NetIQ® AppManager® for Microsoft Windows Management Instrumentation

Management Guide

January 2013

Contents

About this Book and the Library

The NetIQ AppManager product (AppManager) is a comprehensive solution for managing, diagnosing, and analyzing performance, availability, and health for a broad spectrum of operating environments, applications, services, and server hardware.

AppManager provides system administrators with a central, easy-to-use console to view critical server and application resources across the enterprise. With AppManager, administrative staff can monitor computer and application resources, check for potential problems, initiate responsive actions, automate routine tasks, and gather performance data for real-time and historical reporting and analysis.

Intended Audience

This guide provides information for individuals responsible for installing an AppManager module and monitoring specific applications with AppManager.

Other Information in the Library

The library provides the following information resources:

Installation Guide for AppManager

Provides complete information about AppManager pre-installation requirements and step-by-step installation procedures for all AppManager components.

User Guide for AppManager Control Center

Provides complete information about managing groups of computers, including running jobs, responding to events, creating reports, and working with Control Center. A separate guide is available for the AppManager Operator Console.

Administrator Guide for AppManager

Provides information about maintaining an AppManager management site, managing security, using scripts to handle AppManager tasks, and leveraging advanced configuration options.

Upgrade and Migration Guide for AppManager

Provides complete information about how to upgrade from a previous version of AppManager.

Management guides

Provide information about installing and monitoring specific applications with AppManager.

Help

Provides context-sensitive information and step-by-step guidance for common tasks, as well as definitions for each field on each window.

The AppManager library is available in Adobe Acrobat (PDF) format from the AppManager Documentation page of the NetIQ Web site.

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. If you have suggestions for improvements, click **Add Comment** at the bottom of any page in the HTML versions of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

Qmunity, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, Qmunity helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit http://community.netiq.com.

1 Introducing AppManager for Microsoft WMI Management Instrumentation

Microsoft Windows Management Instrumentation (WMI) is the infrastructure for managing data and operations on Windows-based operating systems. It provides an interface through which scripting languages such as VBScript or Windows PowerShell manage local and remote computers.

AppManager for WMI provides Knowledge Scripts that enable you to monitor WMI processes and services, and execute WMI queries. You can configure each Knowledge Script to raise an event, collect data for reporting, and perform automated problem management when an event occurs.

With AppManager for WMI, you can perform the following tasks:

- Configure the frequency of the WMI repository backup and the type, size, and location of the logging files
- Monitor for events generated by the WMI event provider and search for events in the database
- Monitor the size of WMI log files
- Monitor the size of the WMI repository
- Monitor the CPU and memory consumption for WMI processes
- Run WQL statements
- Monitor the availability of the WMI CIMOM service
- Add, delete, or edit a WMI user account or WMI group account for the WMI service

2 Installing AppManager for WMI

This chapter provides installation instructions and describes system requirements for AppManager for WMI.

This chapter assumes you have AppManager installed. For more information about installing AppManager or about AppManager system requirements, see the *Installation Guide for AppManager*, which is available on the AppManager Documentation page.

2.1 System Requirements

For the latest information about supported software versions and the availability of module updates, visit the AppManager Supported Products page. Unless noted otherwise, this module supports all updates, hotfixes, and service packs for the releases listed below.

AppManager for WMI has the following system requirements:

Software/Hardware	Version
NetIQ AppManager installed on the AppManager repository (QDB) computers, on the servers you want to monitor (AppManager agents), and on all console computers	7.0 or later Support for Windows Server 2008 on AppManager 7.x requires AppManager Windows Agent hotfix 71704 or later. For more information, see the AppManager Suite Hotfixes page.
Microsoft Windows operating system on agent computers	One of the following: ◆ Windows Server 2012 ◆ Windows 8 (32-bit or 64-bit) ◆ Windows Server 2008 R2 ◆ Windows Server 2008 (32-bit or 64-bit) ◆ Windows 7 (32-bit or 64-bit) ◆ Windows Server 2003 (32-bit or 64-bit) ◆ Windows XP (32-bit or 64-bit)
Windows Management Instrumentation on agent computers	6.1, 6.0, 5.2, or 5.1
AppManager for Microsoft Windows module installed on repository, agent, and console computers	The most recent version, for support of Windows Server 2008. For more information, see the AppManager Module Upgrades & Trials Web page.

If you encounter problems using this module with a later version of your application, contact NetIQ Technical Support.

