NetIQ® AppManager® for IP Phone Quality

Management Guide

September 2010

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 NetIQ Web site: www.netiq.com/support/am/extended/documentation/default.asp?version=AMDocumentation.

Conventions

The library uses consistent conventions to help you identify items throughout the documentation. The following table summarizes these conventions.

Convention	Use
Bold	• Window and menu items • Technical terms, when introduced
Italics	• Book and CD-ROM titles • Variable names and values • Emphasized words
`Fixed Font`	• File and folder names • Commands and code examples • Text you must type • Text (output) displayed in the command-line interface
Brackets, such as *[value*]	• Optional parameters of a command
Braces, such as {*value}*	• Required parameters of a command
Logical OR, such as *value1*|*value2*	• Exclusive parameters. Choose one parameter.

About NetIQ Corporation

NetIQ, an Attachmate business, is a global leader in systems and security management. With more than 12,000 customers in over 60 countries, NetIQ solutions maximize technology investments and enable IT process improvements to achieve measureable cost savings. The company's portfolio includes award-winning management products for IT Process Automation, Systems Management, Security Management, Configuration Audit and Control, Enterprise Administration, and Unified Communications Management. For more information, please visit www.netiq.com.

Contacting Sales Support

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

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

Contacting Technical Support

For specific product issues, please 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 the Online User Community

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

Chapter 1

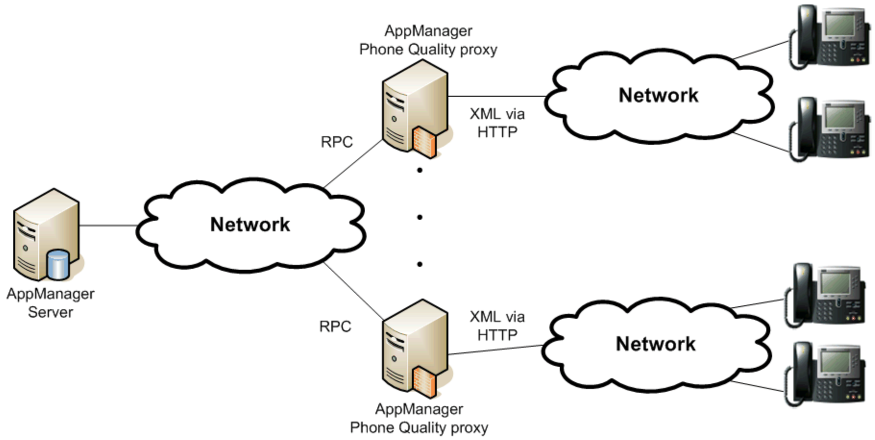
Introducing AppManager for IP Phone Quality

AppManager for IP Phone Quality allows you to obtain call quality statistics directly from Cisco IP phones. This chapter briefly describes the module, explaining how it works and identifying key features and benefits.

Brief Overview

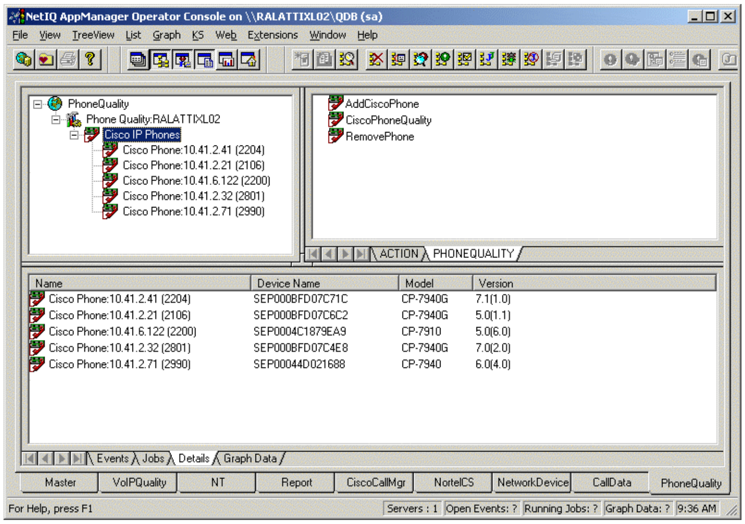
The AppManager for IP Phone Quality module obtains call quality statistics directly from Cisco IP phones by regularly polling selected phones to determine whether they are active (that is, engaged in a phone call). Once activity has been established, AppManager polls more frequently to collect data points as the call progresses. If any call quality metric falls below or exceeds a threshold, AppManager raises an event (only one per call) while the call is active.

