NetIQ® AppManager® for Session Initiation Protocol (SIP) Server

Management Guide

March 2014

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
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

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

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. 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 community.netiq.com.

1 Introducing AppManager for SIP Server

This chapter introduces AppManager for Session Initiation Protocol Server (AppManager for SIP Server). This module defines a *SIP server* as any VoIP server or device that reports call quality data using RFC6035 and other standard formats. These formats enable SIP phones and gateways to report the quality of every call made, allowing you to monitor end-to-end quality for your VoIP activity.

This chapter provides an overview of the module and describes how you can use AppManager for SIP Server to monitor Quality of Service (QoS) data and call data from a variety of SIP servers, such as Avaya Session Manager.

1.1 Features and Benefits

With this initial release of AppManager for SIP Server, you can:

- Receive, process, and alert on QoS packets coming from SIP servers, including Avaya Session Manager.
- Discover a server that uses SIP and discover resources for that server.
- Discover a SIP Server either by SNMP query or by manual configuration.
- Create a supplemental database that stores call detail records, including voice quality reports.
- Start SIP call data collection, and monitor the collection of data once started.
- Continuously monitor calls for quality metrics, including jitter, latency, packet loss, Mean Opinion Score (MOS), and R-Value.
- Continuously monitor real-time user agent voice-quality statistics, including Mean Opinion Score (MOS), R-Value, jitter, latency, and packet loss.

1.2 Devices Supported by This Module

The AppManager for SIP Server module supports the following devices:

- Avaya Session Manager 6.3, 6.2, 6.1, and 6.0
- Avaya SIP phones capable of providing RFC6035 voice quality reports
- Polycom SIP phones capable of providing RFC6035 voice quality reports

This module might support other RFC6035-compliant phones not included in the preceding list. If you need further information or if you have suggestions for devices not in the preceding list, contact Technical Support or your NetIQ sales team.

The following table lists the different combinations of call servers, phone types, and AppManager modules you can use to monitor that phone on that call server:

Call Server or Vendor	Phone Type	Recommended AppManager Module
Avaya	1120E, 1140E, 1165E, 1220, and 1230 using SIP firmware Release 2.1 or later	AppManager for SIP Server (for voice quality monitoring)
Avaya	1120E, 1140E, 1165E, 1220, and 1230 phones using Unistim firmware	AppManager for CS1000 and AppManager for CS2x modules (for voice quality monitoring)
Avaya	96xx series SIP phones	AppManager for Avaya Communications Manager module (for voice quality monitoring)
Avaya (heritage-Nortel)	Communication Server 1000 (CS1000) phones	AppManager for CS1000 (for voice quality monitoring) AppManager for SIP Server supports CS1000 when the phones are connected through the Avaya Session Manager.
Polycom	SoundPoint IP 321/331/335, 430, 450, 550, 560, 650, and 670 phones	AppManager for SIP Server (for voice quality monitoring)
Polycom	SoundStation IP 5000 conference phones	AppManager for SIP Server (for voice quality monitoring)
Polycom	VVX 1500 phones	AppManager for SIP Server (for voice quality monitoring)

1.3 Known Interoperability Limitations

At the time of this release, NetIQ Corporation discovered the following interoperability limitations:

- The Avaya (heritage-Nortel) CS1000 Line Terminal Proxy Server (LTPS) does not forward RFC6035 Publish messages, and phone registered through a CS1000 LTPS cannot report SIP voice quality statistics to the AppManager agent for use with this module. However, you can monitor Avaya Session Manager phones in a deployment scenario that includes LTPS, but not phones directly connected to the LTPS. If you want to monitor voice quality on CS1000 11xx phones, Avaya recommends that you configure these phones with a Unistim load, for which voice quality monitoring is provided by the AppManager for CS1000 module.
- If you want to monitor voice quality for Polycom phones with a CS1000 server, connect these phones directly to the Avaya Session Manager instead of the CS1000 LTPS. You can use AppManager for SIP Server to monitor voice quality on these phones.
- Voice quality reporting for Avaya 11xx and 12xx series phones is not compliant to RFC6035 in one respect: although voice quality reports may be sent as interval, alert or end of call reports, the phones do not identify the reports as `vqIntervalReport`, `vqAlertReport` and `vqSessionReport` as called for in RFC6035. Instead, all three reports are sent as type `vqSessionReport`. As a result, SIPServer_CollectCallData Knowledge Script does not know when a report is received if it is the final report for the call.

- Avaya 11xx phones report voice quality metrics with an end-of-session designator, so that multiple end-of-session metrics are received for a single call on these phones. As a result, the metrics for these calls might appear in multiple SIPServer_CallQuality intervals for a single call that extends through multiple monitoring intervals.
- Asterisk phone servers do not yet support Publish forwarding, so only RFC6035 phones which send voice quality reports directly instead of through the proxy server can be used with the Asterisk.

1.4 Counting AppManager Licenses

The AppManager for SIP Server module consumes one AppManager license for every active SIP user agent.

A *SIP user agent* is a logical network endpoint that can send and receive SIP messages. The SIP user agent ID is configured for a specific SIP user agent, usually a phone or endpoint device. The user agent ID persists for all transactions from that device. The user agent ID identifies a configured user in the same way that a numeric extension identifies the user on a non-SIP call server. Extension numbers are frequently part of a user agent ID, as well as the call server name, such as `1234@Callserver.location`. The exact formatting and usage might vary from vendor to vendor.

2 Installing and Configuring AppManager for SIP Server

This chapter provides installation instructions, and describes system requirements and configuration information for AppManager for SIP Server.

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

2.1 System Requirements

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

AppManager for SIP Server has the following system requirements:

Software/Hardware	Version
NetIQ AppManager installed on the AppManager repository (QDB) computer, on each proxy agent computer, and on all console computers	7.0 or later Support for Windows Server 2008 on AppManager 7.x requires AppManager Windows Agent hotfix 71704 or later. For more information, see the AppManager Suite Hotfixes page.
Microsoft operating system installed on the proxy agent computers	One of the following versions: ◆ Windows Server 2012 ◆ Windows Server 2008 R2 ◆ Windows Server 2008 (32-bit and 64-bit) ◆ Windows 7 (32-bit and 64-bit) ◆ Windows Server 2003 (32-bit and 64-bit)
AppManager for Microsoft Windows module installed on repository, proxy agent, and console computers	Support for Windows Server 2008 R2 on AppManager 7.x requires the AppManager for Windows module, version 7.6.170.0 or later. For more information, see the AppManager Module Upgrades & Trials page.

Software/Hardware	Version
Microsoft SQL Server installed on the proxy agent computers	One of the following versions installed on each proxy agent computer: ◆ SQL Server 2012 or SQL Server 2012 Express ◆ SQL Server 2008 R2 (32-bit and 64-bit) ◆ SQL Server 2008 (32-bit and 64-bit) or SQL Server 2008 Express (32-bit and 64-bit) ◆ SQL Server 2005 (32-bit and 64-bit) Service Pack 4 or SQL Server 2005 Express (32-bit and 64-bit) Service Pack 4.
SQLXML on the proxy agent computers	4.0
Microsoft .NET Framework on proxy agent computer	4.0 or later ◆ You can download Microsoft .NET Framework 4.0 from the Microsoft Download Center. ◆ You can download Microsoft .NET Framework 4.5 from the Microsoft Download Center. If your version of Windows already includes Microsoft .NET Framework 4.5, go to the Add Roles and Features Wizard and select .**NET Framework 4.5** and **ASP.NET 4.5** under **.NET Framework 4.5 Features**. As a result of these selections, the **TCP Port Sharing** component required by this module is also selected. You can find this component below the **WCF Services** component under **.NET Framework 4.5 Features** in the Add Roles and Features Wizard.
Microsoft Core XML Services (MSXML)	6.0 **NOTE:** This component is required mainly for Windows 2003, as later versions of Windows have MSXML installed by default.

2.2 Scalability Considerations and Examples

This topic covers scalability issues you should consider before using this module, followed by a set of sample configurations.

2.2.1 Scalability Considerations

Consider the following before installing the AppManager for SIP Server module:

- For Knowledge Script tasks related to RFC6035 SIP processing, the primary scalability factor is the rate, in *messages per second*, of SIP Publish voice quality reports which are received by the system for processing.
- SIP voice quality reports are generated by RFC6035-enabled user agents which are actively engaged in a call, or off-hook. The number of reports per call is determined by the phone and agent configuration (whether mid-call reporting is enabled), and the length of the call.
- Inactive phones or agents that are configured but not in use do not increase CPU usage on the proxy agent computer, and as a result are not a scalability consideration.

- Phones or agents that are not RFC6035-capable that are in use but do not send SIP Publish reports do not increase CPU usage on the proxy agent computer, and as a result are not a scalability consideration.
- One proxy agent computer can support the monitoring of up to 150 SIP Publish messages per second on one SIP server or across multiple SIP Servers.
- Because the SIP Publish message rate is influenced by the agent and phone configuration, you can monitor more active agents and phones by changing the phone configuration to send fewer mid-call interval reports, or to send only end-of-call session reports. Decreasing the number of SIP Publish messages per call reduces the strain on the proxy agent computer and allows it to monitor more active phones and agents.

