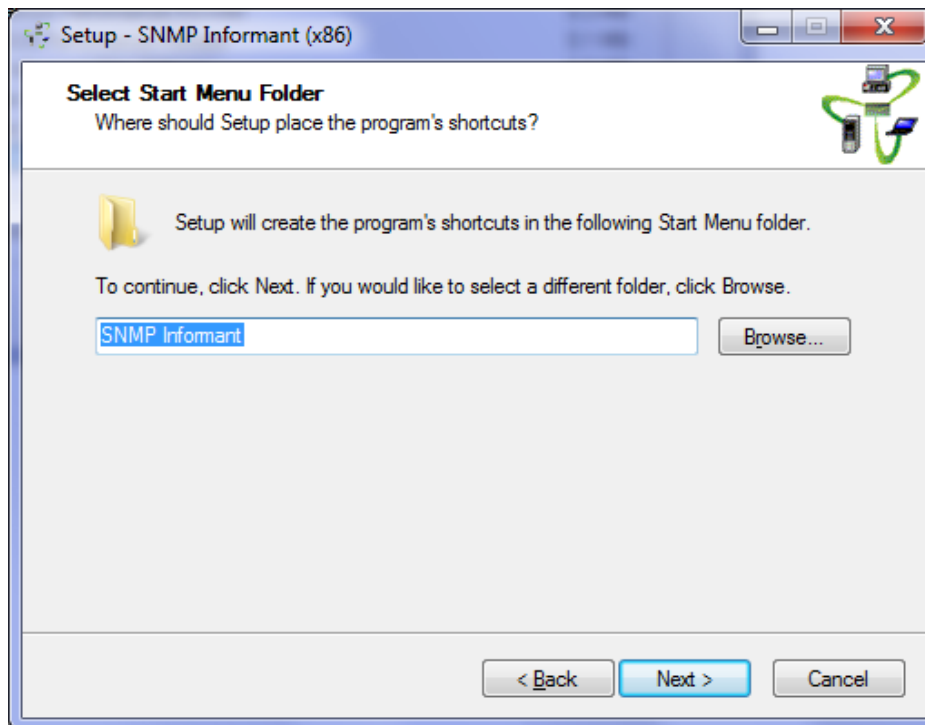


Select the components you would like to install. The validation key you receive with your purchase will determine which components can be selected and installed.

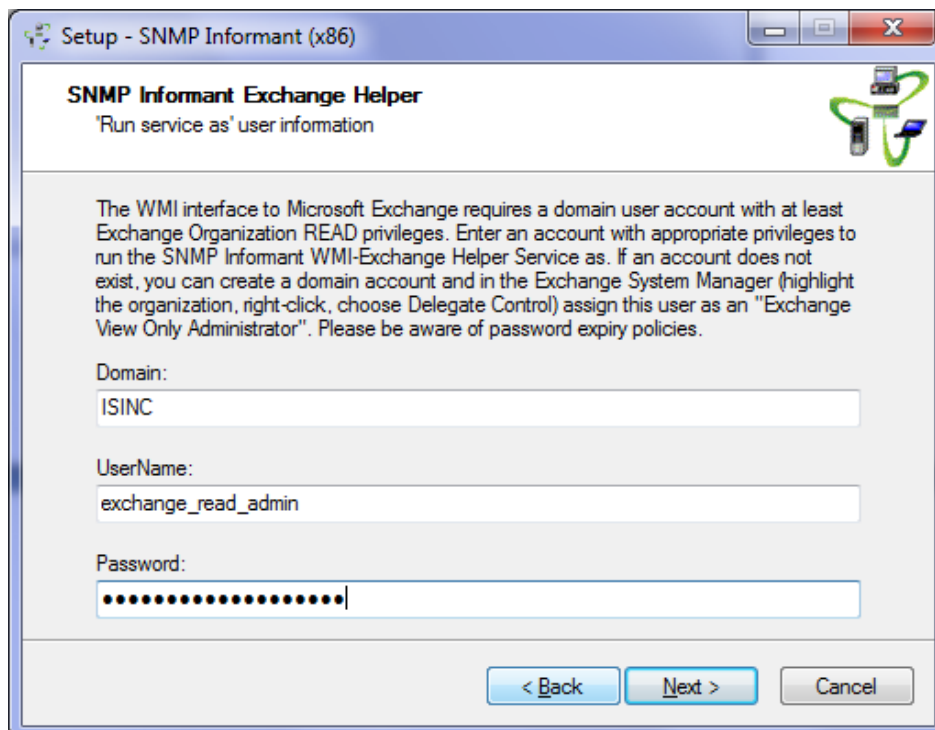




Enter the name of the menu under the Start Menu Folder. Click the Next button.

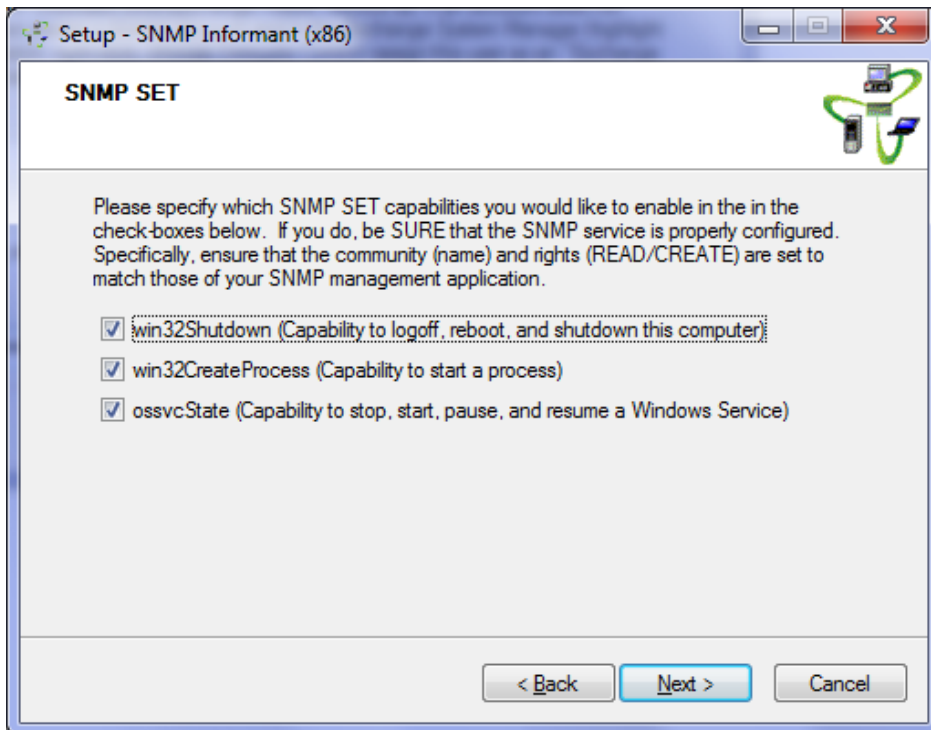


If you are installing the Exchange 2000/2003/2007 WMI Provider agent, you will be prompted for a domain account to start the SNMP Informant Exchange Helper Service. Enter a valid domain user that has delegated Exchange View Only Administrator privileges at least.

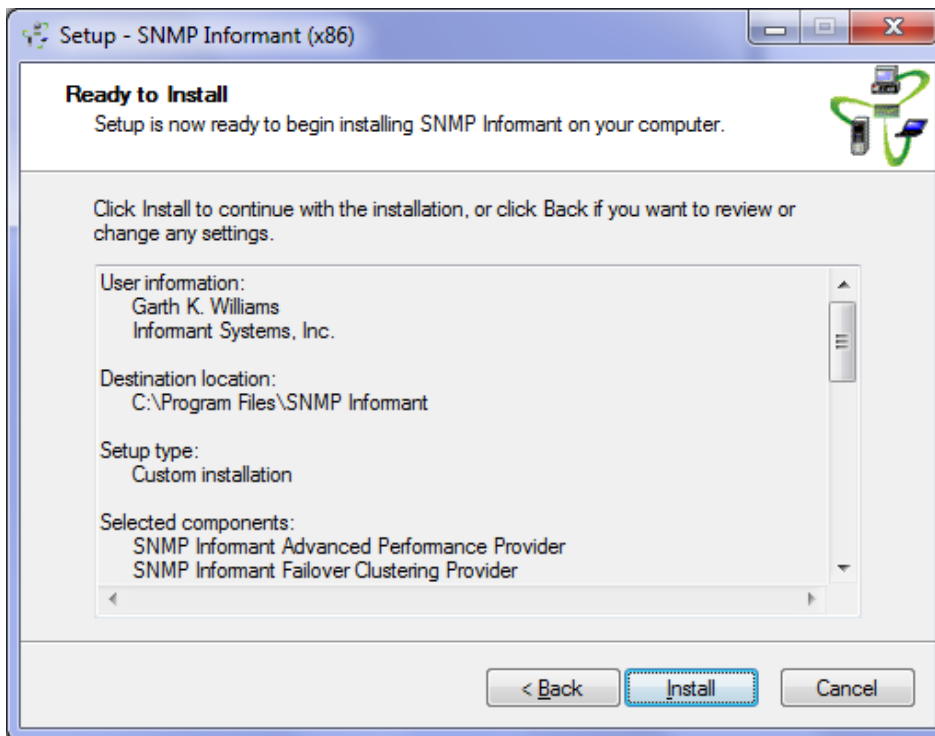


For more information about configuring Exchange 2003/2007 and SNMP Informant, see "Using SNMP Informant".

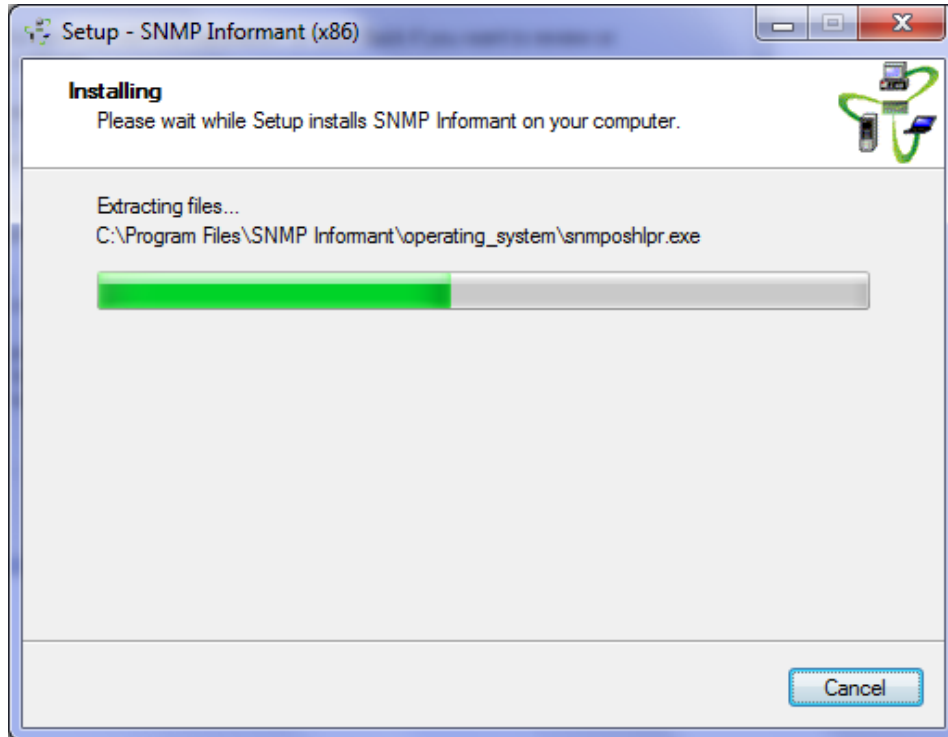
If you are installing the Operating System WMI provider, you will be prompted for SNMP SET functionality you desire to activate within SNMP Informant.



