Management Guide
NetIQ® AppManager® ResponseTime for Microsoft SQL Server

January 2020

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 Documentation Support

Our goal is to provide documentation that meets your needs. If you have suggestions for improvements, please 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, please visit http://community.netiq.com.

1 Introducing AppManager ResponseTime for Microsoft SQL Server

NetIQ AppManager ResponseTime for SQL Server provides a set of transactions that can be run from client computers to SQL servers. These transactions resemble actual SQL Server transactions and appear the same to the network components that process them. The Knowledge Scripts use these transactions to measure and monitor the availability and response time of typical SQL queries. You can deploy them from any sites served by SQL Server.

This chapter provides a brief introduction to SQL Server and an overview of important concepts and terminology. It also summarizes the key ways AppManager can help you monitor and manage your SQL Server.

1.1 Why Measure Response Time?

Response time is perhaps the best metric for reporting on the performance of your IT infrastructure, because response time is the metric most often experienced by and comprehensible to end users, those for whom the infrastructure must perform as well as it possibly can.

Along with measuring response time, the AppManager ResponseTime modules also measure availability, another key metric that really matters to end users.

With the AppManager ResponseTime modules, the response time and availability of key servers, along with that of the network itself, can be measured or tested for different geographies and applications, providing the information you really need for both managing and reporting on network performance.

1.2 The AppManager ResponseTime Modules

The AppManager ResponseTime modules were designed to measure the response time and availability of a client/server transaction from the client's perspective. In that sense, they are different from other AppManager monitoring modules, which run on the server to measure and monitor server activities.

Therefore, ResponseTime modules are normally only installed on client computers, often on computers actually used by end-users, and not on the application servers themselves. In most cases, their operation therefore requires some pre-requisite client software to be installed and configured, such as Microsoft Outlook or an Oracle client.

The following discrete modules make up the AppManager ResponseTime family:

Module Name	Knowledge Script Category Name	What Is Monitored
AppManager ResponseTime for Microsoft Active Directory	AD-RT	Microsoft Active Directory and DNS transactions.
AppManager ResponseTime for Microsoft Exchange	Exchange-RT	Microsoft Outlook transactions.

Module Name	Knowledge Script Category Name	What Is Monitored
AppManager ResponseTime for Networks	Networks-RT	Simulated transactions for many popular applications to measure network performance.
AppManager ResponseTime for Oracle Database	Oracle-RT	ODBC and ADO Transactions to Oracle Servers.
AppManager ResponseTime for Microsoft SQL Server	SQL-RT	ODBC and ADO transactions to Microsoft SQL Server.
AppManager ResponseTime for Web	Web-RT	Web, Internet Mail, and News (NNTP) transactions. This module allows you to record a Web-browsing session, "play back" synthetic Web transactions, and measure the response time.
AppManager ResponseTime for Windows	Windows-RT	Windows transactions. This module allows you to record and "play back" synthetic transactions from any 32-bit Windows or Citrix client.

1.3 How AppManager ResponseTime for SQL Server Works

The strategy that the AppManager ResponseTime modules deploy for measuring network and server response time and availability is based on *synthetic network transactions*.

Whenever you run a job using one of the ResponseTime Knowledge Scripts, a software agent performs a transaction involving the real application server you want to test. Transactions performed for response-time testing are "synthetic" only in the sense that no actual user is involved—the transactions are performed purely in order to monitor performance and availability.

The ResponseTime modules all rely on unique technology developed by NetIQ for monitoring system performance at the application layer. So you not only find out how well the system is performing; you also find out how well SQL Server or Outlook transactions in particular are performing.

ResponseTime Module Architecture

Most AppManager ResponseTime modules have two parts:

◆ A shared **managed object** component, QCMA.dll, installed in NetIQ\AppManager\bin. The managed object handles tasks associated with initializing and spawning the ResponseTime engine process, used by most ResponseTime modules.

This component requires the netiqmc agent process to run as Local System, which allows the agent to start the engine processes as different users. For more information, see Section 2.4, "Permissions for Running Knowledge Scripts," on page 17.

◆ The specific **ResponseTime engine process** that handles the actual transaction. Module-specific engines are installed in %CommonProgramFiles%\Netiq\ResponseTime.

These engine processes run under the user account you specify for the **Run As** user parameters in the Knowledge Script.

Depending on the application transaction to be simulated by the Knowledge Script job, the ResponseTime engine may need to impersonate a user and/or log on to the application server. In some module-specific instances, this engine requires you to supply account and authentication information when you configure the job:

- The values you supply for the **Run As** parameters in a Knowledge Script are used to impersonate a logged-in user and instantiate the application. If required, the ResponseTime engine process will be launched as this user.

- The **values you supply for the SQL Logon** Knowledge Script parameters are used for authenticating the user on the application server, as, for example, with an Exchange Server or SQL Server user logon.

Sometimes these two options equate to the same information, especially in the case where the logon to the application server is handled by Windows Authentication, also called NTLM or "Integrated Security".

Response-Time Test Results

The results you get from response-time testing with one of the SQL-RT Knowledge Scripts are extremely accurate because the network or server is "seeing," and AppManager is timing, a transaction that looks just like a real transaction from the monitored server or client. Client-server emulation also lets you test your system the way end-users "test" it every day—and see the same results, and the same performance, that end-users are seeing.

When a response-time transaction runs, the agent measures the time taken to complete the transaction. This value is then returned as the ResponseTime data point. For most ResponseTime Knowledge Scripts, you can collect two types of data points:

- **Availability**. The Availability data point is always created if the transaction is initialized and starts, meaning that the ResponseTime engine process is started. If the transaction completes without error, a data point of 1 or 100 is created, depending on the data stream format. Otherwise, the data point is 0.

  If the ResponseTime engine process is not started due to initialization errors, no Availability data point is created, and a Transaction Initialization Error Event is raised.

- **Response Time**. The Response Time data point is only created if the transaction completes successfully. The value of the data point is the total time required to run the transaction (in seconds).

  In addition, you can collect up to six additional "**Response Time Breakdown**" data streams, individual data points for the different parts of the Knowledge Script transaction that are timed. These are explained in detail in the Help for each SQL-RT Knowledge Script.

1.4 Understanding SQL Server

Microsoft SQL Server supports XML and can query across the Internet and beyond firewalls. Its programming model is integrated with Windows DNA architecture to allow for developing Web applications.

SQL Server includes many features and services that AppManager can help you monitor:

- The **SQL database** stores event information. Its query structure allows you to pose SQL queries that perform actions on the SQL database, such as issuing an SQL command.

- **Open Database Connectivity** (ODBC) is a standard for applications to access different databases, such as Access, Paradox, dBase, and Excel, regardless of the database management system in place.
- **ActiveX Data Objects** (ADO) **is Microsoft's high-level interface to all kinds of data.**
- **Data Source Name** (DSN) is used by ODBC and ADO. It refers to the associated database connection information for a specific database server.

Knowledge Scripts in the SQL-RT category were designed to monitor these SQL Server components.

Understanding the SQL Database

The SQL database has extensive programming capabilities that are built on Web standards. It supports XML and Internet standards that let you store and retrieve data in XML format with built-in stored procedures. Also, XML allows for easy data updating and deletion. Using any Visual Studio tool, you can design and code database applications.

Centralized database management lets you stay online while moving or copying databases across computers or between instances.

Understanding ODBC

ODBC (Open Database Connectivity) refers to a standard open database access method developed by Microsoft. Its goal is to make any data access possible from any application, regardless of the database management system (DBMS) managing the data. ODBC inserts a middle layer, called a database driver, between an application and the DBMS. This layer translates the application's data queries into commands that the DBMS understands. For this layer to function, both the application and the DBMS must be ODBC-compliant (the application must be capable of issuing ODBC commands and the DBMS must be able to respond to them). Since version 2.0, the standard supports SAG SQL.