2.2.2 Scalability Example 1: Mid-call reports enabled for every 20 seconds

In this example, you have a phone configured to send mid-call reports every 20 seconds, which is the default for Polycom phones. This phone is also set up to send an end-of-call report. The phone typically sends five to six mid-call reports and one end-of-call report in a 120-second call.

If the phone starts a new call every 540 seconds on average, which is equivalent to an 8 ccs call rate, the phone will generate seven messages every 540 sec, for a overall SIP Publish rate of 0.013 messages per second.

Conclusion: One proxy agent computer processing 150 messages per second could support 11,570 phones or agents.

((150 messages/sec) / (0.013 messages/sec/phone)) = 11,570 phones.

2.2.3 Scalability Example 2: End-of-call reports enabled, but not mid-call

In this example, you have a phone configured to send only end-of-call reports, and it will send a single SIP Publish in a 120-second call.

If the phone starts a new call every 540 seconds on average, which is equivalent to an 8 ccs call rate, the phone will generate one message every 540 seconds, for a overall SIP Publish rate of 0.0019 messages per second.

Conclusion: One proxy agent computer processing 150 messages per second could support 81,000 phones or agents in this configuration.

((150 messages/sec) / (0.0019 messages/sec/phone)) = 81,000 phones.

2.2.4 Scalability Example 3: End-of-call reports, threshold alerting enabled

In this example, you have a phone configured to send end-of-call reports, but not mid-call reports, and it will send a single SIP Publish in a 120-second call. Also, the phone might send an additional threshold crossing report, for two messages in a call.

If the phone starts a new call every 540 seconds on average, which is equivalent to an 8 ccs call rate, the phone will generate two messages every 540 seconds, for an overall SIP Publish rate of 0.037 messages per second.

Conclusion: One proxy agent computer processing 150 messages/sec could support 40,500 phone or agents in this configuration

((150 messages/sec) / (0.037 messages/sec/phone)) = 40,500 phones.

2.3 Installing the Module

Run the module installer on all proxy agent computers you want to monitor to install the agent components, and run the module installer on all console computers to install the Help and console extensions.

Access the `AM70-SIPServer-8.x.x.0.msi` module installer from the `AM70_SIPServer_8.x.x.0.` self-extracting installation package on the AppManager Module Upgrades & Trials page.

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

- Log in to the server using the account named Administrator. Then, run 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 to the `\AppManager\qdb\kp` folder as in previous releases of AppManager.

NOTE: Microsoft .NET Framework 3.5 is required if you want to install the QDB component remotely from a Windows Server 2012 computer.

You can install the module manually, or you can use Control Center to deploy the module to a remote computer where an agent is installed. For more information, see Section 2.4, "Deploying the Module with Control Center," on page 17. However, if you do use Control Center to deploy the module, Control Center only installs the *agent* components of the module. The module installer installs the QDB and console components as well as the agent components on the agent computer.

To install the module manually:

1. On all proxy agent computers, stop the NetIQ AppManager Client Resource Monitor (`NetIQmc`) service to ensure that any existing version of `qNQSIPServer.dll` is updated correctly during installation of the module.
2. Double-click the module installer `.msi` file.
3. Accept the license agreement.
4. 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 will install the agent components.

5 To install the Knowledge Scripts into the QDB:

5a Select **Install Knowledge Scripts** to install the repository components, including the Knowledge Scripts, object types, and SQL stored procedures.

5b 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.

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

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

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

9 Run the module installer on all proxy agent computers to install the agent components.

10 Configure all necessary SNMP community strings in AppManager Security Manager to enable access of remote SIP servers. For more information, see Section 2.6.1, "Configuring Security Manager with SNMP Credentials," on page 19.

11 (Conditional) If you have not discovered SIP Server resources, run the Discovery_SIPServer Knowledge Script on all proxy agent computers where you installed the module. For more information, see Section 2.6, "Discovering SIP Server Resources," on page 19.

12 Run the SIPServer_SetupSupplementalDB Knowledge Script to:

12a Apply improved SQL indices to the CDR and Traceroute tables in the SIP Server supplemental database.

12b Update the CDR table in the SIP Server supplemental database to allow the database to store Facilities Restriction Level (FRL) codes.

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

2.4 Deploying the Module with Control Center

You can use Control Center to deploy the module to 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.

2.4.1 Deployment Overview

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

To deploy the module on a proxy agent computer:

1 Verify the default deployment credentials.

2 Check in an installation package. For more information, see Section 2.4.2, "Checking In the Installation Package," on page 18.

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

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

5 Approve the deployment task.

6 View the results.

2.4.2 Checking In the Installation Package

You must check in the installation package, `AM70-SIPServer-8.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 and navigate to the Administration pane.

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

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

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

5 Navigate to the folder where you saved `AM70-SIPServer-8.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.

2.5 Silently Installing the Module

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

```
msiexec.exe /i "AM70-SIPServer-8.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-SIPServer-8.x.x.0.msi.log"
```

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

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

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

Windows authentication:

```
AM70-SIPServer-8.x.x.0.msi /qn MO_B_QDBINSTALL=1 MO_B_MOINSTALL=0
MO_B_SQLSVR_WINAUTH=1 MO_SQLSVR_NAME=SQLServerName MO_QDBNAME=AM-RepositoryName
```

SQL authentication:

```
AM70-SIPServer-8.x.x.0.msi /qn MO_B_QDBINSTALL=1 MO_B_MOINSTALL=0
MO_B_SQLSVR_WINAUTH=0 MO_SQLSVR_USER=SQLLogin MO_SQLSVR_PWD=SQLLoginPassword
MO_SQLSVR_NAME=SQLServerName MO_QDBNAME=AM-RepositoryName
```

2.6 Discovering SIP Server Resources

Use the Discovery_SIPServer Knowledge Script to discover a server that uses Session Initiation Protocol (SIP), such as Avaya Session Manager, and to discover resources for that server.

You can discover a SIP server either by SNMP query or by manual configuration.

- If you are using the **SNMP Query** option for the *Discovery method* parameter to discover SIP servers, specify a comma-separated list of SIP Server addresses, or you can specify the full path to a file containing a list of servers. All devices to be discovered must support RFC1213-MIB, including the `sysObjectID`, the `sysName`, and the `sysDesc` properties.
- If SNMP `get` operations cannot be performed against the SIP server itself, use the **Manual Configuration** option for the *Discovery method* parameter to specify the IP address, system name, and system type for the server. You can also include a description for the discovered server that will display in the TreeView object for that server.

In addition, you can use Discovery_SIPServer to create the supplemental database needed by the SIP server by selecting the *Set up supplemental database?* parameter. You can also set up a supplemental database by running SIPServer_SetupSupplementalDB Knowledge Script. Regardless of the method you use, when you set up the supplemental database, you also create the underlying tables and the stored procedures.

By default, this script runs **once a day**.

2.6.1 Configuring Security Manager with SNMP Credentials

AppManager uses SNMP queries to access remote SIP servers when you select **SNMP Query** for the *Discovery method* parameter in the Discovery_SIPServer Knowledge Script. Before discovering a SIP Server resource, configure SNMP community string information for each SIP server you want to monitor with AppManager Security Manager.

AppManager for SIP Server supports SNMP versions 2 and 3.

Configuration for SNMP Version 2

For SNMP v2 configuration, complete the following fields in the **Custom** tab of Security Manager for the proxy agent computer:

Field	Description
Label	`SIPServer` or `SNMP`
Sub-label	Indicate whether the community string information will be used for a single device or for all devices: ◆ For a community string for a single device for a proxy agent computer, specify the device name for the community string. ◆ For a community string for all devices for a proxy agent computer, type `default`.
Value 1	Specify the appropriate read-only community string value, such as `private` or `public`.

Configuration for SNMP Version 3

AppManager for SNMP supports the following modes for SNMP v3:

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

In addition, the module 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 device that is being monitored by each proxy computer.

For SNMP v3 configuration, complete the following fields in the **Custom** tab of Security Manager for the proxy agent computer:

Field	Description
Label	`SIPServer` or `SNMP`
Sub-label	Indicate whether the SNMP credential string information will be used for a single device or for all devices: ◆ For SNMP credentials for a single device for a proxy agent computer, specify the device name. ◆ For SNMP credentials for all devices for a proxy agent computer, type `default`.
Value 1	Specify the SNMP user name, or *entity*, configured for the device. All SNMP v3 modes require an entry in this field.
Value 2	Specify the name of the context associated with the user name or entity entered in **Value 1**. 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, or if the configuration does not require the use of a specific named context, type an asterisk (*) for a wildcard. All SNMP v3 modes require an entry in this field.
Value 3	Specify the combination of protocol and password appropriate for the SNMP v3 mode you have implemented. ◆ For *no authentication/no privacy mode*, leave this field blank. ◆ For *authentication/no privacy mode*, enter `md5` or `sha` and the password for the protocol, separating each entry with a comma. For example: `md5,abcdef` ◆ For *authentication/privacy mode*, enter `md5` or `sha` and the associated password, and then enter `des` and the associated password, separating each entry with a comma. For example: `sha,hijklm,des,nopqrs`

2.6.2 Configuring Security Manager for Supplemental Database Setup

To avoid an error message when running the Discovery_SIPServer Knowledge Script after selecting the *Set up supplemental database?* parameter for a server requiring a SQL login and password, use AppManager Security Manager to store the SQL user name and password information for the SQL Server hosting the supplemental database.

