Management Guide

NetIQ® AppManager® SNMP Toolkit

December 2018

Legal Notice

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
Website:	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
Website:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ web site in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **comment on this topic** at the bottom of any page in the HTML version 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

NetIQ Communities, 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, NetIQ Communities 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 SNMP Toolkit

This chapter introduces the AppManager SNMP Toolkit and describes how you can use AppManager to better monitor SNMP-enabled devices.

1.1 Brief Overview

SNMP is perhaps the most prevalent network management protocol in use today. Most network devices (such as routers, switches, phones, and printers) and many host systems provide SNMP Management Agents. These agents allow you to monitor and manage the devices and host systems from a remote location.

The AppManager SNMP Toolkit helps you gain easy access to SNMP device data, and to help you analyze and manage that data. The module also minimizes the cost of maintaining SNMP devices, aids in capacity planning, and prevents downtime.

The SNMP Toolkit lets you reorganize and centralize all SNMP capabilities by providing extensive and flexible features. The Toolkit includes Knowledge Scripts for creating jobs that monitor the health, availability, and performance of key SNMP devices. These scripts allow you to monitor and manage crucial SNMP device properties at a depth unparalleled by any other solution. You can configure each Knowledge Script to send an alert, collect data for reporting, and perform automated problem management when an event occurs.

1.2 Features and Benefits

The following are just a few of the features and benefits of monitoring SNMP-enabled devices with AppManager:

- Centralize all SNMP generic GET/SET capabilities under a single AppManager tab
- Retrieve or poll anything from individual SNMP OIDs up to portions of (or entire) SNMP Tables
- Easily add new MIBs, without entering raw OID values
- Support SNMP versions 1, 2, and 3.
- Store and protect SNMP community strings and security profiles
- Use Knowledge Scripts to perform common SNMP calculations such as deltas on iteratively retrieved values
- Poll SNMP attributes at short fixed-time intervals, and report critical data about the polled values
- Use Knowledge Scripts to perform an SNMP walk retrieval of SNMP tables and process/report results across the table
- Create customized Knowledge Scripts tailored to serve your particular environment
- Checks for SNMP traps forwarded from NetIQ SNMP Trap Receiver
- Monitors SNMP version 1, 2, and 3 devices by using SNMP to poll MIBs (management information bases).

1.3 Proxy Architecture

With AppManager support for SNMP, the agent does not need to be installed on every SNMP-enabled device you want to monitor. With this *proxy* architecture, the module is installed on a proxy agent computer. When you run a Knowledge Script job, the managed object runs on the proxy computer and sends messages to or from the network devices (using the SNMP `GET/GetNext/Set` commands) for which you have designated the managed client as the proxy.

In order for the SNMP proxy to function, SNMP must be enabled on the SNMP devices that you want to monitor. Only one computer should act as a proxy for any given SNMP device.

1.4 Counting AppManager Licenses

The AppManager SNMP Toolkit consumes one AppManager license per proxy agent computer. Each proxy agent computer is allowed to manage 50 devices.

2 Installing and Using the AppManager SNMP Toolkit

This chapter provides installation instructions and describes system requirements for the AppManager SNMP Toolkit.

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 SNMP Toolkit has the following system requirements:

Software/Hardware	Version
NetIQ AppManager installed on the AppManager repository (QDB) computers, on all proxy agent computers, and on all console computers	8.0.3, 8.2, 9.1, 9.2, 9.5, or later One of the following AppManager agents are required: ◆ AppManager agent 7.0.4 with hotfix 72616 or later ◆ AppManager agent 8.0.3, 8.2, 9.1, 9.2, 9.5, or later
Microsoft Windows operating system on each proxy agent computers	One of the following: ◆ Windows 10 (32-bit or 64-bit) ◆ Windows Server 2016 ◆ Windows Server 2012 R2 ◆ 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)
SNMP running on the devices that you want to monitor	Versions 1, 2, or 3
AppManager for Microsoft Windows module installed on the AppManager repository (QDB) computer, on all proxy agent computers, and on all console computers	7.6.170.0 or later

Software/Hardware	Version
Microsoft SQL Server Native Client 11.0	11.3.6538.0 or later
(for TLS 1.2 support)	**NOTE:** The SQL Server Native client can be installed from this Microsoft download link.

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

NOTE: If you want TLS 1.2 support and are running AppManager 9.1 or 9.2, then you are required to perform some additional steps. To know about the steps, see the article.

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.

NOTE: Installing the module automatically installs NetIQ SNMP Trap Receiver. For more information, see Section 2.10, “Working with NetIQ SNMP Trap Receiver,” on page 21.

Access the `AM70-SNMP-7.x.x.0.msi` module installer from the `AM_SNMP_7.x` 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 the module installer `.msi` file from a command prompt or by double-clicking it.
- Log in to the server as a user with administrative privileges and run the module installer `.msi` file 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 into local or remote AppManager repositories (QDBs). Install these components only once per QDB.

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

You can install the module manually, or you can use Control Center to deploy the module on a remote computer that has an agent 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, and 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.

NOTE: Microsoft .NET Framework 3.5 is required on the computer where you run the installation program for the QDB portion of the module. For computers running more recent versions of Windows operating systems that use a newer version of .NET, install .NET 3.5 with the Add Roles and Features wizard in Windows Server Manager, as described in this Microsoft article.

---

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

6 ***If you use Control Center 8.x or later,*** run the module installer only for the primary QDB, and Control Center will automatically replicate 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 all proxy agent computers to install the agent components.

9 Configure SNMP community strings and permissions for the devices you want to monitor. For more information, see Section 2.5, “Configuring SNMP Permissions,” on page 15.

10 ***If you have not discovered SNMP enabled devices***, run the Discovery_Snmp Knowledge Script on all proxy agent computers where you installed the module. For more information, see Section 2.6, “Discovering SNMP Resources,” on page 17.

11 To get the updates provided in this release, upgrade any running Knowledge Script jobs. For more information, see Section 2.7, “Upgrading Knowledge Script Jobs,” on page 19.

12 After the installation has completed, the `SNMP_Install.log` file, located in the `\NetIQ\Temp\NetIQ_Debug\`<*ServerName*> folder, lists any problems that occurred.

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 14
3. Configure an e-mail 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-SNMP-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 later) 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 later) or **Check in Packages** (for AppManager 7.*x*).
5. Navigate to the folder where you saved AM70-SNMP-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.

2.4 Silently Installing the Module

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

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

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

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

```
/L* "AM70-SNMP-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 2008 R2, 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-SNMP-7.x.x.0.msi /qn MO_B_QDBINSTALL=1 MO_B_SQLSVR_WINAUTH=1
MO_SQLSVR_NAME=SQL_Server_Name MO_QDBNAME=AM-Repository Name
```

SQL authentication:

```
AM70-SNMP-7.x.x.0.msi /qn MO_B_QDBINSTALL=1 MO_B_SQLSVR_WINAUTH=0
MO_SQLSVR_USER=SQL login MO_SQLSVR_PWD=SQL Login Password MO_SQLSVR_NAME=SQL
Server Name MO_QDBNAME=AM-Repository Name
```

2.5 Configuring SNMP Permissions

For each SNMP-enabled device that you want to discover, configure SNMP information in AppManager Security Manager *before* you run the Discovery_Snmp Knowledge Script. The type of information you configure varies according to the version of SNMP implemented on the device. The SNMP Toolkit supports SNMP versions 1, 2, and 3.

Configuring SNMP information provides AppManager the permissions it needs to access the MIBs on SNMP-enabled devices.

If you do not explicitly configure SNMP information for AppManager SNMP Toolkit, the Snmp category of Knowledge Scripts search for and use community strings you may have already configured for use by the AppManager for Network Device module.

2.5.1 Configuration for SNMP Versions 1 and 2

Configure community string and version information for each device that is being monitored by each proxy agent computer. On the Custom tab in Security Manager, complete the following fields:

Field	Description
Label	`SNMP`
Sub-label	Indicate whether the community string information will be used for a single device or for all devices: ◆ For a single device, type the <*device name*>. ◆ For all devices, type `default`.
Value 1	The appropriate community string value, such as private or public. **NOTE:** The SyncSet Knowledge Script requires read/write permission. All other Knowledge Scripts require read-only permissions.
Value 3	Type v1 or 1 if the device supports SNMP v1. Type v2 or 2 if the device supports SNMP v2. **NOTE:** If you do not specify an SNMP version, AppManager attempts to determine the version during the Discovery job. This process can be time consuming.

2.5.2 Configuration for SNMP Version 3

The AppManager SNMP Toolkit supports the following modes for SNMP v3:

- No authentication; no privacy
- Authentication; no privacy
- Authentication and privacy

In addition, the modules supports the following protocols for SNMP v3:

- MD5 (Message-Digest Algorithm 5, an authentication protocol)
- SHA (Secure Hash Algorithm, an authentication protocol)
- DES (Data Encryption Standard, an encryption protocol)
- AES (Advanced Encryption Standard, an encryption protocol, 128-bit keys only)

Your SNMP v3 implementation may support one or more combinations of mode and protocol. that combination dictates the type of information you configure in AppManager Security Manager: User Name (or entity), Context name, protocol name, and protocol passwords.

Configure SNMP v3 information for each SNMP device that is being monitored by each proxy computer. On the Custom tab in Security Manager, complete the following fields:

Field	Description
Label	`SNMP`
Sub-label	Indicate whether the User Name and Context will be used for a single device or for all devices: ◆ For a single device, type the <*device name*>. ◆ For all devices, type `default`.
Value 1	The SNMP User Name or entity configured for the device. **NOTE:** All SNMP v3 modes require an entry in the **Value 1** field.
Value 2	The name of a context associated with the user name or entity you entered in the **Value 1** field. A context is a collection of SNMP information that is accessible by an entity. If possible, enter a context that provides access to all MIBs for a device. If the device does not support context, type an asterisk (*). All SNMP v3 modes require an entry in the **Value 2** field.
Value 3	The combination of protocol and password appropriate for the SNMP v3 mode you have implemented: ◆ For no authentication/no privacy mode, leave the **Value 3** field blank. ◆ For authentication/no privacy mode, type `md5` or `sha` and the password for the protocol, separating each entry with a comma. For example, type `md5,abcdefgh` ◆ For authentication/privacy mode, type `md5` or `sha` and the associated password, and then type `des` or `aes` and the associated password, separating each entry with a comma. For example, type `sha,hijklmno,des,nopqrstu`

2.6 Discovering SNMP Resources

Use the Discovery_Snmp Knowledge Script to discover SNMP-enabled devices running on a network. The AppManager agent on which the SNMP Toolkit is installed acts as a proxy for discovering SNMP devices. This AppManager agent is referred to as the SNMP proxy agent computer.

By default, this script runs once for each computer.