Understanding ADO

ADO (ActiveX Data Objects) **is a high-level data interface from Microsoft.** ADO **provides consistent, high-performance data access for either front-end database client or middle-tier business objects using an application, tool, language, or even an Internet browser.**

Understanding DSN

DSN (Data Source Name) is a logical name used by the ODBC driver that refers to the drive and other information required to access data. Internet Information Services (IIS) uses the DSN to connect to an ODBC data source, such as the SQL Server database. The DSN contains the database name, directory, driver, user ID, password, and more. After creating a DSN for a database, you can use that DSN in an application to retrieve information from the database.

1.5 How AppManager Can Help

AppManager ResponseTime for SQL Server provides a comprehensive solution for monitoring SQL Server response time from a client perspective. Using AppManager, you can perform the following procedures and find out the response time for each:

- Query your Microsoft SQL Server using ADO.
- Query your Microsoft SQL Server using ADO and a system DSN.
- Query your Microsoft SQL Server using ADO and advanced connection parameters.
- Query your Microsoft SQL Server using ODBC DSN.
- Query your Microsoft SQL Server using ODBC.

2 Installing AppManager ResponseTime for SQL Server

This chapter provides installation instructions and describes system requirements for AppManager ResponseTime for SQL 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 NetIQ Web site or in the `\Documentation` folder of the AppManager installation kit.

2.1 System Requirements

AppManager ResponseTime for Microsoft SQL Server has the following system requirements:

Software/Hardware	Version
NetIQ AppManager installed on the repository, agent, and console computers	6.0, at minimum AppManager provides setup programs to install this module on AppManager 6.x and AppManager 7.x. Run the setup program appropriate for your version of AppManager.
NetIQ AppManager installed on the repository, agent, and console computers	7.x, at minimum
Microsoft operating system installed on the agent computers	One of the following versions: ◆ Windows Server 2019 ◆ Windows Server 2012 R2 ◆ Windows Server 2012 ◆ Windows Server 2008 R2 ◆ 32-bit and 64-bit Windows Server 2008 ◆ 32-bit Windows 2000, SP2 ◆ 32-bit Windows XP Professional, SP2 ◆ 32-bit Windows Vista ◆ 32-bit Windows Server 2003

Software/Hardware	Version
Microsoft SQL Server	One of the following versions: ◆ SQL Server 2019 ◆ SQL Server 2014 ◆ SQL Server 2012 ◆ SQL Server 2008 R2 ◆ SQL Server 2008 ◆ SQL Server 2005 ◆ SQL Server 2000
MDAC installed on the agent computers	Version 2.5 or later
Data Source Names	Configured with ODBC to use either the ADO or ODBC DSN Knowledge Scripts.
Microsoft SQL Server Native Client 11.0 **(for TLS 1.2 support)**	11.3.6538.0 or later **NOTE:** The SQL Server Native client can be installed from this Microsoft download link.

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

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*.

2.2 Upgrade Considerations

Starting with version 7.0 of AppManager ResponseTime for Microsoft SQL Server, backlevel SQL-RT Knowledge Scripts are not supported. Backlevel Knowledge Scripts are defined as Knowledge Scripts from versions earlier than version 6.4, which became available in May 2006. If you have backlevel Knowledge Scripts running in your AppManager management site, upgrade them to the present version.

In addition, if you have multiple AppManager ResponseTime modules installed on a computer and you upgrade one of them, you cannot run Knowledge Scripts earlier than version 6.4 for any of them because the AppManager ResponseTime modules share certain files. Therefore, when you upgrade one module, you must upgrade all of them on a given computer.

Be sure to stop all running SQL-RT jobs before you try to upgrade the AppManager agent.

Because AppManager ResponseTime for SQL Server runs out-of-process from the agent (specifically, from the `netiqmc` service), ensure that all ResponseTime processes associated with running jobs are also stopped. If not, those processes may not get installed or registered as part of the upgrade installation. The installer does not have the privileges necessary to stop these processes.

Upgrade your agent computers before upgrading your repository computer. AppManager ResponseTime for SQL Server will run backlevel versions of the SQL-RT Knowledge Scripts, but the reverse is not true: the new SQL-RT Knowledge Scripts are *not* supported on backlevel versions of the module.

Installation on the repository saves a copy of your existing Knowledge Scripts in the `AppManager\Backup` directory. If you want to keep them, copy them to another location because subsequent installations will delete all files in the `AppManager\Backup` directory.

Once you install the new Knowledge Scripts on the repository and clients, re-run the Discovery_SQL-RT Knowledge Script.

2.3   Determining Where to Install the Module

To ensure the availability and performance of SQL Server resources from the perspective of an end user, install the ResponseTime for SQL module at carefully selected network locations.

If AppManager agents are distributed geographically and topologically on the network, installing the ResponseTime for SQL module on these agents can help you determine whether problems are related to the geographical location of the user, or whether the problem is related to the user's network connection. A WAN link is quite often the source of slower response times and should be included in your planning before you decide where to install the managed objects. Pay particular attention to the back-end resources that support your Web site, which needs rapid response from SQL Server.

Installing the module on computers using connections of different types and speeds, such as DSL or various types of modems, will determine how accessible critical servers are from a range of client connections. For example, you may want to verify a rapid server access time for the slowest connection speed that you expect network users to have. Or you can compile a statistically averaged view of server response time from multiple, distributed agents.

When the ResponseTime for SQL module is deployed behind a firewall, it sends data and events back to the AppManager management server, which forwards the data to the AppManager repository. The management server can be located behind the firewall, or outside the firewall.

2.4   Permissions for Running Knowledge Scripts

AppManager ResponseTime for SQL Server requires that the AppManager Windows agent— specifically, the `netiqmc` process—run as Local System. This requirement stems from the fact that most AppManager ResponseTime applications run out-of-process from the AppManager agent. The separate process for the module run with the authority of the user ID specified in each Knowledge Script. The agent must have the authority to start a new process as any user ID specified in a Knowledge Script parameter. Therefore, the agent must run with Local System authority.

When you install AppManager ResponseTime for SQL Server on computers with existing AppManager agents, you need to update any agents that are not running as Local System. Even though the requirement to run with this authority only applies to the `netiqmc` service, you should update both agent services so that they run with the same authority. If you do not update these services to run as Local System, the Discovery_SQL-RT Knowledge Script will fail.

To update the logon authority:

1 On each computer where you will install the module, navigate to the **Control Panel**, double-click **Administrative Tools**, and then double-click **Services**.

2 In the list of services, right-click **NetIQ AppManager Client Communication Manager** (`netiqccm`) and select **Properties**.

3 On the Logon tab, select ...**Local System account**.

4 Take the same steps for the **NetIQ AppManager Client Resource Monitor** (`netiqmc`) service.

5 Restart both services.

2.5 Installing the Module

This section describes how to install the module in an AppManager version 7.x environment.

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.

Installing the Module on AppManager 7.x

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

◆ Run the AppManager setup program, `amsetup.exe`, which you can begin from `Setup.exe`. You can start these programs from the distribution computer on which you saved the setup files or from the AppManager installation kit.

◆ Run the module setup program, `AM70-SQL-RT-7.x.x.0.msi`, which you downloaded from the Web or from the AppManager installation kit.

◆ Use Control Center to install the module on the remote computer where an agent is installed.

To install the module:

1 If you downloaded the module from the Web, ensure you 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*.

2 Run the AppManager or module setup program on all repository computers to install the Knowledge Scripts and reports. For repositories running in a clustered environment, run the setup program on the node that currently owns the cluster resource.

3 Install the module on the SQL Server computers you want to monitor. Use one of the following methods:

◆ Run the AppManager setup program.

◆ Run the module setup program.

◆ Use Control Center Console to deploy the installation package. Ensure you check in and then configure the installation package, which is the `.XML` file included with the module setup program. For more information about the `.XML` file, see the *AppManager ResponseTime for Microsoft SQL Server ReadMe*. For more information about deploying modules on agents, see the *Control Center User Guide for AppManager*.

4 Run the AppManager or module setup program on all Operator Console and Control Center Console computers to install the Help.

5 If you have not already discovered SQL Server resources, run the Discovery_SQL-RT Knowledge Script on all agent computers where you installed the module.

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

2.6 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 ResponseTime for Microsoft SQL Server ReadMe* matches the version number shown in the System information pane.