2.2 Installing the Module

Run the module installer only once on any computer. The module installer automatically identifies and updates all relevant AppManager components on a computer.

Access the `AM70-WMI-7.`*`x.x`*`.0.msi` module installer from the `AM70_WMI_7.`*`x.x`*`.0` self-extracting installation package on the AppManager Module Upgrades & Trials page.

For Windows environments where User Account Control (UAC) is enabled, install the module using an account with administrative privileges. Use one of the following methods:

- Log in to the server using the account named Administrator. Then, run `AM70-WMI-7.`*`x.x`*`.0.msi` from a command prompt or by double-clicking it.
- Log in to the server as a user with administrative privileges and run `AM70-WMI.`*`x.x`*`.0.msi` as an administrator from a command prompt. To open a command-prompt window at the administrative level, right-click a command-prompt icon or a Windows menu item and select **Run as administrator**.

You can install the Knowledge Scripts and the Analysis Center reports into local or remote AppManager repositories (QDBs). Install these components only once per QDB.

The module installer now installs Knowledge Scripts for each module directly into the QDB instead of installing the scripts in the `\AppManager\qdb\kp` folder as in previous releases of AppManager.

You can install the module manually, or you can use Control Center to deploy the module on a remote computer where an agent is installed. For more information, see Section 2.3, "Deploying the Module with Control Center," on page 13. However, if you use Control Center to deploy the module, Control Center only installs the *agent* components of the module. The module installer installs the QDB and console components as well as the agent components on the agent computer.

To install the module manually:

1 Double-click the module installer `.msi` file.

2 Accept the license agreement.

3 Review the results of the pre-installation check. You can expect one of the following three scenarios:

- **No AppManager agent is present:** In this scenario, the pre-installation check fails, and the installer does not install agent components.
- **An AppManager agent is present, but some other prerequisite fails:** In this scenario, the default is to not install agent components because of one or more missing prerequisites. However, you can override the default by selecting Install agent component locally. A missing application server for this particular module often causes this scenario. For example, installing the AppManager for Microsoft SharePoint module requires the presence of a Microsoft SharePoint server on the selected computer.
- **All prerequisites are met:** In this scenario, the installer installs the agent components.

4 To install the Knowledge Scripts into the QDB:

4a Select **Install Knowledge Scripts** to install the repository components, including the Knowledge Scripts, object types, and SQL stored procedures.

4b Specify the SQL Server name of the server hosting the QDB, as well as the case-sensitive QDB name.

5 (Conditional) If you use Control Center 7.x, run the module installer for each QDB attached to Control Center.

6 (Conditional) If you use Control Center 8.x, run the module installer only for the primary QDB. Control Center automatically replicates this module to secondary QDBs.

7 Run the module installer on all console computers to install the Help and console extensions.

8 Run the module installer on the WMI computers you want to monitor (agents) to install the agent components.

9 If you have not already discovered WMI resources, run the Discovering WMI Resources Knowledge Script on all agent computers where you installed the module.

10 To get the updates provided in this release, upgrade any running Knowledge Script jobs. For more information, see Section 2.6, "Upgrading Knowledge Script Jobs," on page 15.

After the installation has completed, you can find a record of problems encountered in the `WMI_Install.log` file, located in the `\NetIQ\Temp\NetIQ_Debug\<ServerName>` folder.

2.3 Deploying the Module with Control Center

You can use Control Center to deploy the module on a remote computer where an agent is installed. This topic briefly describes the steps involved in deploying a module and provides instructions for checking in the module installation package. For more information, see the *Control Center User Guide for AppManager*, which is available on the AppManager Documentation page.

2.3.1 Deployment Overview

This section describes the tasks required to deploy the module on an agent computer.

To deploy the module on an agent computer:

1 Verify the default deployment credentials.

2 Check in an installation package. For more information, see Section 2.3.2, "Checking In the Installation Package," on page 13.

3 Configure an email address to receive notification of a deployment.

4 Create a deployment rule or modify an out-of-the-box deployment rule.

5 Approve the deployment task.

6 View the results.

2.3.2 Checking In the Installation Package

You must check in the installation package, `AM70-WMI-7.x.x.0.xml`, before you can deploy the module on an agent computer.

To check in a module installation package:

1 Log on to Control Center using an account that is a member of a user group with deployment permissions.

2 Navigate to the **Deployment** tab (for AppManager 8.x) or **Administration** tab (for AppManager 7.x).

3 In the Deployment folder, select **Packages**.

4 On the Tasks pane, click **Check in Deployment Packages** (for AppManager 8.x) or **Check in Packages** (for AppManager 7.x).