In a successful discovery, the following details are discovered:

Object	Discovered Details
SNMP Proxy object	Agent Address – The hostname or IP address of the SNMP proxy agent computer.
SNMP Device object	◆ Device Address — The hostname or IP address under which the device was discovered. ◆ SNMP Version – The SNMP version being used by the SNMP device. ◆ System Name – The value of the SNMP attribute `sysName.0`. ◆ Vendor – The name of the vendor that manufactured the device. ◆ OID – System OID from `sysObjectID.0` that uniquely identifies the type of device. ◆ Services – The network services supported by the device, as specified by `sysServices.0`. ◆ Contact – The value of `sysContact.0`. ◆ Location – The value of `sysLocation.0`.

Devices can only be discovered by providing a device list of hostnames, IP addresses, or IP address ranges for the corresponding Discovery Knowledge Script parameters. There is no automatic discovery capability. The Discovery_Snmp script iteratively attempts to contact and retrieve the System MIB from each supplied hostname or IP address.

Because many SNMP devices (typically routers) have multiple IP addresses, some devices may be discovered multiple times. In these cases, the Discovery script automatically detects and removes the duplicates from the list of discovered devices. Any duplicates are shown in the AppManager TreeView pane under the IP address or hostname under which they were first discovered.

During discovery, various errors may occur with the supplied list of SNMP devices. Most commonly, these will be SNMP timeouts from devices failing to respond because they are down, because they are not running an SNMP agent, or because the community string supplied for the device is incorrect. Other common errors can be due to bad hostnames or SNMP Response errors.

Discovery can take anywhere from a few seconds to several minutes or hours. The time taken largely depends on how many SNMP timeouts occur. For example, if this script is configured with an *SNMP timeout* value of 5 seconds and 3 *SNMP retries*, it takes 20 seconds to determine that the device is not responding.

The time taken for discovery to finish can be controlled by limiting the use of IP address ranges, which are likely to contain many addresses that are not used or that do not correspond to SNMP devices, and to keep the SNMP timeout/retry values as small as is practical for the local network environment.

Set the following parameters as needed:

Parameter	How to Set It
Discovery Parameters	
List of SNMP devices	Supply a list of SNMP device hostnames or IP addresses for the devices you want to monitor. An attempt is made to discover each device listed. The default is `localhost`.
Address ranges of SNMP devices	Instead of listing each IP address individually in *List of SNMP devices*, you can provide a list of IP address ranges. Each range must be in the format "`A.B.C.D-W.X.Y.Z`". For example, enter "`10.1.1.0-10.1.1.254`". Each address range may be no longer than 255 addresses, although they can span subnets. For example, "`10.1.1.250-10.1.2.30`" spans a subnet.
Full path to file with list of SNMP devices	Instead of listing each IP address or range of addresses, you can supply the full path to a file containing a list of individual IP addresses or hostnames. The file path supplied must be accessible from the SNMP proxy agent computer, not the AppManager Operator Console or repository computer. Device names can be separated by commas, blank characters or new lines.
Maximum number of devices to discover	Specify the maximum number of devices to be discovered. Discovery stops when this limit is reached, even if not all IP addresses and hostnames have been attempted. The default and maximum allowed value is 250.
SNMP port number	Specify the UDP port number on the remote SNMP device to which you want to send SNMP requests. The default is 161.
SNMP retries	Specify the number of retries to attempt if a timeout occurs on an SNMP request. The default is 1 retry.
SNMP timeout	Specify the number of seconds to wait for a response before timing out an SNMP request. The default is 3 seconds.
Trap Receiver Discovery	
Discover Trap Receiver?	Set to **Yes** to discover NetIQ SNMP Trap Receiver. The default is Yes. For more information, see Section 2.10, "Working with NetIQ SNMP Trap Receiver," on page 21.
Trap Receiver IP address	Specify the IP address of the computer on which Trap Receiver is installed. The default is `localhost`.
Trap Receiver TCP port	Specify the TCP port number through which Trap Receiver will communicate with AppManager. The default is port 2735.
Event Notification	
Raise event if discovery succeeds?	Set to **Yes** to raise an event if discovery is completely successful. The details of the event contain the list of discovered devices.
Event severity when discovery succeeds	Set the event severity level, from 1 to 40, to indicate the importance of an event in which discovery succeeds. The default is 25. Discovery is successful if all devices in the device list are discovered and no errors occur. The event message indicates any duplicate IP addresses or hostnames. Duplicates are not considered to be an error.

Parameter	How to Set It
Raise event if discovery partially succeeds?	Set to **Yes** to raise an event if discovery is partially successful. The details of the event contain the list of discovered devices and the discovery errors that occurred.
Event severity when discovery partially succeeds	Set the event severity level, from 1 to 40, to indicate the importance of an event in which discovery partially succeeds. The default is 15. A partial discovery occurs if only some of the SNMP devices are discovered, or if the maximum device limit is reached. The event message indicates any duplicate IP addresses or hostnames. Duplicates are not considered errors.
Raise event if discovery fails?	Set to **Yes** to raise an event if no SNMP devices are discovered. The details of the event contain the discovery errors that occurred.
Event severity when discovery fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which no SNMP devices are discovered. The default is 10.

2.7 Upgrading Knowledge Script Jobs

This release of the SNMP Toolkit might 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.

2.7.1 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 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.

2.7.2 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. Customized script parameters may have reverted to default parameters during the installation of the module. New parameters may need to be set appropriately for your environment or application.

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. Click **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.

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 20.

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. On the KS menu, select **Properties propagation** > **Ad Hoc Jobs**.
3. ***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, including the schedule, actions, and Advanced properties.

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.

2.8 Recovering Lost MIB Files from the Backup Folder

When AppManager SNMP Toolkit is installed, the setup program creates two copies of the MIB file. These files are installed in the following default folders:

- *installation folder*`\AppManager\bin\AMSnmpMIBs`
- *installation folder*`\AppManager\bin\AMSnmpMIBBackups`

The SNMP Toolkit actively uses the MIB files in the `\AppManager\bin\AMSnmpMIBs` folder, but not the files in the `\AppManager\bin\AMSnmpMIBBackups` folder.

If you accidentally delete one of the original MIB files (for example, by running the RemoveMIBs Knowledge Script), you can recover it from the backup folder. Simply use the *Full MIB file path*

parameter in the AddMIBs Knowledge Script to supply the path to the `AMSnmpMIBBackups` folder.

2.9 Limiting Knowledge Script Targets

You should limit the number of target objects for any given SNMP Knowledge Script job. Running large jobs will test the limits of your system's CPU and memory resources. For instance, running the SyncGetTable Knowledge Script on multiple devices may seem inconsequential, but some devices can have 200 interfaces or more.

2.10 Working with NetIQ SNMP Trap Receiver

NetIQ SNMP Trap Receiver (Trap Receiver) is installed automatically when you install the SNMP Toolkit. Trap Receiver runs as a service, `NetIQTrapReceiver.exe`, and may compete for port usage with any other trap receiver installed on the same computer. For more information, see Section 2.10.3, "Understanding Trap Receiver Architecture," on page 22.

2.10.1 What is NetIQ SNMP Trap Receiver?

At its most basic, a trap receiver is an application that receives traps from SNMP agents. Trap Receiver receives, filters, and forwards SNMP traps to AppManager. When you use Trap Receiver with the SNMP Toolkit, the SNMPTrap_Async Knowledge Script raises events when SNMP traps are received.

2.10.2 What is an SNMP Trap?

Simple Network Management Protocol (SNMP) is a protocol-based system used to manage devices on TCP/IP-based networks. From devices on which an SNMP agent resides, such as routers and switches, SNMP sends unsolicited notifications, called traps, to network administrators when thresholds for certain conditions are exceeded. These conditions are defined by the vendor in a device's Management Information Base (MIB); the network administrator sets the thresholds.

Traps are composed of protocol data units (PDUs). Each PDU contains the following information, organized in various ways depending on the version of SNMP in use:

- SNMP version number
- Community name of the SNMP agent
- PDU type

- Enterprise OID (object identifier), a unique number that identifies an enterprise and its system objects in the MIB
- IP address of the SNMP agent
- Generic trap type: Cold start, Warm start, Link down, Link up, Authentication failure, and Enterprise
- Specific trap type. When the Generic trap type is set to “Enterprise,” a specific trap type is included in the PDU. A specific trap is one that is unique or specific to an enterprise.
- Time the event occurred
- Varbind (variable binding), a sequence of two fields that contain the OID and a value

2.10.3 Understanding Trap Receiver Architecture

Trap Receiver operates on a Client-Server architecture: the *Server*—the stand-alone Trap Receiver application—receives, filters, and forwards SNMP traps to the *Client*—an application that receives traps, such as AppManager. The Server can receive traps on standard UDP port 162 or on any other configured port. The Client and the Server can reside on the same computer or on separate computers.

Communication between Client and Server is implemented as XML messages over a TCP connection. Only one Server is allowed per computer, however, several Clients are allowed per computer. Clients that are registered to the same Server share the same TCP connection. The Server TCP port should be known to all potential Clients.

2.10.4 Coexisting with Microsoft SNMP Trap Service

Two trap receivers cannot be in use on the same computer while using the same standard UDP port (162). If NetIQ SNMP Trap Receiver and another trap receiver such as Microsoft SNMP Trap Service are installed on the same computer and both are receiving traps, configure Trap Receiver to use the standard UDP port and to forward incoming traps (UDP forwarding) to the other trap receiver. For more information, see Section 2.10.5, “Understanding the Trap Receiver Configuration File,” on page 23.

Then, configure the other trap receiver to use a different, non-standard, UDP port that is not in use by another application. The following are instructions for configuring Microsoft SNMP Trap Service.

To configure Microsoft SNMP Trap Service to use another port:

1. Navigate to `c:\Windows\system32\drivers\etc`.
2. Open the **services** file.
3. In the row for `snmptrap`, change the value for **udp** from 162 to another port number that is not in use by any other application. Use the same port number you set as the forwarding port in the Trap Receiver configuration file. For more information, see Section 2.10.5, “Understanding the Trap Receiver Configuration File,” on page 23.
4. Save and close the **services** file.
5. Restart Windows SNMP Trap Service. In Control Panel, double-click **Administrative Tools** and then double-click **Services**. Right-click **SNMP Trap Service** and select **Restart**.

TIP: To see which ports are in use, run `netstat.exe` from a command prompt. Then select an available port as the port for the other trap receiver service.

2.10.5 Understanding the Trap Receiver Configuration File

The configuration file for Trap Receiver, `NetIQTrapReceiver.conf`, identifies the UDP and TCP ports used by Trap Receiver: the UDP port is used for receiving traps; the TCP port is used for communicating with the Client, such as AppManager or another supported NetIQ application. The configuration file also identifies the level of logging you want to use and whether port forwarding is enabled.

By default, the configuration file is installed in [*installation directory*]`\config`.