2.7 Uninstalling the Module

An uninstallation executable file is installed in the `AppManager\bin` directory when you install this module. Run `Uninstall_SQL-RT.exe` to uninstall AppManager ResponseTime for SQL Server.

2.8 Upgrading Knowledge Script Jobs

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

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

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.

Propagating Knowledge Script Changes

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

Before propagating script changes, verify that the script parameters are set to your specifications. Customized script parameters may have reverted to default parameters during the installation of the module. New parameters may need to be set appropriately for your environment or application.

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

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

Propagating Changes to Ad Hoc Jobs

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

To propagate changes to ad hoc Knowledge Script jobs:

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

2 Click **Properties Propagation** > **Ad Hoc Jobs**.

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

Select	To propagate
Script	The logic of the Knowledge Script.
Properties	Values from the Knowledge Script Schedule and Values tabs, such as schedule, monitoring values, actions, and advanced options.

Propagating Changes to Knowledge Script Groups

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

After you propagate script changes to Knowledge Script Group members, you can propagate the updated Knowledge Script Group members to associated running jobs. For more information, see .

To propagate Knowledge Script changes to Knowledge Script Groups:

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

2 On the KS menu, select **Properties propagation** > **Ad Hoc Jobs**.

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

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

Select	To propagate
Script	The logic of the Knowledge Script.
Properties	Values from the Knowledge Script Schedule and Values tabs, including the schedule, actions, and Advanced properties.

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

3 SQL-RT Knowledge Scripts

AppManager ResponseTime for SQL Server provides Knowledge Scripts for monitoring SQL Server response time.

The ADO Knowledge Scripts use Microsoft ActiveX Data Objects (ADO) that are built on the top of Microsoft OLE Database (OLEDB). If you are using ADO or OLEDB on production, you may find it inappropriate to use ODBC to evaluate client/server database performance.

The SQL-RT category of Knowledge Scripts support both ODBC and ADO. You can set ADO parameters to match those in the applications you are testing. You should be able to configure an ADO Knowledge Script in the same way you configure an `ADODB::Connection` on an in-house application.

ADO and ODBC Knowledge Scripts support SQL statements. Be aware that some risk exists in doing continuous `INSERT` and `DELETE` statements on a short schedule.

These Knowledge Scripts support both System and User DSN. Use the appropriate Knowledge Script for the application you are testing.

From within the SQL-RT view of the Operator Console, you can select a Knowledge Script on the **SQL-RT** tab of the Knowledge Script pane.

The following are the Knowledge Scripts in the SQL-RT category:

Knowledge Script	What It Does
ADODSNQuery	Queries a Microsoft SQL Server using ADO and a system DSN.
ADOQuery	Queries a Microsoft SQL Server using ADO.
AdvancedADOQuery	Queries a Microsoft SQL Server using ADO and advanced connection parameters.
ODBCDSNQuery	Queries a Microsoft SQL Server using ODBC DSN.
ODBCQuery	Queries a Microsoft SQL Server using ODBC.
Report_SQL-RT	Summarizes availability and response time for SQL-RT Knowledge Scripts.
Report_SQL-RT_DSN	Summarizes availability and response time for the SQL-RT Knowledge Scripts that use DSN.
Discovery_SQL-RT	Discovers SQL Server configuration and resources.

3.1 ADODSNQuery

Use this Knowledge Script to query a Microsoft SQL Server using ActiveX Data Objects (ADO) and a system Data Source Name (DSN).

This script generates the following data streams:

- ◆ **Response time**
  - ◆ **Overall response time**. The information returned by this data stream is also saved with the data point, and can be viewed by double-clicking the data point in the Graph Pane or Chart Console.
  - ◆ **Response-time Breakdown**. If enabled as separate parameters, up to three response-time breakdown data streams. These are individual data points for the different parts of the Knowledge Script transaction that are timed.
- ◆ **Availability.** This data stream returns one of two values, depending on the data stream format you selected:
  - ◆ 1 or 100 = transaction was successful
  - ◆ 0 = transaction was not successful

  The Availability data point is an indication of whether the transaction succeeded or failed.

This script raises an event whenever one of the following occurs:

- ◆ A threshold that you have specified as an event parameter is exceeded.
- ◆ The SQL-RT engine cannot be initialized. An initialization error is raised, but neither an Availability nor a Response Time data stream is generated.
- ◆ The job transaction does not complete successfully. A transaction error is raised. Only an Availability data stream is generated, with a value of 0.

You can select where some of the possible events are displayed in the Operator Console TreeView or Control Center Console Server view. This event proxying feature is useful in Control Center Service Map views. It is not supported for jobs that are started in the Operator Web Console.

Resource Object

The SQL response time ADO client

Default Schedule

The default interval for this Knowledge Script is **Every 15 minutes**.

Windows NT Authentication

The ADOQuery and AdvancedADOQuery Knowledge Scripts support both SQL Server authentication and Windows NT authentication; however, the ADODSNQuery Knowledge Script is limited using the security settings in the DSN configuration because ADO cannot overwrite this kind of DSN setting.

Therefore, the job may not use the type of authentication you expected it to use. For example, say you set an invalid SQL username and/or password for the ADODSNQuery parameters, but the job still runs successfully. This can happen when the DSN is set to use Windows authentication, also called "integrated security.". In such a case, if the *Run As Username* parameter has appropriate privileges on the SQL Server database, the OLEDB driver ignores the specific logon user and

password supplied in the Knowledge Script parameters. The logon works because the Windows authentication configured in the DSN takes precedence over the SQL Server authentication specified in the Knowledge Script.

Setting Parameter Values

When running this script, use the security model that was set when you defined the DSN.

◆ When using a DSN configured to use SQL authentication, use the *SQL Logon Username* and *Password* parameters. Or specify a valid Windows account for the *Run As* parameters.

◆ When using a DSN defined with Windows NT authentication, leave the *SQL Logon Username* and *Password* parameters blank, and specify the valid Windows account for the *Run As* parameters.

Set the following parameters as needed:

Description	How to Set It
Availability	
Collect data for availability?	Select **Yes** to collect data for graphs and reports. If enabled, data collection returns: ◆ 1 or 100 -- Transaction completed successfully ◆ 0 -- Transaction did not complete successfully ◆ Time taken to execute the query (in seconds) The default is Yes.
Data stream format	Select a format for the Availability data stream. You can select a 0 ("not available") or 100 ("available") format. The default value is 0-100.
Raise event if transaction fails?	Select **Yes** to raise an event when the server cannot be contacted. The default is Yes.
Event severity when transaction fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the server cannot be contacted. The default is 5.
Response Time	
Collect data for response time?	Select **Yes** to collect response-time data for graphs and reports. The default is Yes. If you enable data collection, you also have the option to see a breakdown in the response times for the component parts of the query, such as the time taken to connect to the SQL Server.
Threshold -- Maximum response time (seconds)	Specify the maximum response time in seconds. When response time exceeds this value, an event is raised. The default is 15 seconds.
Raise event if threshold is exceeded?	Select **Yes** to raise an event when the threshold is exceeded. The event message contains a breakdown of the total response time. The default is Yes.
Event severity when threshold is exceeded	Set the event severity level, from 1 to 40, to indicate the importance of an event in which response time exceeds the threshold. The default is 15.
Response Time Breakdown	

Description	How to Set It
Collect data for connecting to SQL Server?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to establish a connection to the SQL Server. The default is unselected.
Collect data for executing SQL statement?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to execute the SQL statement. The default is unselected.
Collect data for fetching data?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to perform a `fetch` of the query data. The default is unselected.
Target computer	Provide the computer name and instance of the SQL Server. This is an optional field, and is used to enable retrieval of data streams by AppManager Analysis Center v2.0 and later. If specified, it will also be used in place of the DSN in the data stream legend. If you set the *Event on* parameter, the *Target computer* parameter lets you select the server where the event will appear in your console. Enter the name of the server, or click **Browse [...]** to select from a list of available servers. The server you select must already be in the TreeView.
Data source (DSN)	Provide the Data Source DSN used for the connection.
Attributes	Specify the `ADODB::Connection` attributes as follows: ◆ `NONE` (the default). No attributes. ◆ `ABORTRETAINING`: Performs retaining aborts; i.e., calls `RollbackTrans` and automatically starts a new transaction. Not supported by all providers. ◆ `COMMITRETAINING`: Performs retaining commits; i.e., calls `CommitTrans` and automatically starts a new transaction. Not supported by all providers. ◆ `ABORTCOMITRETAINING`: A combination of `ABORTRETAINING` and `COMMITRETAINING`.
Cursor location	Specify the location of the cursor service: ◆ `CLIENT`: Uses client-side cursors supplied by a local cursor library. (Local services may allow many features not allowed by driver-supplied cursors; using this setting may provide some advantage in enabling features.) ◆ `SERVER`: Uses data provider- or driver-supplied cursors. (These cursors may be flexible and allow for additional sensitivity to changes made to the data source by others.) This is the default.
SQL statement	Provide a SQL statement (1024-character maximum) that is compatible with the provider. Calling stored procedures is also supported. The format is `execute stored_procedure`.

