Management Guide

NetIQ® AppManager® for Windows Terminal Server

April 2016

Legal Notice

For information about NetIQ legal notices, disclaimers, warranties, export and other use restrictions, U.S. Government restricted rights, patent policy, and FIPS compliance, see https://www.netiq.com/company/legal/.

For information about NetIQ trademarks, see https://www.netiq.com/company/legal/. All third-party trademarks are the property of their respective owners.

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.

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

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

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

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

Enabling critical business services, better and faster

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

Our Philosophy

Selling intelligent solutions, not just software

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

Driving your success is our passion

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

Our Solutions

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

Contacting Sales Support

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

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

Contacting Technical Support

For specific product issues, contact our Technical Support team.

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

Contacting Documentation Support

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

Contacting the Online User Community

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

1 Introducing AppManager for Windows Terminal Server

AppManager for Windows Terminal Server allows you to manage and monitor the various sessions hosted by a Windows terminal server and to monitor the memory and CPU usage for those sessions.

Remote Desktop Services, formerly known as “Terminal Services” for Windows 2008 R2, is one of the components of Microsoft Windows (both server and client versions) that allows a user to access applications and data on a remote computer over a network. Terminal Services is Microsoft’s implementation of thin-client terminal server computing, where Windows applications, or even the entire desktop of the computer running terminal services, are made accessible to a remote client machine. AppManager for Windows Terminal Server provides several Knowledge Scripts designed to give you a comprehensive view of how the various Windows Terminal Server (WTS) sessions perform on your servers. The Knowledge Scripts in the WTS category monitor the following:

- Completed and active WTS sessions
- Total number of input and output bytes used in a WTS session
- Number of failed and timed-out WTS sessions
- Session errors
- Inactive sessions
- CPU usage for all processes and sessions
- Memory usage for all sessions

With these Knowledge Scripts, you can set thresholds that specify the boundaries of optimal performance. You can also configure AppManager to raise events when those thresholds are breached.

AppManager for Windows Terminal Server also allows you to:

- Send a message to a user who is currently logged into a Remote Desktop Connection or Terminal Services session
- Terminate or reset a session on the Windows terminal server
- View information about all users currently logged on to the Windows terminal server

In addition to monitoring, you can use WTS Knowledge Scripts to collect performance data for use in reports. AppManager lets you generate reports that range in scope from minute-by-minute values to monthly values over a period of years. These reports range in purpose from evaluating a narrow window of performance data to illustrating trends that aid in effective planning.

2 Installing AppManager for Windows Terminal Server

This chapter provides installation instructions and describes system requirements for AppManager for Windows Terminal 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 Windows Terminal Server has the following system requirements:

Item	Requirement
NetIQ AppManager installed on the AppManager repository (QDB) computers, on the Windows terminal servers you want to monitor (AppManager agents), and on all console computers	7.0 or later Support for Windows Server 2008 on AppManager 7.x requires AppManager Windows Agent hotfix 71704 or later. For more information, see the AppManager Suite Hotfixes page.
Microsoft Windows Terminal Server or Remote Desktop Services on agent computers	Windows Terminal Server or Remote Desktop Services installed with one of the following: ◆ Windows Server 2012 R2 ◆ Windows Server 2012 ◆ Windows 8 (32-bit and 64-bit) ◆ Windows Server 2008 R2 ◆ Windows Server 2008 (32-bit and 64-bit) ◆ Windows Server 2003 R2 (32-bit and 64-bit) **NOTE:** On Windows Server 2008 R2 and later, Windows Terminal Services is known as Remote Desktop Services.
AppManager for Microsoft Windows module installed on repository, 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.

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

2.2 Installing the Module

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

Access the `AM70-WTS-7.x.x.0.msi` module installer from the `AM70_WTS_7.x.x.0` self-extracting installation package on the AppManager Module Upgrades & Trials page.