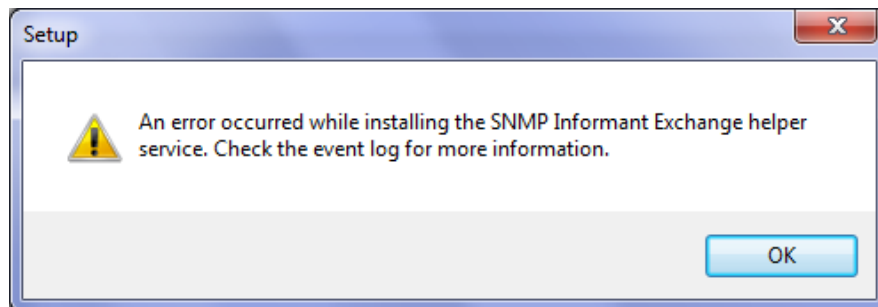
Verify the installation parameters and click the Install button.



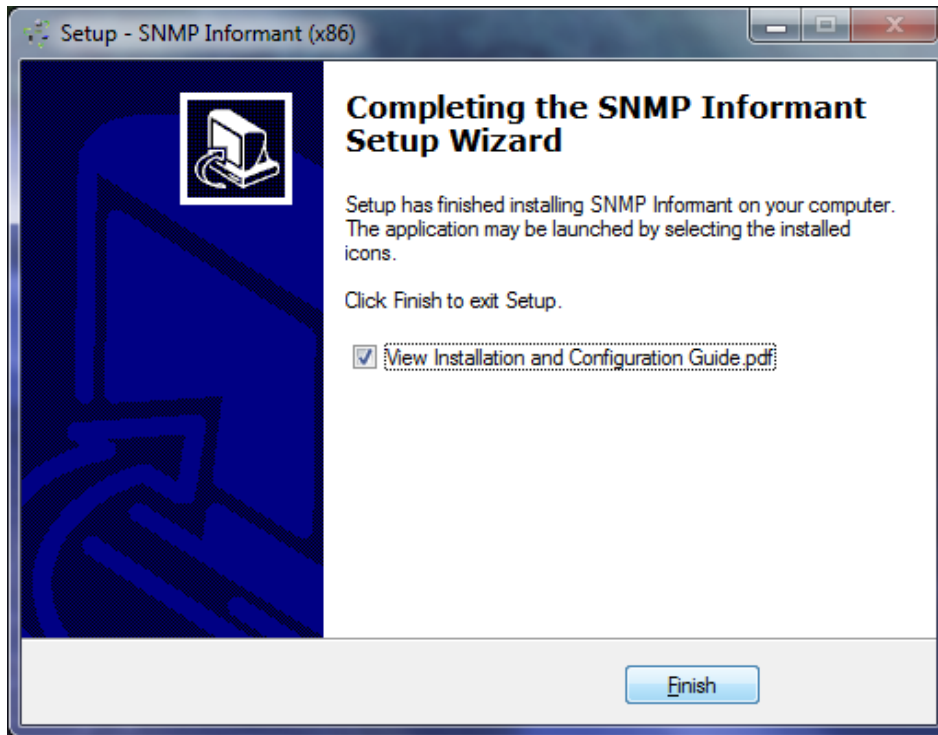
The installer will proceed to install the selected providers, and you will see an installation status bar moving across the install window.



If you have an error with the Exchange Helper Service credentials, you will see this error message during the install (it occurs when the service is started):

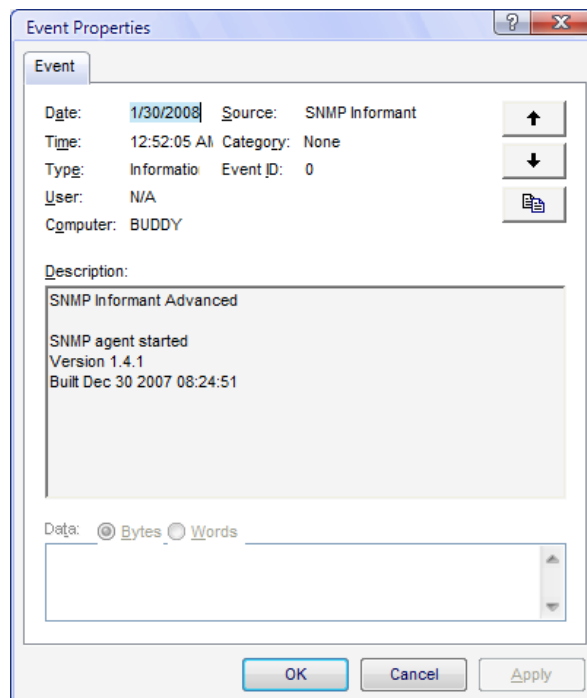


Click the Finish button after installing the SNMP Informant agent. Clear the "View Installation and Configuration Guide.pdf" check-box if you do not want to view this document.



Congratulations! You have now completed the installation of SNMP Informant!

Next, check the Windows Application Event Log to confirm successful SNMP Informant component start-up. Each component will add its own startup message to the Application Event log.



Note: when the SNMP service is stopped, the SNMP Informant extension agents will also shut down, and will post information to that effect in the Application Event Log as well.

Command Line Installation

The Setup program accepts optional command line parameters. These can be useful to system administrators, and to other programs calling the Setup program.

/SILENT, /VERYSILENT

Instructs Setup to be silent or very silent. When Setup is silent the wizard and the background window are not displayed but the installation progress window is. When a setup is very silent this installation progress window is not displayed. Everything else is normal so for example error messages during installation are displayed.

If a restart is necessary and the '/NORESTART' command isn't used (see below) and Setup is silent, it will display a "Reboot now?" message box. If it's very silent it will reboot without asking.

/SUPPRESSMSGBOXES

Instructs Setup to suppress message boxes. Only has an effect when combined with '/SILENT' and '/VERYSILENT'.

The default response in situations where there's a choice is:

- Yes in a 'Keep newer file?' situation.
- No in a 'File exists, confirm overwrite.' situation.
- Abort in Abort/Retry situations.
- Cancel in Retry/Cancel situations.

Yes (=continue) in the following situations:

- DiskSpaceWarning
- DirExists
- DirDoesntExist
- NoUninstallWarning
- ExitSetupMessage
- ConfirmUninstall

Yes (=restart) in a FinishedRestartMessage/UninstalledAndNeedsRestart situation.

5 message boxes are not suppressible:

- The About Setup message box.
- The Exit Setup? message box.
- The FileNotInDir2 message box displayed when Setup requires a new disk to be inserted and the disk was not found.
- Any (error) message box displayed before Setup (or Uninstall) could read the command line parameters.
- Any message box displayed by [Code] support function MsgBox.

/LOG

Causes Setup to create a log file in the user's TEMP directory detailing file installation and [Run] actions taken during the installation process. This can be a helpful debugging aid. For example, if you suspect a file isn't being replaced when you believe it should be (or vice versa), the log file will tell you if the file was really skipped, and why.

The log file is created with a unique name based on the current date. (It will not overwrite or append to existing files.)

The information contained in the log file is technical in nature and therefore not intended to be understandable by end users. Nor is it designed to be machine-parseable; the format of the file is subject to change without notice.

/LOG="filename"

Same as /LOG, except it allows you to specify a fixed path/filename to use for the log file. If a file with the specified name already exists it will be overwritten. If the file cannot be created, Setup will abort with an error message.

/NOCANCEL

Prevents the user from cancelling during the installation process, by disabling the Cancel button and ignoring clicks on the close button. Useful along with '/SILENT' or '/VERYSILENT'.

/NORESTART

Instructs Setup not to reboot even if it's necessary.

/RESTARTEXITCODE=exit code

Specifies the custom exit code that Setup is to return when a restart is needed. Useful along with '/NORESTART'. Also see Setup Exit Codes.

/LOADINF="filename"

Instructs Setup to load the settings from the specified file after having checked the command line. This file can be prepared using the '/SAVEINF=' command as explained below. Don't forget to use quotes if the filename contains spaces.

/SAVEINF="filename"

Instructs Setup to save installation settings to the specified file. Don't forget to use quotes if the filename contains spaces.

/DIR="x:\dirname"

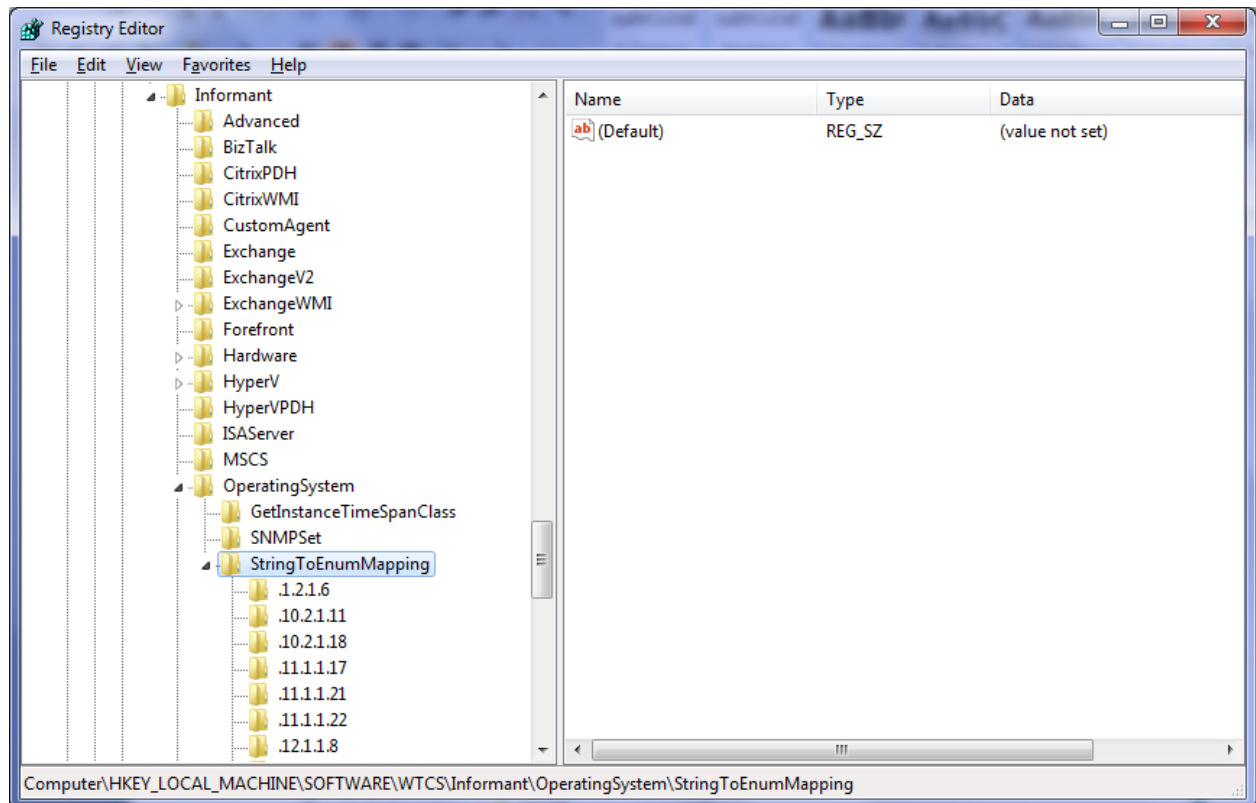
Overrides the default directory name displayed on the Select Destination Location wizard page. A fully qualified pathname must be specified.

/GROUP="folder name"

Overrides the default folder name displayed on the Select Start Menu Folder wizard page. If the [Setup] section directive DisableProgramGroupPage was set to yes, this command line parameter is ignored.

/NOICONS

Note: The WTCS/Informant/Hardware and WTCS/Informant/OperatingSystem keys will have sub-hives called StringToEnumMapping. **DO NOT MODIFY THESE SUB-HIVES!**



Registry Settings and their Meanings

This section describes the registry settings used to control SNMP Informant's behaviour. First of all, let's define what we call a query...

Query: A request made by SNMP Informant to the local subsystem (PDH - Performance Data Helper or WMI – Windows Management Instrumentation, or custom API), based on the SNMP GET, GETNEXT, or WALK request that SNMP Informant receives from a network management application or tool.