The following diagram illustrates the architecture of the AppManager for IP Phone Quality module.



This module is intended for troubleshooting and to supplement information gathered by other VoIP modules, such as AppManager for Cisco CallManager. You should use it to monitor a limited number of phones. For more information, see "Reviewing Supported Phones" on page 5.

To gather call quality statistics, drop a Monitoring Knowledge Script on the phones underneath the PhoneQuality object in the TreeView pane. The TreeView clearly identifies the phones that are available for monitoring.


NetIQ AppManager Operator Console on \\RALATTIXL02\QDB (sa)
File View TreeView List Graph KS Web Extensions Window Help
PhoneQuality
Phone Quality:RALATTIXL02
Cisco IP Phones
Cisco Phone:10.41.2.41 (2204)
Cisco Phone:10.41.2.21 (2106)
Cisco Phone:10.41.6.122 (2200)
Cisco Phone:10.41.2.32 (2801)
Cisco Phone:10.41.2.71 (2990)
AddCiscoPhone
CiscoPhoneQuality
RemovePhone
ACTION
PHONEQUALITY
Events
Jobs
Details
Graph Data
Master
VoIPQuality
NT
Report
CiscoCallMgr
NortelCS
NetworkDevice
CallData
PhoneQuality
For Help, press F1
Servers : 1
Open Events: ?
Running Jobs: ?
Graph Data: ?
9:36 AM

Name	Device Name	Model	Version
Cisco Phone:10.41.2.41 (2204)	SEP000BFD07C71C	CP-7940G	7.1(1.0)
Cisco Phone:10.41.2.21 (2106)	SEP000BFD07C6C2	CP-7940G	5.0(1.1)
Cisco Phone:10.41.6.122 (2200)	SEP0004C1879EA9	CP-7910	5.0(6.0)
Cisco Phone:10.41.2.32 (2801)	SEP000BFD07C4E8	CP-7940G	7.0(2.0)
Cisco Phone:10.41.2.71 (2990)	SEP00044D021688	CP-7940	6.0(4.0)

Features and Benefits

The following are just a few of the features and benefits of monitoring IP phone call quality with AppManager:

- Because datapoints are collected while a call is active, you can see the progress of a call
- If a threshold is crossed, an event can be raised while a call is active, allowing you to troubleshoot the problem *before* the call ends
- Call metrics that are not available from the CallManager call detail records, such as maximum jitter, *are* available directly from the phone

Licensing

AppManager for IP Phone Quality is provided for free with the purchase and license of any of the following AppManager modules: Cisco CallManager, Cisco CallManager Express, VoIP Quality, and Call Data Analysis.

Chapter 2

Installing AppManager for IP Phone Quality

This chapter provides installation instructions and describes system requirements for AppManager for IP Phone Quality.

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 Web site: https://www.netiq.com/support/am/extended/documentation/default.asp.

System Requirements

AppManager for IP Phone Quality has the following system requirements:

Software/Hardware	Version
NetIQ AppManager installed on the repository, console, and proxy agent computers	7.x
Microsoft operating system installed on the proxy agent computers	One of the following: • 32-bit Windows 2000 • 32- or 64-bit Windows Server 2003 • 32- or 64-bit Windows Server 2008, including R2 **Note** Support for Windows Server 2008 requires the AppManager agent for Windows Server 2008. For more information, see the NetIQ Technical Support Web site.
Cisco IP phones	NetIQ has tested AppManager for IP Phone Quality with Cisco IP phone models 7910, 7940, 7960, and 7970, using Cisco CallManager versions 3.3(4), 4.0, and 4.1.

For the latest information about supported software versions and the availability of module updates, visit the AppManager Supported Products page at www.netiq.com/support/am/supportedproducts/default.asp. If you encounter problems using this module with a later version of your application, contact NetIQ Technical Support.

For more information about system requirements for the AppManager agent, repository, and management server, see the *Installation Guide for AppManager.*

Installing the Module

The setup program automatically identifies and updates all relevant AppManager components on a computer. Therefore, run the setup program only once on any computer. The pre-installation check also runs automatically when you launch the setup program.