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

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

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

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

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

To install the module manually:

1. Double-click the module installer `.msi` file.
2. Accept the license agreement.
3. Review the results of the pre-installation check. You can expect one of the following three scenarios:
   - **No AppManager agent is present:** In this scenario, the pre-installation check fails, and the installer does not install agent components.
   - **An AppManager agent is present, but some other prerequisite fails:** In this scenario, the default is to not install agent components because of one or more missing prerequisites. However, you can override the default by selecting Install agent component locally. A missing application server for this particular module often causes this scenario. For example, installing the AppManager for Microsoft SharePoint module requires the presence of a Microsoft SharePoint server on the selected computer.
   - **All prerequisites are met:** In this scenario, the installer installs the agent components.
4. To install the Knowledge Scripts into the QDB:

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

   4b Specify the SQL Server name of the server hosting the QDB, as well as the case-sensitive QDB name.
5. (Conditional) If you use Control Center 7.x, run the module installer for each QDB attached to Control Center.

6. (Conditional) If you use Control Center 8.x or later, run the module installer only for the primary QDB. Control Center automatically replicates this module to secondary QDBs.
7. Run the module installer on all console computers to install the Help and console extensions.
8. Run the module installer on the Windows Terminal Server computers you want to monitor (agents) to install the agent components.
9. If you have not discovered Windows Terminal Server resources, run the Discovery_WTS Knowledge Script on all agent computers where you installed the module. For more information, see Section 2.5, “Discovering Windows Terminal Server Resources,” on page 14.
10. To get the updates provided in this release, upgrade any running Knowledge Script jobs. For more information, see Section 2.6, “Upgrading Knowledge Script Jobs,” on page 15.

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

2.3 Deploying the Module with Control Center

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

2.3.1 Deployment Overview

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

To deploy the module on an agent computer:

1. Verify the default deployment credentials.
2. Check in an installation package. For more information, see Section 2.3.2, “Checking In the Installation Package,” on page 13.
3. Configure an email address to receive notification of a deployment.
4. Create a deployment rule or modify an out-of-the-box deployment rule.
5. Approve the deployment task.
6. View the results.

2.3.2 Checking In the Installation Package

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

To check in a module installation package:

1. Log on to Control Center using an account that is a member of a user group with deployment permissions.
2. Navigate to the **Deployment** tab (for AppManager 8.x or later) or **Administration** tab (for AppManager 7.x).
3. In the Deployment folder, select **Packages**.
4. On the Tasks pane, click **Check in Deployment Packages** (for AppManager 8.x or later) or **Check in Packages** (for AppManager 7.x).

5. Navigate to the folder where you saved `AM70-WTS-7.x.x.0.xml` and select the file.
6. Click **Open**. The Deployment Package Check in Status dialog box displays the status of the package check in.
7. To get the updates provided in this release, upgrade any running Knowledge Script jobs. For more information, see Section 2.6, "Upgrading Knowledge Script Jobs," on page 15.

2.4 Silently Installing the Module

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

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

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

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

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

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

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

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

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

Windows authentication:

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

SQL authentication:

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

2.5 Discovering Windows Terminal Server Resources

Use the Discovery_WTS Knowledge Script to discover Microsoft Windows Terminal Server (WTS) resources. If you have Citrix MetaFrame installed on a Windows terminal server, you can run the Discovery_MFXP Knowledge Script to discover MetaFrame resources as well.

Run Discovery_WTS on WTS server objects. By default, this script runs once for each computer.

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

Description	How to Set It
Raise an event when discovery succeeds?	This Knowledge Script always raises an event when the job fails for any reason. In addition, you can set this parameter to **y** to raise an event when discovery is successful. The default is n.
Event severity when discovery succeeds	Set the event severity level, from 1 to 40, to reflect the importance when the job succeeds. The default is 25.
Event severity when discovery fails	Set the event severity level, from 1 to 40, to reflect the severity when the discovery fails. The default is 5.
Event severity when discovery partially succeeds	Set the event severity level, from 1 to 40, to reflect the severity when the discovery partially succeeds, such as when a discovery returns some data but also generates warning messages. The default is 10.
Event severity when discovery is not applicable	Set the event severity level, from 1 to 40, to reflect the severity when the discovery is not applicable, usually because the target computer does not have a supported version of WTS installed on it. The default is 15.