Below is a list of registry settings that can be adjusted by the user. Registry setting modifications for SNMP Informant are made at HKEY_LOCAL_MACHINE/SOFTWARE/WTCS/informant/<product>. The changes you make are at the <product> level are unique for that agent. **Any other registry settings not described below within the WTCS/informant registry should not be changed and modifying the value may cause unpredictable results.**

Setting: EventFilterMask

Applies to: All Agents

Registry Type: DWORD

Default Value: 7

Units: numeric (decimal)

The EventFilterMask value controls the level of messages SNMP Informant posts into the Application Event Log. Valid values and their meanings are:

Value	Meaning
7	Log Information, Error and Warning messages
6	Log Warning and Information messages
5	Log Information and Error messages
4	Log Information messages
3	Log Error and Warning messages
2	Log Warning messages
1	Log Error messages
0	Log no messages

Setting: PdhInterface

Applies to: PDH Agents (Advance, BizTalk, Exchange PDH, ISA Server, SQL Server).

Registry Type: STRING

Default Value: .

The PDH interface used to gather performance information through. The dot (.) represents the local machine interface. Set the value to "null" to use an alternative interface. The dot local machine interface is slower than the null interface, but the null interface has demonstrated symptoms of memory leakage in rare circumstances.

Setting: MaxQueryCacheSize

Applies to: PDH Agents (Advance, BizTalk, Exchange PDH, ISA Server, SQL Server).

Registry Type: DWORD

Default Value: 300 PDH

Units: Number of queries

The number of different queries that can be cached for **both** GET and GETNEXT queries per agent. When a request comes in, it looks for the query associated with the OID in the cache. If it doesn't exist, then it creates a query and caches it. The cache only contains entries that require multiple samples. For example, the CPU object will be in the cache, but the Memory usage will not, because the memory object counters are an "as at" (right now) sample. CPU on the other hand, is a calculated average value based on two separate samples. Both the last value and the query itself is stored. The query is used to take another sample. The last value is used for the computation to determine the average value. Increase this value for the necessary agent if you are receiving an error message from SNMP Informant stating that the query cache size was exceeded.

Setting: QueryLifeSpan**Applies to:** PDH Agents (Advance, BizTalk, Exchange PDH, ISA Server, SQL Server).**Registry Type:** DWORD**Default Value:** 21600000**Units:** milliseconds

This is the length of time a query (and the accompanying value) can exist in the cache without being requested before it is purged. Default time is 6 hours. If the query lifespan expires, then the query (and accompanying value) is deleted. Once this query is purged from the cache, a computation between it and a new query cannot be performed. Should this be the case, the new query is stored in the cache with a sample value of 0 (in preparation for a second query, where the new value and 0 will be used to calculate an average). If a query that exists in the cache is re-requested, the QueryLifeSpan counter restarts for that query. Increase this value if you are querying the same OID more than 6 hours between samples.

Setting: MinimumQueryRate**Applies to:** PDH and WMI Agents**Registry Type:** DWORD**Default Value:** 5000 for PDH/30000 for WMI**Units:** milliseconds

This registry setting determines how often a new value is gathered and a calculation is performed. SNMP uses the UDP (a lossy network protocol) to communicate with the managing station. Since the response can be lost or the managing station would timeout on the SNMP query and many calculation are based on the difference between the last raw value and the current raw value, the SNMP Informant agent will return the previous calculated value if the same request is made within the MinimumQueryRate registry defined period. This is done to prevent returning false calculated due to the SNMP Managing Station re-querying the request assuming that the packet was lost. A user would reduce this value if they are querying the same OID less than every 5 seconds.

Setting: GetInstanceTimeSpan**Applies to:** PDH and WMI Agents**Registry Type:** DWORD**Default Value:** 60000 for PDH/21600000 for WMI**Units:** milliseconds

This registry setting is used to identify when to look for new instances a PDH object. For example, when iterating across the "process" PDH object, there is a performance hit whenever you looked for a new instances. To minimize response time, we only look for new instances whenever the GetNextInstanceTimeSpan (default time is 60000 seconds) expires or we switch to a different PDH counter/object. Setting this value to a lower number will keep your instance list more accurate (current), but will do so at the expense of performance.

Setting: GetInstanceTimeSpanClass\...**Applies to:** WMI Agents (OS, Hardware, Exchange, Hyper-V, Virtual Server)**Registry Type:** DWORD**Default Value:** 21600000**Units:** milliseconds

This registry setting is the same as GetInstanceTimeSpan above, but for a specific WMI class. This allows the user to set the instance refresh interval on independent MIB branches. This value overrides the global GetInstanceTimeSpan value.

Setting: QueryKeepAlive**Applies to:** WMI Agents (OS, Hardware, Exchange, Hyper-V, Virtual Server)**Registry Type:** DWORD**Default Value:** 10000**Units:** milliseconds

This registry setting specifies the maximum amount of time between two SNMP queries of the same MIB branch that prevents a new iterator to be generated. This prevents an iterator from changing in the middle of an SNMP walk, however, this will also prevent a new instances to be retrieved if the MIB branch is being queried constantly at a rate less than this registry setting.

Setting: WMIFunctionTimeout**Applies to:** WMI Agents (OS, Hardware, Exchange, Hyper-V, Virtual Server)**Registry Type:** DWORD**Default Value:** 4250**Units:** milliseconds

The amount of time to wait for a WMI function call to return before abandoning the call. This registry setting should be increased if you are having difficulty iterating across a MIB object due to the WMI call not being able to complete in time. Ensure that you also increase the HelperResponseTimeout registry setting for WMI-OS and WMI-Exchange agents.

Setting: HelperResponseTimeout

Applies to: WMI-OS and WMI-Exchange Agents

Registry Type: DWORD

Default Value: 4500

Units: milliseconds

This registry setting applies to SNMP Informant WMI agents that have a "helper service" only. It refers to the number of milliseconds the agent (extension DLL) should wait for a response from the SNMP Informant helper service before timing out. The WMIFunctionTimeout should be increased appropriately if this registry setting is changed.

Setting: SpawnDirectory

Applies to: WMI-OS Agent

Registry Type: REG_SZ

Default Value: <installdirectory\spawn> (eg. C:\Program Files\SNMP Informant\operating_system\spawn\)

Units: alphanumeric

This registry setting tells the SNMP Informant WMI-OS agent where scripts and executables that might be remotely spawned should start from.

Using SNMP Informant

For the most part, once the SNMP service is properly installed and configured, SNMP Informant providers are ready immediately after install. Most providers require little or no configuration at all. If you need to “tune” SNMP Informant, see the “Configuring SNMP Informant” section.

After the SNMP Informant provider(s) have been installed and started (by the base SNMP agent), they can be queried by your SNMP Manager software.

General Usage Notes

OID Tree Listings

Please see the file in [install loc]\SNMP Informant\[product]\mibs\informant-[product]-tree.txt for a complete tree listing of the OIDs supported by the various versions of SNMP Informant. For example:

- [install loc]\SNMP Informant\advanced\mibs\informant-adv-tree.txt
- [install loc]\SNMP Informant\sqlserver\mibs\informant-sqlserver-tree.txt
- [install loc]\SNMP Informant\ExchangeWMI\mibs\informant-exchange-tree.txt

Use the Correct MIBS

Be sure to select the correct SNMP version of MIBS for your monitoring application or MIB Browser. SNMP Informant comes with both SMIV1 (SNMPv1) and SMIV2 (SNMPv2) MIBS. You can access the SNMP Informant MIBS in the product install directory. Their location will be in directories similar to the following:

- C:\Program Files\SNMP Informant\[product]\mibs\SMIV1 or SMIV2

Uninstalling SNMP Informant

The uninstall program included with SNMP Informant will remove the registry entries and clean up quite nicely, but you may need to manually remove the \Program Files\SNMP Informant\[product] directory yourself after the uninstall program has completed. This is because you may have created scripts etc. in subdirectories, and we don't want to delete your work.

Using the PDH Providers

The SNMP Informant PDH providers are a bridge between the standardized SNMP protocol and the non-standard Windows performance information. Understanding how Performance Counters are referenced is necessary before grasping the SNMP OID structure.

As seen when adding a performance counter using the Windows Performance Monitor, a specific counter item is referenced using at least two names (object and counter) and where required a third name (instance).

The object name is the group the performance item is associated with (i.e. memory, processor, process, etc.).

The counter name is the specific type of performance information queried for that object (e.g., the percentage of CPU time for the processor object).

The instance name is the specific instance that the query is being performed on (e.g., CPU 0 for the processor object, the lsass.exe process, etc). *The instance name is always referenced as a string.*

Refer to <http://technet2.microsoft.com/windowsserver/en/library/3fb01419-b1ab-4f52-a9f8-09d5eb9ef21033.mspx?mfr=true> for more information on performance object, counters, and instances.

The illustration below shows how to relate Performance Counters, Objects and Instances to an SNMP Informant OID (in this case for Memory: Available bytes):

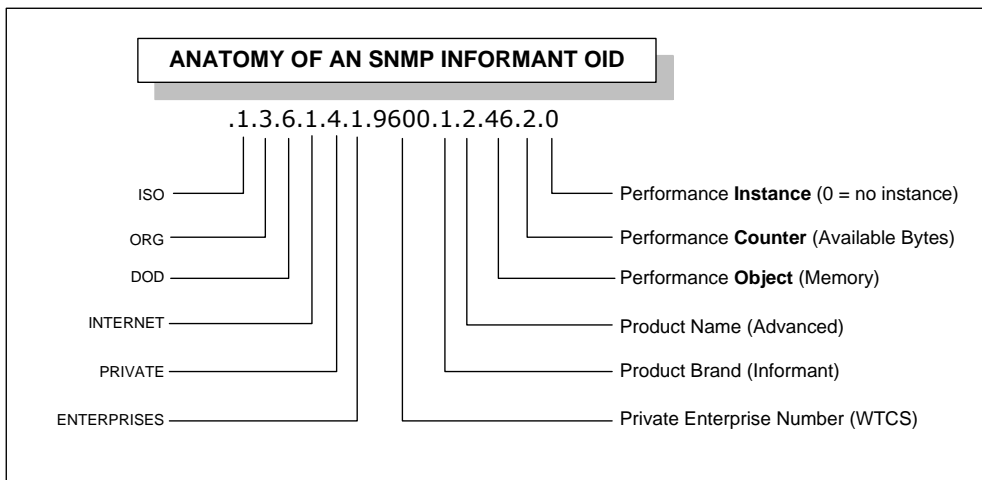


Figure 5 – Anatomy of an SNMP Informant OID

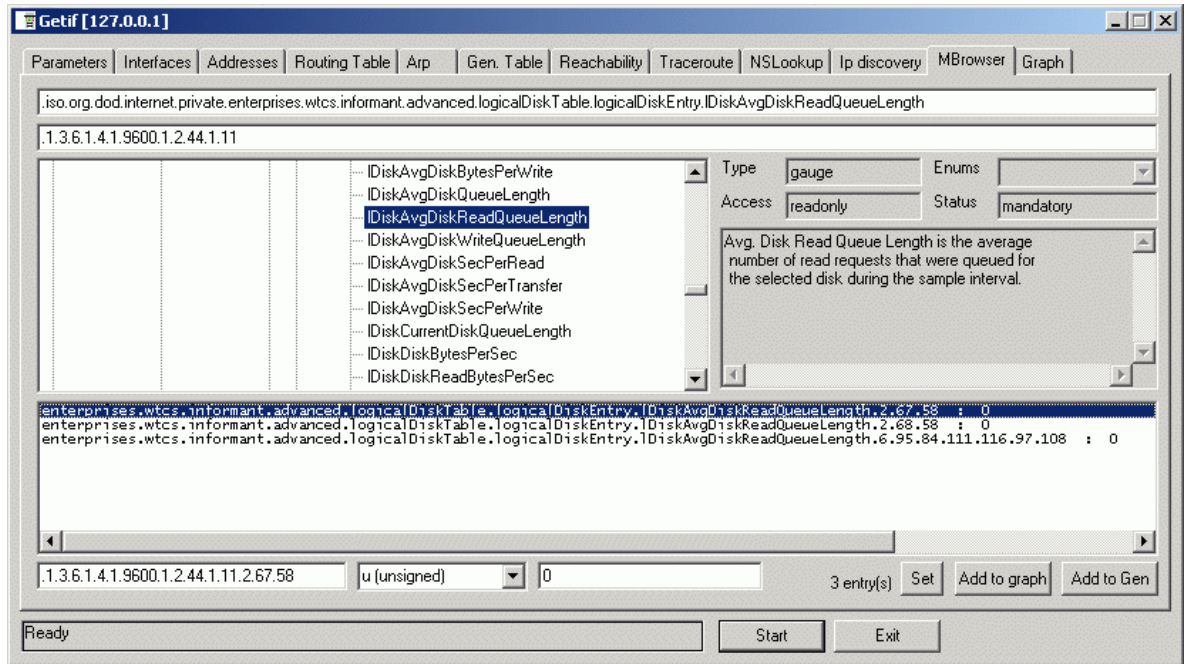
More detailed OIDs can contain instance names. For example, it is not uncommon for a server to have multiple disks, processors and network adapters. Therefore, OIDs for these performance objects will have multiple instances.