You can install the module in one of the following ways:

- Run the module setup program, `AM70-PhoneQuality-7.x.x.0.msi`, which you downloaded from the Web. Save the module setup files on the distribution computer, and then delete the older versions of the module setup files. For more information about the distribution computer, see the *Installation Guide for AppManager.*
- Use Control Center to install the module on the remote computer where an agent is installed. For more information, see "Deploying the Module with Control Center" on page 4.

To install the module:

1. Run the module setup program on all AppManager repository (QDB) computers to install the Knowledge Scripts and reports.
   - Run the setup program on the primary repository computer first. Then run the setup program on all other repository computers.
   - For repositories running in active/active and active/passive clusters, run the setup program on the active node. Then, copy the following Registry key to the non-active node.

     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0
     ```
2. Install the module on the Phone Quality computer you want to monitor (agent computer). Use one of the following methods:
   - Run the module setup program.
   - Use Control Center to deploy the installation package.
3. Run the module setup program on all Operator Console and Control Center computers to install the Help and console extensions.
4. *If you have not already discovered resources*, run the Discovery_PhoneQuality Knowledge Script on all agent computers where you installed the module. For more information, see "Discovering IP Phone Quality Resources" on page 6.
5. Upgrade running jobs for any Knowledge Script changes. For more information, see "Upgrading Knowledge Script Jobs" on page 7.

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

Deploying the Module with Control Center

You can use Control Center to deploy the module on a remote computer where an agent is installed. This topic briefly describes the steps involved in deploying a module and provides instructions for checking in the module installation package. For more information, see the *Control Center User Guide for AppManager*, which is available on the AppManager Documentation Web site: https://www.netiq.com/support/am/extended/documentation/default.asp.

Deployment Overview

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

To deploy the module on an agent computer:

1. Verify the default deployment credentials.
2. Check in an installation package.
3. Configure an e-mail address to receive notification of a deployment.
4. Create a deployment rule or modify an out-of-the-box deployment rule.
5. Approve the deployment task.
6. View the results.

Checking In the Installation Package

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

To check in a module installation package:

1. Log on to Control Center and navigate to the Administration pane.
2. In the Deployment folder, select **Packages**.
3. On the Tasks pane, click **Check in Packages**.
4. Navigate to the folder where you saved `AM70-PhoneQuality-7.x.x.0.xml` and select the file.
5. Click **Open**. The Deployment Package Check in Status dialog box displays the status of the package check in.

Silently Installing the Module

You can run the module setup program, `AM70-PhoneQuality-7.x.x.0.msi`, silently (without user intervention) from a command prompt on the local computer.

Run the following command from the directory in which you saved the module setup program. This command installs the module using default settings.

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

where *x.x* is the actual version number of the module setup program.

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

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

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

For more information, see “Performing a Silent Installation” in the *Installation Guide for AppManager*.

Verifying Your Installed Module

To verify installation on many computers, run the ReportAM_CompVersion Knowledge Script. Ensure you discover a report-enabled agent before running this script. For more information, see the Help for the script.

To verify installation on one or only a few computers, use the Operator Console.

To verify your installed module with the Operator Console:

1. In the TreeView pane, select the computer for which you want to verify your installed module.
2. From the TreeView menu, select **Properties**. On the System tab, the System information pane displays the version numbers for all modules installed on the computer.
3. Verify that the version number from the *AppManager for Phone Quality Readme* matches the version number shown in the System information pane.

Discovering IP Phone Quality Resources

Use the Discovery_PhoneQuality Knowledge Script to discover a Phone Quality managed object on the computer that will be used for monitoring Cisco IP phones. This script always raises an event when the job fails for any reason. By default, this script runs once.

When polling a phone to get device information, the managed object has a 20-second timeout period for each phone that it is attempting to contact. If you are rediscovering a lot of phones at one time, the Discovery job may take quite a while if phones cannot be contacted.

Supported Phones

AppManager for IP Phone Quality monitors a phone by accessing certain URLs on the phone's Web interface. Whether the phone supports these URLs depends on the phone's model and firmware version, which also regulate the type of information available from the phone.

NetIQ has tested AppManager for IP Phone Quality with Cisco IP phone models 7910, 7940, 7960, and 7970, using Cisco CallManager versions 3.3(4), 4.0, and 4.1.