2.6 Upgrading Knowledge Script Jobs

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

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

You can push the changes for updated scripts to running Knowledge Script jobs in one of the following ways:

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

2.6.1 Running AMAdmin_UpgradeJobs

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

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

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

2.6.2 Propagating Knowledge Script Changes

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

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

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

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

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

Propagating Changes to Ad Hoc Jobs

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

To propagate changes to ad hoc Knowledge Script jobs:

1. In the Knowledge Script view, select the Knowledge Script for which you want to propagate changes.
2. Right-click the script and select **Properties propagation** > **Ad Hoc Jobs**.
3. Select the components of the Knowledge Script that you want to propagate to associated ad hoc jobs:

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

Propagating Changes to Knowledge Script Groups

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

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

To propagate Knowledge Script changes to Knowledge Script Groups:

1. In the Knowledge Script view, select the Knowledge Script Group for which you want to propagate changes.
2. Right-click the Knowledge Script Group and select **Properties propagation** > **Ad Hoc Jobs**.

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

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

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

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

3 WTS Knowledge Scripts

AppManager provides the following Knowledge Scripts for monitoring Windows Terminal Server (WTS) application 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
LoggedOffSessions	Collects data about completed WTS sessions.
Messenger	Sends a message to a user currently connected to a WTS session.
SessionsInfo	Reports information about the current terminal sessions connected to the terminal server.
SessionsLogoff	Terminates a WTS session.
SessionsReset	Resets a client session on a terminal server.
SessionsTimeout	Monitors the current number of WTS sessions that have timed out.
SessionsTotalActive	Monitors the current number of sessions that are currently logged onto the terminal server.
SessionsTotalBytes	Monitors the total number of input and output bytes used in a WTS session.
SessionsTotalDisconnected	Monitors the total number of sessions disconnected on a terminal server.
SessionsTotalErrors	Monitors the total number of WTS session errors.
SessionsTotalFrames	Monitors the total number of frames (packets) transmitted in a WTS session.
SessionsTotalInactive	Monitors the number of inactive WTS sessions.
SessionsTotalProtocolHitRatio	Monitors the overall hit ratio for all protocol objects per WTS session.
TopCpuProcs	Monitors CPU usage for all WTS processes and the CPU used by each of a specified number of processes.
TopCPUSessions	Monitors CPU usage for all WTS sessions and the CPU usage used by each of a specified number of sessions.
TopMemorySessions	Monitors total memory usage for all sessions and the memory used by each of the top n number of sessions.
UsersInfo	Displays information about the users currently logged on to the terminal server.

3.1 LoggedOffSessions

This Knowledge Script collects data about the completed sessions hosted by a Windows Terminal Server.

This Knowledge Script scans all user sessions and makes a note of a user’s name, session ID, and logon time. On subsequent scans, it compares the current user sessions with its notes from previous sessions.

For each user that is no longer logged on, the Knowledge Script returns the user name, the session ID, the logon time, and the current time, which is the time when the Knowledge Script ran and detected that the session has ended. The script also calculates a maximum session duration by subtracting the logon time from the current time. You can set a threshold for the maximum number of sessions that can end in any interval.

The accuracy of the data returned by this Knowledge Script improves depending on how frequently you set it to run:

- The Knowledge Script cannot detect sessions that start and end *between* the intervals when it runs. For example, if you schedule the script to run every hour, a session that starts at 10:15 and ends at 10:45 will not be detected by the script when it runs at 10:00 and 11:00.
- The maximum duration is calculated using the time when the script runs and determining that a previous session is no longer active. If, for example, the script runs at 30-minute intervals and a session ends 1 minute after the script runs, 29 minutes elapse until the next time the script runs, and the maximum duration is 29 minutes longer than the actual session time.

In both cases, the accuracy of the data returned improves substantially when you schedule it to run at every minute.

TIP: One common use of this Knowledge Script is to provide billing information for the use of Windows Terminal Server (WTS) services. Each data point returned by the Knowledge Script represents a complete, billable session. When using the Knowledge Script for this purpose, remember that the accuracy of the “maximum duration” returned by this script depends on how frequently you run the Knowledge Script.

3.1.1 Resource Object

WTS Sessions

3.1.2 Default Schedule

The default interval is **Every hour**.

3.1.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if number of completed user sessions exceeds the threshold?	Set to **y** to raise an event if the number of completed user sessions exceeds the threshold you specify. The default is y.

Description	How to Set It
Collect data for completed user sessions?	Set to **y** to collect data for charts and reports. If set to y, the Knowledge Script returns the user name, the session ID, the logon time, the current time (the time when the Knowledge Script ran and detected that the session ended), and the maximum session duration (the current time minus the logon time). The default is y.
Threshold -- completed user sessions	Specify the maximum number of user sessions that can end in any given interval before an event is raised. The default is 0.
Event severity when number of completed user sessions exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event in which the number of completed user sessions exceeds the threshold. The default severity level is 8.