Description	How to Set It
Number of rows per fetch	Specify any positive integer or -1. Use an appropriate value according to the size of the result and of a single row. The default value is 1. If the SQL statement is a select, the engine uses the `GetRows()` method to retrieve the data, so you can fetch thousands or records at once. This could mean a huge performance improvement in production. A value of -1 attempts to retrieve all rows on a single fetch. Although this may show interesting results on a small database, it can easily become catastrophic if the result is large and the client machine has limited memory. You should change this value only if the `GetRows ()` method is used in production with a value different than 1.
Event on	Select the TreeView location where events should be displayed. Select either: ◆ **Agent** (the client computer in the response-time tests). This is the default. ◆ **Server** (the AppManager server being tested--see the **Target computer** parameter, above) ◆ **Both.** The event will be shown in two locations in the TreeView. **Notes** This setting does not apply to events related to the Knowledge Script itself, such as Knowledge Script failure or initialization problems. Such events are always displayed on the computer where the job ran. You must select `Agent` when starting jobs in the Operator Web Console. If you select `Server`, no events are generated. If you select `Both`, an event is only shown on the agent.
SQL Logon	
Username	Set this value appropriately based upon your DSN settings.
Password	Set this value appropriately based upon your DSN settings.
Run As	
Username	Provide the username associated with a specific user who has the required permissions to run this application. **Required**. `Interactive User` is a possible value. Leave the Password and Domain parameters blank if you specify "Interactive User". `Interactive User` requires a user to be physically logged into the computer for the test to run. You might want to do this in environments where a firewall is preventing access to an Active Directory domain controller, or where the test computer is part of a workgroup and not part of a domain. With this feature, the user is not validated, so the test can proceed despite the lack of access to the domain.
Password	Specify the password associated with this user that is required to log on to the network and run the application.
Domain	Specify the domain associated with this user that is the domain name you are logging onto. **Required**.
Administrators group on managed client	Provide the name of the Administrators Group on the managed client. Typically, this name is "`Administrators`", except on some foreign language operating systems. The default is "`Administrators`".

Description	How to Set It
Timeouts	
Command timeout	Specify the number of seconds to wait while establishing a connection before terminating the event and raising an event. The default is 30 seconds.
Connection timeout	Specify the number of seconds to wait while executing a command before terminating the attempt and raising an event. The default is 15 seconds.

3.2   ADOQuery

Use this Knowledge Script to query a Microsoft SQL Server using ADO.

This script generates the following data streams:

- **Response time**
  - **Overall response time**. The information returned by this data stream is also saved with the data point, and can be viewed by double-clicking the data point in the Graph Pane or Chart Console.
  - **Response-time Breakdown**. If enabled as separate parameters, up to three response-time breakdown data streams. These are individual data points for the different parts of the Knowledge Script transaction that are timed. See "Setting Parameter Values" on page 25 below for more information.
- **Availability**--Returns one of two values:
  - 1 or 100 = transaction was successful
  - 0 = transaction was not successful.

  The Availability data point is an indication of whether the transaction succeeded or failed.

An event is raised whenever one of the following occurs:

- A threshold that you have specified as an event parameter is exceeded.
- The SQL-RT engine cannot be initialized. An initialization error is raised, but neither an Availability nor a Response Time data stream is generated.
- The job transaction doesn't complete successfully. A transaction error is raised. Only an Availability data stream is generated, with a value of 0.

You can select where some of the possible events are displayed in the Operator Console TreeView or Control Center Console Server view. This event proxying feature is useful in Control Center Service Map views. It is not supported for jobs that are started in the Operator Web Console.

Resource Object

The SQL-RT ADO client

Default Schedule

The default interval for this Knowledge Script is **Every 15 minutes**.

Windows NT Authentication

The ADOQuery and AdvancedADOQuery Knowledge Scripts support both SQL Server authentication and Windows NT authentication; however, the ADODSNQuery Knowledge Script is limited to the DSN configuration because ADO cannot override this kind of DSN setting.

Therefore, the job may not use the type of authentication you expected it to use. For example, say you set an invalid SQL username and/or password for ADODSNQuery Knowledge Script parameters. The job may still run successfully if the DSN is set to use Windows authentication. In such a case, if the *Run As Username* parameter has appropriate privileges on the SQL Server database, the OLEDB driver ignores the specific logon username and password. The logon always works because the Windows authentication configured in the DSN takes precedence over the SQL Server authentication set in the Knowledge Script.

Setting Parameter Values

Set the Integrated security option according to the security model you want to use:

- **For SQL authentication**: Clear the **Yes** check box to disable integrated security, then specify the SQL Logon Username and Password parameters. Also, specify the *Run As* parameters to supply a valid account under which to run the Knowledge Script.

- **For Windows NT authentication**: Select the **Yes** check box for integrated security, and leave the *SQL Username* and *Password* parameters blank. Specify the valid account under which to run the Knowledge Script and perform the Windows authentication for the *Run As* parameters.

Set the following parameters as needed.

Description	How to Set It
Availability	
Collect data for availability?	Select **Yes** to collect data for graphs and reports. If enabled, data collection returns: • 1 or 100 -- Transaction completed successfully • 0 -- Transaction did not complete successfully • Time taken to execute the query (in seconds) The default is Yes.
Data stream format	Select a format for the Availability data stream. You can select a 0 ("not available") or 100 ("available") format. The default value is 0-100.
Raise event if transaction fails?	Select **Yes** to raise an event when the server cannot be contacted. The default is yes.
Event severity when transaction fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the server cannot be contacted. The default is 5.
Response Time	

Description	How to Set It
Collect data for response time?	Select **Yes** to collect response-time data for graphs and reports. The default is Yes.
	If you enable data collection, you also have the option to see a breakdown in the response times for the component parts of the query, such as the time taken to connect to the SQL Server.
Threshold -- Maximum response time (seconds)	Specify the maximum response time in seconds. When response time exceeds this value, an event is raised. The default is 15 seconds.
Raise event if threshold is exceeded?	Select **Yes** to raise an event when the threshold is exceeded. The event message contains a breakdown of the total response time. The default is Yes.
Event severity when threshold is exceeded	Set the event severity level, from 1 to 40, to indicate the importance of an event in which response time exceeds the threshold. The default is 15.
Response Time Breakdown	
Collect data for connecting to SQL Server?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to establish a connection to the SQL Server.
	The default is unselected.
Collect data for executing SQL statement?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to execute the SQL statement. The default is unselected.
Collect data for fetching data?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to perform a `fetch` of the query data. The default is unselected.
Server and instance name	Provide the names of the server and instance where the transaction will be run. Use the following syntax: `Server/Instance`.
	If you set the *Event on* parameter, this parameter lets you select the server where the event will appear in your console. The instance name is stripped out.
	Provide the name of the server and instance, or click **Browse [...]** to select from a list of available servers. The server you select must already be in the TreeView.
Database name	Specify the database name on the SQL Server. This is the "Initial Catalog" part of the Connection Properties collection. If the provider does not support this property, use the default database.
Cursor location	Specify the location of the cursor service:
	◆ `CLIENT`: Uses client-side cursors supplied by a local cursor library. (Local services may allow many features not allowed by driver-supplied cursors; using this setting may provide some advantage in enabling features.)
	◆ `SERVER`: Uses data provider- or driver-supplied cursors. (These cursors may be flexible and allow for additional sensitivity to changes made to the data source by others.) This is the default.
SQL statement	Provide a SQL statement (128-character maximum) that is compatible with the provider. The format is `execute stored_procedure`.