The configuration file has the following format:

```
################################################################
#
# NetIQTrapReceiver.conf
#
# A configuration file for NetIQ SNMP Trap Receiver
#
################################################################
#########################
# TCP port
# Syntax: tcp_port [port]
# E.g. : tcp_port 2735
#########################
tcp_port 2735
#########################
# UDP port
# Syntax: udp_port [port]
# E.g. : udp_port 162
#########################
udp_port 162
#########################
# Forwarding
# Syntax: forward [address]:[port] [v1]
# E.g. : forward 127.0.0.1:1000 v1
#########################
#########################
# Log level
# Syntax: log_level error|warning|info|debug|xml
# E.g. : log_level info
#########################
log_level debug
```

If the configuration file cannot be found, cannot be parsed, or does not contain one of the required values, Trap Receiver is initialized with the default configuration as shown above.

When changing values in the configuration file, take into account the following:

- If you change the TCP port number, stop all asynchronous Knowledge Script jobs associated with the modules that support Trap Receiver. Run the appropriate Discovery Knowledge Script on all monitored devices to enable the devices to recognize the new TCP port number.
- If you change the UDP port number, also change the UDP port number configured on the devices that send traps to Trap Receiver.
- If another service uses port 2735 or port 162, Trap Receiver will not start. The Trap Receiver log file will contain the error message. Either change the port numbers in the configuration file, stop the service that is using the default Trap Receiver port numbers, or forward the traps coming in to UDP port 162.

- To forward incoming traps to another trap receiver, such as Microsoft SNMP Trap Service, set the Forwarding values as follows:
  `forward [IP address of other trap receiver]:[port number of other trap receiver] [SNMP version]`.
  For example: `forward 10.40.40.25:167 v1`. By default, incoming traps are not forwarded. For more information, see Section 2.10.4, “Coexisting with Microsoft SNMP Trap Service,” on page 22.
- Restart Trap Receiver after any change to the configuration file. From Control Panel, double-click Administrative Tools and then double-click Services. Right-click NetIQ Trap Receiver and select Restart.

2.10.6 Trap Receiver Log File

When installed as a stand-alone application on a computer that is not running an AppManager agent, Trap Receiver saves its log file, `trap.log`, in the `[installation directory]\log directory`.

When installed along with an AppManager module or on a computer with a previous installation of an AppManager agent, Trap Receiver saves its log file in the default AppManager location: `\\Program Files\NetIQ\Temp\NetIQ_debug`.

The `trap.log` file contains initialization data and error messages.

2.10.7 Tips for Using Trap Receiver

This topic provides tips for sending and receiving SNMP traps from one agent computer to another.

1. Establish which server is the Sender (source) and which one is the Receiver (destination). Note the hostnames of the servers and their roles. The Receiver should be the server on which the NetIQ SNMP Trap Receiver is installed.
2. On the Sender server, configure the Receiver server as a trap destination.
3. Before running Discovery_Snmp, add the required SNMP community strings in AppManager Security Manager. For more information, see Section 2.5, “Configuring SNMP Permissions,” on page 15.
4. Run Discovery_Snmp on the Sender server and provide the following details in the Values tab:
   - **4a** *List of SNMP devices* parameter: Hostname of the Sender server.
   - **4b** *Trap Receiver IP address* parameter: IP address of the Receiver server.
5. Run SNMPTrap_Async on the Sender server. Leave the *List of trap OIDs* parameter blank the first time you run the script. You can set up filtering after testing.
6. Send an SNMP trap from the Sender to the Receiver to ensure an event is raised. If you send a trap from any other server, an event is not raised.

NOTE

- If the Sender has a community string of `public`, then the traps should contain the same community string that is specified in Security Manager. The traps are also designed to filter on community string.
- Traps are designed to filter on the Sender IP address, the community string, and the OIDs, if specified. If one of these is incorrect, no event is raised.

---

3 Snmp Knowledge Scripts

The Snmp category provides a set of Knowledge Scripts for monitoring SNMP-enabled devices. From the Knowledge Script view of Control Center, you can access more information about any NetIQ-supported Knowledge Script by selecting it and clicking **Help**. Or in the Operator Console, click any Knowledge Script in the Knowledge Script pane and press **F1**.

Many Snmp Knowledge Scripts use the terms “ODE” or “OID”:

- **ODE**: SNMP Object Description in its readable, named format. For example, “`system.sysDescr.0`” is the ODE for the numeric OID “`.1.3.6.1.2.1.1.0`”.
- **OID**: SNMP Object Identifier in its numeric format. For example, “`.1.3.6.1.2.1.1.0`” is the OID for the SNMP attribute “`system.sysDescr.0`”.

For more information, see Section 3.1, “Customizing Snmp Knowledge Scripts,” on page 26.

The Snmp category includes the following Knowledge Scripts:

Knowledge Script	What It Does
AddMIBs	Copies the specified list of MIB files to the AppManager agent host where you installed AppManager SNMP Toolkit. Reloads the MIB tree for the module so the new MIBs are recognized.
DeviceReboot	Monitors device uptime for an SNMP device and raises an event if the device has restarted since the last monitoring interval.
InterfaceState	Monitors the contents of the Interface MIB from an SNMP device and raises an event when an interface changes state.
RemoveMIBs	Removes the specified list of MIB files from the AppManager agent host where you installed AppManager SNMP Toolkit. Reloads the MIB tree for the module so the removed MIBs are no longer accessible.
SNMPTrap_Async	Checks for incoming SNMP traps forwarded from NetIQ SNMP Trap Receiver.
SyncGet	Attempts an SNMP `Get` or `GetNext` for the specified SNMP attributes. Thresholds can be checked, and mathematical conversions can be performed.
SyncGetTable	Retrieves a specified set of SNMP table columns, and reports results for each row retrieved. Thresholds can be checked, and mathematical conversions can be performed.
SyncPoll	Polls specified SNMP attributes at a prescribed time interval and number of polling attempts. Thresholds can be checked, and mathematical conversions can be performed.
SyncPollTable	Polls a specified set of SNMP table columns at a prescribed time interval and number of polling attempts, and reports summary results for each row polled. Thresholds can be checked, and mathematical conversions can be performed.
SyncSet	Attempts an `SNMP Set` for the specified SNMP attributes of the specified values.

3.1 Customizing Snmp Knowledge Scripts

There are several simple ways to customize Snmp Knowledge Scripts to enhance or add functionality. You can check out an existing script from the `\AppManager\bin\kp\SNMP` folder, perform customizations, and check in the modified script or rename the file to create a new Knowledge Script.

The following list describes how to customize existing Snmp scripts to take advantage of additional functions and features of AppManager SNMP Toolkit.

Enabling SNMP Traffic Tracing

You can enable SNMP Traffic Tracing for all Knowledge Scripts. Turning on Traffic Tracing prints the contents of all SNMP requests or SNMP responses to the `mctrace.log` file on the AppManager agent. This capability can be used as a debugging tool to determine what is actually being sent or received by AppManager SNMP Toolkit.

To enable SNMP Traffic Tracing for a specific Knowledge Script, locate the `Const gintSNMPSectionTraceOn` entry and change the value from “0” to “1”. Check out the script, then locate and edit the following block of text accordingly:

```
'#
'# Constants for SNMP Traffic Tracing
'# Set gintSNMPSectionTraceOn to 1 to trace SNMP Traffic.
'# Output goes to the AppManager Agent's mctrace.log file.
'#
Const gintSNMPSectionTraceBit        = 1
Const gintSNMPSectionTraceOn         = 0
```

Changing default locations of community strings

You can change the default locations of community strings (as defined in the **Custom** tab of AppManager Security Manager) for all Knowledge Scripts except for AddMIBs and RemoveMIBs:

- `gstrSNMPSecurityLabels` lists the labels under which the Knowledge Script searches for community strings, in the order they are listed.
- `gstrSNMPDefaultDevice` specifies the **Sub-Label** used to supply the default community string if one has not been configured for a specific device address.

To edit these values, check out the desired Knowledge Script, then locate and edit the following block of text accordingly:

```
'#
'# String(s) to use for finding SNMP Community Strings
'#
Const gstrSNMPSecurityLabels      = "SNMP,NetworkDevice"
Const gstrSNMPDefaultDevice       = "Default"
```

Changing default list separator characters

Many Snmp Knowledge Script parameters are lists: devices, SNMP OIDs, and the like. The characters that serve as list separators can be changed. The defaults are blank and comma.

You can add additional list separator characters, but you cannot delete the comma character.

To edit these values, check out the desired script, then locate and edit the following block of text accordingly:

```
'#
'# Separators for KS Parameters
'#
Const gstrSNMPListSeparators      = " ,"
```

If you change `gstrSNMPListSeparators`, you must also add the `<Delim></Delim>` option to each script parameter's XML definition to specify the same separators defined by `gstrSNMPListSeparators`. If it is not changed in both places, the script will not work correctly.

For the SyncSet script, there is an additional constant: `gstrSNMPStringValueSeparators`. The default is "`,`" (comma). This specifies how SNMP values for this script are delimited. Because setting string values may include strings with spaces, only a comma is allowed as a separator. For example:

```
'#
'# Separators for KS Parameters
'#
Const gstrSNMPListSeparators         = " ,"
Const gstrSNMPStringValueSeparators  = ","
```

Default empty strings in the Discovery_Snmp script

For the Discovery_Snmp Knowledge Script, there are default strings defined to be used if any of the SNMP device details are empty strings. If `sysName.0` is an empty string, "`No Name`" is shown in the TreeView and the object details. All other object details will show "`No Value`" if they did not have a value. For example:

```
'#
'# String(s) to use for SNMP Devices with empty values in System MIB.
'#
Const gstrSNMPNoDeviceName        = "No Name"
Const gstrSNMPNoDeviceValue       = "No Value"
```

Changing default legend prefix and units for the DeviceReboot script

For the DeviceReboot Knowledge Script, the legend prefix and units in which device uptime is reported can be changed. For example, you could change it to Days by editing the text below and changing the divisor value to `86400`.

```
'#
'# Constants for Datastreams
'#
Const DEVICE_UPTIME_LEGEND       = "Device UpTime"
Const DEVICE_UPTIME_AC           = "Device UpTime (Hours)"
Const DEVICE_UPTIME_UNITS        = "Hours"
Const DEVICE_UPTIME_DIVISOR      = 3600
```

Changing default legend prefix for SyncGet and SyncGetTable scripts

For the SyncGet and SyncGetTable Knowledge Scripts, the legend prefix can be changed. Locate and edit the text below:

```
'#
'# Constant for SNMP Generic Scripts Legend Prefix
'#
Const gstrSNMPLegendPrefix        = "SNMP"
```

NOTE: If you rename a customized Knowledge Script, you cannot access Help using the **Help** button in the Knowledge Script Properties dialog box. Refer to the Help for the original script for assistance with parameter configuration.