3.2 Messenger

This Knowledge Script sends a message to a user currently connected to a Windows Terminal Server (WTS) session.

You can specify the recipients for the message by user name, session name, session ID, or by entering a file name that contains a list of recipients by user name (user names must not include spaces), session name, or session ID.

3.2.1 Resource Object

WTS Server

3.2.2 Default Schedule

The default interval is **Run once**.

3.2.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Recipients (separate with commas)	Specify a list of recipients for the message, separated by commas with no spaces. Type an asterisk (`*`) to send a message to all sessions. The default is `*` (all sessions). You can specify WTS sessions by user name, session name, session ID, or file name. Names cannot include spaces. For example: `wsmith,jcarter,001003` If you enter a file name that contains a list of recipients, specify a complete path, including the `@` symbol. For example: `@c:\recipient.txt` **Notes** ◆ Ensure that the list of recipients listed in the file are separated by line breaks and not special characters. ◆ The recipient text file must contain only the list of recipients currently connected to a Windows terminal server.
Raise event if message fails to send?	Set to **y** to raise an event if the message is not successfully delivered. The default is y.
Collect data for messages sent?	Set to **y** to make the script return the number of sessions that received the message. The default is n.
Delay (time to wait for the receiver to acknowledge the message)	Specify the number of seconds to wait for the recipient to acknowledge receiving the message. After the number of seconds has passed, the message automatically closes. The default is 0 seconds, which means that the message is displayed until it the user acknowledges it.
Message	Specify the text of the message you want to send. The default message is `work`. To avoid an error with your message, if you want to include characters in double quotes in the text of your message (such as `"Server"`), use *two* double quote characters before and after the word(s) you want to enclose. For example:" . The length of the text message must not exceed 246 characters.
Event severity when message fails to send	Set the event severity level, from 1 to 40, to indicate the importance of the event raised when the message is not successfully sent. The default severity level is 5.

3.3 SessionsInfo

This Knowledge Script displays information about the current terminal sessions connected to the Windows terminal server. The information returned in the detailed data message depends on the version of Windows Terminal Server (WTS):

- With some versions, the data message contains the session name, user name, session ID, current status, session type, and device.
- With other versions, the data message contains the session name, user name, session ID, current status, client address, and client name.

If the Knowledge Script cannot get information about the WTS sessions, an event is raised.

3.3.1 Resource Objects

WTS Sessions

3.3.2 Default Schedule

The default interval is **Every hour**.

3.3.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Collect data on terminal session information?	Set to **y** to collect data for charts and reports. If set to y, the script returns data on the current number of sessions connected to the target computer. The default is y.
Event severity if information cannot be retrieved	Set the event severity level, from 1 to 40, to indicate the importance of the event in which data cannot be retrieved. The default severity level is 8.

3.4 SessionsLogoff

This Knowledge Script terminates a Windows Terminal Server session. You can specify the sessions to log off by session name or session ID. If the Knowledge Script fails, an event is raised.

3.4.1 Resource Objects

WTS Sessions

3.4.2 Default Schedule

The default interval is **Run once**.

3.4.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Session Name or Session ID	Specify a list of sessions you want to terminate, separated by commas with no spaces. Type an asterisk (`*`) to terminate all sessions. Specify sessions by session name or session ID number. For example: `RDP-TCP#8,RDP-TCP#9,4`. The default is blank.
Raise event if a session cannot be logged off?	Set to **y** to raise an event if the operation fails for any reason. The default is y.
Collect data for sessions logged off?	Set to **y** to collect data for charts and reports. If set to y, the script returns the number of sessions terminated. The default is n.
Event severity when a session cannot be logged off?	Set the event severity level, from 1 to 40, to indicate the importance of the event in which a session cannot be logged off. The default severity level is 5.

3.5 SessionsReset

This Knowledge Script resets a client session on a Windows Terminal Server. When you run this Knowledge Script, the session's hardware and software subsystems are reset to the known initial values.

You can identify the session to reset by session name or session ID number. If the Knowledge Script fails, an event is raised.

3.5.1 Resource Objects

WTS Sessions

3.5.2 Default Schedule

The default interval is **Run once**.