Description	How to Set It
Number of rows per fetch	Specify any positive integer or -1. Use an appropriate value according to the size of the result and of a single row. The default value is 1. If the SQL statement is a select, the engine uses the `GetRows()` method to retrieve the data, so you can fetch thousands or records at once. This could mean a huge performance improvement in production. A value of -1 attempts to retrieve all rows on a single fetch. Although this may show interesting results on a small database, it can easily become catastrophic if the result is large and the client computer has limited memory. Change this value only if the `GetRows()` method is used in production with a value different than 1.
Integrated security?	Select **Yes** to specify whether the authentication should be done on the Windows integrated security model. The default is unselected.
Event on	Select the TreeView location where events should be displayed. Select either: ◆ **Agent** (the client computer in the response-time tests). This is the default. ◆ **Server** (the AppManager server being tested--see the *Server and instance name* parameter) ◆ **Both.** The event will be shown in two locations in the TreeView. **Notes** This setting does not apply to events related to the Knowledge Script itself, such as Knowledge Script failure or initialization problems. Such events are always displayed on the computer where the job ran. You must select `Agent` when starting jobs in the Operator Web Console. If you select `Server`, no events are generated. If you select `Both`, an event is only shown on the agent.
SQL Logon	
Username	Set this value when you do *not* use integrated security. This is the User ID part of the Connection Properties collection.
Password	Set this value when you do *not* use integrated security. This is the Password part of the Connection Properties collection. Hard encryption is always used.
Run As	
Username	Provide the username associated with a specific user who has the required permissions to run this application. **Required**. `Interactive User` is a possible value. Leave the Password and Domain parameters blank if you specify "Interactive User". `Interactive User` requires a user to be physically logged into the computer for the test to run. You might want to do this in environments where a firewall is preventing access to an Active Directory domain controller, or where the test computer is part of a workgroup and not part of a domain. With this feature, the user is not validated, so the test can proceed despite the lack of access to the domain.
Password	Provide the password associated with this user that is required to log on to the network and run the application.
Domain	Provide the domain associated with this user that is the domain name you are logging onto. **Required**.

Description	How to Set It
Administrators group on managed client	Provide the name of the Administrators Group on the managed client. Typically, this name is "Administrators", except on some foreign-language operating systems. The default is "Administrators".
Timeouts	
Command timeout	Specify the number of seconds to wait while establishing a connection before terminating the event and generating an error. The default is 30 seconds.
Connection timeout	Specify the number of seconds to wait while executing a command before terminating the attempt and generating an error. The default is 15 seconds.

3.3 AdvancedADOQuery

Use this Knowledge Script to check the ability to query a Microsoft SQL Server using ADO and advanced connection parameters.

This script generates the following data streams:

- **Response time**
  - **Overall response time**. The information returned by this data stream is also saved with the data point, and can be viewed by double-clicking the data point in the Graph Pane or Chart Console.
  - **Response-time Breakdown**. If enabled as separate parameters, up to three response-time breakdown data streams. These are individual data points for the different parts of the Knowledge Script transaction that are timed. See "Setting Parameter Values" on page 25 below for more information.

- **Availability**--Returns one of two values:
  - 1 or 100 = transaction was successful
  - 0 = transaction was not successful.

  The Availability data point is an indication of whether the transaction succeeded or failed.

An event is raised whenever one of the following occurs:

- A threshold that you have specified as an event parameter is exceeded.
- The SQL-RT engine cannot be initialized. An initialization error is raised, but neither an Availability nor a Response Time data stream is generated.
- The job transaction does not complete successfully. A transaction error is raised. Only an Availability data stream is generated, with a value of 0.

You can select where some of the possible events are displayed in the Operator Console TreeView or Control Center Console Server view. This event proxying feature is useful in Control Center Service Map views. It is not supported for jobs that are started in the Operator Web Console.

Resource Object

The SQL response time ADO client

Default Schedule

The default interval for this Knowledge Script is **Every 15 minutes**.

Setting Parameter Values

Be sure to set the *Integrated security?* parameter according to the security model you want to use:

- **For SQL authentication:** Clear the **Yes** check box to disable integrated security, then specify the *SQL Logon Username* and *Password.* Also specify the *Run As* account information to supply a valid account under which to run the Knowledge Script.

- **For Windows NT authentication:** Select the **Yes** check box for integrated security, and leave the *SQL Logon Username* and *Password* parameters blank. Specify a valid user account under which to run the Knowledge Script for the *Run As* Knowledge Script parameters.

Set the following parameters as needed:

Description	How to Set It
Availability	
Collect data for availability?	Select **Yes** to collect data for graphs and reports. If enabled, data collection returns: • 1 or 100 -- Transaction completed successfully • 0 -- Transaction did not complete successfully • Time taken to execute the query (in seconds) The default is Yes.
Data stream format	Select the data stream format for the Availability data stream. You can use a 0 ("not available") or 100 ("available") format. The default value is 0-100.
Raise event if transaction fails?	Select **Yes** to raise an event when the server cannot be contacted. The default is Yes.
Event severity when transaction fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the server cannot be contacted. The default is 5.
Response Time	
Collect data for response time?	Select **Yes** to collect response-time data for graphs and reports. By default, data is collected. If you enable data collection, you also have the option to see a breakdown in the response times for the component parts of the query, such as the time taken to connect to the SQL Server.
Threshold -- Maximum response time (seconds)	Specify the maximum response time in seconds. When response time exceeds this value, an event is raised. The default is 15 seconds.
Raise event if threshold is exceeded?	Select **Yes** to raise an event when the threshold is exceeded. The event message contains a breakdown of the total response time. The default is Yes.
Event severity when threshold is exceeded	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the response time threshold is exceeded. The default is 15.
Response Time Breakdown	

Description	How to Set It
Collect data for connecting to SQL Server?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to establish a connection to the SQL Server. The default is unselected.
Collect data for executing SQL statement?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to execute the SQL statement. The default is unselected.
Collect data for fetching data?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to perform a `fetch` of the query data. The default is unselected.
Server and instance name	Provide the names of the server and instance where the transaction will be run. Use the following syntax: `Server/Instance`. If you set the Event on parameter, this parameter lets you select the server where the event will appear in your console. The instance name is stripped out. Provide the name of the server and instance, or click **Browse [...]** to select from a list of available servers. The server you select must already be in the TreeView.
Database name	Provide the database name on the SQL Server. This is the "Initial Catalog" part of the Connection Properties collection. If the provider does not support this property, use the default database.
Attributes	Specify the `ADODB::Connection` attributes as follows: ◆ `NONE` (the default). No attributes. ◆ `ABORTRETAINING`: Performs retaining aborts; i.e., calls `RollbackTrans` and automatically starts a new transaction. Not supported by all providers. ◆ `COMMITRETAINING`: Performs retaining commits; i.e., calls `CommitTrans` and automatically starts a new transaction. Not supported by all providers. ◆ `ABORTCOMITRETAINING`: A combination of `ABORTRETAINING` and `COMMITRETAINING`.
Cursor location	Specify the location of the cursor service: ◆ `CLIENT`: Uses client-side cursors supplied by a local cursor library. (Local services may allow many features not allowed by driver-supplied cursors; using this setting may provide some advantage in enabling features.) ◆ `SERVER`: Uses data provider- or driver-supplied cursors. (These cursors may be flexible and allow for additional sensitivity to changes made to the data source by others.) This is the default.