5 Navigate to the folder where you saved `AM70-WMI-7.x.x.0.xml` and select the file.

6 Click **Open**. The Deployment Package Check in Status dialog box displays the status of the package check in.

7 To get the updates provided in this release, upgrade any running Knowledge Script jobs. For more information, see Section 2.6, "Upgrading Knowledge Script Jobs," on page 15.

2.4 Silently Installing the Module

To silently (without user intervention) install a module using the default settings, run the following command from the folder in which you saved the module installer:

```
msiexec.exe /i "AM70-WMI-7.x.x.0.msi" /qn
```

where *x.x* is the actual version number of the module installer.

To get the updates provided in this release, upgrade any running Knowledge Script jobs. For more information, see Section 2.6, "Upgrading Knowledge Script Jobs," on page 15.

To create a log file that describes the operations of the module installer, add the following flag to the command noted above:

```
/L* "AM70-WMI-7.x.x.0.msi.log"
```

The log file is created in the folder in which you saved the module installer.

NOTE: To perform a silent install on an AppManager agent running Windows Server 2008 R2 or Windows Server 2012, open a command prompt at the administrative level and select **Run as administrator** before you run the silent install command listed above.

To silently install the module on a remote AppManager repository, you can use Windows authentication or SQL authentication.

Windows authentication:

```
AM70-WMI-7.x.x.0.msi /qn MO_B_QDBINSTALL=1 MO_B_MOINSTALL=0 MO_B_SQLSVR_WINAUTH=1
MO_SQLSVR_NAME=SQLServerName MO_QDBNAME=AM-RepositoryName
```

SQL authentication:

```
AM70-WMI-7.x.x.0.msi /qn MO_B_QDBINSTALL=1 MO_B_MOINSTALL=0 MO_B_SQLSVR_WINAUTH=0
MO_SQLSVR_USER=SQLLogin MO_SQLSVR_PWD=SQLLoginPassword
MO_SQLSVR_NAME=SQLServerName MO_QDBNAME=AM-RepositoryName
```

2.5 Discovering WMI Resources

Use the Discovery_WMI Knowledge Script to discover the Microsoft Windows Management Instrumentation (WMI) server configuration and resources.

By default, this script is only run once for each computer.

Set the **Values** tab parameters as needed.

Description	How to Set It
Raise event if discovery succeeds?	This Knowledge Script always raises an event when the job fails for any reason. In addition, you can set this parameter to y to raise an event when the job succeeds. The default is n.

Description	How to Set It
Event severity when discovery succeeds	Set the event severity level, from 1 to 40, to reflect the severity when the discovery succeeds. The default is 25.
Event severity when discovery fails	Set the event severity level, from 1 to 40, to reflect the severity when the discovery fails. The default is 5.
Event severity when discovery is not applicable	Set the event severity level, from 1 to 40, to reflect the severity when the discovery is not applicable, usually because the target computer does not have WMI installed. The default is 15.

2.6 Upgrading Knowledge Script Jobs

If you are using AppManager 8.x or later, the module upgrade process now *retains* any changes you may have made to the parameter settings for the Knowledge Scripts in the previous version of this module. Before AppManager 8.x, the module upgrade process *overwrote* any settings you may have made, changing the settings back to the module defaults.

As a result, if this module includes any changes to the default values for any Knowledge Script parameter, the module upgrade process ignores those changes and retains all parameter values that you updated. Unless you review the management guide or the online Help for that Knowledge Script, you will not know about any changes to default parameter values that came with this release.

This release of AppManager for WMI may contain updated Knowledge Scripts. You can push the changes for updated scripts to running Knowledge Script jobs in one of the following ways:

- Use the AMAdmin_UpgradeJobs Knowledge Script.
- Use the Properties Propagation feature.

2.6.1 Running AMAdmin_UpgradeJobs

The AMAdmin_UpgradeJobs Knowledge Script can push changes to running Knowledge Script jobs. Your AppManager repository (QDB) must be at version 7.0 or later. In addition, the repository computer must have hotfix 72040 installed, or the most recent AppManager Repository hotfix. To download the hotfix, see the AppManager Suite Hotfixes Web page.

Upgrading jobs to use the most recent script version allows the jobs to take advantage of the latest script logic while maintaining existing parameter values for the job.