Use the modified ASCII chart below to make it easier to read SNMP Informant instance OID values, and convert them to their ASCII equivalent. We have removed the Hex and Octal values, leaving only the Decimal values.

Decimal to ASCII conversion applies to many SNMP Informant PDH agent tables, where the information (name) is pulled directly from the Performance Data Helper (we don't make the names up).

Here are four examples of converting an SNMP Informant Instance to an ASCII (character) equivalent. For ease of reading, we will always assume that SNMP Informant agent is the Advanced version, and the prefix will be .iso.org.dod.internet.private.enterprises.wtcs.informant.advanced (.1.3.6.1.4.1.9600.1.2), and the walk will occur below that point. The first number after the fully qualified OID (.1.3.6.1.4.1.9600.1.2) is the OBJECT. After that comes the COUNTER. After THAT comes the INSTANCE. The **first number** after the INSTANCE tells us how many **characters** follow, and the characters make up the NAME of the INSTANCE. The dots between the **characters** can be removed to make up the name (Character (ASCII) Equivalent) but are required from an SNMP perspective)

Example 1: LogicalDisk: Logical Disk Average Read Queue Length
 (we've included a Getif Screenshot in this example to provide further detail)



Fully qualified SNMP Informant OID (walk from here)	SNMP Informant Instance (Decimal) OID response	Character (ASCII) Equivalent
.logicalDiskTable.logicalDiskEntry.IDiskAvgDiskReadQueueLength	.2.67.58	C:
.logicalDiskTable.logicalDiskEntry.IDiskAvgDiskReadQueueLength	.2.68.58	D:
.logicalDiskTable.logicalDiskEntry.IDiskAvgDiskReadQueueLength	.6.95.84.111.116.97.108	_Total

.2 indicates that 2 **characters** follow
.6 indicates that 6 **characters** follow

Example 2: Processor: % Processor Time

Fully qualified SNMP Informant OID (walk from here)	SNMP Informant Instance (Decimal) OID response	Character (ASCII) Equivalent
.processorTable.processorEntry.cpuPercentProcessorTime	.1.48	0
.processorTable.processorEntry.cpuPercentProcessorTime	.1.49	1
.processorTable.processorEntry.cpuPercentProcessorTime	.6.95.84.111.116.97.108	_Total

- .1** indicates that 1 **character** follows
- .6** indicates that 6 **characters** follow

Example 3: Network Interface: netBytesTotalPerSecond

Fully qualified SNMP Informant OID (walk from here)	.networkInterfaceTable.networkInterfaceEntry.netBytesTotalPerSec	
SNMP Informant Instance (Decimal) OID response	.25.77.83.32.84.67.80.32.76.111.111.112.98.97.99.107.32.105.110.116.101.114.102.97.99.101	
Character (ASCII equivalent)	MS TCP Loopback interface	

- .25** indicates that 25 **characters** follow

Fully qualified SNMP Informant OID (walk from here)	.networkInterfaceTable.networkInterfaceEntry.netBytesTotalPerSec	
SNMP Informant Instance (Decimal) OID response	.27.72.80.32.78.67.51.49.54.51.32.70.97.115.116.32.69.116.104.101.114.110.101.116.32.78.73.67	
Character (ASCII equivalent)	HP NC3163 Fast Ethernet NIC	

- .27** indicates that 27 **characters** follow

Example 4: PhysicalDisk: Physical Disk Average Disk Queue Length

Fully qualified SNMP Informant OID (walk from here)	SNMP Informant Instance (Decimal) OID response	Character (ASCII) Equivalent
.physicalDiskTable.physicalDiskEntry.pDiskAvgDiskQueueLength	.4.48.32.67.58	0 C:
.physicalDiskTable.physicalDiskEntry.pDiskAvgDiskQueueLength	.4.49.32.68.58	1 D:
.physicalDiskTable.physicalDiskEntry.pDiskAvgDiskQueueLength	.6.95.84.111.116.97.108	_Total

- .4** indicates that 4 **characters** follow
- .6** indicates that 6 **characters** follow

SNMP Informant Decimal OID instance to ASCII Character Conversion Table

Decimal Value	Character Value
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	SPACE

Decimal Value	Character Value
33	!
34	"
35	#
36	\$
37	%
38	&
39	'
40	(
41)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?

Decimal Value	Character Value
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
91	[
92	\
93]
94	^
95	_
96	`

Decimal Value	Character Value
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	{
124	
125	}
126	~
127	DEL

= commonly seen values

Using the WMI-Exchange Provider

Exchange Server 2003 and 2007 has different levels of administrative responsibility, and supports three types of administrative roles: Exchange Full Administrator, Exchange Administrator and Exchange View Only Administrator.

Exchange Full Administrator: This role has total control over the Exchange organization, and can delegate administrative roles to other users.

Exchange Administrator: This role is identical to the Exchange Full Administrator role, but the Exchange Administrator role lacks have the power to delegate administrative roles to other users.

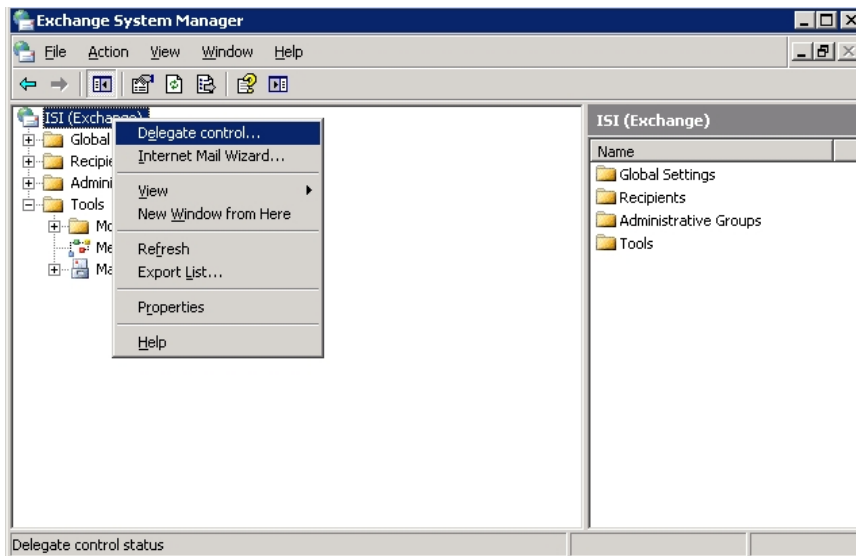
The Exchange View Only Administrator: In Exchange Server 2003, this role is intended for administrators to use during training. The Exchange View Only Administrator role gives administrators-in-training the ability to browse through the Exchange System Manager (ESM), but no power to make any changes.

The SNMP Informant Exchange Helper service must be configured to automatically start and run (Log On As) a user with at least Exchange View Only Administrator privileges.

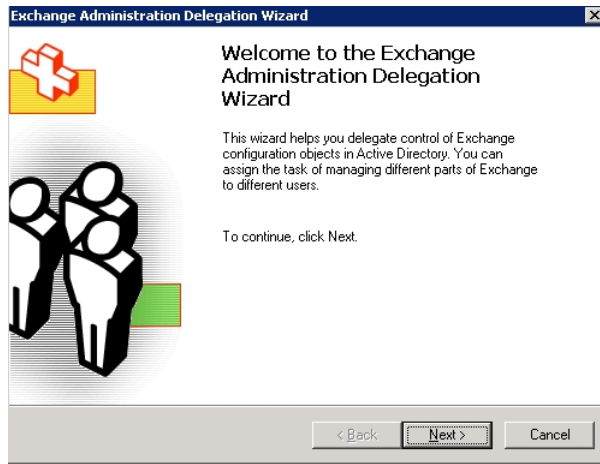
Assuming that no Exchange View Only Administrator exists, one can be created. Simply create a domain account called exchange-read-admin. Assign a password that you can remember, and set it to no expiry. While this is not optimal, if password expiry is allowed, the SNMP Informant Exchange Helper service will eventually stop working.

Next, create a domain group called Exchange-Read-Admins, and put the user you just created above in that group.

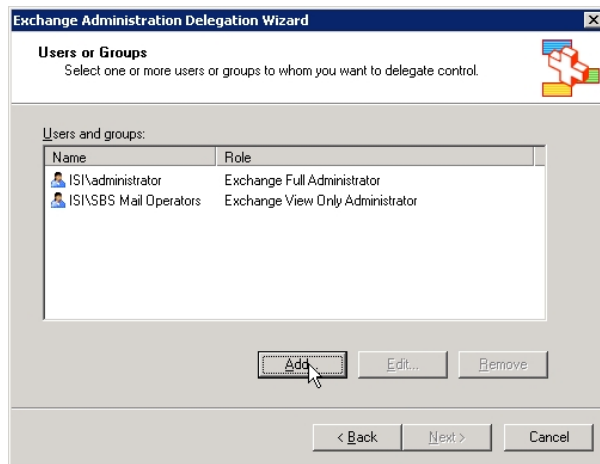
Then, start the Exchange System Manager (ESM) as a user with Exchange Full Administrator privileges. Then, select the Organization name, right click it, and choose Delegate Control.



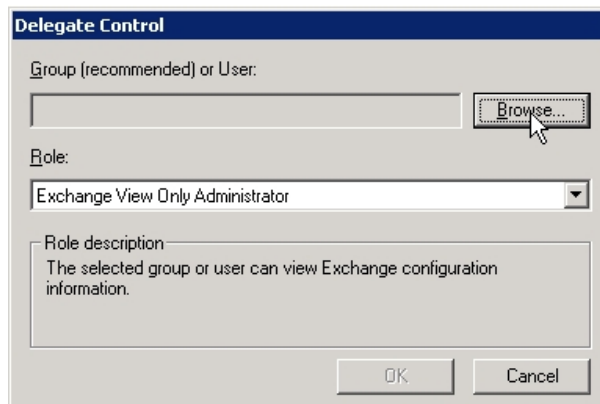
This will start the Exchange Administration Delegation Wizard.



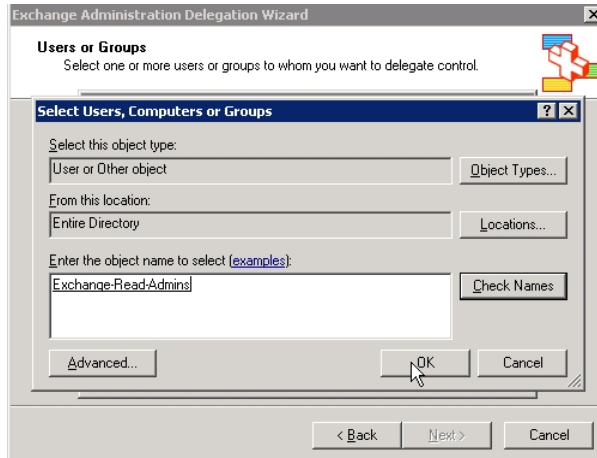
Click Next, and add the Exchange-Read-Admins domain group you just created.