Description	How to Set It
Isolation level	Specify the level of transaction isolation for a Connection object as follows:
	◆ UNSPECIFIED: The provider is using a different isolation level than specified, but that level cannot be determined.
	◆ CHAOS: You cannot overwrite pending changes from more highly isolated transactions.
	◆ BROWSE: You can view uncommitted changes in other transactions from one transaction.
	◆ READUNCOMMITTED: Same as BROWSE.
	◆ CURSORSTABILITY: You can view changes in other transactions from one transaction only after they have been committed.
	◆ READCOMMITTED: Same as BROWSE. This is the default.
	◆ REPEATABLEREAD: You cannot see changes made in other transactions from one transaction; however, querying again can retrieve new Recordset objects.
	◆ ISOLATED: Transactions are conducted in isolation from other transactions.
	◆ SERIALIZABLE: Same as ISOLATED.
Mode	Set the available permissions for modifying data in a connection, as follows:
	◆ UNKNOWN: Permissions have either not yet been set or cannot be determined. This is the default.
	◆ READ: Read-only permissions.
	◆ WRITE: Write-only permissions.
	◆ READWRITE: Read/write permissions.
	◆ RECURSIVE: Not supported at this time.
	◆ SHAREDENYNONE: Allows others to open a connection with any permission. Neither read nor write access can be denied to others.
	◆ SHAREDENYREAD: Prevents others from opening a connection with read permissions.
	◆ SHAREDENYWRITE: Prevents others from opening a connection with write permissions.
	◆ SHAREEXCLUSIVE: Prevents others from opening a connection.
SQL Statement	Provide a SQL statement (128-character maximum) that is compatible with the provider. The format is execute stored_procedure.
Number of rows per fetch	Specify any positive integer or -1. Use an appropriate value according to the size of the result and of a single row. The default value is 1.
	If the SQL statement is a select, the engine uses the GetRows() method to retrieve the data, so you can fetch thousands or records at once. This could mean a huge performance improvement in production.
	A value of -1 attempts to retrieve all rows on a single fetch. Although this may show interesting results on a small database, it can easily become catastrophic if the result is large and the client machine has limited memory. You should change this value only if the GetRows() method is used in production with a value different than 1.

Description	How to Set It
Integrated security?	Select **Yes** to specify whether the authentication should be done on the Windows integrated security model. The default is unselected.
Event on	Select the TreeView location where events should be displayed. Select either: ◆ **Agent** (the client computer in the response-time tests). This is the default. ◆ **Server** (the AppManager server being tested--see the *Server and instance name* parameter) ◆ **Both.** The event will be shown in two locations in the TreeView. **Notes** This setting does not apply to events related to the Knowledge Script itself, such as Knowledge Script failure or initialization problems. Such events are always displayed on the computer where the job ran. You must select Agent when starting jobs in the Operator Web Console. If you select Server, no events are generated. If you select Both, an event is only shown on the agent.
SQL Logon	
Username	Set this value when you do *not* use integrated security. This is the User ID part of the Connection Properties collection.
Password	Set this value when you do *not* use integrated security. This is the Password part of the Connection Properties collection. Hard encryption is always used.
Run As	
Username	Provide the username associated with a specific user who has the required permissions to run this application. **Required**. Interactive User is a possible value. Leave the Password and Domain parameters blank if you specify "Interactive User". Interactive User requires a user to be physically logged into the computer for the test to run. You might want to do this in environments where a firewall is preventing access to an Active Directory domain controller, or where the test computer is part of a workgroup and not part of a domain. With this feature, the user is not validated, so the test can proceed despite the lack of access to the domain.
Password	Provide the password associated with this user that is required to log on to the network and run the application.
Domain	Provide the domain associated with this user that is the domain name you are logging onto. **Required**.
Administrators group on managed client	Enter the name of the Administrators Group on the managed client. Typically, this name is "Administrators", except on some foreign-language operating systems. The default is "Administrators".
Timeouts	
Command timeout	Specify the number of seconds to wait while establishing a connection before terminating the event and generating an error. The default is 30 seconds.
Connection timeout	Specify the number of seconds to wait while executing a command before terminating the attempt and generating an error. The default is 15 seconds.

3.4 ODBCDSNQuery

Use this Knowledge Script to query a Microsoft SQL Server using Open Database Connectivity (ODBC) and a Data Source Name (DSN).

This script generates the following data streams:

- **Response time**
    - **Overall response time**. The information returned by this data stream is also saved with the data point, and can be viewed by double-clicking the data point in the Graph Pane or Chart Console.
    - **Response-time Breakdown**. If enabled as separate parameters, up to three response-time breakdown data streams. These are individual data points for the different parts of the Knowledge Script transaction that are timed. See "Setting Parameter Values" on page 25 below for more information.
- **Availability**--Returns one of two values:
    - 1 or 100 = transaction was successful
    - 0 = transaction was not successful.

    The Availability data point is an indication of whether the transaction succeeded or failed.

This script raises an event whenever one of the following occurs:

- A threshold that you have specified as an event parameter is exceeded.
- The SQL-RT engine can't be initialized. An initialization error is raised, but neither an Availability nor a Response Time data stream is generated.
- The job transaction doesn't complete successfully. A transaction error is raised. Only an Availability data stream is generated, with a value of 0.

You can select where some of the possible events are displayed in the Operator Console TreeView or Control Center Console Server view. This event proxying feature is useful in Control Center Service Map views. It is not supported for jobs that are started in the Operator Web Console.

Resource Object

The SQL response time ODBC client

Default Schedule

The default interval for this Knowledge Script is **Every 15 minutes**.

About Windows NT Authentication

ODBC Knowledge Scripts do not support Windows authentication (integrated security), but DSN does. Therefore, the job may not use the type of authentication you wanted it to use. For example, if you set an invalid SQL username and/or password in the ODBCDSNQuery Knowledge Script, the job may run successfully if the DSN is set to use Windows authentication.

In such a case, if the value supplied for the *Run As Username* parameter has appropriate privileges on the SQL Server database, the ODBC driver ignores the specific logon username and password you supplied. The logon then succeeds in this case because the Windows authentication in the DSN takes precedence over the authentication method set in the Knowledge Script.

Setting Parameter Values

Depending on what security model was set when you defined the DSN, you should only use that security method when running this Knowledge Script.

- When using a DSN defined with SQL authentication, use the *SQL Logon Username* and *Password* fields only.

- When using a DSN configured to use Windows authentication, leave the *SQL Logon Username* and *Password* fields blank, and specify the valid Windows account for the *Run As* Knowledge Script parameters.

Set the following parameters as needed:

Description	How to Set It
Availability	
Collect data for availability?	Select **Yes** to collect data for graphs and reports. If enabled, data collection returns: • 1 or 100 -- Transaction completed successfully • 0 -- Transaction did not complete successfully • Time taken to execute the query (in seconds) The default is yes.
Data stream format	Select the data stream format for the Availability data stream. You can use a 0 ("not available") or 100 ("available") format. The default value is 0-100.
Raise event if transaction fails?	Select **Yes** to raise an event when the server cannot be contacted. The default is Yes.
Event severity when transaction fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the server cannot be contacted. The default is 5.
Response Time	
Collect data for response time?	Select **Yes** to collect response time data for graphs and reports. The default is Yes. If you enable data collection, you also have the option to see a breakdown in the response times for the component parts of the query, such as the time taken to connect to the SQL Server.
Threshold -- Maximum response time (seconds)	Specify the maximum response time in seconds. When response time exceeds this value, an event is raised. The default is 15 seconds.
Raise event if threshold is exceeded?	Select **Yes** to raise an event when the threshold is exceeded. The event message contains a breakdown of the total response time. The default is Yes.
Event severity when threshold is exceeded	Set the event severity level, from 1 to 40, to indicate the importance of an event in which response time exceeds the threshold. The default is 15.
Response Time Breakdown	
Collect data for connecting to SQL Server?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to establish a connection to the SQL Server. The default is unselected.