For more information, see the **Help** for the AMAdmin_UpgradeJobs Knowledge Script.

2.6.2 Propagating Knowledge Script Changes

You can propagate script changes to jobs that are running and to Knowledge Script Groups, including recommended Knowledge Script Groups and renamed Knowledge Scripts.

Before propagating script changes, verify that the script parameters are set to your specifications. New parameters may need to be set appropriately for your environment or application.

If you are not using AppManager 8.x or later, customized script parameters may have reverted to default parameters during the installation of the module.

You can choose to propagate only properties (specified in the Schedule and Values tabs), only the script (which is the logic of the Knowledge Script), or both. Unless you know specifically that changes affect only the script logic, you should propagate both properties and the script.

For more information about propagating Knowledge Script changes, see the "Running Monitoring Jobs" chapter of the *Operator Console User Guide for AppManager*.

Propagating Changes to Ad Hoc Jobs

You can propagate the properties and the logic (script) of a Knowledge Script to ad hoc jobs started by that Knowledge Script. Corresponding jobs are stopped and restarted with the Knowledge Script changes.

To propagate changes to ad hoc Knowledge Script jobs:

1 In the Knowledge Script view, select the Knowledge Script for which you want to propagate changes.

2 Right-click the script and select **Properties propagation** > **Ad Hoc Jobs**.

3 Select the components of the Knowledge Script that you want to propagate to associated ad hoc jobs:

Select	To propagate
Script	The logic of the Knowledge Script.
Properties	Values from the Knowledge Script Schedule and Values tabs, such as schedule, monitoring values, actions, and advanced options. If you are using AppManager 8.x or later, the module upgrade process now *retains* any changes you may have made to the parameter settings for the Knowledge Scripts in the previous version of this module.

Propagating Changes to Knowledge Script Groups

You can propagate the properties and logic (script) of a Knowledge Script to corresponding Knowledge Script Group members.

After you propagate script changes to Knowledge Script Group members, you can propagate the updated Knowledge Script Group members to associated running jobs. For more information, see "Propagating Changes to Ad Hoc Jobs" on page 16.

To propagate Knowledge Script changes to Knowledge Script Groups:

1 In the Knowledge Script view, select the Knowledge Script Group for which you want to propagate changes.

2 Right-click the Knowledge Script Group and select **Properties propagation** > **Ad Hoc Jobs**.

3 (Conditional) If you want to exclude a Knowledge Script member from properties propagation, deselect that member from the list in the Properties Propagation dialog box.

4 Select the components of the Knowledge Script that you want to propagate to associated Knowledge Script Groups:

Select	To propagate
Script	The logic of the Knowledge Script.
Properties	Values from the Knowledge Script Schedule and Values tabs, such as schedule, monitoring values, actions, and advanced options. If you are using AppManager 8.x or later, the module upgrade process now *retains* any changes you may have made to the parameter settings for the Knowledge Scripts in the previous version of this module.

5 Click **OK**. Any monitoring jobs started by a Knowledge Script Group member are restarted with the job properties of the Knowledge Script Group member.

3 WMI Knowledge Scripts

AppManager provides the following Knowledge Scripts for monitoring Microsoft Windows Management Instrumentation (WMI) services and executing WMI queries.

From the Knowledge Script view of the Control Center, you can access more information about any Knowledge Script by selecting it and clicking **Help**. In the Operator Console, click any Knowledge Script in the Knowledge Script pane and press **F1.**

Knowledge Script	What It Does
Configure	Configures the frequency of the WMI repository backup and the type, size, and location of the logging files.
EventConsumer	Monitors for events generated by the WMI event provider and allows you to search for events in the database.
LogSizes	Monitors the size of WMI log files.
RepositoryUsage	Monitors the size of the WMI repository.
ResourceHigh	Monitors the CPU and memory consumption for WMI processes.
RunWQL	Allows you to run WQL statements.
ServiceDown	Monitors the availability of the WMI CIMOM service.
UserManager	Allows you to add, delete, or edit a WMI user account or WMI group account for the WMI service.

3.1 Configure

Use this Knowledge Script to configure the frequency of the WMI repository backup and the type, size, and location of the logging files.

NOTE: This Knowledge Script is not supported for WMI servers running Windows Server 2012, Windows 8, Windows Server 2008 R2, Windows Server 2008, Windows Vista, or Windows 7.