Testing provided two noteworthy findings:

- Older versions of the 7910 firmware do not provide the model number in the device details.
- The 7970 phone provides maximum jitter but not average jitter.

Setting Parameter Values

Set the Values tab 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 reflect the importance of the failure of the Discovery_PhoneQuality job. The default is 5.

Parameter	How To Set It
Update phone details for existing phones?	Select **Yes** to poll device information from phones already in the TreeView pane. The default is Yes.
Raise event if discovery succeeds?	Select **Yes** to raise an event if discovery is successful. The default is Yes.
Event severity when discovery succeeds	Set the event severity level, from 1 to 40, to reflect the importance of an event in which discovery succeeds. The default is 25.
Raise event if discovery fails?	Select **Yes** to raise an event if discovery fails. The default is Yes.
Event severity when discovery fails	Set the event severity level, from 1 to 40, to reflect the importance when discovery fails. The default is 5.
Raise event if phone details cannot be updated?	Select **Yes** to raise an event if discovery encounters an error during when polling one or more phones for device information. The default is Yes.
Event severity when phone details cannot be updated	Set the event severity level, from 1 to 40, to reflect the importance when phone details cannot be updated. The default is 15.
Raise event if Phone Quality managed object not installed?	Select **Yes** to raise an event when the Phone Quality managed object is not installed. The default is Yes.
Event severity when Phone Quality managed object not installed	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the Phone Quality module is not installed at the time the discovery job is run. The default is 5.

Upgrading Knowledge Script Jobs

This release of AppManager for IP Phone Quality may contain updated Knowledge Scripts. You can push the changes for these updated script to running Knowledge Script jobs in one of the following ways:

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

Running AMAdmin_UpgradeJobs

The AMAdmin_UpgradeJobs Knowledge Script can push changes to running Knowledge Script jobs. Your AppManager repository (QDB) must be at version 7.0 or later and must have hotfix 71583 or later installed. To download the hotfix, see the AppManager Suite Hotfixes Web page.

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

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

Propagating Knowledge Script Changes

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

Before propagating script changes, verify that the script parameters are set to your specifications. Customized script parameters may have reverted to default parameters during the installation of the module. New parameters may need your attention.

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

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

Propagating Changes to Running Jobs

You can propagate the properties and the logic of a Knowledge Script to running jobs started by that Knowledge Script. Corresponding jobs are stopped and restarted with the Knowledge Script changes.

To propagate changes to running Knowledge Script jobs:

1. In the Knowledge Script pane, select the Knowledge Script for which you want to propagate changes.
2. On the KS menu, select **Properties propagation**, and then select **Ad Hoc Jobs**.
3. Select the components of the Knowledge Script that you want to propagate to associated running jobs:

Select	To propagate
Script	The logic of the Knowledge Script.
Properties	Values from the Knowledge Script Schedule and Values tabs.

Propagating Changes to Knowledge Script Groups

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

After you propagate script changes to Knowledge Script Group members, you can propagate the updated Knowledge Script Group members to associated running jobs. For more information, see “Propagating Changes to Running Jobs” on page 8.

To propagate Knowledge Script changes to Knowledge Script Groups:

1. In the Knowledge Script pane, select the Knowledge Script for which you want to propagate changes.
2. On the KS menu, select **Properties propagation**, and then select **Knowledge Script Group Members**.

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

Select	To propagate
Script	The logic of the Knowledge Script.
Properties	Values from the Knowledge Script Schedule and Values tabs.

Chapter 3
PhoneQuality Knowledge Scripts

AppManager for IP Phone Quality provides the following Knowledge Scripts for monitoring the voice quality of Cisco IP phones. From within the Operator Console, you can select a Knowledge Script in the Knowledge Script pane and press **F1** for complete details.

Knowledge Script	What It Does
AddCiscoPhone	Adds a Cisco IP phone to the TreeView pane.
CiscoPhoneQuality	Polls Cisco phones for voice quality statistics.
RemovePhone	Removes an IP phone from the TreeView pane.

AddCiscoPhone

Use this Knowledge Script to add a Cisco IP phone to the TreeView pane. You must add a phone before you can monitor it with the CiscoPhoneQuality script.

When polling a phone to get device information, the Knowledge Script job has a 20-second timeout period for each phone that it is attempting to contact. If you are adding many phones at one time, this Knowledge Script job may take quite a while if phones cannot be contacted.