3.2 AddMIBs

Use this Knowledge Script to install additional MIB files on an SNMP proxy agent computer. The specified files are copied to a default MIBs folder for the AppManager SNMP Toolkit module, and the module reloads the MIB tree so the new MIBs take effect.

This script copies files to and reloads both `AppManager/bin/AMSnmpMIBs` and `AppManager/bin/MIBs` directories. Both managed objects generate events for this script, and the short event messages include the name of the relevant managed object.

Run this script when no other SNMP scripts are active on the target proxy agent computer, because reloading the MIB tree can take place only when no SNMP sessions are active. If this script is unable to reload the MIB tree because of active SNMP sessions, an appropriate event can be raised. If no path or MIB files are supplied, by default this script creates an event containing a list of currently installed MIBs.

This script raises an event if the specified MIB files cannot be found, or an `ASN.1` compilation error occurs while reloading the MIB tree.

SNMP MIBs tend to be organized hierarchically. One MIB is often dependent on another, from which it imports more generic data definitions. Therefore, add a MIB only if all dependent MIBs are already present or are supplied in the same execution of this script.

3.2.1 Resource Object

SNMP Proxy Agent computer

3.2.2 Default Schedule

By default, this script runs once.

3.2.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Full MIB file path	Specify the directory path where the MIB files to be installed are located. This script does not transfer the specified MIB files over the network. The directory path specified for the MIB files must already be locally accessible by the SNMP proxy agent computer.
List of MIB files	Supply a list of filenames for the MIBs you want to install. Do not include the directory path.
Reload MIB tree?	Set to **Yes** to reload the MIB tree after MIB files have been copied. A MIB tree reload is attempted only if all MIB files are installed successfully.
MIB reload timeout	Specify how long this script should wait to try and reload the MIB tree. This can only be performed if no other script has active SNMP sessions. The default is 10 seconds.
Event Notification	

Parameter	How to Set It
Raise event if installation of MIBs succeeds?	Set to **Yes** to raise an event if all MIBs are installed successfully. The details of the event contain the list of installed MIB files.
Event severity when installation of MIBs succeeds	Set the event severity level, from 1 to 40, to indicate the importance of an event in which MIB installation succeeds. The default is 25.
Raise event if installation of MIBs fails?	Set to **Yes** to raise an event if any MIB files fail to install. This can occur if the path of MIB filenames specified is incorrect.
Event severity when installation of MIBs fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which MIB installation fails. The default is 10.
Raise event if reloading of MIB tree succeeds?	Set to **Yes** to raise an event if all MIB files install successfully and the MIB tree is reloaded successfully. This can only occur if the *Reload MIB tree?* parameter is enabled.
Event severity when reloading of MIB tree succeeds	Set the event severity level, from 1 to 40, to indicate the importance of an event in which MIB tree reloading is successful. The default is 25.
Raise event if reload MIB parser warnings received?	Set to **Yes** to raise an event if all MIB files install successfully and reloading the MIB tree reports `ASN.1` parsing errors. This can only occur if the *Reload MIB tree?* parameter is enabled. Details of the parsing errors are in the event.
Event severity when reload MIB parser warnings received	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the MIB reload generates warnings. The default is 15.
Raise event if reloading of MIB tree fails?	Set to **Yes** to raise an event if all MIB files install successfully and reloading the MIB tree fails. This can occur if *Reload MIB tree?* is enabled and the *MIB reload timeout* expires.
Event severity when reloading of MIB tree fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which MIB tree reload fails. The default is 10.

3.3 DeviceReboot

Use this Knowledge Script to monitor whether an SNMP device or its network management component has rebooted between job intervals. This script tracks the uptime value of the host (`hrSystemUptime.0`) or the network management component (`sysUpTime.0`) across job iterations to determine whether devices have rebooted. If the uptime value is less than the last iteration, the device has either rebooted or the uptime counter value has wrapped.

This script attempts to track `sysUpTime.0` on all SNMP devices where the script is run, regardless of whether any previous attempts have failed. If failures do occur, successful retrievals on other devices are not discarded.

This script generates data streams for each monitored SNMP device. The data value saved is the current number of hours the device has been up.

3.3.1 Resource Objects

SNMP Device objects

3.3.2 Default Schedule

By default, this script runs every hour.

3.3.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
SNMP Parameters	
SNMP port number	Specify the UDP port number on the remote SNMP device to which you want to send SNMP requests. The default is 161.
SNMP retries	Set the number of retries to attempt if a timeout occurs on an SNMP request. The default is 2 retries.
SNMP timeout	Set the number of seconds to wait for a response before timing out an SNMP request. The default is 5 seconds.
Collect data for device uptime?	Set to **Yes** to collect data for use in graphs and reports. If data collection is enabled, returns the device uptime in hours. The default is unchecked.
Monitor host uptime or uptime of the network management portion of the system?	Specify whether you want to monitor the uptime of the SNMP device or its network management component. The default is Host uptime.
Event Notification	
Raise event if device has rebooted?	Set to **Yes** to raise an event if a device reboot is detected. The details of the event contain the retrieved data.
Event severity when device has rebooted	Set the severity level, from 1 to 40, to indicate the importance of an event in which a device reboots. The default is 15.
Raise event if SNMP timeout exceeded?	Set to **Yes** to raise an event if the SNMP `Get` request receives no response from the device, and all retries fail.
Event severity when SNMP timeout exceeded	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP timeout threshold is exceeded. The default is 15.
Raise event if SNMP Response error received?	Set to **Yes** to raise an event if an SNMP Response error is received from the device. The type of error is reported in the event details.
Event severity when SNMP Response error received	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP Response error is received. The default is 10.
Raise event if device uptime baseline established?	Set to **Yes** to raise an event when initial data values are retrieved, setting the baseline for comparison on the next retrieval.
Event severity when device uptime baseline established	Set the severity level, from 1 to 40, to indicate the importance of an event in which baseline is established. The default is 25.
Raise event if the host uptime is not available?	Set to Yes to raise an event when the host uptime is not available. The default is Yes.
Event severity when the host uptime is not available.	Set the severity level, from 1 to 40, to indicate the importance of an event in which the host uptime is not available. The default is 15.

3.4 InterfaceState

Use this Knowledge Script to monitor the state of all interfaces in a device. This script tracks the values of `ifAdminStatus` and `ifOperStatus` for each interface across job iterations. If the operational status of an interface changes, an event is raised indicating the time the change occurred. This script also verifies whether `ifAdminStatus` and `ifOperStatus` are in sync.

The script attempts to track interface state on all SNMP devices on which the script is run, regardless of whether any previous attempts failed. If failures do occur, successful retrievals on other devices are not discarded.

This script collects separate data streams for each interface in a device. The data value saved is an integer representing the current operational state of each interface.

3.4.1 Resource Objects

One or more SNMP Device objects

3.4.2 Default Schedule

By default, this script runs every hour.

3.4.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
SNMP Parameters	
List of interface indices	By default, all interfaces are monitored. However, supplying a list of SNMP interface indices can restrict monitoring to those specific interfaces. Indices must be a single integer value. For example, "1 4 7" monitors only interfaces with an SNMP interface index of 1, 4 and 7.
Maximum number of interfaces to monitor	Specify the maximum number of interfaces to monitor. The default is 100.
SNMP port number	Specify the UDP port number on the remote SNMP device to which you want to send SNMP requests. The default is 161.
SNMP retries	Set the number of retries to attempt if a timeout occurs on an SNMP request. The default is 2 retries.
SNMP timeout	Set the number of seconds to wait for a response before timing out an SNMP request. The default is 5 seconds.
Collect data for interface state?	Set to **Yes** to collect data for use in graphs and reports. If enabled, returns a data stream for each interface in the device indicating its state. The default is unchecked.
Event Notification	
Raise event if interface state mismatch detected?	Set to **Yes** to raise an event if an interface has values for `ifAdminStatus` and `ifOperStatus` that are not the same. The details of the event contain the retrieved data.

Parameter	How to Set It
Event severity when interface state mismatch detected	Set the event severity level, from 1 to 40, to indicate the importance of an event in which Interface State Mismatch exists. The default is 10.
Raise event if interface state is up?	Set to **Yes** to raise an event if an interface has transitioned to the Up state since the last Knowledge Script iteration. The details of the event contain the retrieved data.
Event severity when interface state is up	Set the severity level, from 1 to 40, to indicate the importance of an event in which the interface state is Up. The default is 25.
Raise event if interface state is down?	Set to **Yes** to raise an event if an interface has transitioned to the Down state since the last Knowledge Script iteration. The details of the event contain the retrieved data.
Event severity when interface state is down	Set the severity level, from 1 to 40, to indicate the importance of an event in which the interface state is Down. The default is 15.
Raise event if SNMP timeout exceeded?	Set to **Yes** to raise an event if the SNMP `Get` or `GetNext` request receives no response from the device, and all retries fail.
Event severity when SNMP timeout exceeded	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP timeout is exceeded. The default is 15.
Raise event if SNMP Response error received?	Set to **Yes** to raise an event if an SNMP Response error is received from the device. The type of error is reported in the event details.
Event severity when SNMP Response error received	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP Response error is received. The default is 10.
Raise event if interface state baseline established?	Set to **Yes** to raise an event when initial data values are retrieved, setting the baseline for comparison on the next retrieval.
Event severity when interface state baseline established	Set the severity level, from 1 to 40, to indicate the importance of an event in which baseline is established. The default severity level is 25.

3.5 RemoveMIBs

Use this Knowledge Script to remove one or more MIB files from an SNMP proxy agent computer. The specified files are deleted from the installation folder, which by default is *installationfolder*`\AppManager\bin\AMSnmpMIBs`.

Run this script when no other SNMP Knowledge Scripts are active on the target AppManager SNMP Toolkit installation because reloading the MIB tree can only be done when no SNMP sessions are active. If the RemoveMIBs Knowledge Script is unable to reload the MIB tree because of active SNMP sessions, the script can raise an appropriate event.

This script raises an event if the specified MIB files cannot be deleted, or an `ASN.1` compilation error occurs while reloading the MIB tree.

SNMP MIBs tend to be hierarchical. That is, one MIB is often dependent on another, from which it imports more generic data definitions. Therefore, delete a MIB only if it has no other dependent MIBs installed.

3.5.1 Resource Object

SNMP Proxy Agent computer

3.5.2 Default Schedule

By default, this script runs once.