3.1.1 Resource Object

WMI server

3.1.2 Default Schedule

The default schedule for this script is **Run once**.

3.1.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Raise event if operation succeeds?	Set to **y** to raise an event when the selected operation succeeds. The default is **y**. **NOTE**: This script always raises an event if a selected operation fails.
Time between repository backup attempts	Enter the number of minutes between WMI backup repository attempts. The default is 60 minutes.
Logging type: disable(d)/ error(e)/verbose(v)	Specify the type of log files you want created by the WMI repository backup operation. Valid values are: ◆ **d** to disable logging (no log file is created) ◆ **e** to enable error logging (log file records any errors encountered) ◆ **v** to enable verbose logging (log file includes error and informational messages) The default is **e**.
Maximum log file size	Enter the maximum size of the log file in bytes. If the log file exceeds this size, the file is truncated. The default is 65535 bytes.
Logging directory	Enter the name of the directory to use for log files.
Event severity level for...	Set the event severity level, from 1 to 40, to indicate the importance of: ◆ **...configuration succeeded**. The default is 25 (blue event indicator). ◆ **...configuration failed**. The default is 5 (red event indicator).

3.2 EventConsumer

Use this Knowledge Script to search a specific WMI repository namespace for events generated by the event provider. Enter the search criteria using the Windows Management Instrumentation Query Language (WQL). You can perform event queries with this script using the SELECT statement and related WITHIN, GROUP, and HAVING clauses. For more information about WQL, refer to the Microsoft documentation.

This script raises an event when events matching your query are found in the WMI repository. You specify the WMI event properties to display in the AppManager event.

3.2.1 Resource Object

WMI server

3.2.2 Default Schedule

The default schedule for this script is **Asynchronous**. This script will run indefinitely until you stop the script.

3.2.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Path to the WMI namespace	Specify the path to the WMI namespace you want to monitor. The default is `root\cimv2`.
WMI event query	Specify the information you want to find in the WMI repository using the WQL query format. The default query is: `select * from __InstanceCreationEvent within 1 where TargetInstance is a 'Win32_NTLogEvent'`
Fields to display for the event in the List pane	Specify the type of event information to display in the Message field in the List pane of the Operator Console. By default, the Source Name and Event ID fields are displayed. For example, if the source of an event is `MSSQLSERVER` and the Event ID is `17055`, the information displayed in the Message field would be `MSSQLSERVER - 17055`. **NOTE**: This parameter requires case-sensitive entries. For example, do not enter `sourcename` if the actual field name is `SourceName`.
Event severity level	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default is 5 (red event indicator).

3.3 LogSizes

Use this Knowledge Script to monitor the size of the following WMI log files:

- cimom.log
- mofcomp.log
- wbemcore.log
- wbemprox.log

This Knowledge Script allows you to set a maximum log file size for individual log files and a maximum size for the sum of all log files. If either threshold is exceeded, an event is raised.

NOTE: This Knowledge Script is not supported for WMI servers running Windows Server 2012, Windows 8, Windows Server 2008 R2, Windows Server 2008, Windows Vista, or Windows 7.

3.3.1 Resource Object

WMI Log file object

3.3.2 Default Schedule

The default interval for this script is **Once every hour**.

3.3.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Event?	Set to **y** to raise events. The default is **y**.
Collect data for all log files?	Set to **y** to collect data for all log files. If set to **y**, the script returns the total file size used by all log files. The default is **n**.
Collect data for individual log files?	Set to **y** to collect data for individual log files. If set to **y**, the script returns the file size used by each log file. The default is **n**.
All log files size maximum threshold	Enter a threshold in MB for the maximum total file size used by all log files. The default is 1000 MB.
Individual log files size maximum threshold	Enter a threshold in MB for the maximum size of each log file. The default is 50 MB.
Event severity level	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default is 5 (red event indicator).

3.4 RepositoryUsage

Use this Knowledge Script to monitor the size of the WMI repository. If the repository size exceeds the threshold you set, an event is raised.

3.4.1 Resource Object

WMI Repository folder

3.4.2 Default Schedule

The default interval for this script is **Once every hour**.

3.4.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Event?	Set to **y** to raise events. The default is **y**.
Collect data?	Set to **y** to collect data for graphs and reports. If set to **y**, the script returns the size of the WMI repository in MB. The default is **n**.
Repository size	Enter a threshold in MB for the repository size. The default is 500 MB.
Event severity level	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default is 5 (red event indicator).