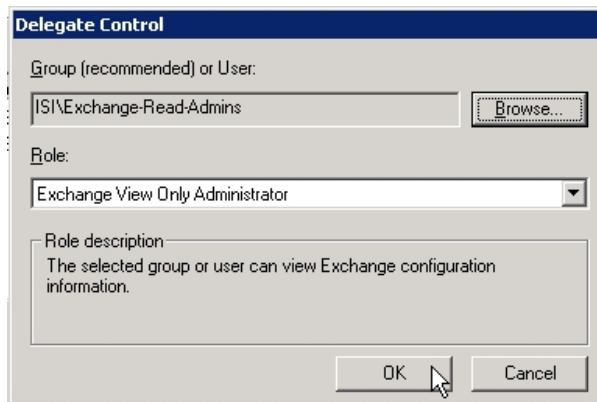
Click Add, then Browse



Type **Exch** in the box, and press Check Names. You should then be able to pick the domain group called Exchange-Read-Admins, created earlier.

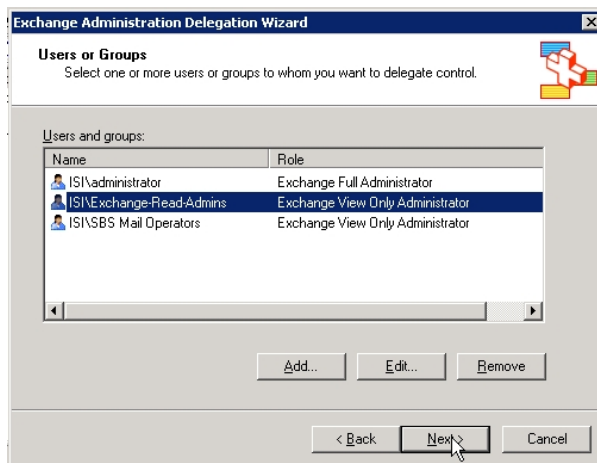


Click OK.



Verify that the right group is in the Exchange View Only Administrator's Role, and press OK.

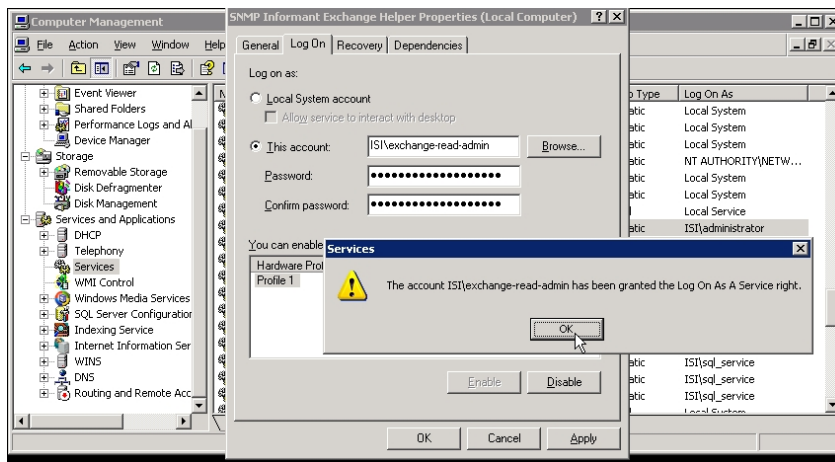
Press Next



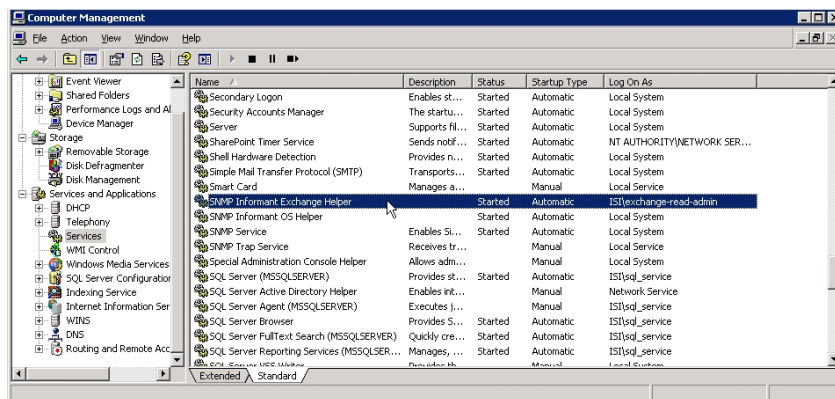
Press Finish to complete the role assignment.



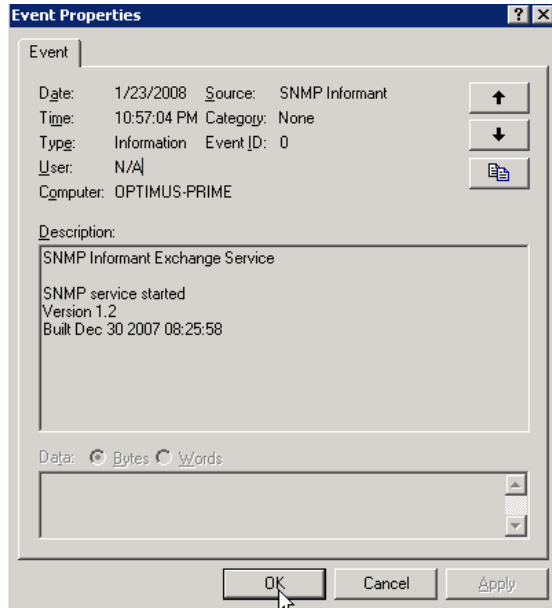
Now, set the SNMP Informant Exchange Helper service to start and run logged in as the exchange-read-admin account created earlier.



Once you have completed the service configuration, restart the SNMP Informant Exchange Helper Service. Check to make sure it started with no error.



Finally, check the Application event log for a successful start message from SNMP Informant.



Using the WMI-OS Provider

Like the WMI-Exchange provider, the WMI-OS provider also has a helper service. It includes support for SET commands sent to it from the NMS. By populating an OID with a value, and sending a SET command, you can control the WMI-OS provider.

Win32Shutdown (.1.3.6.1.4.1.9600.1.22.8.9.0)

The win32Shutdown OID allows a user to remotely logoff, restart, or shutdown a destination computer using SNMP SET. Make sure to add the .0 to the end of the OID when setting the value. The OID must be set to an INTEGER value representing one of the following states:

Value	Meaning
12	Forced Power Off
8	Power Off
6	Forced Reboot
5	Forced Shutdown
4	Forced Log Off
2	Reboot
1	Shutdown
0	Log Off

Win32CreateProcess (.1.3.6.1.4.1.9600.1.22.9.3.0)

The win32CreateProcess OID allows a user to remotely start a process on the destination computer using SNMP SET. Make sure to add the .0 to the end of the OID when setting the value. The OID must be set to a STRING value representing the start command you wish to run. For security reasons, the initial program can ONLY be run out of the "spawn" sub-directory below where the agent is installed on the computer (example below). You can configure the location of the spawn directory using SNMP

Informant registry settings (See the "Configuring SNMP Informant" section for more information on how to do this).

C:\Program Files\SNMP Informant\operating_system\spawn\).

The initial program (in the "spawn" directory could then call or execute programs in other directories if necessary/as required.

Below are some examples of acceptable and not acceptable forms of strings for the SNMP SET command.

Acceptable Forms

```
Start_prog.cmd
progA.vbs
mytool.exe
start_prog.cmd C:\winnt\system32\notepad.exe
mytool.exe C:\temp\input.xml
junk.txt
```

NON-Acceptable Forms

```
C:\winn\system32\notepad.exe
C:\winnt\system32\cmd /c test.cmd
\\computerA\c$\temp\run.exe
..\..\test.cmd
```

The process will be created with the user account that the SNMP service runs with, normally the SYSTEM account. If you wish to create the process a different account than what the SNMP Service is executed as, you can write a script which would run a tool like "su.exe" to execute as a different user.

By default, GUI application will not be displayed on the console window. If the GUI application must be shown, then you must enable "Allow service to interact with desktop". An administrator can do this by running the Windows Service Manager, double clicking on the SNMP Service entry, clicking on the "Log On" tab, and enabling the "Allow service to interact with desktop" checkbox, then clicking the "Okay" button, and restarting the SNMP service. Doing this will cause GUI applications started by the SNMP Informant Operating System agent to appear on the destination computer's console when executed.

ossvcState (.1.3.6.1.4.1.9600.1.22.11.1.1.21.x)

The ossvcState OID allows a user to remotely start, stop, pause, or resume a Windows Service on the destination computer using SNMP SET. The .x at the end of the OID is the instance number of the service you wish to perform the action on. The OID must be set to an INTEGER value representing the final state you wish to put the service into. Below is a list of valid final states you can set the service to:

Value	Meaning
7	Paused
4	Running (or resume)
1	Stopped

For example, if the service is in the "stopped" state and you wish to start it, you would set the state to "running". If you would want to pause a service, you would set it to the "paused" state. If you wish to resume a service, you would set it to the "running" state.

Here's an example:

First, you need WALK the Service Caption OID (.1.3.6.1.4.1.9600.1.22.11.1.1.4) to identify the service you want to manipulate. There you will see a list of all the services identified by an instance number. Let's assume that `ossvcCaption.71`, or `.1.3.6.1.4.1.9600.1.22.11.1.1.4.71` = Task Scheduler

Then navigate to the `ossvcState` OID (.1.3.6.1.4.1.9600.1.22.11.1.1.21). WALK this OID, and you will see the instance numbers again, and their current states. Find the instance number that matches the process you want to manipulate (as identified in the first step). In this example, `.ossvcState.71` = running

SO ... in order to stop the Task Scheduler service, you would send an SNMP SET to `.1.3.6.1.4.1.9600.1.22.11.1.1.21.71` with a value of 1 (Stop).

Using the MSCS Provider

The SNMP Informant Cluster Server Agent allows you to collect Microsoft Cluster Services information remotely using SNMP, by linking into the Cluster Services components (on your Windows 2000 or Windows 2003 cluster server, of course).

Cluster Services *must* be present on the server in order for the SNMP Informant Cluster Server Agent to install.

First, install the MSCS Informant agent on all the clustered computers. Then, when you want to collect cluster server information, query the *cluster name* rather than each computer individually. The information is redundant on all clustered computers, so by querying the cluster name, you will get the information from the active node.

Second, clustered resources are assigned to cluster groups. Whenever a cluster resource fails (e.g., a node failure or a SCSI interface failure), the entire group is moved to one of the other nodes in the cluster. Since SNMP Informant currently does not support any trap based notifications, the best way to monitor for a failure is to poll either `mscsResGroupOwnerNode` or `mscsResourceOwnerNode`. If the owner ever changes, it's usually because the owner node has shutdown/restarted, a hardware failure occurred, somebody manually moved the cluster resources through the Cluster Administrator interface, or a failback occurred. If a cluster resource has permanently failed (e.g., a permanent clustered disk failure), then you can monitor `mscsResGroupState` or `mscsResourceState` for this failure.

The "State" OIDs will not pick up transient changes (e.g., a resource group moving from one computer to another successfully) unless you poll at a fairly high frequency. You can also monitor `mscsNodeState` to see if a specific cluster node is up or down.

Using the Custom Provider

The SNMP Informant Custom Agent allows you define your own SNMP query structure and query rules. This agent is only available in Premium licenses of the SNMP Informant software. The agent currently supports three different types of query rules:

- **RegistryQuery** – Query a remote system's registry and get the contents back using SNMP.
- **ExecuteQuery** – Remotely execute a program or script, and get the return value back using SNMP.
- **PerformanceQuery** – Query any performance counter on a remote system.

This functionality adds a new level of flexibility to SNMP Informant and further extends SNMP capability on Windows systems.

How it Works

From a user perspective, the SNMP Informant Custom Provider consists of two components, the Agent Definitions File and the Custom MIB. SNMP Informant comes with samples of each, allow you to rapidly begin using the tool. It is important to realize that the two files are intimately connected.

- The **Agent Definitions File**, located in the "custom" directory where you install the SNMP Informant software (i.e. C:\Program Files\SNMP Informant\custom by default) contains information about what the custom agent will do, and how it will respond (i.e. What OIDs to listen for and what actions to take).
- The **Corresponding Custom MIB** file provides integration into an NMS and should be created in such a way that it matches the Agent Definitions file, so that query types entered into the Agent Definitions File are linked to sections in the MIB.

When the custom provider starts up, it looks for the Agent Definitions file, and if not present or not correctly formatted, will not load successfully.