If you create the supplemental database on a server that uses Windows authentication instead of SQL authentication, you do not need to create the Security Manager entry.

On the **Custom** tab in Security Manager, complete the following fields for each SQL Server you are using for this module:

Field	Description
Label	Specify the SQL Server name and the instance name of the SQL Server hosting the supplemental database. Use the following structure: `sql$`*`SQL Server Name`*`\`*`Instance Name`* For example: `sql$HOUSERVER2\DB1`
Sub-label	Specify the SQL Server user name.
Value 1	Specify the SQL Server password.
Extended application support	Required field. Select this option to encrypt the user name and password in Security Manager. Do not leave this option unselected.

2.6.3 Setting Parameter Values for Discovery_SIPServer

Set the parameters on the **Values** tab as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the severity level, from 1 to 40, to indicate the importance of an event in which the Discovery_SIPServer job fails. The default is 5.
Raise event if discovery succeeds?	Select **Yes** to raise an event if discovery succeeds in finding a SIP server. The default is unselected.
Event severity when discovery succeeds	Set the severity level, from 1 to 40, to indicate the importance of an event in which this script succeeds in finding SIP Servers. The default is 25.
Raise event if discovery fails?	Select **Yes** to raise an event if discovery fails to find some or all of your SIP servers. The default is Yes.
Event severity when discovery fails	Set the severity level, from 1 to 40, to indicate the importance of an event in which discovery fails to find some or all of your SIP servers. The default is 10.

Parameter	How to Set It
Raise event if database setup succeeds?	Select **Yes** to raise an event if the creation of the supplemental database is successful. The default is unselected.
Event severity when database setup succeeds	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the supplemental database is created successfully. The default is 25.
Raise event if database setup fails?	Select **Yes** to raise an event if creation of the supplemental database fails. The default is Yes.
Event severity when database setup fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the creation of the supplemental database fails. The default is 15.
Discover SIP Servers	
Discovery method	Specify the method you want to use to discover the SIP server. The default is SNMP Query. If you choose **SNMP Query**, you can specify the identity of the target server or servers using either a comma-separated list or a file listing each of the servers to discover. The devices in the list are queried using SNMP, and the system properties are used to create treeview objects for each system. With the SNMP Query option, configure all necessary SNMP community strings in AppManager Security Manager to enable access of remote SIP servers. For more information, see Section 2.6.1, “Configuring Security Manager with SNMP Credentials,” on page 19. **Note** A device to be discovered must support RFC1213-MIB. If the device to be discovered does not support RFC1213, or if the device does not respond to SNMP queries for the RFC1213 properties listed, you need to use the manual discovery method. If you choose **Manual Configuration**, the script can only discover one SIP server per job. Use the parameters in the *System Properties for Manual Configuration* section of this script to populate the object details for the SIP server that this script discovers. The script does not send SNMP to the selected server.
SNMP Settings	
Comma-separated list of SIP servers	Specify the DNS name or IP address for the SIP servers you want to discover, using a comma to separate multiple items. For example: `10.0.1.1,10.0.1.7,10.0.1.100` Leave this parameter blank if you want to use the *Path to file with list of SIP servers* parameter to specify a file containing this information. **Note** If any of the servers are located behind a firewall using Network Address Translation, the proxy agent must be able to access the address listed here from the agent side of the firewall.

Parameter	How to Set It

Full path to file with list of SIP servers	Instead of listing each server separately in the previous parameter, you can specify the full path to a file on the agent that contains a list of DNS names or IP addresses of SIP servers.
	
	In the file, list the servers on multiple lines, and ensure that each line contains only one entry.
	
	For example:
	
	`10.0.1.1`
	
	`10.0.1.7`
	
	`10.0.1.100`
	
	The default location for this file is `/netiq/AppManager/bin/SIPServer`. If you save the file in this location, specify just the file name in this parameter.
	
	If you save the file in any other location, specify the full path name.

SNMP message timeout	Specify the number of seconds discovery should attempt an SNMP message request to an *individual* SIP Server server before retrying the connection. The minimum value is 10 seconds and the maximum is 2000 seconds. The default is 120 seconds. The value you set here is the timeout value for *all* SNMP message requests for *all* SIPServer Knowledge Script jobs.
SNMP task timeout	Specify the number of seconds discovery should attempt an SNMP retrieve request to an *individual* SIP Server server before retrying the connection. The minimum value is 900 seconds and the maximum is 999999 seconds. The default is 3600 seconds. The value you set here is the timeout value for *all* SNMP retrieve requests for *all* SIPServer Knowledge Script jobs.
SNMP retries	Specify the number of times discovery should attempt an SNMP connection to an individual SIP Server before attempting an SNMP connection to the next SIP Server in the list. The default is 4 attempts, and the maximum is 10 attempts. The value you set here will be the number of retries for *all* SNMP connections for *all* SIPServer Knowledge Script jobs.
System Properties for Manual Configuration	
SIP server IP address	Specify the IP address of the SIP server you want to discover. You cannot leave this parameter blank if you selected *Manual Configuration* for the *Discovery method* parameter. The IP address you specify in this parameter becomes part of the SIPServer TreeView object name.
System type	Select the type of server you want to discover. This value displays in the SIP Server TreeView object for the server you discover. You cannot leave this parameter blank if you selected *Manual Configuration* for the *Discovery method* parameter. You can choose Custom or AvayaSM.

Parameter	How to Set It
System name	Specify the name of the instance you want to display for the SIP Server TreeView object for the server you discover. You cannot leave this parameter blank if you selected *Manual Configuration* for the *Discovery method* parameter.
System description	Specify the descriptive text you want to display for the SIP Server TreeView object for the server you discover. This parameter is optional.
Discover SIP Quality Of Service Reporting Interface?	Select **Yes** to set up SIP quality of service report data collection. If you select Yes, the script creates an object in the TreeView for quality of service report data. The default is Yes.
SIP identity of collector	Specify the Uniform Resource Identifier (URI) where the proxy agent will accept SIP quality of service reports. If you leave this field blank, the discovery job raises an event, and the discovery fails. Format the URI using RFC 3261: `sip:`*`username@address`*`:`*`port`*`;transport=`*`transportType`* For example: `sip:collector@raldvap710.us.houqe.lab:5060;transport=udp` The *`username`* is any string using characters that are compliant with RFC 3261: A-Z, a-z, and &=+$,;?/ The *`address`* is where the SIP reports will be received. This value is the fully qualified domain name (FQDN) that will appear in the SIP message for phones using this SIP server. The *`port`* is the port number that will receive the SIP reports, and it should be should be a number between 1 to 65535. The *`transportType`* is the protocol used to receive SIP reports. This protocol is the transport type of the SIP trunk between the SIP server and the agent, not the transport type used by the phones themselves. The protocol must be TCP or UDP; TLS and SCTP are not supported.
Set up supplemental database?	Select **Yes** to create the supplemental database, including the tables and stored procedures needed to store call detail records and phone deregistration information. The default is unselected. For more information, see Section 3.3.4, “Understanding the Supplemental Database,” on page 46.

Parameter	How to Set It
Start pruning job on supplemental database?	**For all supported versions of SQL Server, except SQL Server Express versions**: Set to **Yes** to create a SQL job that deletes data from the supplemental database. The SQL job runs every night. Data is deleted from the supplemental database based on the value you specify in the *Number of days to keep call detail records* parameter. The default is Yes. **For SQL Server Express versions**: Set to **No**. The pruning job is not supported for SQL Server Express versions. **To manually delete data from the supplemental database**: 1. Run the following stored procedure from a command line: `osql -E -S <`*sql server*`> -n -d <`*database*`> -Q "exec dbo.Task_SIPServer_Pruning"` where <*sql server*> is the name of the server that hosts the supplemental database, and where <`database`> is the name of the supplemental database. For example: `osql -E -S SuppDBSIPServer -n -d SIPServer_S8300-Cluster -Q "exec dbo.Task_SIPServer_Pruning"` 2. Configure a Windows Scheduled Task to schedule pruning at an interval of your choosing. The process for configuring a Windows Scheduled Task varies according to your version of Microsoft Windows. For more information, consult your Windows documentation.
Number of days to keep call detail records	Specify the number of days' worth of CDRs to keep in the SIP Server supplemental database. Data older than what you specify is discarded. The minimum number of days is 1, and the maximum is 30. The default is 7 days.
SQL Server Information	
SQL Server instance name	Specify the instance name of the SQL Server where you want to create the new SIP Server supplemental database. If you specify both the SQL Server instance name for this parameter and the SQL Server database user name in the following parameter, these values must match the values you specified in Section 2.6.2, "Configuring Security Manager for Supplemental Database Setup," on page 21. Leave this parameter blank to use the default SQL server instance on the proxy agent computer.
SQL database user name	Specify the user name for the SQL Server where you want to create the new SIP Server supplemental database. Leave this parameter blank to use Windows authentication instead of SQL authentication.