Description	How to Set It
Collect data for executing SQL statement?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to execute the SQL statement. The default is unselected.
Collect data for fetching data?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to perform a `fetch` of the query data. The default is unselected.
Target computer	Specify the computer name and instance of the SQL Server. This is an optional field, and is used to enable retrieval of data streams by AppManager Analysis Center v2.0 and later. If specified, it is also used in place of the DSN in the data stream legend. If you set the *Event on* parameter, the *Target computer* parameter lets you select the server where the event will appear in your console. Specify the name of the server and instance, or click **Browse [...]** to select from a list of available servers. The server you select must already be in the TreeView.
Data source (DSN)	Provide the system DSN on which the Knowledge Script will be run. **Required**.
SQL Statement	Provide a SQL statement (128-character maximum) that is compatible with the provider. The format is `execute stored_procedure`.
Event on	Select the TreeView location where events should be displayed. Select either: ♦ **Agent** (the client computer in the response-time tests). This is the default. ♦ **Server** (the AppManager server being tested--see the *Target computer* parameter) ♦ **Both.** The event will be shown in two locations in the TreeView. **Notes** This setting does not apply to events related to the Knowledge Script itself, such as Knowledge Script failure or initialization problems. Such events are always displayed on the computer where the job ran. You must select `Agent` when starting jobs in the Operator Web Console. If you select `Server`, no events are generated. If you select `Both`, an event is only shown on the agent.
SQL Logon	
Username	Set this value appropriately based upon your DSN settings.
Password	Set this value appropriately based upon your DSN settings.
Run As	
Username	Provide the username associated with a specific user who has the required permissions to run this application. **Required**. `Interactive User` is a possible value. Leave the Password and Domain parameters blank if you specify "Interactive User". `Interactive User` requires a user to be physically logged into the computer for the test to run. You might want to do this in environments where a firewall is preventing access to an Active Directory domain controller, or where the test computer is part of a workgroup and not part of a domain. With this feature, the user is not validated, so the test can proceed despite the lack of access to the domain.

Description	How to Set It
Password	Provide the password associated with this user that is required to log on to the network and run the application.
Domain	Provide the domain associated with this user that is the domain name you are logging onto. Required.
Administrators group on managed client	Provide the name of the Administrators Group on the managed client. Typically, this name is "`Administrators`", except on some foreign language operating systems. The default is "`Administrators`".
Connection timeout	Specify the number of seconds to wait while executing a command before terminating the attempt and generating an error. The default is 15 seconds.

3.5 ODBCQuery

Use this Knowledge Script to query a Microsoft SQL Server using Open Database Connectivity (ODBC) and measure the response time.

This script does not support Windows authentication. Use the ODBCDSNQuery Knowledge Script if you need to use Windows authentication. That script lets you supply a DSN to use. You can then configure the Data Source Name (DSN) file to use Windows authentication.

This script generates the following data streams:

- **Response time**
  - **Overall response time**. The information returned by this data stream is also saved with the data point, and can be viewed by double-clicking the data point in the Graph Pane or Chart Console.
  - **Response-time Breakdown**. If enabled as separate parameters, up to three response-time breakdown data streams. These are individual data points for the different parts of the Knowledge Script transaction that are timed. See "Setting Parameter Values" on page 25 below for more information.
- **Availability**--Returns one of two values:
  - 1 or 100 = transaction was successful
  - 0 = transaction was not successful.

  The Availability data point is an indication of whether the transaction succeeded or failed.

This script raises an event whenever one of the following occurs:

- A threshold that you have specified as an event parameter is exceeded.
- The SQL-RT engine cannot be initialized. An initialization error is raised, but neither an Availability nor a Response Time data stream is generated.
- The job transaction does not complete successfully. A transaction error is raised. Only an Availability data stream is generated, with a value of 0.

You can select where some of the possible events are displayed in the Operator Console TreeView or Control Center Console Server view. This event proxying feature is useful in Control Center Service Map views. It is not supported for jobs that are started in the Operator Web Console.

Resource Object

The SQL response time ODBC client.

Default Schedule

The default interval for this Knowledge Script is **Every 15 minutes**.

Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Availability	
Collect data for availability?	Select **Yes** to collect data for graphs and reports. If enabled, data collection returns: ◆ 1 or 100 -- Transaction completed successfully ◆ 0 -- Transaction did not complete successfully ◆ Time taken to execute the query (in seconds) The default is Yes.
Data stream format	Select a format for the Availability data stream. You can use a 0 ("not available") or 100 ("available") format. The default value is 0-100.
Raise event if transaction fails?	Select **Yes** to raise an event when the server cannot be contacted. The default is Yes.
Event severity when transaction fails	Set the event severity level, from 1 to 40, to indicate the importance of an event in which the server cannot be contacted. The default is 5.
Response Time	
Collect data for response time?	Select **Yes** to collect response time data for graphs and reports. The default is Yes. If you enable data collection, you also have the option to see a breakdown in the response times for the component parts of the query, such as the time taken to connect to the SQL Server.
Threshold -- Maximum response time (seconds)	Specify the maximum response time in seconds. When response time exceeds this value, an event is raised. The default is 15 seconds.
Raise event if threshold is exceeded?	Select **Yes** to raise an event when the threshold is exceeded. The event message contains a breakdown of the total response time. The default is Yes.
Event severity when threshold is exceeded	Set the event severity level, from 1 to 40, to indicate the importance of an event in which response time exceeds the threshold. The default is 15.
Response Time Breakdown	
Collect data for connecting to SQL Server?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to establish a connection to the SQL Server. The default is unselected.

Description	How to Set It
Collect data for executing SQL statement?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to execute the SQL statement. The default is unselected.
Collect data for fetching data?	Select **Yes** to collect response-time data showing how much of the overall response time could be attributed to the time taken to perform a `fetch` of the query data. The default is unselected.
Server and instance name	Provide the names of the server and instance where the transaction will be run. Use the following syntax: `Server/Instance`.
	If you set the *Event on* parameter, this parameter lets you select the server where the event will appear in your console. The instance name is stripped out.
	Provide the name of the server and instance, or click **Browse [...]** to select from a list of available servers. The server you select must already be in the TreeView.
Database name	Provide the database name on the SQL Server. This is the "Initial Catalog" part of the Connection Properties collection. If the provider does not support this property, use the default database.
SQL statement	Set this value appropriately based on your DSN settings. The format is `execute stored_procedure`.
Event on	Select the TreeView location where events should be displayed. Select either:
	◆ **Agent** (the client computer in the response-time tests). This is the default.
	◆ **Server** (the AppManager server being tested--see the **Server and instance name** parameter, above)
	◆ **Both.** The event will be shown in two locations in the TreeView.
	Notes This setting does not apply to events related to the Knowledge Script itself, such as Knowledge Script failure or initialization problems. Such events are always displayed on the computer where the job ran.
	You must select `Agent` when starting jobs in the Operator Web Console. If you select `Server`, no events are generated. If you select `Both`, an event is only shown on the agent.
SQL Logon	
Username	Set this value appropriately based on your DSN settings.
Password	Set this value appropriately based on your DSN settings.
Run As	
Username	Provide the username associated with a specific user who has the required permissions to run this application. **Required**.
	`Interactive User` is a possible value. Leave the Password and Domain parameters blank if you specify "Interactive User".
	`Interactive User` requires a user to be physically logged into the computer for the test to run. You might want to do this in environments where a firewall is preventing access to an Active Directory domain controller, or where the test computer is part of a workgroup and not part of a domain. With this feature, the user is not validated, so the test can proceed despite the lack of access to the domain.

Description	How to Set It
Password	Provide the password associated with this user that is required to log on to the network and run the application.
Domain	Provide the domain associated with this user that is the domain name you are logging onto. **Required**.
Administrators group on managed client	Provide the name of the Administrators Group on the managed client. Typically, this name is "`Administrators`", except on some foreign language operating systems. The default is "`Administrators`".
Connection timeout	Specify the number of seconds to wait while executing a command before terminating the attempt and generating an error. The default is 15 seconds.

3.6 Report_SQL-RT

Use this Report Knowledge Script to generate a report detailing availability and response time for the following SQL-RT Knowledge Scripts:

- ADOQuery
- AdvancedADOQuery
- ODBCQuery

Resource Object

AppManager repository

Default Schedule

The default schedule is **Run once**.

Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Data Source	
KS for report	Click **Browse [...]** to select the Knowledge Scripts you want to include in the report.
SQL-RT client(s)	Click **Browse [...]** to select the SQL-RT client computers to include in the report. Your selections are limited to computers or server groups that are visible in the selected views.
	Select one of the filter options:
	- **View:** Includes all computers in the views you selected.
	- **Computer:** Select from individual computers in the views you selected.
	- **Server Group:** Select from server groups in the views you selected.
	NOTE: Selecting a server group includes all computers in that group.