Agent Definitions File breakdown:

The agent definitions file is divided into two parts:

- The root MIB definition section
- The query specifiers section

The **root MIB definition section** has one entry. It has to be the first non-comment entry (comment entries start with a #) within the file, and it identifies the root OID prefix that applies to ALL queries within the custom agent.

- This flexibility allows SNMP Informant to respond to queries on a company's Private Enterprise OID if they have one registered with IANA.
 - Default Setting:
 - ROOT = .1.3.6.1.4.1.9600.1.99

The **query specifiers section** allows you to add numerous custom query rules. The generic format of a query rules is defined below, with each field separated by a whitespace (space or tab).

<OID_SUFFIX> <QUERY_TYPE> <ASN_DATATYPE> <STRING_ARGS>...

- **The OID_SUFFIX** is appended to the root MIB defined at the beginning of the file to define the entire MIB. For example, if the root MIB is defined as .1.3.6.1.4.1.9600.1.99 (default) and the OID_SUFFIX is set to 1.1.0, then the entire OID path for that particular query will be .1.3.6.1.4.1.9600.1.99.1.1.0.
 - **Note:** RegistryQuery, ExecuteQuery, and PerformanceQuery are all non-SNMP tables and thus should end with a .0 specifying that it is a SNMP node.

- **The QUERY_TYPE** defines what kind of query is being made, and is either a PerformanceQuery, an ExecuteQuery or a RegistryQuery.
- **PerformanceQuery** – A performance query is one that performs a Windows performance query and calculation (based on PDH, the Performance Data Helper). A performance query is composed of two to three string identifiers (depending if the performance object has instances associated with it or not). The first string argument is the performance object followed by the performance counter. Performance objects that are singletons (e.g., Memory, Objects, TCP, etc) do not have an instance associated with them. Performance objects that have multiple items (e.g., Process, LogicalDisk, Network Interface) must have the instance column populated. The best way to determine these entries is to use the Windows Performance tool in Administrative Tools to determine the necessary string argument values. The format of performance queries is shown below depending on if the an instance is required or not:

```
<OID_SUFFIX> PerformanceQuery <ASN_DATATYPE> '<PERF_OBJECT>' '<PERF_COUNTER>'
```

```
<OID_SUFFIX> PerformanceQuery <ASN_DATATYPE> '<PERF_OBJECT>' '<PERF_COUNTER>' '<PERF_INSTANCE>'
```

- **RegistryQuery** – A Registry query is one that retrieves a registry value from the HKEY_LOCAL_MACHINE registry hive. A RegistryQuery is composed of two arguments, the registry key and valueName. The RegistryQuery is in the following format:

```
<OID_SUFFIX> RegistryQuery <ASN_DATATYPE> '<REG_KEY >' '<REG_VALUENAME>'
```

The ASN Datatypes need to be:

- OCTETSTRING for registry datatypes of REG_SZ and REG_EXPAND_SZ
- COUNTER64 for registry datatypes of REG_QWORD
- All other ASN datatypes for registry datatypes of REG_DWORD.

- **ExecuteQuery** – A ExecuteQuery provides the capability to run a binary executable, VBScript, or JavaScript, capture the standard output and standard error streams, and return those values back to the SNMP Manager Station. The agent will terminate the process and return whatever captured data has been received if the executed process does not exit within 3 seconds. The ExecuteQuery is in the following format:

```
<OID_SUFFIX> ExecuteQuery <ASN_DATATYPE> '<SPAWN_COMMAND>'
```

In the case of SPAWN_COMMAND, the script or executable program to run must be present in the "spawn" directory located below where SNMP Informant custom is installed (i.e. C:\Program Files\SNMP Informant\custom\spawn).

- **The ASN_DATATYPE** defines how the data should be returned. The following data type values are supported:
 - **COUNTER32** - integer which increases until a maximum value and goes back (wraps) to zero, range is 0 to $2^{32}-1$
 - **COUNTER64** – identical to counter 32, range is 0 to 2^{64}
 - **GAUGE32** - integer which increases and decreases, range is 0 to $2^{32}-1$
 - **INTEGER32** a whole number, range is -2^{31} to $2^{31}-1$
 - **OCTETSTRING** - a string of octets which is used to represent hexadecimal data (i.e. physical address of an interface).

- **UNSIGNED32** - unsigned integer, range is 0 to $2^{32} - 1$

A sample Agent Definitions File is shown on the next few pages.

An important note about SNMP Informant Helper Services

The OS Provider, Exchange WMI provider and Custom Provider must be restarted each time you restart the SNMP service. In addition if the Agent Definitions file (for the Custom Provider) is ever modified, then the SNMP Informant Custom Helper service **MUST** be restarted, since the .INI file is only read on startup.

Remainder of page intentionally left blank.

Agent Definitions File

```
#####  
# Copyright (c) 2010 Informant Systems, Inc.  
# All rights reserved.  
#####  
#  
# Sample INI file for custom agent  
#  
# Comments are specified with a '#' as the first non-whitespace character  
#  
#####  
  
#####  
#  
# ROOT MIB  
#  
# The first entry in the INI file needs to be the root MIB OID prefix. The  
# custom agent will fail to initialize if this value is in an invalid format  
# or does not exist. The value is specified in a numeric OID format with or  
# without a preceeding dot.  
#  
#####  
  
ROOT = .1.3.6.1.4.1.9600.1.99  
  
#####  
#
```

```
# QUERY RULES
#
# After the root mib is specified, the user will specify the different custom
# query rules. All query rules contain the following common format:
#
# <OID Suffix> <Query Type> <ASN Datatype> '<STRING ARG>' ...
#
# - OID Suffix is appended to the root MIB OID defined above to generate the
#   entire OID structure. For example, given a ROOT of .1.3.6.1.4.1.9600.1.99
#   and an OID Suffix of .1.1.0, the entire OID for that query will be
#   .1.3.6.1.4.1.9600.1.99.1.1.0. All entries should end with a .0 specifying
#   that it is an SNMP node entry (SNMP tables are currently not supported).
#
# - The ASN Datatype defines the format of the data being retrieved. The
#   following data type values are supported:
# * Counter32
# * Counter64
# * Gauge32
# * Integer32
# * OctetString
# * Unsigned32
#   For PerformanceQueries, an OctetString will retrieve the floating-point
#   performance value and embed it within a string and return that back to
#   the SNMP monitor station. This is useful when a value is normally a
#   fraction (less than one) and will be returned as zero for all other ASN
#   datatypes.
#
# The following query types are supported:
#
# - REGISTRYQUERY
```

```

# A RegistryQuery is one that retrieves a registry value from the
# HKEY_LOCAL_MACHINE registry hive. A RegistryQuery is composed of two
# arguments, the registry key and ValueName. The RegistryQuery is in the
# following format:
#
# <OID Suffix> RegistryQuery <ASN Datatype> '<REG_KEY>' '<REG_VALUENAME>'
#
# Each field needs to be seperated by a whitespace (space or tab)
# along with the registry key and valuenam entries being
# single-quoted. The ASN datatype needs to be OctetString for registry
# datatypes of REG_SZ and REG_EXPAND_SZ, Counter64 for registry datatypes
# of REG_QWORD, and all other ASN datatypes for registry datatypes of
# REG_DWORD.
#
#
# - EXECUTEQUERY
# A ExecuteQuery provides the capability to execute a binary
# executable, VBScript, or JavaScript, capture the standard out and error
# streams, and return those values back to the SNMP Manager Station. The
# agent will terminate the process and return whatever captured data has
# been received if the spawned process does not exit within 3 seconds. The
# ExecuteQuery is in the following format:
#
# <OID Suffix> EXECUTEQUERY <ASN Datatype> '<SPAWN COMMAND>'
#
# Each field needs to be seperated bya whitespace (space or tab) along
# with the spawn command in single quotes. The maximum possible returned
# length is 8192 characters for OctetString ASN datatypes. All other data
# types will attempt to convert the standard-out or standard-error to that
# ASN datatype. If the conversion fails a value of zero will be returned.

```

```
#
# For security reasons, the initial program can ONLY be run out of the
# "spawn" sub-directory where the agent is installed on the computer.
# You can configure the location of the spawn directory using SNMP
# Informant registry settings (See the "Configuring SNMP Informant"
# section for more information on how to do this).
#
#
# - PERFORMANCEQUERY
# A performance query is one that performs a Windows performance query
# and calculation (based on PDH). A performance query is composed of
# two to three string identifiers (depending if the performance object
# has instances associated with it or not). The first string argument is the
# performance object followed by the performance counter. Performance
# objects that are singletons (e.g., Memory, Objects, TCP, etc) do not
# have an instance associated with them but performance objects that
# have multiple items (e.g., Process, LogicalDisk, Network Interface)
# must have the instance column populated. The best way to determine
# these entries is to use the Windows Performance tool in Administrative
# Tools to determine the necessary entries. The format of performance
# queries is shown below depending on if an instance is required or not.
#
# <OID Suffix> PerformanceQuery <ASN Datatype> '<PERF_OBJECT>' '<PERF_COUNTER>'
# <OID Suffix> PerformanceQuery <ASN Datatype> '<PERF_OBJECT>' '<PERF_COUNTER>' '<PERF_INSTANCE>'
#
# Each field needs to be seperated by a whitespace (space or tab)
# along with each performance string entry being single-quoted.
#
#
#####
```

#####

# Suffix	RegistryQuery	ASN Type	Registry Key	Registry ValueName
.1.1.0	RegistryQuery	OctetString	'SOFTWARE\Microsoft\Windows NT\CurrentVersion'	'ProductName'
.1.2.0	RegistryQuery	OctetString	'SOFTWARE\Microsoft\Windows NT\CurrentVersion'	'ProductId'
.1.3.0	RegistryQuery	OctetString	'SOFTWARE\Microsoft\Windows NT\CurrentVersion'	'CurrentBuild'
.1.4.0	RegistryQuery	OctetString	'SOFTWARE\Microsoft\Windows NT\CurrentVersion'	'CurrentBuildNumber'
.1.5.0	RegistryQuery	OctetString	'SOFTWARE\Microsoft\Windows NT\CurrentVersion'	'CurrentVersion'
.1.6.0	RegistryQuery	OctetString	'SOFTWARE\Microsoft\Windows NT\CurrentVersion'	'BuildLab'
.1.7.0	RegistryQuery	Unsigned32	'SOFTWARE\Microsoft\Windows NT\CurrentVersion'	'InstallDate'

#####

# Suffix	ExecuteQuery	ASN Type	Command Line
.2.1.0	ExecuteQuery	OctetString	'hello_string.js "javascript arg1 example"'
.2.2.0	ExecuteQuery	OctetString	'hello_string.vbs "vbscript arg1 example"'
.2.3.0	ExecuteQuery	OctetString	'hello_string.exe "executable arg1 example"'
.2.4.0	ExecuteQuery	Counter32	'date_integer.js'
.2.5.0	ExecuteQuery	Counter64	'date_integer.vbs'
.2.6.0	ExecuteQuery	Integer32	'date_integer.exe'
.2.7.0	ExecuteQuery	Unsigned32	'date_integer.exe'
.2.8.0	ExecuteQuery	Gauge32	'date_integer.exe'

#####

# Suffix	PerformanceQuery	ASN Type	OBJECT	COUNTER	INSTANCE
.3.1.0	PerformanceQuery	Gauge32	'Memory'	'% Committed Bytes In Use'	
.3.2.0	PerformanceQuery	Gauge32	'Memory'	'Available Bytes'	
.3.3.0	PerformanceQuery	Gauge32	'Memory'	'Available KBytes'	