2.7 Configuring SIP Servers for Voice Quality Monitoring

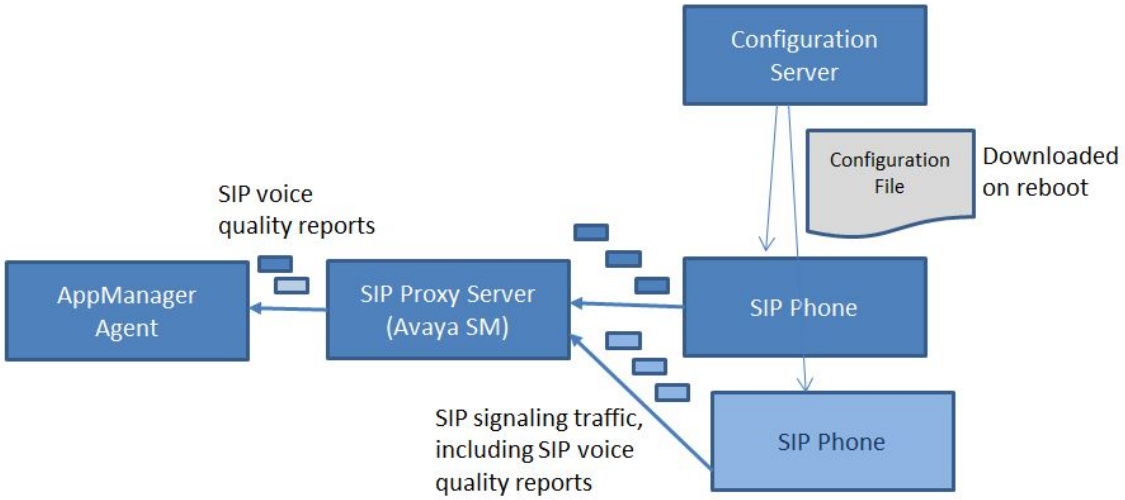
This topic covers how to configure Avaya phones and Polycom IP phones for voice quality monitoring.

Avaya and Polycom IP phones that support RFC6035 voice quality monitoring do not send voice quality reports directly to the AppManager agent (the central report collector). Instead, these phones send the reports to the SIP proxy server where the phone is registered, which then forwards the reports to the agent.

Each phone is registered to a particular SIP proxy server, such as an Avaya Session Manager, and that phone contains the network location of a configuration server in the initial phone setup. On reboot, the phone contacts the configuration server to download a configuration file specific to that phone type.

The phone uses the configuration file to determine when to send a voice quality report, and the report includes the address of the agent (the central report collector) when it sends the report to the SIP Proxy server. The SIP proxy server, such as the Avaya Session Manager, delivers the report to the AppManager agent.

Figure 2-1 *How AppManager for SIP Server generates voice quality reports*



2.7.1 Configuring Avaya Session Manager for Voice Quality Monitoring

Phones configured for voice quality monitoring send SIP Publish messages containing voice quality information to the Avaya Session Manager, which then forwards the voice quality reports to the AppManager agent. To enable this, you need to configure a connection between the session manager and the agent.

To configure a connection between the Avaya Session Manager and the AppManager agent:

1. Log into the System Manager web interface associated with Avaya Session Manager.
2. From the **Elements** menu, select **Routing**.
3. From the **SIP Entities** page, click **New**.
4. In the **Name** field, specify a name for the monitoring agent.
5. In the **FQDN or IP Address** field, specify the network address of the AppManager agent.

6. In the **Type** field, select **SIP Trunk**.
7. In the **Entity Links** section, click **Add**.
8. In the **SIP Entity 1** field of the new entity record, specify the name of the Avaya Session Manager.
9. In the **Protocol** field, select **UDP**.
10. In the **Port** field, specify **5060** as the port.
11. In the **SIP Entity 2** field, specify the monitoring agent name. This value should match what you specified in the **Name** field in the **General** section.
12. In the **Port** field, specify the port configured for the data collector when you ran Discovery_SIPServer. For more information, see "Discovering SIP Server Resources" on page 19. The Avaya Session Manager is now ready to forward SIP voice quality reports to the AppManager agent.

NOTE: If you enabled *SIP Link Monitoring* at the Avaya Session Manager for the entity link you just created, the link only displays as **up** when a SIPServer_CollectCallData Knowledge Script is running. Otherwise, the SIP entity link displays as **down**. You can check the link status by selecting **Session Manager > System status > SIP entity monitoring**.

2.7.2 Configuring Voice Quality Monitoring for Avaya phones

Voice quality monitoring configuration information is provided to the 11xx and 12xx series phones from a configuration server, along with other information necessary to phone operation. For more information on setting up the configuration server for 11xx and 12xx series phones, see the following PDF documents:

- SIP Software for Avaya 1100 Series IP Deskphones-Administration
- SIP Software for Avaya 1200 Series IP Deskphones-Administration

You can find the specific voice quality monitoring settings for the 11xx and 12xx series phones in the device configuration file, `asmDeviceConfig.txt`, which is retrieved from the configuration server when the phone reboots.

11xx and 12xx series phones require that you name the designated collector `pvqmservice`. Do not include that information in their device file setup. For example, if the SIP data collector in the TreeView is configured as `SIP:pvqmservice@10.1.2.3:5060`, you only need to enter `10.1.2.3` for the agent address in the configuration file.

The following is an example of the `asmDeviceConfig.txt` file that enables voice quality monitoring on an 11xx phone from an AppManager agent. In this particular example, the AppManager agent address is `10.1.2.3` and the SIP Server RFC6035 data collector identity has been configured at discovery to be `SIP:pvqmservice@10.1.2.3:5060`.

```
VQMON_PUBLISH YES
VQMON_PUBLISH_IP 10.1.2.3

SESSION_RPT_EN YES
SESSION_RPT_INT 30

LISTENING_R_ENABLE YES
LISTENING_R_WARN 70
LISTENING_R_EXCE 60

PACKET_LOSS_ENABLE YES
PACKET_LOSS_WARN 256
PACKET_LOSS_EXCE 1280
```

```
DELAY_ENABLE YES
DELAY_WARN 150
DELAY_EXCE 175

JITTER_ENABLE YES
JITTER_WARN 3276
JITTER_EXCE 32760

# Server and Network configuration commands
DNS_DOMAIN    us.qe.ral
SIP_DOMAIN1    netiqavayasm.ral
SERVER_IP1_1   10.4.5.6
SERVER_IP1_2   10.4.5.6
# TCP is used in the sample configuration
SERVER_UDP_PORT1_1 5060
SERVER_UDP_PORT1_2 5060
SERVER_RETRIES1  3

SNTP_ENABLE YES
SNTP_SERVER 10.5.6.7
AUTO_UPDATE YES
AUTO_UPDATE_TIME 3600
# 48600 = 1:30pm
# 3600 = 1:00am
# 28800 = 8:00am 29700
# 54900 = 3:15pm

# Voice mail Feature configuration commands
VMAIL       56245
VMAIL_DELAY    300

MADN_DIALOG  YES

# Time Zone - (GMT-0500)Eastern Time (US & Canada)
FORCE_TIME_ZONE  YES
TIMEZONE_OFFSET  -18000
DST_ENABLED    YES

# Voice Application commands
DEF_LANG     English
DEF_AUDIO_QUALITY High
ENABLE_BT YES
ENABLE_3WAY_CALL YES
SIP_PING     YES

 VMAIL 123-1234
 VMAIL_DELAY 2000

 DST_ENABLED 1
 TIMEZONE_OFFSET 0

 MAX_INBOX_ENTRIES 100
 MAX_OUTBOX_ENTRIES 100
 MAX_REJECTREASONS 20
 MAX_CALLSUBJECT 20
 MAX_PRESENCENOTE 20

DEF_LANG English
 DSCP_CONTROL -1
 802.1P_CONTROL -1
 DSCP_MEDIA -1
 802.1P_MEDIA -1
 DSCP_DATA -1
 802.1P_DATA -1
 LOG_LEVEL 2
 RECOVERY_LEVEL 255
 AUTO_UPDATE 0
 AUTO_UPDATE_TIME 0
 AUTO_UPDATE_TIME_RANGE 1
 DOS_PACKET_RATE 5
```

```
DOS_MAX_LIMIT 100
DOS_LOCK_TIME 20
DEF_AUDIO_QUALITY High
DEF_DISPLAY_IM NO
MAX_IM_ENTRIES 999
MAX_ADDR_BOOK_ENTRIES 100
ADDR_BOOK_MODE NETWORK
IM_MODE ENCRYPTED
ADMIN_PASSWORD 123456
HOLD_TYPE rfc3261
AUTH_METHOD AUTH
ENABLE_3WAY_CALL 1
DISABLE_PRIVACY_UI 0
DIALTONE
RINGINGTONE
BUSYTONE
FASTBUSYTONE
CONGESTIONTONE
ENABLE_BT 0
NAT_SIGNALLING NONE
NAT_MEDIA NONE
NAT_TYPE NONE
NAT_TTL 120
STUN_SERVER_IP1 0.0.0.0
STUN_SERVER_IP1 0.0.0.0
STUN_SERVER_PORT1 3478
STUN_SERVER_PORT1 3478

TRANSFER_TYPE MCS
ENABLE_PRACK 0
PROXY_CHECKING 1
ENABLE_BT 0
REDIRECT_TYPE MCS
AUTOLOGIN_ENABLE 1
MADN_PRIVACY
MADN_TIMER 1800
MADN_DIALOG 0
IM_NOTIFY 1
DISABLE_OCT_ENDDIAL 0
FORCE_OCT_ENDDIAL 0
DISPLAY_CALL_SNDR_IM_KEY1
FORCE_CFWD_NOTIFY 0
DEFAULT_CFWD_NOTIFY 0
FORCE_TIME_ZONE 0

SESSION_TIMER_ENABLE NO
```