Tips

- If you use the Cisco CallManager Extension Mobility feature, then you know that it allows a Cisco IP phone to assume different configurations based on the user who is logged into the phone. When you add an Extension Mobility-enabled phone with the AddCiscoPhone script, the details of the phone, including directory number, are also based on the user who is logged in. If no user is logged in, then phone details are based on your CallManager configuration. In some cases, the phone details may indicate a directory number of 0000. In others, the phone may not be added at all because the CallManager configuration for the directory number is blank. You should add the phone at a time when the regular user is logged on.
- After running AddCiscoPhone for the first time, press [F5] to refresh the Operator Console. Refreshing ensures that all PhoneQuality Knowledge Scripts are visible on the **PhoneQuality** tab of the Knowledge Script pane.

Resource Object

PHONEQ_CISCOPHONEF

Default Schedule

By default, this script runs once.

Setting Parameter Values

Set the following parameters as needed:

Description	How To Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the AddCiscoPhone job fails. The default is 5.
Configuration Settings	
List of Cisco phone IP addresses	Type the IP addresses (in dotted notation) of the Cisco phone that you want to monitor. You can type one address or a list of addresses. If you type a list, separate the addresses with a comma, like so: `10.46.4.15,10.46.4.149`. **Note** If you have more addresses than is convenient to enter in this field, you can list the addresses in a separate file and then use the following parameter to access that file.

Description	How To Set It

Full path to file with list of Cisco phone IP addresses	Type the full path to a file that contains a list of the IP addresses. Each address in the file should be on a separate line, like so:
	`10.46.4.15`
	`10.46.4.149`
	Because the file must be accessible from the agent, the path must be a local directory on the agent computer or a UNC path.
	Important If you type a UNC path, then the `netiqmc` service must be running as a user that has access to the path.

Event Notification	
Raise event if phone added successfully?	Select **Yes** to raise an event if the phone is successfully added to the TreeView pane. The default is Yes.
Event severity when phone added successfully	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the phone is added successfully. The default is 25.
Raise event if phone cannot be added?	Select **Yes** to raise an event if the phone cannot be added to the TreeView pane. The default is Yes.
Event severity when phone cannot be added	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the phone cannot be added to the TreeView pane. The default is 40.

CiscoPhoneQuality

Use this Knowledge Script to poll Cisco IP phones for voice quality statistics on active calls.

AppManager for IP Phone Quality collects or calculates the following call quality metrics:

- **Jitter**. Both average and maximum jitter are collected, if available on the monitored phone. Jitter, known to adversely affect call quality, indicates a variance in the arrival rate of datagrams sent during a VoIP call.

  When a datagram is sent, the sending phone gives it a timestamp. When the datagram is received, the receiving phone adds another timestamp. These two timestamps are used to calculate the datagram's transit time. If the transit times for datagrams within the same call are different, the call contains jitter. In a telephone call, jitter's effect may be similar to the effect of packet loss: some words may be missing or garbled.

  The amount of jitter in a call depends on the degree of difference between the datagrams' transit times. If the transit time for all datagrams is the same (no matter how long it took for the datagrams to arrive), the call contains no jitter. If the transit times differ slightly, the call contains some jitter. Jitter values provide a short-term measurement of network congestion and can also show the effects of queuing within the network.

- **Interval packet loss**. The percentage of packet loss is calculated from the number of packets received and the number of packets lost since the last poll. Measuring the packet loss that occurs during each interval, instead of over the entire call, provides a better indication of whether packet loss is occurring in short, dense periods. Short, dense periods of packet loss have a more severe impact on VoIP quality than loss that is spread over a longer period of time.

- **Listening R-value and listening MOS**. Using the performance metrics collected from monitored call, AppManager calculates an R-value, which summarizes the quality of the VoIP transmission.

  NetIQ uses a modified version of the ITU G.107 standard E-Model equation to make the R-value calculation. The E-Model algorithm evaluates the quality of a voice transmission by factoring in the "mouth-to-ear" characteristics of a speech path. These characteristics were derived from studies of user satisfaction with varying levels of transmission clarity and stability.

  The output of the E-Model's complex calculation is a single score called an *R-value*, which is derived from delays and equipment impairment factors.