.3.4.0	PerformanceQuery	Gauge32	'Memory'	'Available MBytes'	
.3.5.0	PerformanceQuery	Gauge32	'Memory'	'Cache Bytes'	
.3.6.0	PerformanceQuery	Gauge32	'Memory'	'Cache Bytes Peak'	
.3.7.0	PerformanceQuery	Gauge32	'Memory'	'Cache Faults/sec'	
.3.8.0	PerformanceQuery	Gauge32	'Memory'	'Commit Limit'	
.3.9.0	PerformanceQuery	Gauge32	'Memory'	'Committed Bytes'	
.3.10.0	PerformanceQuery	Gauge32	'Memory'	'Demand Zero Faults/sec'	
.3.11.0	PerformanceQuery	Gauge32	'Memory'	'Free System Page Table Entries'	
.3.12.0	PerformanceQuery	Gauge32	'Memory'	'Page Faults/sec'	
.3.13.0	PerformanceQuery	Gauge32	'Memory'	'Page Reads/sec'	
.3.14.0	PerformanceQuery	Gauge32	'Memory'	'Page Writes/sec'	
.3.15.0	PerformanceQuery	Gauge32	'Memory'	'Pages Input/sec'	
.3.16.0	PerformanceQuery	Gauge32	'Memory'	'Pages Output/sec'	
.3.17.0	PerformanceQuery	Gauge32	'Memory'	'Pages/sec'	
.3.18.0	PerformanceQuery	Gauge32	'Memory'	'Pool Nonpaged Allocs'	
.3.19.0	PerformanceQuery	Gauge32	'Memory'	'Pool Nonpaged Bytes'	
.3.20.0	PerformanceQuery	Gauge32	'Memory'	'Pool Paged Allocs'	
.3.21.0	PerformanceQuery	Gauge32	'Memory'	'Pool Paged Bytes'	
.3.22.0	PerformanceQuery	Gauge32	'Memory'	'Pool Paged Resident Bytes'	
.3.23.0	PerformanceQuery	Gauge32	'Memory'	'System Cache Resident Bytes'	
.3.24.0	PerformanceQuery	Gauge32	'Memory'	'System Code Resident Bytes'	
.3.25.0	PerformanceQuery	Gauge32	'Memory'	'System Code Total Bytes'	
.3.26.0	PerformanceQuery	Gauge32	'Memory'	'System Driver Resident Bytes'	
.3.27.0	PerformanceQuery	Gauge32	'Memory'	'System Driver Total Bytes'	
.3.28.0	PerformanceQuery	Gauge32	'Memory'	'Transition Faults/sec'	
.3.29.0	PerformanceQuery	Gauge32	'Memory'	'Transition Pages RePurposed/sec'	
.3.30.0	PerformanceQuery	Gauge32	'Memory'	'Write Copies/sec'	
.4.1.0	PerformanceQuery	Gauge32	'Processor'	'% C1 Time'	'_Total'
.4.2.0	PerformanceQuery	Gauge32	'Processor'	'% C2 Time'	'_Total'

.4.3.0	PerformanceQuery	Gauge32	'Processor'	'% C3 Time'	'_Total'
.4.4.0	PerformanceQuery	Gauge32	'Processor'	'% DPC Time'	'_Total'
.4.5.0	PerformanceQuery	Gauge32	'Processor'	'% Idle Time'	'_Total'
.4.6.0	PerformanceQuery	Gauge32	'Processor'	'% Interrupt Time'	'_Total'
.4.7.0	PerformanceQuery	Gauge32	'Processor'	'% Privileged Time'	'_Total'
.4.8.0	PerformanceQuery	Gauge32	'Processor'	'% Processor Time'	'_Total'
.4.9.0	PerformanceQuery	Gauge32	'Processor'	'% User Time'	'_Total'
.4.10.0	PerformanceQuery	Gauge32	'Processor'	'C1 Transitions/sec'	'_Total'
.4.11.0	PerformanceQuery	Gauge32	'Processor'	'C2 Transitions/sec'	'_Total'
.4.12.0	PerformanceQuery	Gauge32	'Processor'	'C3 Transitions/sec'	'_Total'
.4.13.0	PerformanceQuery	Gauge32	'Processor'	'DPC Rate'	'_Total'
.4.14.0	PerformanceQuery	Gauge32	'Processor'	'DPCs Queued/sec'	'_Total'
.4.15.0	PerformanceQuery	Gauge32	'Processor'	'Interrupts/sec'	'_Total'
.5.1.0	PerformanceQuery	Gauge32	'LogicalDisk'	'% Disk Read Time'	'C:'
.5.2.0	PerformanceQuery	Gauge32	'LogicalDisk'	'% Disk Time'	'C:'
.5.3.0	PerformanceQuery	Gauge32	'LogicalDisk'	'% Disk Write Time'	'C:'
.5.4.0	PerformanceQuery	Gauge32	'LogicalDisk'	'% Free Space'	'C:'
.5.5.0	PerformanceQuery	Gauge32	'LogicalDisk'	'% Idle Time'	'C:'
.5.6.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Avg. Disk Bytes/Read'	'C:'
.5.7.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Avg. Disk Bytes/Transfer'	'C:'
.5.8.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Avg. Disk Bytes/Write'	'C:'
.5.9.0	PerformanceQuery	OctetString	'LogicalDisk'	'Avg. Disk Queue Length'	'C:'
.5.10.0	PerformanceQuery	OctetString	'LogicalDisk'	'Avg. Disk Read Queue Length'	'C:'
.5.11.0	PerformanceQuery	OctetString	'LogicalDisk'	'Avg. Disk Write Queue Length'	'C:'
.5.12.0	PerformanceQuery	OctetString	'LogicalDisk'	'Avg. Disk sec/Read'	'C:'
.5.13.0	PerformanceQuery	OctetString	'LogicalDisk'	'Avg. Disk sec/Transfer'	'C:'
.5.14.0	PerformanceQuery	OctetString	'LogicalDisk'	'Avg. Disk sec/Write'	'C:'
.5.15.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Current Disk Queue Length'	'C:'
.5.16.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Disk Bytes/sec'	'C:'

.5.17.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Disk Read Bytes/sec'	'C:'
.5.18.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Disk Reads/sec'	'C:'
.5.19.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Disk Transfers/sec'	'C:'
.5.20.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Disk Write Bytes/sec'	'C:'
.5.21.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Disk Writes/sec'	'C:'
.5.22.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Free Megabytes'	'C:'
.5.23.0	PerformanceQuery	Gauge32	'LogicalDisk'	'Split IO/Sec'	'C:'

Common SNMP Informant OIDs

The following table lists some common SNMP Informant OIDs which can be used to monitor different types of Windows Servers.

Category	Performance Counter	SNMP Informant Agent Required	SNMP Informant OID (with comments where applicable)	
(see SMTP Notes below)	SMTP Server\Categorizer Queue Length	Advanced Agent	.1.3.6.1.4.1.9600.1.2.76.1.70.6.95.84.111.116.97.108 (_Total)	
	SMTP Server\Local Queue Length	Advanced Agent	.1.3.6.1.4.1.9600.1.2.76.1.81.6.95.84.111.116.97.108 (_Total)	
	SMTP Server\Remote Queue Length	Advanced Agent	.1.3.6.1.4.1.9600.1.2.76.1.110.6.95.84.111.116.97.108 (_Total)	
	SMTP Server\Messages Delivered/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.76.1.93.6.95.84.111.116.97.108 (_Total)	
	SMTP Server\Messages Received/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.76.1.96.6.95.84.111.116.97.108 (_Total)	
	SMTP Server\Messages Sent/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.76.1.101.6.95.84.111.116.97.108 (_Total)	
	MSExchange\IS Mailbox\Receive Queue Length	Exchange Server Application Plus Pack		
	MSExchange\IS Mailbox\Send Queue Length	Exchange Server Application Plus Pack		
	MSExchange\IS Mailbox\Folder Opens/sec	Exchange Server Application Plus Pack	.1.3.6.1.4.1.9600.1.5.15.1.6.6.95.84.111.116.97.108 (_Total)	
	MSExchange\IS Mailbox\Message Opens/sec	Exchange Server Application Plus Pack	.1.3.6.1.4.1.9600.1.5.15.1.18.6.95.84.111.116.97.108 (_Total)	
	MSExchange\IS Mailbox\Local Delivery Rate	Exchange Server Application Plus Pack	.1.3.6.1.4.1.9600.1.5.15.1.16.6.95.84.111.116.97.108 (_Total)	
	MSExchange\IS Public\Receive Queue Size	Exchange Server Application Plus Pack	.1.3.6.1.4.1.9600.1.5.16.1.27.6.95.84.111.116.97.108 (_Total)	
	MSExchange\IS Public\Send Queue Size	Exchange Server Application Plus Pack	.1.3.6.1.4.1.9600.1.5.16.1.45.6.95.84.111.116.97.108 (_Total)	
(see Exchange Notes below)	MSExchange\IS\RPC Operations/sec	Exchange Server Application Plus Pack	.1.3.6.1.4.1.9600.1.5.14.68.0	
	MSExchange\IS\RPC Requests	Exchange Server Application Plus Pack	.1.3.6.1.4.1.9600.1.5.14.70.0	
(see SQL Notes below)	SQLServer:Buffer Manager\Buffer cache hit ratio	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.22.1.6.x (where x increments per SQL instance)	
	SQLServer:Buffer Manager\Page reads/sec	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.22.1.14.x (where x increments per SQL instance)	
	SQLServer:Buffer Manager\Page writes/sec	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.22.1.15.x (where x increments per SQL instance)	
	SQLServer:Cache Manager\Cache Hit Ratio	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.24.1.2.1.6.95.84.111.116.97.108 (_Total)	
	SQLServer:Databases\Active Transactions	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.27.1.2.1.6.95.84.111.116.97.108 (_Total)	
	SQLServer:Databases\Transactions/sec	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.27.1.32.1.6.95.84.111.116.97.108 (_Total)	
	SQLServer:General Statistics\User Connections	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.29.1.17.x (where x increments per SQL instance)	
	SQLServer:General Statistics\Logins/sec	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.29.1.4.x (where x increments per SQL instance)	
	SQLServer:General Statistics\Logouts/sec	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.29.1.5.x (where x increments per SQL instance)	
	SQLServer:Memory Manager\Total Server Memory (KB)	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.32.1.13.x (where x increments per SQL instance)	
	SQLServer:Memory Manager\SQL Cache Memory (KB)	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.32.1.12.x (where x increments per SQL instance)	
	SQLServer:Locks\Lock Requests/sec	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.31.1.1.1.6.95.84.111.116.97.108 (_Total)	
	SQLServer:Locks\Average Wait Time(ms)	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.31.1.2.1.6.95.84.111.116.97.108 (_Total)	
	SQLServer:SQL Statistics\Batch Requests/sec	SQL Server Application Plus Pack	.1.3.6.1.4.1.9600.1.3.39.1.2.x (where x increments per SQL instance)	

Category	Performance Counter	SNMP Informant Agent Required	SNMP Informant OID (with comments where applicable)
Active Directory	NTDS\DS Threads in Use	Advanced Agent	.1.3.6.1.4.1.9600.1.2.55.88.0
	NTDS\LDAP Client Sessions	Advanced Agent	.1.3.6.1.4.1.9600.1.2.55.94.0
	NTDS\LDAP Searches/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.55.95.0
	NTDS\DRA Inbound Bytes Total/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.55.22.0
	NTDS\DRA Outbound Bytes Total/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.55.40.0
IIS Web (see IIS Notes below)	Web Service\Connection Attempts/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.86.1. 7 .6.95.84.111.116.97.108 (_Total)
	Web Service\Current Connections	Advanced Agent	.1.3.6.1.4.1.9600.1.2.86.1. 14 .6.95.84.111.116.97.108 (_Total)
	Web Service\Logon Attempts/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.86.1. 27 .6.95.84.111.116.97.108 (_Total)
	Web Service\Bytes Received/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.86.1. 3 .6.95.84.111.116.97.108 (_Total)
	Web Service\Bytes Sent/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.86.1. 4 .6.95.84.111.116.97.108 (_Total)
IIS FTP (see IIS Notes below)	FTP Service\Bytes Sent/sec	Advanced Agent	.1.3.6.1.4.1.9600.1.2.23.1. 3 .6.95.84.111.116.97.108 (_Total)
	FTP Service\Current Connections	Advanced Agent	.1.3.6.1.4.1.9600.1.2.23.1. 6 .6.95.84.111.116.97.108 (_Total)
	FTP Service\Total Logon Attempts	Advanced Agent	.1.3.6.1.4.1.9600.1.2.23.1. 17 .6.95.84.111.116.97.108 (_Total)

SMTP Service Notes: SMTP service counters support all named SMTP server instances as created. The first number after the bolded number indicates the number of characters in the named instance, and the remaining numbers are ASCII representations of the characters in the name. For example, .6.95.84.111.116.97.108 means that **6 numbers follow**, and that they (in ASCII) spell out **_Total**. To find out the instance name and the numbers that follow the bolded number, walk OID .1.3.6.1.4.1.9600.1.2.76.1.1.