You can adjust the following voice quality monitoring settings in the configuration file for 11xx and 12xx SIP phones:

Table 2-1 *Voice quality monitoring settings for 11xx and 12xx SIP phones*

Setting	Allowed Values	Default Value	Notes
VQMON_PUBLISH	YES | NO	NO	YES enables the Publish message containing the voice quality monitoring metrics sent to the Proactive Voice Quality Monitoring (PVQMoN) collecting server (the Appmanager agent).

Setting	Allowed Values	Default Value	Notes
VQMON_PUBLISH_IP	xxx.xxx.xxx.xxx		Specify the IP address of the PVQMoN server that collects voice quality monitoring metrics from the Publish message (the Appmanager agent). This IP address is used only within the report, and the message is relayed through the SIP server rather than being sent directly.
LISTENING_R_ENABLE	YES | NO		Enable or disable the alerts based on the Listening R Minor and Major thresholds. YES enables the sending of the alert report based on the Listening R-Value. NO disables the sending of the alert report.
LISTENING_R_WARN	[xx]	VOCODER_G711_ULAW VOCODER_G711_ULAW PLP: 80 VOCODER_G723 VOCODER_FLAG_G723_RATE_53 VOCODER_FLAG_G723_RATE_63: 60 VOCODER_G729 VOCODER_PCM8 vqmonVocoderTypeUnknown: 70 (default if not configured and unknown type)	Specify the threshold to send a report on Listening R less than [xx]. The default value is 70. Using a value of 0 resets it to the default value, based on the far-end VOCODER. xx is an integer value used as the threshold.
LISTENING_R_EXCE	[xx]	VOCODER_G711_ULAW VOCODER_G711_ULAW PLP: 70 VOCODER_G723 VOCODER_FLAG_G723_RATE_53 VOCODER_FLAG_G723_RATE_63: 50 VOCODER_G729 VOCODER_PCM8 vqmonVocoderTypeUnknown: 60	Specify the threshold to send a report on Listening R less than [xx]. The default value is 60. Using a value of 0 resets it to the default value, based on the far-end VOCODER. xx is an integer value used as the threshold.

Setting	Allowed Values	Default Value	Notes
PACKET_LOSS_ENABLE	YES | NO		Enable or disable the alerts based on the packet loss thresholds. Packet loss is the fraction of RTP data packets from the source lost since the beginning of reception. The value is an integer scaled by 256. The range is 1 to 25600. YES enables the sending of the alert report based on the packet loss NO disables the sending of the alert report.
PACKET_LOSS_WARN	1 to 25600	256 (1%)	Specify the threshold to send a report on packet loss greater than [xx]. Using a value of 0 resets the threshold to the default, an integer scaled by 256. The range is 1 to 25600.
PACKET_LOSS_EXCE	1 to 25600	1280 (5%)	Specify the threshold to send a report on packet loss greater than [xx]. Using a value of 0 resets the threshold to the default, an integer scaled by 256.
JITTER_ENABLE	YES | NO		Enable or disable alerts based on the inter-arrival jitter on incoming RTP packets time. The value is represented in 1/65536 of a second. YES enables the sending of an alert report based on jitter detection. NO disables the sending of an alert report based.
JITTER_WARN	[xx]	3276 (50 ms)	Specify the threshold to send a report on inter-arrival Jitter greater than [xx]. 1 second is broken up into 65535 (0xffff hex) parts. [xx] / 65535 is the threshold in seconds. Using a value of 0 resets the threshold to the default, an integer scaled by 256.

Setting	Allowed Values	Default Value	Notes
JITTER_EXCE	[xx]	32760 (500 ms)	Specify the threshold to send a report on inter-arrival Jitter greater than [xx]. 1 second is broken up into 5535 (0xffff hex) parts. [xx] / 65535 is the threshold in seconds. Using a value of 0 resets the threshold to the default, an integer scaled by 256.
DELAY_ENABLE	YES | NO		Enable or disable alerts based on excessive delay detection. This is the one-way delay (including system delay) for the call, measured in milliseconds. YES enables excessive delay detection. NO disables excessive delay detection.
DELAY_WARN	[xx]	150 ms	Specify the threshold to send a report on excessive delay detection. Using a value of 0 resets the threshold to the default value. xx is an integer value used as a threshold, measured in 1/1000 of a second.
DELAY_EXCE	[xx]	175 ms	Specify the threshold to report unacceptable excessive delay greater than [xx]. Using a value of 0 resets the threshold to the default value. xx is an integer value used as a threshold, measured in 1/1000 of a second.

Setting	Allowed Values	Default Value	Notes
SESSION_RPT_EN	YES | NO	disabled (NO)	Enable or disable periodic VQMon session reports. Session report enable (SESSION_RPT_EN) and session report interval (SESSION_RPT_INT) must be configured if the IP Deskphone software has been upgraded to SIP Release 3.0 or later. Otherwise, the SESSION_RPT_INT default of 60 seconds is used automatically. YES enables periodic VQMon session reports. NO disables periodic VQMon session reports.
SESSION_RPT_INT	[xx]	60 seconds	Specify the interval for the periodic VQMon session report in seconds. The minimum acceptable value is 60 seconds. The maximum acceptable value is 600 seconds. xx is an integer value used as a threshold,

2.7.3 Configuring Voice Quality Monitoring for Polycom Phones

Voice quality monitoring configuration and licensing information is provided to Polycom phones from a configuration server. You can find basic Polycom phone setup information, including how to designate a configuration server, at the Polycom Support page.

Polycom VVX 1500 phones, and SpectraLink handsets include voice quality monitoring in the device's firmware image and do not require a specific voice quality monitoring license. All other SPIP/SSIP/VVX devices require a license file from Polycom to enable voice quality monitoring. Install this license file on the configuration server in the same directory as the `sip.cfg` configuration file. If this license file is absent, the phones will not send SIP Publish reports, even if you have enabled voice quality monitoring in the device configuration file.

For more information about phone feature licensing through Polycom for Polycom phones, including licensing for voice quality monitoring, see Polycom Technical Bulletin 32265.

The specific voice quality monitoring settings for the Polycom phones are in the device configuration file, `sip.cfg`, which is retrieved from the configuration server when the phone reboots.

The following is an example of a `sip.cfg` file which will enable voice quality monitoring to an Appmanager agent.