3.5.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
List of MIB files to uninstall	Supply a list of MIB filenames to be deleted.
Reload MIB tree?	Set to **Yes** to reload the MIB tree after MIB files have been deleted. A MIB tree reload is only be attempted if all MIB files are deleted successfully.
MIB reload timeout	Specify how long this script should wait to try and reload the MIB tree. This can only be performed if no other Knowledge Scripts have active SNMP sessions. The default is 10 seconds.
Event Notification	
Raise event if uninstallation of MIBs succeeds?	Set to **Yes** to raise an event if all MIBs are deleted successfully. The details of the event contain the list of installed MIB files.
Event severity when uninstallation of MIBs succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which MIB uninstallation succeeds. The default is 25.
Raise event if uninstallation of MIBs fails?	Set to **Yes** to raise an event if any MIB files fail to delete. This can occur if any specified MIB filenames correspond to files not currently installed.
Event severity when uninstallation of MIBs fails	Set the severity level, from 1 to 40, to indicate the importance of an event in which MIB uninstallation fails. The default is 10.
Raise event if reloading of MIB tree succeeds?	Set to **Yes** to raise an event if all MIB files uninstall successfully and the MIB tree is reloaded successfully. This can only occur if the *Reload MIB tree?* parameter is set to Yes.
Event severity when reloading of MIB tree succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which MIB tree reload succeeds. The default is 25.
Raise event if reload MIB parser warnings received?	Set to **Yes** to raise an event if all MIB files uninstall successfully and reloading the MIB tree reports `ASN.1` parsing errors. This can only occur if the *Reload MIB tree?* parameter is set to Yes. Details of the parsing errors are in the event.
Event severity when reload MIB parser warnings received	Set the severity level, from 1 to 40, to indicate the importance of an event in which MIB reload warnings are received. The default is 15.
Raise event if reloading of MIB tree fails?	Set to **Yes** to raise an event if all MIB files uninstall successfully but reloading the MIB tree fails. This can occur if the *Reload MIB tree?* parameter is set to Yes and the *MIB Reload timeout* expires.
Event severity when attempt to reload MIB tree fails	Set the severity level, from 1 to 40, to indicate the importance of an event in which MIB tree reload fails. The default is 10.

3.6 SNMPTrap_Async

Use this Knowledge Script to check for SNMP traps forwarded from NetIQ SNMP Trap Receiver. This script raises an event when an SNMP trap is received and when Trap Receiver is unavailable or subsequently becomes available. In addition, this script generates data streams for Trap Receiver availability.

This script checks for SNMP traps in the MIB tree. You can add MIBs (management information bases) to the MIB tree. For more information, see the AddMIBs Knowledge Script.

In general, a trap receiver is an application that receives traps from SNMP agents. NetIQ SNMP Trap Receiver (Trap Receiver) receives SNMP traps, filters them, and then forwards the traps to AppManager.

To run this Knowledge Script, you must configure SNMP permissions in Security Manager. For more information, see Section 2.5, “Configuring SNMP Permissions,” on page 15.

3.6.1 Resource Object

SNMP_TrapReceiver

3.6.2 Default Schedule

By default, this script runs on an asynchronous schedule.

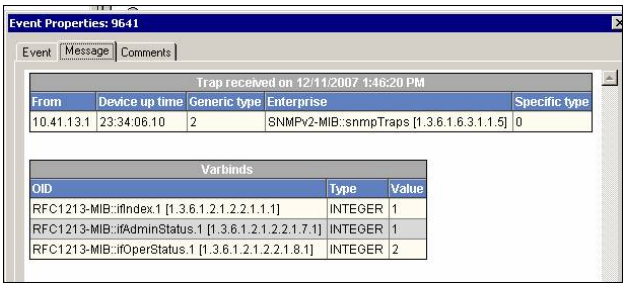
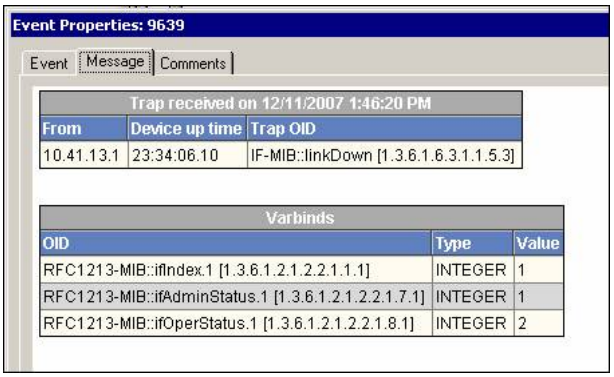
3.6.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Trap Filters	
List of trap OIDs	Type the OIDs (object identifiers) of the traps you want to monitor. You can type one OID or a list of OIDs. Separate multiple OIDs with a comma, for example: `1.3.6.1.2.1.2.2.1.1.1,1.3.6.1.2.1.2.2.1.7.1`

Full path to file with list of trap OIDs	If you have many OIDs to monitor, provide the full path to a file that contains a list of the OIDs. Each OID in the file should be on a separate line, for example:
	`1.3.6.1.2.1.2.2.1.1.1`
	`1.3.6.1.2.1.2.2.1.7.1`
	Because the file must be accessible from the AppManager agent, the path must be a local directory on the agent computer or a UNC path.
	Important If you type a UNC path, then the `netiqmc` service must be running as a user that has access to the path.

Event Notification	
Raise trap events?	Set to **Yes** to raise an event when a trap message is received from Trap Receiver. The default is Yes.

Parameter	How to Set It
Event severity when trap is received	Set the severity level, from 1 and 40, to indicate the importance of an event in which a trap is received. The default is 15.
Format trap data according to SNMP version	Select the version of SNMP whose formatting should be used for trap event messages. The data provided by each format is the same; only the layout is different. An event message in SNMP v1 format looks like this:  An event message in SMMP v2 format looks like this: 
Raise Trap Receiver availability events?	Set to **Yes** to raise an event when Trap Receiver becomes unavailable and when Trap Receiver becomes available once again. The default is Yes.
Event severity when Trap Receiver is unavailable	Set the severity level, from 1 to 40, to indicate the importance of an event in which Trap Receiver becomes unavailable. The default is 5.
Event severity when Trap Receiver becomes available	Set the severity level, from 1 to 40, to indicate the importance of an event in which Trap Receiver becomes available after being unavailable. The default is 25.
Data Collection	
Collect data for Trap Receiver availability?	Set to **Yes** to collect data for charts and reports. If enabled, data collection returns a “1” if Trap Receiver is available and a “0” if Trap Receiver is unavailable. The default is unchecked.
Interval for collecting Trap Receiver availability data	Specify the frequency with which the script collects Trap Receiver availability data. The default is every 5 minutes.

3.7 SyncGet

Use this Knowledge Script to perform an SNMP `Get` or `GetNext` operation for one or more SNMP attributes from one or more SNMP enabled devices. This script raises an event if retrieved values exceed the threshold you set.

The script attempts to get the specified attributes on all supplied SNMP devices, regardless of whether any previous attempts failed. If failures do occur, successful gets on other devices are not discarded.

This script can independently collect separate data streams for each SNMP device/SNMP attribute pairing. Thus, the total number of data streams collected is the number of devices the script has been run on in the TreeView, multiplied by the number of SNMP attributes provided.

NOTE: Supply either all-numeric SNMP attributes or all-string SNMP attributes, because you must choose either a numeric check or string check. SNMP attributes that are octet strings, OIDs, or IP addresses are considered to be string attributes.

If one or more numerical conversions are selected, they are performed in the following order: Multiplication, Division, Delta and Percentage. If Delta and Division are both selected, integer division is performed and any remainder is discarded. If Delta is not selected, real-number division is performed.

Values reported in SNMP Success events show the results of multiplication and division conversions, but not delta and percentage conversions, as these are performed after the Success event has been raised. The final result of all conversions is shown in any threshold events, or in the data points if data is collected.

This script can be run at intervals to periodically poll SNMP attributes. However, if delta calculations are being performed on the retrieved values, the script interval should not be less than one minute, because the accuracy of delta calculations is time dependent and may not produce reliable results at shorter intervals. This is due to many factors including the accuracy of AppManager job scheduling and network traffic delay. The SyncPoll script should be used to accurately poll at intervals of less than one minute.

By default, this script retrieves `sysUpTime.0` from the device, converts the retrieved value to “Days”, and reports the retrieved value in a success event.

3.7.1 Resource Objects

One or more SNMP Device objects

3.7.2 Default Schedule

By default, this script runs once.

3.7.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Get Parameters	

Parameter	How to Set It
SNMP ODE/OIDs	Supply a list of SNMP Attribute ODEs and/or OIDs. Use ODEs only if the SNMP proxy agent computer has the corresponding MIB available. The default is `sysUpTime.0`.
SNMP port number	Specify the UDP port number on the remote SNMP device to which you want to send SNMP requests. The default is 161.
SNMP retries	Specify the number of retries to attempt if a timeout occurs on an SNMP request. The default is 2 retries.
SNMP timeout	Specify the number of seconds to wait for a response before timing out an SNMP request. The default is 5 seconds.
Perform GetNext instead of Get?	Set to **Yes** to perform an SNMP `GetNext` operation instead of a `Get`. The default is to perform a `Get` operation.
OID Value Check	
Collect data for OID value?	Set to **Yes** to collect data for use in graphs and reports. If enabled, returns a data stream for each SNMP device/SNMP attribute pair. The default is unchecked.
Calculated units	If this script is performing numeric conversions, a name for the resulting units calculated can be entered here. If nothing is supplied for this parameter, the default units are the name of the data type retrieved. The default is “Days”.
Calculate delta for numeric OID value?	Set to **Yes** to specify that the retrieved values are numeric and a delta should be calculated. The difference between the new value and the previous value is calculated. Normally this is used to monitor growth of SNMP counter values between iterations. The default is unchecked. Delta calculations are not normalized. Thus, it is usually necessary to perform a division conversion on delta calculations to convert to the desired time units. For example, if the growth of an SNMP counter is being tracked by doing a delta calculation at a script interval of one minute, it is necessary to divide by 60 to track the growth of the counter on a per-second basis. If set to **Yes**, **integer math** (not **floating point math**) is used; thus, any remainder is discarded. **NOTE:** Do not enable this option if the script is running at intervals of less than one minute. The results may not be reliable.
Convert string to numeric value (Advanced)?	Set to **Yes** to convert the string values returned by SNMP to numeric value. The Knowledge Script converts the string value to number if the string represents a valid numeric data. Otherwise an event is raised. The numeric data allows you to perform the mathematical operations as provided by the Knowledge Script. The default is No.
Process OID’s Individually (Advanced)?	Set to **Yes** to see individual success or failure events of each OID. The default is No.
Use multiplier for numeric OID value?	Set to **Yes** to specify that the retrieved values are numeric and should be multiplied by the value specified for the next parameter. The default is unchecked.
Multiplier value	If the *Use multiplier for numeric OID value?* parameter is enabled, the retrieved SNMP attributes are multiplied by this value before being reported. The default is 1.