3.5.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Session Name or Session ID	Specify a list of sessions you want to reset, separated by commas with no spaces. You can specify sessions by session name or session ID number. For example: `RDP-TCP#8,RDP-TCP#9,4`.
Raise event if session reset fails?	Set to **y** to raise events if the operation fails for any reason. The default is y.

Description	How to Set It
Collect data for reset sessions?	Set to **y** to collect data for charts and reports. If set to y, the script returns the number of sessions reset. The default is n.
Event severity when reset fails	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default severity level is 5.

3.6 SessionsTimeout

This Knowledge Script to monitors the total number of timeouts on the communication line from both the host and client sides of the connection. If the number of timed-out sessions exceeds the threshold you set, an event is raised.

On some high-latency networks, the timeout could result from the protocol timeout being too short. Increasing the protocol timeout on these types of lines will improve performance by reducing unnecessary re-transmissions.

3.6.1 Resource Objects

WTS Sessions

3.6.2 Default Schedule

The default interval is **Every 30 minutes**.

3.6.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if the number of session timeouts exceeds the threshold?	Set to **y** to raise an event if the number of session timeouts exceeds the threshold you set. The default is y.
Collect data for session timeouts?	Set to **y** to collect data for charts and reports. If set to y, the script returns the total number of session timeouts. The default is y.
Threshold -- maximum session timeouts per interval	Specify the number of timed-out sessions that will raise an event. The default is 0.
Event severity if the number of session timeouts exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default severity level is 5.

3.7 SessionsTotalActive

This Knowledge Script monitors the current number of client sessions that are currently logged on to Windows Terminal Server (WTS). If the number of active sessions exceeds the threshold you set, an event is raised.

3.7.1 Resource Objects

WTS Sessions

3.7.2 Default Schedule

The default interval is **Every 1 hour**.

3.7.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if the total number of active sessions exceeds the threshold?	Set to **y** to raise an event if the total number of active sessions exceeds the threshold you set. The default is y.
Collect data for active sessions?	Set to **y** to collect data for charts and reports. If set to y, the script returns the total number of active sessions. The default is y.
Threshold -- maximum active sessions	Specify the maximum number of concurrent sessions that can be active before raising an event. The default is 50.
Event severity when the total number of active sessions exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event in which the total number of active sessions exceeds the threshold. The default severity level is 5.

3.8 SessionsTotalBytes

This Knowledge Script monitors the total number of input and output bytes used in a session. The total number of bytes includes memory used to handle any protocol overhead.

If the number of bytes used in a session exceeds the threshold you set, an event is raised.

NOTE: This Knowledge Script is not supported on Windows Server 2012 and Windows Server 2012 R2 because the Total Bytes counter is not available for Terminal Services Session.

3.8.1 Resource Objects

WTS Sessions

3.8.2 Default Schedule

The default interval is **Every 1 hour**.

3.8.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if number of bytes per session exceeds the threshold?	Set to **y** to raise an event if the number of bytes per session exceeds the threshold you set. The default is y.
Collect data for total bytes per session?	Set to **y** to collect data for charts and reports. If set to y, the script returns the number of bytes used by each session. The default is y.
Threshold -- total bytes used per session	Specify the maximum number of bytes to be used per session before an event is raised. The default is 500.
Event severity if the number of bytes per session exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default severity level is 5.

3.9 SessionsTotalDisconnected

This Knowledge Script monitors the total number of sessions that have been disconnected on a Windows terminal server. If the number of disconnected Windows Terminal Server (WTS) sessions exceeds the threshold you set, an event is raised.

3.9.1 Resource Objects

WTS Sessions

3.9.2 Default Schedule

The default interval is **Every 1 hour**.

3.9.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise an event if the total number of disconnected sessions exceeds the threshold?	Set to **y** to raise an event if the total number of disconnected sessions exceeds the threshold. The default is y.
Collect data for disconnected sessions?	Set to **y** to collect data for charts and reports. If set to y, this script returns the total number of disconnected sessions. The default is n.
Threshold -- maximum disconnected sessions	Specify the maximum number of sessions to be disconnected before an event is raised. The default is 10.
Event severity when the total number of disconnected sessions exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event in which the total number of disconnected sessions exceeds the threshold. The default severity level is 8.