In this particular example, the Appmanager agent address is `10.1.2.3` and the SIP Server RFC6035 Data collector identity has been configured at discovery to be `SIP:pvqmservice@10.1.2.3:5060`. Because Avaya 11xx/12xx phones require the name of the designated collector to be named `pvqmservice`, any agent which is monitoring both Avaya and Polycom phones should specify `pvqmservice` as the collector name both at discovery and in the `sip.cfg` file for the phones.)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Generated features.cfg Configuration File -->
polycomConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="polycomConfig.xsd">
<dialplan dialplan.digitmap="[2-9]11|0T|011xxx.T|[0-1][2-
9]xxxxxxxxx|[5]xxxx|[8]xxxx|[2-9]xxxxxxxxx|[5]xxxx|[8]xxxx|[2-9]xxxT">
</dialplan>
<voice>
  <voice.qualityMonitoring>
   <voice.qualityMonitoring.collector
voice.qualityMonitoring.collector.period="20">
   <voice.qualityMonitoring.collector.enable
voice.qualityMonitoring.collector.enable.periodic="1"
voice.qualityMonitoring.collector.enable.session="1" >
    </voice.qualityMonitoring.collector.enable>
    <voice.qualityMonitoring.collector.server
voice.qualityMonitoring.collector.server.1.address=pvqmservice@10.1.2.3.
voice.qualityMonitoring.collector.server.1.port="5060">
    </voice.qualityMonitoring.collector.server>
   </voice.qualityMonitoring.collector>
   <voice.qualityMonitoring.rtcpxr voice.qualityMonitoring.rtcpxr.enable="1">
   </voice.qualityMonitoring.rtcpxr>
  </voice.qualityMonitoring>
 </voice>
</polycomConfig>
```

The specific voice quality monitoring settings that may be configured in the file are:

`Quality Monitoring <quality monitoring/>`

You can adjust the following settings in the Polycom configuration file:

Table 2-2 *Voice quality monitoring settings for Polycom phones*

Parameter	Permitted Values	Default	Description
voice.qualityMonitoring.collector.alert .moslq.threshold.critical	0 to 40	0	Threshold value of listening MOS score (MOS-LQ) that causes the phone to send a critical alert quality report. Configure the desired MOS value multiplied by 10. If 0 or Null, critical alerts are not generated due to MOS-LQ. For example, a configured value of 28 corresponds to the MOS score 2.8.

Parameter	Permitted Values	Default	Description
voice.qualityMonitoring.collector.alert.moslq.threshold.warning	0 to 40	0	Threshold value of listening MOS score (MOS-LQ) that causes the phone to send a warning alert quality report. Configure the desired MOS value multiplied by 10. If 0 or Null, warning alerts are not generated due to MOS-LQ. For example, a configured value of 35 corresponds to the MOS score 3.5.
voice.qualityMonitoring.collector.alert.delay.threshold.critical	0 to 2000	0	Threshold value of one-way delay (in ms) that causes phone to send a critical alert quality report. If 0 or Null, critical alerts are not generated due to one-way delay. One-way delay includes both network delay and end system delay.
voice.qualityMonitoring.collector.alert.delay.threshold.warning	0 to 2000	0	Threshold value of one-way delay (in ms) that causes phone to send a warning alert quality report. If 0 or Null, warning alerts are not generated due to one-way delay. One-way delay includes both network delay and end system delay.
voice.qualityMonitoring.collector.enable.periodic	0 or 1	0	If 0, periodic quality reports are not generated. If 1, periodic quality reports are generated throughout a call.
voice.qualityMonitoring.collector.enable.session	0 or 1	0	If 0, quality reports are not generated at the end of each call. If 1, reports are generated at the end of each call.
voice.qualityMonitoring.collector.enable.triggeredPeriodic	0 to 2	0	If 0, alert states do not cause periodic reports to be generated. If 1, periodic reports are generated if an alert state is critical. If 2, period reports are generated when an alert state is either warning or critical.
voice.qualityMonitoring.collector.period	5 to 20	0	The time interval between successive periodic quality reports.

Parameter	Permitted Values	Default	Description
voice.qualityMonitoring.collector.server.x.address	Dotted-decimal IP address or hostname	Null	The server address of the monitoring agent (report collector) that accepts voice quality reports contained in SIP Publish messages. Set x to 1 as only one report collector is supported by Polycom at this time. This should match the value entered for the SIP data collector ID at Discovery. If the data collector is used for both Avaya and Polycom phones, use `pvqmservice@`*`IPaddress`*, such as `pvqmservice@10.41.5.1`.
voice.qualityMonitoring.collector.server.x.port	1 to 65535	5060	The port of the monitoring agent (report collector) that accepts voice quality reports contained in SIP Publish messages. This should match the port configured at discovery for your RFC6035 data collector. Set x to 1 as only one report collector is supported by Polycom at this time.
voice.qualityMonitoring.rtcpxr.enable	0 or 1	0	If 0, RTCP-XR packets are not generated. If 1, the packets are generated.

3 SIPServer Knowledge Scripts

AppManager provides the following Knowledge Scripts for monitoring SIP servers and resources. 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**. In the Operator Console, click any Knowledge Script in the Knowledge Script pane and press **F1**.

Knowledge Script	What It Does
CallQuality	Monitors calls for quality metrics, including jitter, latency, packet loss, Mean Opinion Score (MOS), and R-Value.
CollectCallData	Starts and monitors SIP call data collection.
SetupSupplementalDB	Creates a supplemental database in which to store call detail records, including voice quality reports.
UserAgentQuality	Monitors real-time user agent voice-quality statistics, including jitter, latency, packet loss, Mean Opinion Score (MOS), and R-Value.

3.1 CallQuality

Use this Knowledge Script to monitor SIP packet information stored in the SIP Server supplemental database for call quality statistics. These statistics include jitter, latency (one-way delay), lost data, Mean Opinion Score (MOS), and R-Value. MOS and R-Value are computed only for calls that use one of the following codecs: G.711u, G.711a, or G.729.

This script raises an event if a monitored call quality statistic, such as MOS, crosses a threshold that you specified. The script generates data streams for all monitored call quality statistics.

NOTE: You can trigger NetIQ Vivinet Diagnostics to diagnose the problem indicated by an event in which the percentage lost data threshold is exceeded. For more information, see Section 3.1.6, "Triggering Call and Phone Quality Diagnoses," on page 42.

The purpose of this script is two-fold:

- **Troubleshooting**. In troubleshooting mode, this script runs once and checks the supplemental database tables for calls that disconnected within the range you select in the *Call disconnect time range* parameter. Select **Run once** on the **Schedule** tab to run this script in troubleshooting mode.
- **Diagnosing**. In diagnostic mode, this script works in conjunction with NetIQ Vivinet Diagnostics to diagnose VoIP quality problems detected during monitoring. If a call quality threshold is exceeded, then, by default, this script launches Action_DiagnoseVoIPQuality, a Knowledge Script that in turn launches Vivinet Diagnostics to generate a diagnosis of the problem.

To turn off diagnostic mode, click the Actions tab, select **Action_DiagnoseVoIPQuality**, and click **Delete**. Turning diagnostic mode off or on does not affect the events raised by this script.

- **Monitoring**. In monitoring mode, this script checks the supplemental database tables at each specified interval for new records that match your query. You always run the script in monitoring mode unless you select **Run once** on the **Schedule** tab.

3.1.1 Understanding Data Streams and Threshold Events

This script generates data streams for average MOS, R-Value, jitter, latency, and packet loss. These average values are based on data from each phone involved in calls that completed during the script's interval, which is, by default, every 5 minutes. For example, in a given call, party 1's phone jitter was 30 milliseconds and party 2's phone jitter was 75 milliseconds. For this call, the data stream would be a calculated average of the jitter for both phones: 52.5 milliseconds.

This calculated average is below the default threshold value of 60 milliseconds. However, AppManager raises threshold events based on values for each phone in a call, not on the average value. Therefore, for this call, AppManager would raise one event based on the 75 milliseconds of jitter for the called phone.

3.1.2 Prerequisites

- Run SIPServer_SetupSupplementalDB to create the SIP Server supplemental database.
- Because the SIPServer_CallQuality script reports on data stored in the supplemental database by a data collector service, data must exist in the supplemental database before the reporting can be successful. To place data in the supplemental database, run SIPServer_CollectCallData on the SIP Server being monitored before you run the CallQuality script. If the CollectCallData script stops, the data collection also stops, even if the CallQuality script is still running.

3.1.3 Resource Object

SIP Server Call Data folder

3.1.4 Default Schedule

By default, this script runs **every 5 minutes**.

3.1.5 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the CallQuality job. The default is 5.

Parameter	How to Set It
Raise event if no records found?	Select **Yes** to raise an event if there are no SIP packets to monitor in the SIP Server supplemental database. The script only raises an event when no records exist in the database. The script does not raise an event if it finds records, but those records do not have call quality data. If you select Yes and this script raises this event, check the status of the job run by the SIPServer_CollectCallData Knowledge Script. The default is unselected.
Event severity when no records found	Set the event severity level, from 1 to 40, to indicate the importance of an event in which no SIP packets were found. The default is 25.
Call Details	
Include call details?	Select **Yes** to include call details in the events raised by this script. The default is Yes. Leave this parameter unselected to suppress the following call details: ◆ Calling Party ◆ Called Party ◆ Caller and Called Average MOS ◆ Caller and Called Average R-Value ◆ Caller and Called Jitter ◆ Caller and Called Latency ◆ Caller and Called Lost Packets ◆ Caller and Called Codec ◆ Connect Time ◆ Disconnect Time ◆ Duration Calling Party and Called Party details usually contain phone numbers, such as station extensions. If calls are made from named SIP user agents or gateways, rather than phones with numbers assigned, the Agent ID or gateway name might display instead of a called or calling phone extension.
Query Filters	Use the following parameters to filter the list of call data records. **Note** Using a quotation mark character (") in a filter parameter causes an error event, unless the character is duplicated. To work around this issue, replace any instance of a quotation mark character with two quotation mark characters. For example, if you want to use `"MyCallerID"` in a filter, write it as `""MyCallerID""`.