Parameter	How to Set It
Use divisor for numeric OID value?	Set to **Yes** to specify that the retrieved values are numeric and should be divided by the value specified for the next parameter. If you enabled the *Calculate delta for numeric OID value* parameter, integer division is performed and any remainder is discarded. Otherwise, values are converted to real numbers to perform the division and retain the precision of any remainder. The default is Yes.
Divisor value	If the *Use divisor for numeric OID value?* parameter is enabled, the retrieved SNMP attributes are divided by this value before being reported. The default is `8640000`.
Calculate percentage of numeric OID value?	Set to **Yes** to specify that the retrieved values are numeric and should be converted to a percentage of the maximum value supplied in the next parameter. Values are converted to real numbers to perform the percentage calculation and retain the precision of any remainder. Calculated values are restricted to a real number between 0% and 100%. The default is unchecked.
Maximum value	If the *Calculate percentage of numeric OID value?* parameter is enabled, the retrieved SNMP attributes are converted to a percentage of the maximum value entered here. The default is 100.
Raise event when maximum threshold exceeded?	Set to **Yes** to specify that the retrieved values should be compared against the maximum threshold value supplied in the next parameter. If enabled, the check is performed after any mathematical conversions that may have been selected above. The default is unchecked.
Threshold -- Maximum OID value (post-calculation)	This parameter has no effect unless *Raise event when maximum threshold exceeded?* is enabled. For numeric attributes, specify the maximum threshold value. If this value is exceeded, an event is raised. The threshold value is restricted to whole integer values. The default is 1000.
Raise event when minimum threshold not met?	Set to **Yes** to specify that the retrieved values should be compared against the minimum threshold value supplied in the next parameter. If enabled, the check is performed after any mathematical conversions that may have been selected above. The default is unchecked.
Threshold -- Minimum OID value (post-calculation)	This parameter has no effect unless *Raise event if threshold not met?* is enabled. For numeric attributes, specify the minimum threshold value. If this threshold is not met, an event is raised. The threshold value is restricted to whole integer values. The default is 100.
Raise event when returned value equals “Numeric value”?	Set to **Yes** to raise an event when the retrieved numeric values equal the value supplied in the *Numeric value* parameter. If enabled, the check is performed after any mathematical conversions that may have been selected above.
Check for “not equal” instead of “equals”?	Use this parameter to change the function of the previous parameter. To raise an event when the retrieved numeric values do not equal the value supplied in the *Numeric value* parameter, set to **Yes** and ensure that *Raise event when returned value equals “Numeric value”?* is also set to **Yes**.

Parameter	How to Set It
Numeric value	Use this parameter only if *Raise event when returned value equals "Numeric value"?* is enabled. An event is raised if the retrieved SNMP values equal the value you enter here. If *Check for "not equal" instead of "equals"?* is also enabled, an event is raised if the retrieved SNMP values are not equal to the value you enter here. This threshold is restricted to whole integer values. The default is 0.
Raise event when returned string equals "String value"?	Set to **Yes** to raise an event when the retrieved values equal the value supplied in the *String value* parameter. Use this parameter to retrieve SNMP attributes that are strings, octet strings, OIDs, or IP addresses.
Check for "not equal" instead of "equals"?	Use this parameter to change the function of the previous parameter. To raise an event when the retrieved values do not equal the value supplied in the *String value* parameter, set to **Yes** and ensure *Raise event when returned string equals "String value"?* is also set to **Yes**.
Do case-insensitive comparison?	If *Raise event when returned string not equal to value?* is enabled, set to **Yes** to specify that the retrieved values should be compared without regard to character case.
String value	Use this parameter only If *Raise event when returned string equals "String value"?* is enabled. An event is raised if the retrieved string equals the value you enter here. If *Check for "not equal" instead of "equals"?* is also enabled, an event is raised if the retrieved string does not equal the value you enter here. The default is "`String Value`".
Event severity when OID value violates check	Set the severity level, from 1 to 40, to indicate the importance of an event in which a threshold is crossed or an equality/inequality check fails. The default is 5.
Event Notification	
Raise event if SNMP operation succeeds?	Set to **Yes** to raise an event if the SNMP `Get` or `GetNext` operation is successful. The details of the event contain the retrieved data.
Event severity when SNMP operation succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP operation succeeds. The default is 25.
Raise event if SNMP timeout exceeded?	Set to **Yes** to raise an event if the SNMP `Get` or `GetNext` request receives no response from the device, and all retries fail.
Event severity when SNMP timeout exceeded	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP timeout period is exceeded. The default is 15.
Raise event if SNMP Response error received?	Set to **Yes** to raise an event if an SNMP Response error is received from the device. The type of error is reported in the event details.
Event severity when SNMP Response error received	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP Response error is received. The default is 10.
Raise event if delta baseline established?	This event can be raised only if the *Calculate delta for numeric OID value?* parameter is enabled. Set to **Yes** to raise an event when the initial value is retrieved, setting the baseline for a difference calculation on the next job iteration.

Parameter	How to Set It
Event severity when delta baseline established	Set the severity level, from 1 to 40, to indicate the importance of an event in which a delta baseline is established. The default is 25.

3.8 SyncGetTable

Use this Knowledge Script to perform an SNMP table walk along specified columns of an SNMP table. This script raises an event if retrieved values exceed the threshold you set.

NOTE: Because this script walks an SNMP table, do not supply an index value on the ODE/OIDs. Supplying just the attribute name (for example, “ifDescr”) is normally sufficient. If only a portion of the table is to be walked, a parameter is available to specify the subset of table indices to walk.

The table walk is performed with iterative `GetNext` operations. As soon as any of the attributes walk beyond the end of the table, or the table indices become out of sync, the table walk terminates. If a table is fully populated, all attributes or table columns walk beyond the end of the table on the same `GetNext` operation. However, if the table has missing values, the table indices become out of sync as soon as a missing value is reached. A table walk is terminated when the first missing value is detected.

This script attempts to walk the specified attributes on all SNMP devices on which this script has been run, regardless of whether any previous attempts failed. If failures do occur, successful walks on other devices are not discarded.

This script individually collects separate data streams for each SNMP device/data OID pairing. Thus, the total number of data streams collected is the number of devices the Knowledge Script is run on multiplied by the number of data OIDs and the number of rows in the table on each device.

NOTE: Supply either all-numeric SNMP attributes or all-string SNMP attributes because you must choose either a numeric check or string check. SNMP attributes that are octet strings, OIDs, or IP addresses are considered to be string attributes.

If one or more numerical conversions are selected, they are performed in the following order: Multiplication, Division, Delta and Percentage. If Delta and Division are both enabled, integer division is performed, and any remainder is discarded. If Delta is not enabled, real-number division is performed.

Values reported in SNMP Success events show the results of multiplication and division conversions, but not delta or percentage conversions, as these are performed after the Success event has been raised. The final result of all conversions is shown in any threshold-crossing events or reflected in the data streams if data is collected.

This script can be run at intervals to periodically poll SNMP tables. However, if delta calculations are being performed on the retrieved values, the script interval should not be less than one minute, because the accuracy of delta calculations is time dependent and may not produce reliable results at shorter intervals. This unreliability is due to many factors including the accuracy of AppManager job scheduling and network traffic delay. The SyncPollTable Knowledge Script should be used to accurately poll at intervals of less than one minute.

By default, this script retrieves the operational status for all interfaces on a device.

3.8.1 Resource Objects

SNMP Device objects

3.8.2 Default Schedule

By default, this script runs once.

3.8.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Get Table Parameters	
Descriptive ODE/OIDs	Supply a list of SNMP attribute ODEs and/or OIDs. These are descriptive attributes only, which are reported in event and data details for reference purposes. No processing is performed on the retrieved values. The attributes chosen should uniquely identify the retrieved row of an SNMP Table. The default is "`ifIndex, ifDescr`", which identifies the row number and name of a communications interface in the device.
Data ODE/OIDs	Supply a list of SNMP attribute ODEs and/or OIDs. ODEs can only be used if the SNMP proxy agent computer has the corresponding MIB available. The default is `ifOperStatus`.
Optional table indices	By default, the entire SNMP table is walked. However, supplying a list of the table indices to be walked can restrict the walk. Indices can be a single integer value, or multiple integer values separated by dots; just like a numeric OID value but without a leading dot. For example, "1.10.42.1.47" is a valid table index.
Maximum number of table rows to get	Specify the maximum number of table rows to be retrieved. The default is 100.
SNMP port number	Specify the UDP port number on the remote SNMP device to which you want to send SNMP requests. The default is 161.
SNMP retries	Set the number of retries to attempt if a timeout occurs on an SNMP request. The default is 2 retries.
SNMP timeout	Set the number of seconds to wait for a response before timing out an SNMP request. The default is 5 seconds.
OID value check	
Collect data for OID value?	Set to **Yes** to collect data for graphs and reports. The data is stored in the AppManager repository. When enabled, returns a data stream for each SNMP device/SNMP attribute pair. The default is unchecked.
Calculated units	If this script is performing numeric conversions, a name for the resulting units calculated can be entered here. If nothing is supplied for this parameter, the default units are the name of the data type retrieved. The default is blank.

Parameter	How to Set It
Calculate delta for numeric OID value?	Set to **Yes** to specify that the retrieved values are numeric and the difference between the new value and the previous value should be calculated. Normally this is used to monitor growth of SNMP counter values between iterations. The default is unchecked. Delta calculations are not normalized. Thus, it is usually necessary to perform a division conversion on delta calculations to convert to the desired time units. For example, if the growth of an SNMP counter is being tracked by doing a delta calculation at a script interval of one minute, it is necessary to divide by 60 to track the growth of the counter on a per second basis. If set to **Yes**, **integer math** (not **floating point math**) is used, thus any remainder is discarded. **NOTE:** Do not enable this option if the script is running at intervals of less than one minute. The results may not be reliable.
Use multiplier for numeric OID value?	Set to **Yes** to specify that the retrieved values are numeric and should be multiplied by the value in the next parameter. By default, multiplication is not performed on retrieved values.
Multiplier value	If the *Use multiplier for numeric OID value?* parameter is enabled, the retrieved SNMP attributes are multiplied by this value before being reported. The default is 1.
Use divisor for numeric OID value?	Set to **Yes** to specify that the retrieved values are numeric and should be divided by the value in the next parameter. If the *Calculate delta for numeric OID value* parameter has also been enabled, integer division is performed and any remainder is discarded. Otherwise, values are converted to real numbers to perform the division and retain the precision of any remainder. The default is unchecked.
Divisor value	If the *Use divisor for numeric OID value?* parameter is enabled, the retrieved SNMP attributes are divided by this value before being reported. The default is 1.
Calculate percentage of numeric OID value?	Set to **Yes** to specify that the retrieved values are numeric and should be converted to a percentage of the *Maximum value*. Values are converted to real numbers to perform the percentage calculation and retain the precision of any remainder. Calculated values are restricted to a real number between 0% and 100%. The default is unchecked.
Maximum value	If the *Percentage of numeric OID value?* parameter is enabled, the retrieved SNMP attributes are converted to a percentage of the maximum value entered here. The default is 100.
Raise event when maximum threshold exceeded?	Set to **Yes** to specify that the retrieved values should be compared against the maximum threshold value supplied in the next parameter. If enabled, the check is performed after any mathematical conversions that may have been selected above. The default is unchecked.
Threshold -- Maximum OID value (post-calculation)	This parameter has no effect unless *Raise event if threshold exceeded?* is enabled. For numeric attributes, specify the maximum threshold value. If this value is exceeded, an event is raised. The threshold value is restricted to whole integer values. The default is 1000.