  R-values range from 100 (excellent) to 0 (poor). Closely related to an R-value is an estimated Mean Opinion Score (MOS). MOS scores range from 1.0 to 5.0, where 2.6 and below indicate nearly all users are dissatisfied with the call, and 4.3 and above indicate that users are very satisfied with the call.

  AppManager for IP Phone Quality uses metrics available from the phones to calculate VoIP metrics: the codec and the number of lost data packets. The result of the calculation is a *listening* R-value and a *listening* MOS. The term "listening" indicates that the values do not include "interactive" or "conversational" characteristics such as delay.

Among the details within a quality data stream is the following information:

- Remote IP address and port number
- Number of datagrams sent and lost for packet loss statistics
- Codec for MOS and R-value statistics

Although this information can be useful, over time it can take up a significant amount of space in the repository. To gather only data points, and not the data discussed above, click the Advanced tab and select **Collect only data point** in the Data options panel.

Understanding Polling Intervals

The most significant feature of AppManager for IP Phone Quality is the *polling interval*. Polling is the process by which AppManager contacts the IP phone for call quality statistics; the interval is the frequency with which the polling takes place.

You control the polling interval using the parameters in the CiscoPhoneQuality Knowledge Script. Before you set these parameters, however, you should understand the two types of intervals: *no call in progress* and *call in progress*.

- Initially, a phone is polled according to the “no call in progress” polling interval, which is less frequent than the “call in progress” interval, consisting of three possible states:
  - No call is in progress, and there have been no calls since the last polling period. In this state, the Knowledge Script performs no actions, such as gathering data or generating events.
  - No call is in progress, but there was a call since the last polling period. In this state, the Knowledge Script again takes no action. In this state, however, a call has been missed. A call of shorter duration than your polling interval can begin and end before AppManager has a chance to poll again.
  - A call is in progress. The polling interval is speeded up according to the “call in progress” interval, and data points and events are generated.
- Once in the “call in progress” state, polling continues at the faster rate until the phone call ends. The generation of data points and events also continues until the phone call ends. Once the call ends, polling resumes at the “no call in progress” interval.

Understanding When Events are Raised

If, while AppManager is polling an IP phone, a call quality metric falls below or exceeds a threshold, then AppManager raises an event while the call is active. However, AppManager raises *only* one event per call.

For instance, if the jitter threshold is crossed, then AppManager will raise an event. If, 30 seconds later during the same call, the MOS threshold is crossed, AppManager will not raise another event.

Configuring a Diagnosis Action

By raising only one event per call, you are guaranteed to invoke NetIQ Vivinet Diagnostics only once per call, fIf you use Vivinet Diagnostics to diagnose call quality problems between your IP phones. Use the **Actions** tab to configure Action_DiagnoseVoIPQuality to trigger Vivinet Diagnostics run a Diagnosis when an event is raised.

On the Actions tab, click **New**. In the **Action** list, select **Action_DiagnoseVoIPQuality**. Click **Properties** and set the parameters for the Action script. For more information, see the script’s Help.

For more information about triggering a Diagnosis, see the *Vivinet Diagnostics User Guide*.

Deciding Which Phones to Monitor

AppManager for IP Phone Quality is intended to monitor a limited number of phones. You should monitor no more than 100 phones.

If you need to monitor more than 100 phones, consider these guidelines:

- CPU usage on the agent computer is affected by the number of calls in progress.
- A Windows XP computer with a 1.5 GHz processor should be able to handle 100 in-progress calls.

- To monitor more than 100 in-progress calls, consider getting a faster computer or change the Polling interval when call is in progress parameter to a higher value for less-frequent polling.

When deciding which phones to monitor, consider the number of locations that you have and whether a phone is used often. For on-going monitoring, you may, for example, choose to monitor the receptionist's phone at each location. And, you can monitor additional phones on an as-needed basis for troubleshooting.

Prerequisites

- In order for AppManager to collect call quality statistics from Cisco IP phones, you must enable the Web interface on each phone that you want to monitor.
- Run AddCiscoPhone to add phones to the TreeView of the Operator Console.

Resource Object

PHONEQ_CISCOPHONEOBJ

Default Schedule

By default, this script runs on an asynchronous schedule.