Minimum duration	Use this parameter to filter out records whose call duration is less than the value you specify. Accept the default of 0 seconds to ignore the filter for minimum duration.
Maximum table size	Specify the maximum number of detail rows to include in an event message. The default is 50 rows.
Maximum duration	Use this parameter to filter out records whose call duration is greater than or equal to the value you specify. Accept the default of 0 seconds to ignore the filter for maximum duration.

Parameter	How to Set It
Calling party	Specify the calling party involved in the call that you want to find in the supplemental database. Using an asterisk before and after the search string helps you find a specific numbered extension that may include brackets. For example, specifying `*sip:51006@*` would return the following SIP phone: `<sip:51006@netiqavayasm.ral>`. Leave this parameter blank to search for any call party.
Party connector	Set this parameter only if you specified a party for both the *Calling party* parameter and the *Called party* parameter. Your selection indicates how the script will connect the two parameters: AND or OR. The default is AND.
Called party	Specify the second party involved in the call that you want to find in the supplemental database. Using an asterisk before and after the search string helps you find a specific numbered extension that may include brackets. For example, specifying `*sip:51006@*` would return the following SIP phone: `<sip:51006@netiqavayasm.ral>`. Leave this parameter blank to search for any call party.
Troubleshooting	
Call stop time range	Select a range of time and dates in which the query should search for data in the supplemental database. ◆ Select **Fixed Time** to select specific days and times that the query should begin and end. ◆ Select **Sliding** to select a number of hours, days, months, or years in which to search. The query starts its search at the time the job runs, and goes back through the supplemental database for the number of units you select. The default is Fixed Time. **NOTE:** This parameter is valid only when you select **Run once** on the Schedule tab.
Monitor Average MOS	
Event Notification	
Raise event if average MOS falls below threshold?	Select **Yes** to raise an event if the average MOS value falls below the threshold. The default is Yes.
Threshold - Average MOS	Specify the lowest average MOS value, from 1.0 to 5.0, that must occur to prevent an event from being raised. The default is 3.60.
Event severity when average MOS falls below threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the average MOS value falls below the threshold. The default is 5.
Data Collection	
Collect data for average MOS?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the average MOS value during the monitoring period. The default is unselected.
Monitor Average R-Value	

Parameter	How to Set It
Event Notification	
Raise event if average R-Value falls below threshold?	Select **Yes** to raise an event if the average R-Value falls below the threshold. The default is Yes.
Threshold - Average R-Value	Specify the lowest average R-Value, from 0 to 100, that must occur to prevent an event from being raised. The default is 70.
Event severity when average R-Value falls below threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the average R-Value falls below the threshold. The default is 5.
Data Collection	
Collect data for average R-Value?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the average R-Value during the monitoring period. The default is unselected.
Monitor Average Jitter	
Event Notification	
Raise event if jitter exceeds threshold?	Select **Yes** to raise an event if the average jitter value exceeds the threshold. The default is Yes.
Threshold - Maximum jitter	Specify the highest average jitter value, in milliseconds, that can occur before an event is raised. The default is 60 milliseconds.
Event severity when jitter exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the average jitter value exceeds the threshold. The default is 15.
Data Collection	
Collect data for jitter?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the amount of average jitter that occurred during the monitoring period. The default is unselected.
Monitor Average Latency	
Event Notification	
Raise event if latency exceeds threshold?	Select **Yes** to raise an event if the average latency value exceeds the threshold. The default is Yes.
Threshold - Maximum latency	Specify the highest amount of average latency, in milliseconds, that can occur before an event is raised. The default is 400 milliseconds.
Event severity when latency exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the average latency value exceeds the threshold. The default is 15.
Data Collection	
Collect data for latency?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the amount of latency that occurred during the monitoring period. The default is unselected.
Monitor Average Packet Loss	
Event Notification	
Raise event if packet loss exceeds threshold?	Select **Yes** to raise an event if the average packet loss value exceeds the threshold. The default is Yes.

Parameter	How to Set It
Threshold - Maximum packet loss	Specify the highest percentage of average packet loss that can occur before an event is raised. The default is 1%.
Event severity when packet loss exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the packet loss value exceeds the threshold. The default is 15.
Data Collection	
Collect data for packet loss?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of packet loss that occurred during the monitoring period. The default is unselected.

3.1.6 Triggering Call and Phone Quality Diagnoses

You can use NetIQ Vivinet Diagnostics to diagnose problems identified by SIPServer Knowledge Scripts.

Using the existing methodology of launching an Action script based on an event, AppManager can launch Action_DiagnoseVoIPQuality to trigger Vivinet Diagnostics to diagnose the problem for events raised by the SIPServer_CallQuality Knowledge Script. SIPServer_CallQuality events trigger Vivinet Diagnostics to diagnose the problem when average MOS, average R-Value, average jitter, average latency, and average packet loss fall below or exceed their thresholds.

The Action script runs by default only if Vivinet Diagnostics 2.3 or later is installed on the computer on which the script is running.

To trigger Vivinet Diagnostics:

1. When setting parameter values for the CallQuality script, click the **Actions** tab. Action_DiagnoseVoIPQuality is selected by default.
2. Click **Properties** and enter values for all parameters for the Action script. For more information about the parameter values, click **Help** on the Properties for Action_DiagnoseVoIPQuality dialog box.

For more information about Vivinet Diagnostics and call quality diagnoses, see the *User Guide for Vivinet Diagnostics* and the Help for the Action_DiagnoseVoIPQuality Knowledge Script.

3.2 CollectCallData

Use this Knowledge Script to monitor the availability of call data for SIP Quality of Server (QoS) sources. This script raises an event when the SIP QoS call data collection is unavailable or available, and it also raises an event when call data collection raises a warning for any reason, including errors that prevent an individual data record from being saved to the database.

3.2.1 Resource Object

SIPServer Call Data folder

3.2.2 Default Schedule

By default, this script runs **every 5 minutes**. You can set the schedule interval in seconds, minutes or hours, or you select the option to run the script once.

3.2.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the CollectCallData job encounters a problem that prevents it from running, such as an invalid parameter, an invalid object detail, or a missing required system resource. The default is 5.
Raise event if call data is unavailable?	Select **Yes** to raise an event when call data is unavailable for any reason, including a failure to start the collection of call data. The default is Yes. The event message lists the reason for the interruption. The reasons include: ◆ The NetIQ Call Data Collector Server Windows service is stopped and cannot be started. ◆ The SIP call data collector cannot start because the UDP port is in use by another application or collector. ◆ The supplemental database does not exist. ◆ The supplemental database exists, but it is down. ◆ The supplemental database exists and is running, but it cannot be accessed. ◆ The supplemental database exists, is running and can be accessed, but writes to the database fail.
Event severity when call data is unavailable	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the call data is not available for any reason. The default is 5.
Raise event if call data collection warning?	Select **Yes** to raise an event if call data collection raises a warning for any reason, including errors that prevent an individual data record from being saved to the database. The default is Yes.
Event severity when call data collection warning	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the call data collection raises a warning. The default is 15.
Raise event if call data is available?	Select **Yes** to raise an event if call data is available. The default is unselected.
Event severity when call data is available	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the call data is available since the last iteration. The default is 25.
Monitor Call Data Availability	
Data Collection	
Collect data for call data availability?	Select **Yes** to collect data for the availability of call data. A 100 indicates that call data was available throughout the monitoring interval, and a 0 indicates otherwise, such as at least one interruption occurred. The default is unselected. If you select the schedule option of run once, the script will not report data, because no “end of interval” point exists. As a result, the script will not report data even if you selected Yes for this parameter.

3.3 SetupSupplementalDB

Use this Knowledge Script to create an SIP Server supplemental database, including the tables and stored procedures needed to store call detail records (CDRs) and voice quality monitoring for a SIP server. In addition, this script creates a SQL Server job that removes old records from the supplemental database.

You can also create the SIP Server supplemental database using the *Set up supplemental database?* parameters in the Discovery_SIPServer Knowledge Script.

For more information, see Section 3.3.4, "Understanding the Supplemental Database," on page 46.

3.3.1 Resource Object

SIP Server Call Data folder

3.3.2 Default Schedule

By default, this script runs **once**.

3.3.3 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the SetupSupplementalDB job. The default is 5.
Raise event if database set up succeeds?	Select **Yes** to raise an event if creation of the SIP Server supplemental database is successful. The default is unselected.
Event severity when database setup succeeds	Set the event severity level, from 1 to 40, to indicate the importance of the success of the creation of the SIP Server supplemental database. The default is 25.