3.5 ResourceHigh

Use this Knowledge Script to monitor CPU and memory consumption by the WMI service (Winmgmt). This script raises an event if CPU usage or memory usage exceeds the threshold you set.

NOTE: The Winmgmt service runs with several other services under an instance of the svchost.exe process. Therefore, when the ResourceHigh Knowledge Script monitors CPU and memory for the Winmgmt service, it is actually monitoring these same metrics for all services hosted by the instance of the svchost.exe process.

If you set the *Collect data?* parameter to **y**, the values returned for percentage of CPU used and MB of memory used are sums of CPU usage and memory usage for all services hosted by the svchost.exe process. The values do not represent CPU and memory usage for only the Winmgmt service.

3.5.1 Resource Object

WMI server

3.5.2 Default Schedule

The default interval for this script is **Every 10 minutes**.

3.5.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Event?	Set to **y** to raise events. The default is **y**.
Collect data?	Set to **y** to collect data for graphs and reports. If set to **y**, the script returns the percentage of CPU resources and the MB of memory used by the WMI service. The default is **n**.
% CPU maximum threshold	Enter a threshold for the maximum percentage of CPU resources that WMI should be allowed to consume. The default is 60%.
Memory maximum threshold (in MB)	Enter a threshold in MB for the maximum amount of memory WMI should be allowed to consume. The default is 6 MB.
Event severity level	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default is 8 (red event indicator).

3.6 RunWQL

Use this Knowledge Script to run WMI Query Language (WQL) queries. You can enter the WQL query to be executed as a parameter of this Knowledge Script, or you can load the query from a script file. You can choose the data output to be a specified number of data rows (all columns) or the value of the first row of a specific column (the column is specified by either number or name).

Examples of simple WQL queries:

```
/* Command to get the path setting for a computer. */
Select * from Environment where Name = 'Path'

/* Command to get information about the provider CIMWin32. */
Select * from _Win32Provider where Name = 'CIMWin32'
```

3.6.1 Resource Object

WMI server

3.6.2 Default Schedule

The default interval for this script is **Run once**.

3.6.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Event?	Set to **y** to raise events. The default is **y**.
Collect data?	Set to **y** to collect data for graphs and reports. The default is **y**.
WQL query	Enter the WQL query that will run. The default query is: `SELECT Caption, ThreadCount FROM win32_Process` **Tip** Unless you are entering very simple queries, you may find that typing WQL statements in this field is error-prone. To avoid errors, you can use the *Load WQL script from file* parameter. Alternatively, if you have an AppManager Developer's license, you can check this Knowledge Script out of the repository, use the Knowledge Script Editor to paste the desired WQL statements into the WQL query field, and then check in the modified Knowledge Script.
Load WQL script from file?	Set to **y** to load an existing WQL script. The file containing the script must be present on the computer on which the Knowledge Script job will run. The default is **n**.
WQL script file (full path)	Enter the full path to the file that contains the WQL script (for example: `C:\netiq\Sample.wql`). **NOTE**: This path is relative to the computer on which the Knowledge Script job will run.
WMI server\namespace	Enter the name of the managed WMI server and CIM namespace. The default is `root\CIMv2`.

Description	How to Set It
Return N rows (set to 0 for all rows)	Enter the number of rows to return as data output when the *Collect data?* parameter is set to **y**. The default is 10. **NOTE**: You can set this value to 0 to set no limit on the number of rows returned. However, there is a limit of 32K for the total of returned data.
Return first row of specified column?	Set to **y** to use a specified column number or column name from which data will be returned (specify the column number or name in one of the two following parameters). Setting this parameter to **y** will override the *Return N rows* parameter. The default is **n**.
Column number	If the *Return first row of specified column?* parameter is set to **y**, the value of the first row of the column specified here (by number), rather than the number of rows, is used as data output. The default is 0.
Column name	If the *Return first row of specified column?* parameter is set to **y**, the value of the first row of the column specified here (by name), rather than the number of rows, is used as data output. The default is blank.
GivenLegend	String used in the Legend column of graph data. If this value is left blank, the Legend column will read: "WQL query results on WMI Server: <machine name>."
Maximum threshold	Set the high watermark for the return value of the WQL query (this value can be the number of rows returned or the value of the first row of a specified column, depending on the type of data output you chose). If the return value exceeds this limit, an event will be raised. The default is 10000.
Minimum threshold	Set a low watermark for the return value of the WQL query (this value can be the number of rows returned or the value of the first row of a specified column, depending on the type of data output you chose). If the return value is below this limit, an event will be raised. The default is 10.
Event severity level	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default is 5 (red event indicator).