Setting Parameter Values

Set the following parameters as needed:

Description	How To Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to reflect the importance of the failure of the CiscoPhoneQuality job. The default is 5.
Monitor Settings	
Polling interval when no call is in progress	Specify the number of seconds that will elapse between polling instances when no call is in progress. The default is 120 seconds.
Polling interval when call is in progress	Specify the number of seconds that will elapse between polling instances when a call *is* in progress. The default is 30 seconds.
Additional fixed delay for MOS/R-value calculation	Enter an amount of delay (in milliseconds) that you want to add to a call. This amount of delay is in addition to the three other types of delay that are associated with calculating MOS and R-value: • Network delay in one direction. This value is not available from the phones and so is set to 0 for calculating MOS and R-value. A network delay of 0 provides a "Listening" MOS/R-value because it does not take into consideration the interaction of a t wo-way conversation. • Packetization delay. This value is fixed, based on the type of codec being used. • Jitter buffer delay. This value is fixed, based on the type and size of the jitter buffer being used.
Event Notification	

Description	How To Set It
Raise event if error occurs during polling?	Select **Yes** to raise an event if an error occurs during the polling of the selected IP phone. The default is Yes. Among the possible errors are the inability of AppManager to contact the IP phone and the improper formatting of the gathered statistics.
Event severity when error occurs during polling	Set the event severity level, from 1 to 40, to reflect the importance of an event in which an error occurred during phone polling. The default is 5.
Monitor Listening MOS	
Event Notification	
Raise event if Listening MOS falls below threshold?	Select **Yes** to raise an event if the value of Listening MOS falls below the threshold that you set. The default is Yes.
Threshold - Minimum Listening MOS	Specify the lowest Listening MOS value that can be calculated before an event is raised. The default is 3.6.
Event severity when Listening MOS falls below threshold	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the Listening MOS value falls below the threshold you set. The default is 5.
Data Collection	
Collect data for Listening MOS?	Select **Yes** to collect Listening MOS data for charts and graphs. The default is Yes. The Listening MOS data stream is calculated based on the statistics available from the phone.
Monitor Listening R-value	
Event Notification	
Raise event if Listening R-value falls below threshold?	Select **Yes** to raise an event if the Listening R-value falls below the threshold that you set. The default is unchecked.
Threshold - Minimum Listening R-value	Specify the lowest Listening R-value that can be calculated before an event is raised. The default is 70.
Event severity when Listening R-value falls below threshold	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the Listening R-value falls below the threshold you set. The default is 5.
Data Collection	
Collect data for Listening R-value?	Select **Yes** to collect Listening R-value data for charts and graphs. The default is unselected. The Listening R-value data stream is calculated based on the statistics available from the phone.
Monitor Average Jitter	
Event Notification	
Raise event if average jitter exceeds threshold?	Select **Yes** to raise an event if the amount of average jitter exceeds the threshold that you set. The default is Yes.
Threshold - Maximum average jitter	Specify the highest amount of average jitter that can be achieved before an event is raised. The default is 60 milliseconds.
Event severity when average jitter exceeds threshold	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the amount of average jitter exceeds the threshold you set. The default is 15.
Data Collection	
Collect data for average jitter?	Select **Yes** to collect average jitter data for charts and graphs. The default is Yes. The average jitter data stream is calculated by the phone and represents the average jitter, so far, over the duration of the call.

Description	How To Set It
Monitor Maximum Jitter	
Event Notification	
Raise event if maximum jitter exceeds threshold?	Select **Yes** to raise an event if the amount of maximum jitter exceeds the threshold that you set. The default is Yes.
Threshold - Highest maximum jitter	Specify the highest amount of maximum jitter that can be achieved before an event is raised. The default is 60 milliseconds.
Event severity when maximum jitter exceeds threshold	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the amount of maximum jitter exceeds the threshold you set. The default is 15.
Data Collection	
Collect data for maximum jitter?	Select **Yes** to collect maximum jitter data for charts and graphs. The default is Yes. The maximum jitter data stream is calculated by the phone and represents the maximum jitter, so far, over the duration of the call.
Monitor Interval Packet Loss	
Event Notification	
Raise event if interval packet loss exceeds threshold?	Select **Yes** to raise an event if the percentage of interval packet loss exceeds the threshold that you set. 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 interval packet loss exceeds threshold	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the percentage of interval packet loss exceeds the threshold you set. The default is 15.
Data Collection	
Collect data for interval packet loss?	Select **Yes** to collect interval packet loss data for charts and graphs. The default is Yes. The interval packet loss data stream is calculated based on the number of lost and received packets (these metrics are provided by the phone). Represents the percentage of packets lost over the previous polling interval. By calculating the percentage of packet loss on an interval, you get a better indication of the density of any packet loss.