Parameter	How to Set It
Start pruning job on supplemental database?	**For all supported versions of SQL Server, except SQL Server Express versions**: Set to **Yes** to create a SQL job that deletes data from the supplemental database. The SQL job runs every night. The default is Yes. Data is deleted from the supplemental database based on the value you specify in the *Number of days to keep call detail records* parameter. **For SQL Server Express versions**: Set to **No**. The pruning job is not supported for SQL Server Express versions. **To manually delete data from the supplemental database**: 1. Run the following stored procedure from a command line: `osql -E -S <sql server> -n -d <database> -Q "exec dbo.Task_SIPServer_Pruning"` where <*sql server*> is the name of the server that hosts the supplemental database, and where <`database`> is the name of the supplemental database. For example: `osql -E -S SuppDBSIPServer -n -d SIPServer_S8300-Cluster -Q "exec dbo.Task_SIPServer_Pruning"` 2. Configure a Windows Scheduled Task to schedule pruning at an interval of your choosing. The process for configuring a Windows Scheduled Task varies according to your version of Microsoft Windows. Consult your Windows documentation for more information.
Number of days to keep call detail records	Specify the number of days' worth of CDRs you want to keep in the SIP Server supplemental database. Data older than what you specify is discarded. The default is 7 days.
SQL Server Information	
SQL Server instance name	Specify the instance name of the SQL Server where you want to create the new SIP Server supplemental database. Leave this parameter blank to use the default SQL server instance on the proxy agent computer.
SQL database user name	Specify the user name for the SQL Server where you want to create the new SIP Server supplemental database. Leave this parameter blank to use Windows authentication instead of SQL authentication.

3.3.4 Understanding the Supplemental Database

The SIP Server supplemental database is a Microsoft SQL Server database you create for the proxy agent computer. The supplemental database fulfills several functions.

Storage for CDRs and SIP packets

The managed object on the proxy agent computer receives call detail records (CDRs) from SIP servers and SIP packets from phones registered to SIP servers. The SIPServer_CollectCallData Knowledge Script starts and monitors the complete call data collection on the proxy agent computer, and it saves the CDR and SIP packet data to tables in the SIP Server supplemental database. The SIPServer_CallQuality and SIPServer_UserAgentQuality Knowledge Scripts query the supplemental database for the data they need.

When you start the SIPServer_CallQuality Knowledge Script job, the job starts a collection task in the NetIQ Call Data Collector Server Windows service that begins collecting CDR and SIP data to store in the SIP Server supplemental database.

When you create the supplemental database, you specify how long data is retained before being deleted. AppManager automatically deletes CDRs older than the retention age you specify.

To create and use the supplemental database:

1. **Create the database**. Create one SIP Server supplemental database per SIP Server cluster you are monitoring. Use the Discovery_SIPServer or SIPServer_SetupSupplementalDB Knowledge Script for this purpose.
2. **Monitor the data in the database**. Use the SIPServer_CallQuality or SIPServer_UserAgentQuality scripts to monitor jitter, latency, lost data, R-Value, and MOS data in the database.

3.4 UserAgentQuality

A *SIP user agent* is a logical network endpoint that can send and receive SIP messages. A user agent performs the role of a user agent *client*, which sends SIP requests, and the user agent *server*, which receives the requests and returns a SIP response.

Use this Knowledge Script to continuously monitor SIP packet information stored in the SIP Server supplemental database for quality of service (QOS) statistics for a SIP user agent.

This script monitors jitter, latency, packet loss, Mean Opinion Score (MOS), and R-Value for a SIP user agent. This script raises an event if a monitored value exceeds or falls below a threshold. MOS and R-Value are computed only for calls that use one of the following codecs: G.711u, G.711a, or G.729.

NOTE: You can trigger NetIQ Vivinet Diagnostics to diagnose the problem indicated by an event in which the percentage lost data threshold is exceeded. For more information, see Section 3.1.6, "Triggering Call and Phone Quality Diagnoses," on page 42.

The purpose of this script is two-fold:

- **Troubleshooting**. In troubleshooting mode, this script runs once and checks the supplemental database tables for calls that disconnected within the range you select in the *Call disconnect time range* parameter. Select **Run once** on the **Schedule** tab to run this script in troubleshooting mode.

- **Diagnosing**. In diagnostic mode, this script works in conjunction with NetIQ Vivinet Diagnostics to diagnose VoIP quality problems detected during monitoring. If a call quality threshold is exceeded, then, by default, this script launches Action_DiagnoseVoIPQuality, a Knowledge Script that in turn launches Vivinet Diagnostics to generate a diagnosis of the problem.

  To turn off diagnostic mode, click the Actions tab, select **Action_DiagnoseVoIPQuality**, and click **Delete**. Turning diagnostic mode off or on does not affect the events raised by this script.
- **Monitoring**. In monitoring mode, this script checks the supplemental database tables at each specified interval for new records that match your query. You always run the script in monitoring mode unless you select **Run once** on the **Schedule** tab.

3.4.1 Resource Objects

SIP Server Call Data folder

3.4.2 Prerequisites

- Run SIPServer_SetupSupplementalDB to create the SIP Server supplemental database.
- Because the SIPServer_UserAgentQuality script reports on data stored in the supplemental database by a data collector service, data must exist in the supplemental database before the reporting can be successful. To place data in the supplemental database, run SIPServer_CollectCallData on the SIP Server being monitored before you run the SIPServer_UserAgentQuality script. If the SIPServer_CollectCallData script stops, the data collection also stops, even if the SIPServer_UserAgentQuality script is still running.

3.4.3 Default Schedule

By default, this script runs **every 30 seconds**.

3.4.4 Setting Parameter Values

Set the following parameters as needed:

Parameter	How to Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to indicate the importance of the failure of the UserAgentQuality job. The default is 5.
Monitor Settings	
User agent to monitor	Specify the name of the user agent you want to monitor, such as `user@domain`. If you do not specify a domain, the script will include any user agent with the correct name that reports RFC6035 statistics.
Troubleshooting	

Parameter	How to Set It
Call disconnect time range	Select a range of time and dates in which the query should search for data in the supplemental database. This parameter is valid *only* when you select **Run once** on the Schedule tab. ◆ Select **Fixed Time** to select specific days and times for the query to begin and end. ◆ Select **Sliding** to select a number of hours, days, months, or years in which to search. The query starts its search at the time the job runs, and goes back through the supplemental database for the number of units you select. The default is Fixed Time.
Monitor Interval MOS	
Event Notification	
Raise event if interval MOS falls below threshold?	Select **Yes** to raise an event if the interval MOS value falls below the threshold. The default is Yes.
Threshold - Interval MOS	Specify the lowest interval MOS value, from 1.0 to 5.0, that must occur to prevent an event from being raised. The default is 3.60.
Event severity when interval MOS falls below threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the interval MOS value falls below the threshold. The default is 5.
Data Collection	
Collect data for interval MOS?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the interval MOS value during the monitoring period. The default is unselected.
Monitor Interval R-Value	
Event Notification	
Raise event if interval R-Value falls below threshold?	Select **Yes** to raise an event if the interval R-Value falls below the threshold. The default is Yes.
Threshold - Interval R-Value	Specify the lowest interval R-Value, from 0 to 100, that must occur to prevent an event from being raised. The default is 70.
Event severity when interval R-Value falls below threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the interval R-Value falls below the threshold. The default is 5.
Data Collection	
Collect data for interval R-Value?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the interval R-Value during the monitoring period. The default is unselected.
Monitor Interval Jitter	
Event Notification	
Raise event if interval jitter exceeds threshold?	Select **Yes** to raise an event if the interval jitter value exceeds the threshold. The default is Yes.
Threshold - Maximum interval jitter	Specify the highest interval jitter value, in milliseconds, that can occur before an event is raised. The default is 60 milliseconds.

Parameter	How to Set It
Event severity when interval jitter exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the interval jitter value exceeds the threshold. The default is 15.
Data Collection	
Collect data for interval jitter?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the amount of interval jitter that occurred during the monitoring period. The default is unselected.
Monitor Interval Latency	
Event Notification	
Raise event if interval latency exceeds threshold?	Select **Yes** to raise an event if the interval latency value exceeds the threshold. The default is Yes.
Threshold - Maximum interval latency	Specify the highest amount of interval latency, in milliseconds, that can occur before an event is raised. The default is 400 milliseconds.
Event severity when interval latency exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the interval latency value exceeds the threshold. The default is 15.
Data Collection	
Collect data for interval latency?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the amount of latency that occurred during the monitoring period. The default is unselected.
Monitor Interval Packet Loss	
Event Notification	
Raise event if interval packet loss exceeds threshold?	Select **Yes** to raise an event if the interval packet loss value exceeds the threshold. The default is Yes.
Threshold - Maximum interval packet loss	Specify the highest percentage of interval packet loss that can occur before an event is raised. The default is 1%.
Event severity when packet interval loss exceeds threshold	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the packet loss value exceeds the threshold. The default is 15.
Data Collection	
Collect data for interval packet loss?	Select **Yes** to collect data for charts and reports. If enabled, data collection returns the percentage of packet loss that occurred during the monitoring period. The default is unselected.