3.7   ServiceDown

Use this Knowledge Script to monitor the WMI CIMOM (Common Information Model Object Manager) service. If the CIMOM service is not running, an event is raised. Optionally, you can set the Knowledge Script to attempt to restart the service automatically.

3.7.1   Resource Object

WMI service object

3.7.2   Default Schedule

The default interval for this script is **Every 5 minutes**.

3.7.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Collect data?	Set to **y** to collect data for graphs and reports. The default is **n**. If set to **y**, the script returns a value of 100 if the CIMOM service is running and a value of 0 if the service is not running.
Auto-start service?	Set to **y** to automatically restart down services. The default is **y**.
Event severity level for...	Set the event severity level, from 1 to 40, to indicate the importance of:

◆ **...service down; restart failed**. The default is 5 (red event indicator).

◆ **...service down; restart succeeded**. The default is 25 (blue event indicator).

◆ **...service down; don't restart**. The default is 18 (yellow event indicator).

3.8 UserManager

Use this Knowledge Script to add, delete, or edit WMI user or group accounts for the WMI service on the computer where the Knowledge Script job is running. You can also use this Knowledge Script to manage user access to CIM objects.

This script can add or modify domain accounts or groups. The account or group you are modifying must belong to the same domain as the computer on which the script is running.

By default, this script raises an event whose message informs you of the success or failure of the operation.

If you want to use this script to manage an account that was added from the WMI Control, that account must meet the following requirements:

◆ Permissions must be granted for **This namespace and subnamespaces**.
◆ Permissions must allow Provider Write, Enable Account, and Remote Enable.

NOTE: This Knowledge Script is not supported for WMI servers running Windows Server 2012, Windows 8, Windows Server 2008 R2, Windows Server 2008, Windows Vista, or Windows 7.

3.8.1 Resource Object

WMI server

3.8.2 Default Schedule

The default interval for this script is **Run once**.

3.8.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if operation succeeds?	Set to **y** to raise an event when the selected operation succeeds. The default is **y**. **NOTE**: This script always raises an event if a selected operation fails.
Manage user or group?	Set to **u** to manage user accounts. Set to **g** to manage group accounts. The default is **u**.
Operation to perform: add(a)/ delete(d)/edit(e)	Indicate the type of operation you want to perform. Set to one of the following: ◆ **a** to Add users or groups ◆ **d** to Delete users or groups ◆ **e** to Edit users or groups The default is **a**.
User or Group names to be managed	Enter the user or group account name you want to manage. You can enter multiple names, separated by commas with no spaces. For example: `guest,admin,user1` The default is guest.
Domain name	Enter the domain name associated with the user or group account. You must specify the local machine name when deleting or editing a local user.
Enable this account?	Set to **y** to enable the specified accounts if you are adding new user or group accounts or editing existing accounts. Set to **n** to disable an account. The default is **y**.
Can this account execute methods?	Set to **y** to give the specified accounts permission to execute methods exported from the CIM Object Manager. The default is **n**.
Schema access privileges: Read-only(r) / Write instance(i) / Write class(c)?	Indicate the WMI schema access allowed for the specified accounts. Set to: ◆ **r** to authorize read-only access (users can execute queries or retrieve instances and classes, but cannot create, delete, or modify CIM objects). ◆ **i** to allow the account read/write/delete access to instances in the WMI schema and read-only access to classes. ◆ **c** to allow the account full read/write/delete access to all CIM objects, classes, and instances in the WMI schema. The default is **r**.
Can this account edit security?	Set to **y** to give the specified accounts permission to edit security. When set to **y**, the user has read and write access to the ROOT\Security namespace. When set to **n**, the user cannot access the ROOT\Security namespace. The default is **n**.

Description	How to Set It
Group account type: NTLM Local(l)/NTLM Global(g)	Indicate whether the specified group accounts are local to the managed computer or global. Set to: ◆ **l** to create a local Windows group (that can only access the local workstation or domain). ◆ **g** to create a domain global group (that can access its own domain, member servers and workstations in the domain, and trusting domains). This parameter is used only for managing group accounts. The default is **l**.
Event severity level	Set the severity level, from 1 to 40, to indicate the importance of the event. The default is 12 (yellow event indicator).