Parameter	How to Set It
Raise event when minimum threshold not met?	If *Raise event if threshold not met?* is enabled, set to **Yes** to specify that the retrieved values should be compared against the minimum threshold value supplied in the next parameter. If enabled, the check is performed after any mathematical conversions that may have been selected above. The default is unchecked.
Threshold -- Minimum OID value (post-calculation)	If the *Raise event if threshold not met?* parameter is enabled, specify a minimum value for numeric attributes after any calculations have been performed. If this threshold is not met, an event is raised. The threshold value is restricted to whole integer values. The default is 100.
Raise event when returned value equals "Numeric value"?	Set to **Yes** to raise an event when the retrieved numeric values equal the value supplied in the *Numeric value* parameter. If enabled, the check is performed after any mathematical conversions that may have been selected above.
Check for "not equal" instead of "equals"?	Use this parameter to change the function of the previous parameter. To raise an event when the retrieved numeric values do not equal the value supplied in the *Numeric value* parameter, set to **Yes** and ensure that *Raise event when returned value equals "Numeric value"?* is also set to **Yes**.
Numeric value	Use this parameter only if *Raise event when returned value equals "Numeric value"?* is enabled. An event is raised if the retrieved SNMP values equal the value you enter here. If *Check for "not equal" instead of "equals"?* is also enabled, an event is raised if the retrieved SNMP values do not equal the value you enter here. This threshold is restricted to whole integer values. The default is 0.
Raise event when returned string equals "String value"?	Set to **Yes** to raise an event when the retrieved values equal the value supplied in the *String value* parameter. Use this parameter to retrieve SNMP attributes that are strings, octet strings, OIDs, or IP addresses.
Check for "not equal" instead of "equals"?	Use this parameter to change the function of the previous parameter. To raise an event when the retrieved values do not equal the value supplied in the *String value* parameter, set to **Yes** and ensure *Raise event when returned string equals "String value"?* is also set to **Yes**.
Do case-insensitive comparison?	If *Raise event when returned string not equal to value?* is enabled, set to **Yes** to specify that the retrieved values should be compared without regard to character case.
String value	Use this parameter only If *Raise event when returned string equals "String value"?* is enabled. An event is raised if the retrieved string equals the value you enter here. If *Check for "not equal" instead of "equals"?* is also enabled, an event is raised if the retrieved string does not equal the value you enter here. The default is "`String Value`".
Event severity when OID value violates check	Set the event severity level, from 1 to 40, to indicate the importance of an event in which a threshold is crossed or an equality/inequality check fails. The default is 5.
Event Notification	

Parameter	How to Set It
Raise event if SNMP operation succeeds?	Set to **Yes** to raise events if the SNMP `Get` or `GetNext` operation is successful. The details of the event contain the retrieved data.
Event severity when SNMP operation succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP operation succeeds. The default is 25.
Raise event if SNMP timeout exceeded?	Set to **Yes** to raise events if the SNMP `Get` or `GetNext` request receives no response from the device, and all retries fail.
Event severity when SNMP timeout exceeded	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP timeout period is exceeded. The default is 15.
Raise event if SNMP Response error received?	Set to **Yes** to raise an event if an SNMP Response error is received from the device. The type of error is reported in the event details.
Event severity when SNMP Response error received	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP Response error is received. The default is 10.
Raise event if delta baseline established?	If *Calculate delta for numeric OID value?* has been enabled, set to **Yes** to raise an event when the initial value is retrieved, setting the baseline for a difference calculation on the next retrieval.
Event severity when delta baseline established?	Set the severity level, from 1 to 40, to indicate the importance of an event in which a delta baseline is established. The default is 25.

3.9 SyncPoll

Use this Knowledge Script to poll SNMP attributes on a Device at short time intervals during each Knowledge Script iteration. Only numeric SNMP attributes may be polled. For each set of polled values, this script computes a minimum, maximum, average and standard deviation. This script raises an event if computed values exceed the threshold you set.

When this script is run on multiple devices, they are polled successively, and not simultaneously. To poll devices simultaneously, a different job must be created for each device.

NOTE: The number of Polling attempts multiplied by the Polling interval and then multiplied by the number of devices on which the Knowledge Script is run must not exceed the time interval between Knowledge Script iterations. Attempting to do so causes the script to abort, as the polling would not be able to complete before the next iteration is due to execute.

This script continues polling the SNMP device regardless of whether any previous attempts failed. At least two polling attempts must succeed for any meaningful data to be calculated.

This script individually collects separate data streams for each SNMP attribute. By default, this script calculates the percent bandwidth utilization for the first interface listed in the `ifTable` for the SNMP device, assuming the speed of this interface is standard Ethernet of 100 Megabits per second. If the speed of this interface is different, the values returned by the default settings are not valid.

If one or more numerical conversions are selected, they are performed in the following order: Delta, Multiplication, Division and Percentage.

When polling the growth of SNMP counter values using the Delta option, all values are normalized on a per second basis, regardless of the length of the polling interval. For example, this script can be used to calculate the Kilobytes per second flowing through an interface by polling `ifInOctets` and `ifOutOctets` and dividing the returned values by 1024. The values reported are Kilobytes per second regardless of the length of the polling interval.

3.9.1 Resource Objects

SNMP Device objects

3.9.2 Default Schedule

By default, this script runs once.

3.9.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Polling Parameters	
SNMP ODE/OIDs	Supply a list of SNMP attribute ODEs and/or OIDs. ODEs can only be used if the SNMP proxy agent computer has the corresponding MIB available. The default is: "`ifInOctets.1, ifOutOctets.1`".
Polling interval	Specify the time interval between polling attempts during each Knowledge Script iteration. The default is 5 seconds.
Polling attempts	Specify the number of polling attempts to perform during each Knowledge Script iteration. The default is 12. Combined with the default interval of 5 seconds, the script by default polls the SNMP device for 1 minute. The minimum value is 2.
SNMP port number	Specify the UDP port number on the remote SNMP device to which you want to send SNMP requests. The default is 161.
SNMP timeout	Specify the number of seconds to wait for a response before timing out an SNMP request. The default is 5 seconds.
Polled Values Check	
Collect data for polled values?	Set to **Yes** to collect data for use in graphs and reports. The data is stored in the AppManager repository. If enabled, returns a data stream for each SNMP device/SNMP attribute pair. The default is unchecked.
Polling calculation type	For each set of polled values, the Average, Minimum, Maximum and Standard Deviation are calculated. Use this parameter to select which calculated value serves as the data point.
Calculated units	Specify a name to identify the units being polled and calculated by this script. The default is “Percent”.

Parameter	How to Set It
Calculate delta for polled value?	Set to **Yes** to specify that the retrieved values should be considered a delta from the value retrieved by the previous polling attempt. The difference between the new value and the previous value is calculated. Normally this is used to monitor growth of SNMP counter values between polling attempts. By default, a delta calculation is performed. When set to **Yes**, **integer math** (not **floating point math**) is used; thus, any remainder is discarded. **NOTE:** When this parameter is enabled, the script reports one less polling attempt than was specified for *Polling attempts* because the first polling attempt is used to set a baseline for the delta calculations to follow.
Use multiplier for polled value?	Set to **Yes** to specify that the retrieved values are numeric and should be multiplied by the value in the next parameter. The default is unchecked.
Multiplier value	If the *Use multiplier for numeric OID value?* parameter is enabled, the retrieved SNMP attributes are multiplied by this value before being reported. The default is 1.
Use divisor for polled value?	Set to **Yes** to specify that the retrieved values should be divided by the value in the next parameter. Values are converted to real numbers to perform the division and retain the precision of any remainder. The default is unchecked.
Divisor value	If the *Use divisor for numeric OID value?* parameter is enabled, the retrieved SNMP attributes are divided by this value before being reported. The default is 1.
Calculate percentage of polled value?	Set to **Yes** to specify that the retrieved values should be converted to a percentage of the maximum value supplied in the next parameter. Values are converted to real numbers to perform the percentage calculation and retain the precision of any remainder. Calculated values are restricted to a real number between 0% and 100%. The default is Yes.
Maximum value	If the *Percentage of numeric OID value?* parameter is enabled, the retrieved SNMP attributes are converted to a percentage of the maximum value entered here. The default is 12500000 bytes (or 100 Megabits) per second.
Raise event when maximum threshold exceeded?	Set to **Yes** to specify that the retrieved values should be compared against the maximum threshold value supplied in the next parameter. If enabled, the check is performed after any mathematical conversions that may have been selected above. The default is unchecked.
Threshold -- Maximum OID value (post-calculation)	This parameter has no effect unless *Raise event when maximum threshold exceeded?* is enabled. For numeric attributes, specify the maximum threshold value. If this value is exceeded by a retrieved value after any selected calculations have been performed, an event is raised. Polling threshold values are real numbers. The default is 90.
Raise event when minimum threshold not met?	Set to **Yes** to specify that the retrieved values should be compared against the minimum threshold value supplied in the next parameter.If enabled, the threshold check is done after any mathematical conversions that may have been selected above. The default is unchecked.

Parameter	How to Set It
Threshold -- Minimum OID value (post-calculation)	This parameter has no effect unless *Raise event when minimum threshold not met?* is enabled. Specify a minimum threshold value. If the retrieved value fails to meet the threshold value after any calculations have been performed, an event is raised. Polling threshold values are real numbers. The default is 0.
Event severity when polled value violates check	Set the severity level, from 1 to 40, to indicate the importance of an event in which a threshold is crossed or an equality/inequality check fails. The default is 5.
Event Notification	
Raise event if SNMP operation succeeds?	Set to **Yes** to raise an event if the SNMP `Get` or `GetNext` operation is successful. The details of the event contain the retrieved data.
Event severity when SNMP operation succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP operation succeeds. The default is 25.
Raise event if SNMP timeout exceeded?	Set to **Yes** to raise an event if the SNMP `Get` or `GetNext` request receives no response from the device, and all retries fail.
Event severity when SNMP timeout exceeded	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP timeout period is exceeded. The default is 15.
Raise event if SNMP Response error received?	Set to **Yes** to raise an event if an SNMP response error is received from the device. The type of error is reported in the event details.
Event severity when SNMP Response error received	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP Response error is received. The default is 10.