3.10 SessionsTotalErrors

This Knowledge Script monitors the total number of session errors of all types. The total number of errors for a session can include lost acknowledgments, badly formed packets, and transmission problems. If the number of errors exceeds the threshold you set, an event is raised.

NOTE: This Knowledge Script is not supported on Windows Server 2012 and Windows Server 2012 R2 because the Total Errors counter is not available for Terminal Services Session.

3.10.1 Resource Objects

WTS Sessions

3.10.2 Default Schedule

The default interval is **Every 1 hour**.

3.10.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if the number of errors per session exceeds the threshold?	Set to **y** to raise an event if the number of errors per session exceeds the threshold. The default is y.
Collect data for session errors?	Set to **y** to collect data for charts and reports. If set to y, this script returns the total number of errors for each session. The default is y.
Threshold -- maximum session errors	Specify the maximum number of errors that can occur for any session before an event is raised. The default is 500.
Event severity when the number of errors per session exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default severity level is 5.

3.11 SessionsTotalFrames

This Knowledge Script monitors the total number of frames (packets) transmitted in a session. If the number of frames exceeds the threshold you set, an event is raised.

NOTE: This Knowledge Script is not supported on Windows Server 2012 and Windows Server 2012 R2 because the Total Frames counter is not available for Terminal Services Session.

3.11.1 Resource Objects

WTS Sessions

3.11.2 Default Schedule

The default interval is **Every 1 hour**.

3.11.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if the total number of frames per session exceeds the threshold?	Set to **y** to raise an event if the total number of frames per session exceeds the threshold. The default is y.
Collect data for frames transmitted per session?	Set to **y** to collect data for charts and reports. If set to y, this script returns the total number of frames for each session. The default is n.
Threshold -- maximum transmitted frames	Specify the maximum number of frames to be transmitted in a session before an event is raised. The default is 30.
Event severity when the total number of frames per session exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default severity level is 5.

3.12 SessionsTotalInactive

This Knowledge Script monitors the number of inactive Windows Terminal Server (WTS) sessions. An inactive session is one with no users logged on. If the number of inactive WTS sessions exceeds the threshold you set, an event is raised.

3.12.1 Resource Objects

WTS Sessions

3.12.2 Default Schedule

The default interval is **Every 1 hour**.

3.12.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if the number of inactive sessions exceeds the threshold?	Set to **y** to raise an event if the number of inactive sessions exceeds the threshold. The default is y.
Collect data for inactive sessions?	Set to **y** to collect data for charts and reports. If set to y, the script returns the total number of inactive sessions. The default is y.
Threshold -- maximum inactive sessions	Specify the maximum number of sessions to be inactive before an event is raised. The default is 50.

Description	How to Set It
Event severity when the number of inactive sessions exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default severity level is 5.

3.13 SessionsTotalProtocolHitRatio

This Knowledge Script monitors the overall hit ratio for all protocol objects per session. The *hit ratio* is the percentage of time protocol objects that are reused or available in the client cache.

A higher hit ratio indicates reduced data transmission and better performance. A low hit ratio is caused when a session is updated with new information that is not re-used, or is not used within the number of bytes available for the client cache.

If the hit ratio for any session is lower than the threshold you set, an event is raised.

NOTE: This Knowledge Script is not supported on Windows Server 2012 and Windows Server 2012 R2 because the Total Protocol Hit Ratio counter is not available for Terminal Services Session.

3.13.1 Resource Objects

WTS Sessions

3.13.2 Default Schedule

The default interval is **Every 1 hour**.

3.13.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if the protocol hit ratio is below threshold?	Set to **y** to raise an event if the protocol hit ratio is below threshold. The default is y.
Collect data for protocol hit ratio?	Set to **y** to collect data for charts and reports. If set to y, the script returns the overall hit ratio for all protocol objects per session. The default is n.
Threshold -- minimum protocol hit ratio	Specify the minimum protocol hit ratio percentage per session to be reached before an event is raised. The default is 50.
Event severity if the protocol hit ratio is below threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event in which the protocol hit ratio is below threshold. The default severity level is 5.

3.14 TopCpuProcs

This Knowledge Script monitors CPU usage for all Windows Terminal Server (WTS) processes and the CPU used by the top number of processes specified by you. If the total percentage of CPU usage for the set of top processes exceeds the threshold you set, an event is raised.