Exchange Notes: Exchange counters support all named Storage Group names as created. The first number after the bolded number indicates the number of characters in the Storage Group Name, and the remaining numbers are ASCII representations of the characters in the name. For example, .6.95.84.111.116.97.108 means that **6 numbers follow**, and that they (in ASCII) spell out **_Total**. To find out the Storage Group name, and the numbers that follow the bolded number, walk OID .1.3.6.1.4.1.9600.1.5.15.1.1.

SQL Notes: Walk OID .1.3.6.1.4.1.9600.1.3.16.1.2 to determine what instance number (**x**) matches what database instance number.

IIS Notes: IIS Web and FTP counters support all named Web and FTP instances as created. The first number after the bolded number indicates the number of characters in the named instance, and the remaining numbers are ASCII representations of the characters in the name. For example, .6.95.84.111.116.97.108 means that **6 numbers follow**, and that they (in ASCII) spell out **_Total**.

Troubleshooting SNMP Informant

SNMP Informant logs events to the Application Event log. Depending on your actions, and the results of queries performed by SNMP Informant, these messages will differ. *If SNMP Informant does not seem to be working, checking the Application Event Log should be one of your first courses of action.*

- **Check the SNMP Informant Knowledge base** at: <http://www.snmp-informant.com/Knowledgebase.htm>

The table below lists some troubleshooting steps to take if you find SNMP Informant is not working the way it is supposed to:

Troubleshooting Table

Problem	Check	Solution
I can't query any data from SNMP Informant.	Is the Windows SNMP Service installed?	Install the SNMP Service according to this guide.
	Is the Windows SNMP Service running?	Start the SNMP Service using the Windows Service Manager.
	Can you request any SNMP data from the SNMP service?	Check that your community names match your SNMP Manager. Check that the security settings are correct for your environment.
	Check that the Windows Application Event Log for any SNMP Informant errors or warnings.	Check the various SNMP Informant web pages for related information <ul style="list-style-type: none"> • http://www.wtcs.org/informant/support.htm Check the Microsoft Windows support website for related information <ul style="list-style-type: none"> • http://support.microsoft.com
	Check to see if the Windows Performance Monitor works on that computer.	Check the Microsoft Windows support website for related information <ul style="list-style-type: none"> • http://support.microsoft.com
I can't query a specific SNMP Informant OID.	Check to see that you are referencing the SNMP OID correctly by using SNMP GETNEXT/WALK operations.	Use the returned SNMP OID from the GETNEXT/WALK operation.
	That performance counter may not be available on the computer/software you are using.	Check the various SNMP Informant web pages for related information <ul style="list-style-type: none"> • http://www.wtcs.org/informant/support.htm Check the Microsoft Windows support website for related information <ul style="list-style-type: none"> • http://support.microsoft.com
	Check that the Windows Application Event Log for any SNMP Informant errors or warnings.	Check the various SNMP Informant web pages for related information <ul style="list-style-type: none"> • http://www.wtcs.org/informant/support.htm Check the Microsoft Windows support website for related information <ul style="list-style-type: none"> • http://support.microsoft.com

Troubleshooting PDH Providers

If you are trying to do an SNMP GET of a particular OID, and cannot seem to get data, remember that what performance counters you *can* access all depends on the OS version where SNMP Informant is installed.

For example, Windows 2003 has performance counters that do not exist on Windows 2000, so SNMP GET requests to OIDs that map to Windows 2003 performance counters will fail on Windows 2000 systems.

The general "can I use SNMP Informant to collect data from the <insert name here> performance counter?" test is this:

- **Check to see if you can see the local Performance Monitor equivalent of the OID**

Check the Performance Monitor applet (Start/Run/Perfmon) on the system you want to collect data from. If you can see the performance object and counter and instances you want (or are trying) to track, then you should be able to install SNMP Informant on that server, and (using the proper OID, of course) use SNMP to GET that data. **If you are unable to see the performance object, counter and instances, then you will NOT be able to get that data using SNMP Informant.**

Sometimes, for reasons we are unable to explain, certain performance counters get deactivated, and subsequently, SNMP Informant is unable to bridge to them. We suggest that you download the Extensible Performance Counter tool and make sure that they ARE in fact activated.

The Extensible Performance Counter List (exctrlst.exe) GUI utility will list all of the services and applications that provide registry-based performance counters on local and remote Windows systems.

Exctrlst displays the names and locations of the dynamic link libraries (DLLs) that support performance counters, indicates disabled counters, and lets you enable and disable the performance counters for a service.

Exctrlst displays the following information about performance counters:

- Services and applications that provide performance counters
- Names and locations of performance counter DLLs
- Enabled and disabled performance counters
- Indexes of performance objects, counter names, and help text provided by the service
- Names of the open, collect, and close procedure of each counter DLL
- The highest index on the system used for performance counters and their help text.

You can download the tool from this link here:

- http://download.microsoft.com/download/win2000platform/exctrlst/1.00.0.1/nt5/en-us/exctrlst_setup.exe

Troubleshooting WMI Providers

As in the Performance Counters, what WMI classes you can access also depends on the OS/product version where SNMP Informant is installed. For example, Windows 2003 may have WMI classes and objects that do not exist on Windows 2000, so SNMP GET requests to OIDs that map to Windows 2003 WMI classes will fail on Windows 2000 systems. The same applies to the WMI-HW and WMI-Exchange providers.

In order to install successfully, all SNMP Informant WMI Agents requires that the Windows WMI Service is properly installed and configured. A default installation of the Windows WMI service is usually sufficient for SNMP Informant WMI agents to install successfully.

The general "can I access the <insert name here> WMI class?" test is this:

- **Check to see if see if you can see the local WMI class equivalent of the OID**

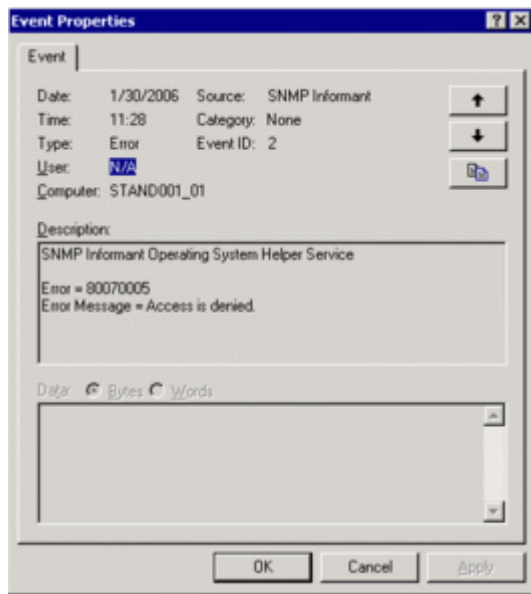
We recommend using the free Microsoft **WMI Object Browser** tool available at <http://www.microsoft.com/downloads/details.aspx?FamilyID=6430F853-1120-48DB-8CC5-F2ABDC3ED314>. This tool allows you to browse WMI classes on a local or remote computer (using IE) sort of like a MIB Browser. Note that you have to have proper Windows permissions to do so. A secured user account will likely not have the access rights to do so. Log in as a local administrator on the computer in question, install the software, and run it.

If you see the WMI classes and objects you want (or are trying) to track, then you should be able to install SNMP Informant on that server, and (using the proper OID, of course) use SNMP to GET that data. If you are unable to see the WMI classes and objects, then you will NOT be able to get that data using SNMP Informant.

A note about the "No endpoint mapper available" message in the Application Event Log:

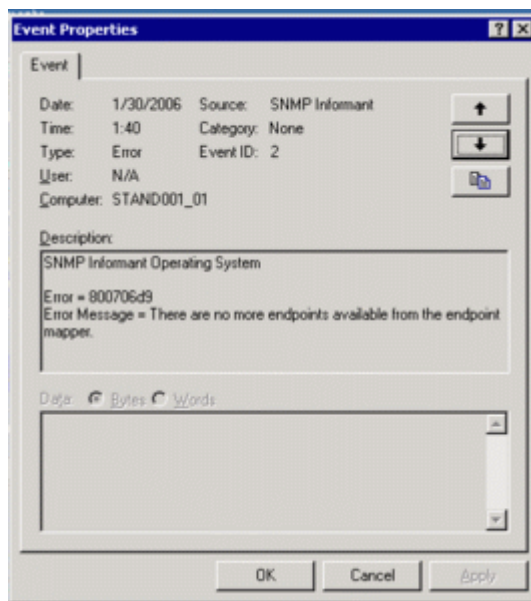
The SNMP Informant WMI-OS, and WMI-Exchange agents install a separate SNMP Informant "helper" service to connect to the WMI subsystem using RPC/DCOM. Sometimes this connection cannot be completed, and the error log might get filled with messages like the one below:

SNMP Informant Operating System Helper Service
Error = 80070005
Error Message = Access is denied



and this ...

SNMP Informant Operating System
Error = 800706d9
Error Message = There are no more endpoints available from the endpoint mapper.

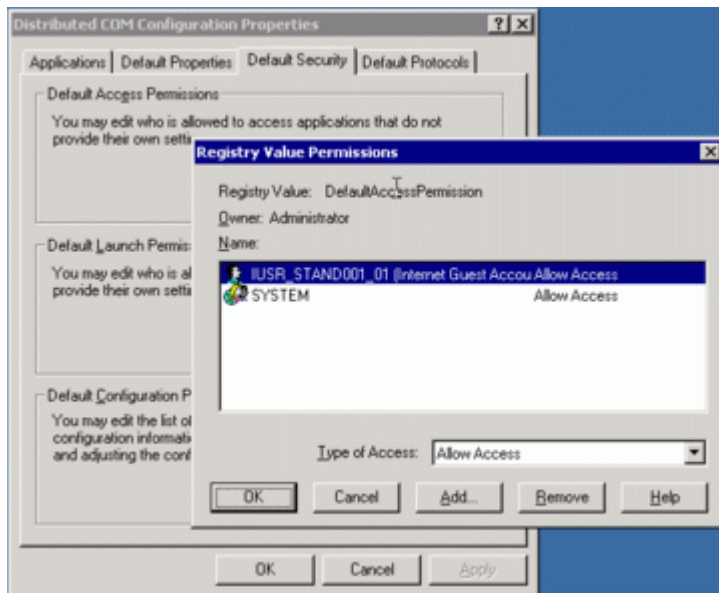


This usually accompanies other WMI-related errors, and seldom exists all by itself. It indicates that SNMP Informant cannot "attach" to WMI. If you are getting these error messages from SNMP Informant, check your event logs to see if you are getting 80070005 messages from *other* services as well, as DCOM security permissions affect more than just SNMP Informant (i.e. .Net runtime. Here are some great starting points:

- <http://support.microsoft.com/kb/839880> - Domain related DCOM errors

- <http://support.microsoft.com/kb/892500> - Other general DCOM errors

Meantime, To resolve this (for SNMP Informant at least), start the DCOM Configuration program (start/run/dcomcnfg), and add the SYSTEM account to the Default Access Permissions group.



You should also check the Default Launch Permissions group to ensure the SYSTEM account is also there.

More information about setting DCOM access permissions can be found here:

http://www.microsoft.com/windows2000/en/advanced/help/default.asp?url=/windows2000/en/advanced/help/set_specacc_dcom.htm

Troubleshooting Custom Providers

Custom providers are a little more complex to troubleshoot. You have to figure out why provider is not returning data.

If the issue is that queries to the Citrix OIDs are not returning data, be sure Citrix is installed on the server where SNMP Informant is installed.

If the issue is that queries to the Cluster Service OIDs are not returning data, be sure Cluster server is installed (and active) on the server where SNMP Informant is installed. If you are querying the virtual name, then you should get a response from whatever node is active.

End of Document