3.10 SyncPollTable

Use this Knowledge Script to perform an SNMP table walk along specified columns of an SNMP table. The retrieved table on the device is then polled at short time intervals at every script iteration. Only numeric attributes are polled. For each table row that is polled, this script computes a minimum, maximum, average and standard deviation. This script raised an event if the computed values exceed the threshold you set.

NOTE: Because this script walks an SNMP table, it is normally not necessary to supply an index value on the ODE/OIDs. Supplying just the attribute name (for example, “ifDescr”) is normally sufficient. If only a portion of the table is to be walked, a parameter is available to specify the subset of table indices to walk.

The table walk is performed with iterative `GetNext` operations. As soon as any attribute walks beyond the end of the table, or the table indices become out of sync, the table walk terminates. If a table is fully populated, all attributes (or table columns) walk beyond the end of the table on the same `GetNext` operation. However, if the table has missing values, the table indices become out of sync as soon as a missing value is reached. A table walk is terminated when the first missing value is detected.

When this script is run on multiple devices, they are polled successively, not simultaneously. To poll devices simultaneously, create a different job for each device.

NOTE: The number of Polling attempts multiplied by the Polling interval and then multiplied by the number of devices on which the script is run must not exceed the time interval between script iterations. If this value does exceed the interval, the script job aborts because the polling would not be able to complete before the next Knowledge Script job is due to execute.

The script continues polling, regardless of whether any previous attempts failed. At least two polling attempts must succeed in order to report meaningful data.

This script collects separate data streams for each SNMP table row. Thus, the number of data streams is the number of SNMP data OIDs, multiplied by the number of rows in the table, and multiplied by the number of devices on which the script is run. By default, this script polls how many Kilobytes per second are flowing through each interface listed in the `ifTable` for the SNMP device.

If one or more numerical conversions are selected, they are performed in the following order: Delta, Multiplication, Division and Percentage. When polling growth of SNMP counter values using the Delta option, all values are normalized on a per second basis, regardless of the length of the polling interval. For example, as stated above by default this script calculates the Kilobytes per second flowing through all interfaces in the device. The values reported are Kilobytes per second regardless of the length of the polling interval.

3.10.1 Resource Object

Host System Folder running an AppManager agent

3.10.2 Default Schedule

By default, this script runs once.

3.10.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
Polling Table Parameters	
Descriptive ODE/OIDs	Supply a list of SNMP Attribute ODEs and/or OIDs. These are descriptive attributes only which are reported in event and data details for reference purposes, and no processing is done on the retrieved values. The attributes chosen should uniquely identify the retrieved row of an SNMP table. The default is "`ifIndex, ifDescr`", which identifies the row number and name of a communications interface in the device.
Data ODE/OIDs	Supply a list of SNMP Attribute ODEs and/or OIDs. ODEs can only be used if the SNMP Toolkit module on the proxy agent computer has the corresponding MIB available. The default is "`ifInOctets,IfOutOctets`".
Optional table indices	By default, the entire SNMP table is walked. However, supplying a list of the table indices to be walked can restrict the walk. Indices can be a single integer value, or multiple integer values separated by dots; just like a numeric OID value but without a leading dot. For example, "1.10.42.1.47" is a valid table index.
Maximum number of table rows to poll	Specify the maximum number of table rows to be polled. The default is 100.

Parameter	How to Set It
Polling interval	Specify the time interval between polling attempts during each Knowledge Script iteration. The default is 5 seconds.
Polling attempts	Specify the number of polling attempts to perform during each Knowledge Script iteration. The default is 12. Combined with the default interval of 5 seconds, the script by default polls the SNMP device for 1 minute. The minimum value is 2.
SNMP port number	Specify the UDP port number on the remote SNMP device to which you want to send SNMP requests. The default is 161.
SNMP timeout	Specify the number of seconds to wait for a response before timing out an SNMP request. The default is 5 seconds.
Polled Values Check	
Collect data for polled values?	Set to **Yes** to collect data for use in graphs and reports. The data is stored in the AppManager repository. When enabled, returns a data stream for each SNMP device/SNMP attribute pair. The default is unchecked.
Polling calculation type	For each set of polled values, the Average, Minimum, Maximum and Standard Deviation are calculated. Use this parameter to select which calculated value serves as the data point for the retrieved values.
Calculated units	Specify a name to identify the units being polled and calculated by this script. The default is “Kbytes/Sec”.
Calculate delta for polled value?	Set to **Yes** to specify that the retrieved values should be considered a delta from the value retrieved by the previous polling attempt. The difference between the new value and the previous value is calculated. Normally this is used to monitor growth of SNMP counter values between polling attempts. The default is Yes. When set to **Yes**, **integer math** (not **floating point math**) is used, thus any remainder is discarded. **NOTE:** When you enable this parameter, this script reports one less polling attempt than was specified for *Polling attempts*, because the first polling attempt is used to set a baseline for the delta calculations to follow.
Use multiplier for polled value?	Set to **Yes** to specify that the retrieved values are numeric and should be multiplied by the value in the next parameter. The default is unchecked.
Multiplier value	If *Use multiplier for numeric OID value?* is enabled, the retrieved SNMP attributes are multiplied by this value before being reported. The default is 1.
Use divisor for polled value?	Set to **Yes** to specify that the retrieved values are numeric and should be divided by the value in the next parameter. Values are converted to real numbers to perform the division and retain the precision of any remainder. The default is unchecked.
Divisor value	If *Use divisor for numeric OID value?* is enabled, the retrieved SNMP attributes are divided by this value before being reported. The default is 1.
Calculate percentage of polled value?	Set to **Yes** to specify that the retrieved values are numeric and should be converted to a percentage of the maximum value supplied in the next parameter. Values are converted to real numbers to perform the percentage calculation and retain the precision of any remainder. Calculated values are restricted to a real number between 0% and 100%. The default is unchecked.

Parameter	How to Set It
Maximum value	If *Calculate percentage of numeric OID value?* is enabled, the retrieved SNMP attributes are converted to a percentage of the maximum value entered here. The default is 100.
Raise event when maximum threshold exceeded?	Set to **Yes** to specify that the retrieved values should be compared against the maximum threshold value supplied in the next parameter. If enabled, the check is performed after any mathematical conversions that may have been selected above. The default is unchecked.
Threshold -- Maximum OID value (post-calculation)	If *Raise event when maximum threshold exceeded?* is selected, specify a maximum threshold value. If a retrieved value exceeds the threshold, an event is raised. Polling threshold values are real numbers. The default is 1000.
Raise event when minimum threshold not met?	Set to **Yes** to specify that the retrieved values should be compared against the minimum threshold value supplied in the next parameter. If enabled, the check is performed after any mathematical conversions that may have been selected above. The default is unchecked.
Threshold -- Minimum OID value (post-calculation)	If *Raise event when minimum threshold not met?* is enabled, specify a minimum threshold value. If a retrieved value fails to meet this threshold, an event is raised. Polling threshold values are real numbers. The default is 100.
Event severity when polled value violates check	Set the event severity level, from 1 to 40, to indicate the importance of an event in which a threshold is crossed or an equality/inequality check fails. The default is 5.
Event Notification	
Raise event if SNMP operation succeeds?	Set to **Yes** to raise an event if the SNMP `Get` or `GetNext` is successful. The details of the event contain the retrieved data.
Event severity when SNMP operation succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP operation is successful. The default is 25.
Raise event if SNMP timeout exceeded?	Set to **Yes** to raise an event if the SNMP `Get` or `GetNext` request receives no response from the device, and all retries fail.
Event severity when SNMP timeout exceeded	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP timeout interval is exceeded. The default is 15.
Raise event if SNMP Response error received?	Set to **Yes** to raise an event if an SNMP Response error is received from the device. The type of error is reported in the event details.
Event severity when SNMP Response error received	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP Response error is received. The default is 10.

3.11 SyncSet

Use this Knowledge Script to set one or more SNMP attributes on one or more SNMP-enabled devices to the specified values. The values can be of different types. However, so that the value types can be determined, the SNMP proxy agent computer that executes the script must have the MIB available for the specified attributes even if numeric OIDs are supplied. If the MIB for the specified

attributes is not available, an SNMP `Failure` event is raised. Although requiring the MIB to be installed is a restriction, it does allow the flexibility to easily set multiple SNMP attributes of different types from a single script.

This script attempts to set the specified attributes on all supplied SNMP devices, regardless of whether any or all of the attempts fail. If failures do occur, successful sets on other devices cannot be reversed or backed out.

Collected data for this script is a Boolean value that specifies whether the SNMP `Set` operation failed or succeeded.

3.11.1 Resource Objects

SNMP Device objects

3.11.2 Default Schedule

By default, this script runs once.

3.11.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
SNMP ODE/OIDs	Supply a list of ODEs and/or OIDs. The default is "`sysContact.0`".
SNMP values (comma-separated)	Supply a comma-separated list of SNMP attribute values. As some SNMP values could be text strings, commas must be used as separators rather than spaces. The list must contain the same number of items as the *SNMP ODE/OIDs* parameter, and should be in the order corresponding to their respective attribute. The default is "Your System Administrator".
SNMP port number	Specify the UDP port number on the remote SNMP device to which you want to send SNMP requests. The default is 161.
SNMP retries	Specify the number of retries to attempt if a timeout occurs on an SNMP request. The default is 0 retries.
SNMP timeout	Specify the number of seconds to wait for a response before timing out an SNMP request. The default is 5 seconds.
Set success	
Collect data for Set success?	Set to **Yes** to collect data for use in graphs and reports. The data is stored in the AppManager repository. When enabled, returns a data stream containing a Boolean value representing whether the `Set` request succeeded or failed.
Raise event if Set success not equal to threshold?	Set to **Yes** to raise an event if the result of the `Set` request is not equal to the threshold. The default is Yes.

Parameter	How to Set It
Threshold – Set success	Set this value to one of the following: ◆ 1 -- the `Set` request succeeds ◆ 0 -- the `Set` request does not succeed. The default is 1 (success).
Event severity when Set success not equal to threshold	Set the severity level, from 1 to 40, to indicate the importance of an event in which Set success is not equal to the threshold you set. The default is 15.
Event Notification	
Raise event if SNMP operation succeeds?	Set to **Yes** to raise an event if the SNMP `Set` operation is successful. The details of the event contain the list of attributes you set.
Event severity when SNMP operation succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP `Set` operation succeeds. The default is 25.
Raise event if SNMP timeout exceeded?	Set to **Yes** to raise an event if the SNMP `Set` request receives no response from the device, and all retries fail.
Event severity when SNMP timeout exceeded	Set the severity level, from 1 to 40, to indicate the importance of an event in which the SNMP timeout interval is exceeded. The default is 15.
Raise event if SNMP Response error received?	Set to **Yes** to raise an event if an SNMP Response error is received from the device. The type of error is reported in the event details.
Event severity when SNMP Response error received	Set the severity level, from 1 to 40, to indicate the importance of an event in which an SNMP Response error is received. The default is 10.