You can specify the number of top processes to display in the detail message. The detail message includes the percentage of CPU used by each of the top processes.

3.14.1 Resource Object

WTS Server

3.14.2 Default Schedule

The default interval is **Every 30 minutes**.

3.14.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if CPU usage exceeds the threshold?	Set to **y** to raise an event if the CPU usage exceeds the threshold. The default is y.
Collect data for CPU usage by top processes?	Set to **y** to collect data for charts and reports. If set to y, the script returns the total CPU usage for the set of top processes. The default is n.
Threshold -- maximum CPU usage	Specify the maximum percentage of CPU resources WTS processes can consume before raising an event. The default is 90.
Number of top CPU usage processes to show	Specify the number of top processes to display in the detail message. Specify 0 if you want to display all processes. The default is 5.
Event severity when CPU usage exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event in which the CPU usage exceeds the threshold. The default severity level is 5.

3.15 TopCPUSessions

This Knowledge Script monitors CPU usage for all WTS sessions and monitors the CPU usage of the top number of sessions specified by you. If the total percentage of CPU usage for these sessions exceeds the threshold you set, an event is raised.

You can specify the number of top sessions to display in the detail message. The detail message includes the percentage of CPU usage used by each of the top sessions.

3.15.1 Resource Objects

WTS Sessions

3.15.2 Default Schedule

The default interval is **Every 30 minutes**.

3.15.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if CPU usage exceeds the threshold?	Set to **y** to raise an event if the CPU usage exceeds the threshold you set. The default is y.
Collect data for session CPU usage?	Set to **y** to collect data for charts and reports. If set to y, the script returns the total CPU usage for the top set of sessions. The default is n.
Threshold -- total CPU usage	Specify a threshold for the maximum percentage of CPU resources WTS sessions can consume before raising an event. The default is 90.
Number of top CPU usage sessions to show	Specify the number of top sessions you want to display in the detail message. Specify 0 if you want to display all sessions. The default is 5.
Event severity when CPU usage exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event in which the CPU usage exceeds the threshold. The default severity level is 5.

3.16 TopMemorySessions

This Knowledge Script monitors total memory usage for all sessions and the memory used by each of a specified set of sessions. If the memory usage of any of the sessions exceeds the threshold you set, an event is raised.

You can specify the number of top sessions to display in the detail message. The detail message includes the total memory used by each of the top set of sessions.

3.16.1 Resource Objects

WTS Sessions

3.16.2 Default Schedule

The default interval is **Every 30 minutes**.

3.16.3 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Raise event if session memory usage exceeds the threshold?	Set to **y** to raise an event if the session memory usage exceeds the threshold. The default is y.

Description	How to Set It
Collect data for session memory usage?	Set to **y** to collect data for charts and reports. If set to y, the script returns the total memory usage for the top *n* number of sessions. The default is n.
Threshold -- maximum memory usage	Specify the maximum memory the top *n* number of sessions can consume before raising an event. The default is 5120.
Number of top usage of memory sessions to show	Specify the number of top sessions you want to display in the detail message. Specify 0 if you want all to display all users. The default is 5.
Event severity when session memory usage exceeds the threshold	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default severity level is 5.

3.17 UsersInfo

This Knowledge Script displays information about the users currently logged on to the Windows Terminal Server (WTS) server. The detailed message includes each user's name, session name, session ID, current status, idle time, and log on time. If the Knowledge Script cannot get information about WTS users, an event is raised.

3.17.1 Resource Objects

WTS Users

3.17.2 Default Schedule

The default interval is **Every 1 hour**.

3.17.3 Example of the Information Returned

The following is an example of the data collected and displayed in the data detail message:

```
USERNAME    SESSIONNAME     ID  STATE  IDLE TIME  LOGON TIME
>shawn      console          0  active         .  09/03/03 18:56
 netiq                       4  disc        none  09/04/03 10:32
 netiq      rdp-tcp#5        5  active        50  09/04/03 12:05
```

3.17.4 Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Collect data for logged-on users?	Set to **y** to collect data for charts and reports. If set to y, the script returns the number of users currently connected to WTS. The default is y.
Event severity if user information cannot be retrieved	Set the event severity level, from 1 to 40, to indicate the importance of the event. The default severity level is 8.