Troubleshooting

Consult the following topics for solutions to problems and answers for frequently asked questions. For help with any issue, contact Technical Support: www.netiq.com/support.

Problem	Description
Phone details shown as “unknown”	**Problem**: In the Details tab, the description of a phone indicates “unknown.” **Solution**: Phone details do not appear in the Details tab if they are not available from the phone’s Web interface. The availability of phone details varies by phone model and firmware version. Ensure you have enabled the Web interface for each phone you want to monitor.

Problem	Description
Datastreams not created for CiscoPhoneQuality job	**Problem**: As you run a CiscoPhoneQuality Knowledge Script job, you notice that no datastreams are being collected, even though you enabled data collection. In addition, no events are raised to indicate that the phone cannot be polled. **Solution**: Verify one or both of the following possible causes: • Was the call in progress for less time than the polling interval that is configured in the Knowledge Script? The script collects data only while a call is in progress. If the call begins and ends between polling instances, AppManager will miss the call. • Is the locale of the phone set to a language other than English? The Phone Quality managed object makes use of the Row Status value found on the phone’s Streaming Statistics Web page to determine when a call is in progress. When the locale is set to English, this value is “Active.” NetIQ has tested the Phone Quality module with several locales, including German, French, Italian, and Japanese. If you believe that the Phone Quality managed object is not recognizing the locale of your phone, contact NetIQ Technical Support and provide the following information:
Directory number is “0000” or blank	**Problem**: After adding a phone, you notice that the Directory Number in the phone details is either “0000” or is blank. **Solution**: More than likely, the phone you added has the Extension Mobility feature enabled. If you use the CallManager Extension Mobility feature, then you know that it allows a Cisco IP phone to assume different configurations based on the user who is logged into the phone. When you add an Extension Mobility-enabled phone with the AddCiscoPhone script, the details of the phone, including directory number, are also based on the user who is logged in. If no user is logged in, then phone details are based on your CallManager configuration. In some cases, the phone details may indicate a directory number of 0000. In others, the phone may not be added at all because the CallManager configuration for the directory number is blank. You should add the phone at a time when the regular user is logged on. In addition, understand that the TreeView and data stream legends are not automatically updated when a directory number changes because a different user has logged on.

RemovePhone

Use this Knowledge Script to remove an IP phone from the TreeView pane.

When this Knowledge Script job runs successfully, the data source object in the TreeView pane is deleted. In addition, the job itself is also deleted, which is a normal side-effect of removing a TreeView object. The event that this job creates is not deleted because it is associated with the parent object (`PhoneQuality:<computer name>` object). However, you can set global preferences to to delet an event when the object and job are deleted.

To remove an event when a job is deleted:

1. On the File menu in the Operator Console, select **Preferences**, select **Repository**, and then select **Event**.
2. Select **Remove associated events when jobs are deleted**.
3. Click **OK**.

Tips

- After running this script on the object of the phone that you want to remove, double-check your selection in the Objects tab. By specifically selecting a phone object from the Objects tab, you will not accidentally remove a phone that you want to keep.
- Before attempting to remove a phone, stop any monitoring jobs that are running on the phone.

Resource Object

PHONEQ_CISCOPHONEOBJ

Default Schedule

By default, this script runs once.

Setting Parameter Values

Set the following parameters as needed:

Description	How To Set It
General Settings	
Job Failure Notification	
Event severity when job fails	Set the event severity level, from 1 to 40, to reflect the importance of the failure of the RemovePhone job. The default is 5.
Event Notification	
Raise event if phone removal succeeds?	Select **Yes** to raise an event if the phone is successfully removed from the TreeView pane. The default is Yes.
Event severity when phone removal succeeds	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the phone is successfully removed from the TreeView pane. The default is 25.
Raise event if phone removal fails?	Select **Yes** to raise an event if the phone removal attempt fails. The default is Yes.
Event severity when phone removal fails	Set the event severity level, from 1 to 40, to reflect the importance of an event in which the phone removal attempt fails. The default is 15.