Description	How to Set It
SQL Server or "All"	Specify the name of the SQL Server or type "All" to designate all computers as SQL Servers. The default is "All".
Select time range	Click **Browse [...]** to set specific start and end report information dates, good for historical or ad hoc reports, or a sliding range that sets the time range of data to include in the report. The sliding range option is the default and is useful for reports running on a regular schedule.
Select peak weekday(s)	Click **Browse [...]** to select a contiguous day range a selection of non-contiguous days.
Aggregation by	Select the time unit by which to aggregate data. The default is Hour. This works in conjunction with the *Aggregation interval* parameter, which determines the number of units for one interval of data aggregation.
Aggregation interval	Select the interval units in which to aggregate data. The default is 1. For example, if you aggregate by the Hour and select 1 here, data is aggregated once every hour.
Report Settings	
Include parameter card?	Select **Yes** to display a table of parameters used in the report. The default is Yes.
Include Availability Detail table?	Select **Yes** to display the Availability Detail table as part of the report. The default is Yes.
Include Availability chart?	Select **Yes** to display the Availability chart as part of the report. The default is Yes.
Availability data stream format	Specify the data stream format. Options are 0-100 or 0-1. ◆ 1 or 100 -- Available ◆ 0 -- Not available The default format is 0-100.
Threshold on Availability chart	Specify an integer for the percentage threshold. The default is 0 (no threshold is displayed).
Include Response Time Detail table?	Select **Yes** to display the Response Time detail table as part of the report. The default is Yes.
Include Response Time chart?	Select **Yes** to display the Response Time chart as part of the report. The default is Yes.
Units for Response Time report	Select the response time unit, msec or sec. The default is msec.
Threshold on Response Time chart (selected units)	Specify the response time threshold to display on the chart in the report. The default is 0, which suppresses the threshold indicator in the chart.
Select chart style	Click **Browse [...]** to set the appearance of the chart. The same parameters are used in both the Availability and Response Time charts, if both are produced. The default is Ribbon.
Select output folder	Select **Browse [...]** to specify the report filename and the report folder. You can specify a specific folder or have the system generate the folder each time the report runs.
Add job ID to output folder name?	Select **Yes** to add a job ID to the output folder name. Use a job ID to correlate a specific instance of a Report Script with the corresponding report. The default is unselected.

Description	How to Set It
Index-Report Title	Select **Browse [...]** to configure report title settings and custom fields.
Add timestamp to title?	Select **Yes** to add a timestamp to the report title, making each title unique. The time stamp is made up of the date and time the report was generated. A time stamp lets you run consecutive iterations of the same report without overwriting previous output. The default is unselected.
Event Notification	
Generate event on success?	Select **Yes** to raise an event when a report is generated. The default is Yes.
Severity level for report success	Set the event severity level to indicate the importance of an event in which the report is generated successfully. The default is 35.
Severity level for report with no data	Set the event severity level to indicate the importance of an event in which the report has no data. The default is 25.
Severity level for report failure	Set the event severity level to indicate the importance of an event in which the report cannot be generated. The default is 5.

3.7 Report_SQL-RT_DSN

Use this Report Knowledge Script to generate a report detailing availability and response time for the following SQL-RT DSN Knowledge Scripts:

- ADODSNQuery
- ODBCDSNQuery

Resource Object

AppManager repository

Default Schedule

The default schedule is **Run once**.

Setting Parameter Values

Set the following parameters as needed:

Description	How to Set It
Data Source	
KS for report	Click **Browse [...]** to select the Knowledge Scripts you want to include in the report.

Description	How to Set It
SQL-RT client(s)	Click **Browse [...]** to select the SQL-RT client computers to include in the report. Your selections are limited to computers or server groups that are visible in the selected views.
	Select one of the filter options:
	◆ **View:** Includes all computers in the views you selected.
	◆ **Computer:** Select from individual computers in the views you selected.
	◆ **Server Group:** Select from server groups in the views you selected.
	NOTE: Selecting a server group includes all computers in that group.
SQL DSN or "All"	Specify the name of the SQL Server or type "All" to designate all computers as SQL Servers. The default is "All".
Select time range	Click **Browse [...]** to set specific start and end report information dates, good for historical or ad hoc reports, or a sliding range that sets the time range of data to include in the report. The sliding range option is the default and is useful for reports running on a regular schedule.
Select peak weekday(s))	Click **Browse [...]** to select a contiguous day range a selection of non-contiguous days.
Aggregation by	Select the time unit by which to aggregate data. The default is Hour. This works in conjunction with the next field (Aggregation interval), which determines the number of units for one interval of data aggregation.
Aggregation interval	Select the interval units in which to aggregate data. The default is 1. For example, if you aggregate by the Hour and select 1 here, data is aggregated once every hour.
Report Settings	
Include parameter card?	Select **Yes** to display a table of parameters used in the report. The default is Yes.
Include Availability detail table?	Select **Yes** to display the Availability Detail table as part of the report. The default is Yes.
Include Availability chart?	Select **Yes** to display the Availability chart as part of the report. The default is Yes.
Availability data stream format	Specify the data stream format. Options are 0-100 or 0-1.
	◆ 1 or 100 -- Available
	◆ 0 -- Not available
	The default format is 0-100.
Threshold on Availability chart	Specify an integer for the percentage threshold. The default is 0 (no threshold is displayed).
Include Response Time Detail table?	Select **Yes** to display the Response Time Detail table as part of the report. The default is Yes.
Include Response Time chart?	Select **Yes** to display the Response Time chart as part of the report. The default is Yes.
Units for Response Time report	Select the response time unit, msec or sec. The default is msec.
Threshold on Response Time chart (selected units)	Specify the response time threshold to display on the chart in the report. The default is 0, which suppresses the threshold indicator in the chart.

Description	How to Set It
Select chart style	Click **Browse [...]** to set the appearance of the chart. The same parameters are used in both the Availability and Response Time charts, if both are produced. The default is Ribbon.
Select output folder	Select **Browse [...]** to specify the report filename and the report folder. You can specify a specific folder or have the system generate the folder each time the report runs.
Add job ID to output folder name?	Select **Yes** to add a job ID to the output folder name. Use a job ID to correlate a specific instance of a Report Script with the corresponding report. The default is unselected.
Index-Report Title	Select **Browse [...]** to configure report title settings and custom fields.
Add timestamp to title?	Select **Yes** to add a timestamp to the report title, making each title unique. The time stamp is made up of the date and time the report was generated. A time stamp lets you run consecutive iterations of the same report without overwriting previous output. The default is unselected.
Event Notification	
Generate event on success?	Select **Yes** to raise an event when a report is generated. The default is Yes.
Severity level for report success	Set the event severity level to indicate the importance of an event in which the report is generated successfully. The default is 35.
Severity level for report with no data	Set the event severity level to indicate the importance of an event in which the report has no data. The default is 25.
Severity level for report failure	Set the event severity level to indicate the importance of an event in which the report cannot be generated. The default is 5.

3.8 Discovery_SQL-RT

Use this Knowledge Script to discover AppManager ResponseTime for SQL Server resources on a specified computer.

After successful discovery, a new object appears in the TreeView pane with a list of computers on which you are performing discovery.

Resource Objects

Windows XP, Windows 2000, Windows 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, or Windows Server 2012 R2.

Default Schedule

The default schedule is **Run once**.

Setting Knowledge Script Parameters

Set the following parameters as needed:

Description	How to Set It
Raise event if discovery succeeds?	This Knowledge Script always raises an event when the job fails for any reason. In addition, you can select the **Yes** check box to raise an event when the job succeeds. By default, events are not raised on success.
Event severity when discovery...	Set the event severity level, from 1 to 40, to reflect the importance when the job: ◆ **... succeeds**. If you set this Knowledge Script to raise an event when the job succeeds, set the event severity level for a successful discovery. The default is 25. ◆ **... fails**. The default is 5. ◆ **... partially succeeds**. This type of failure usually occurs when the target computer does not have all the prerequisites installed. The default is 10.

4 Troubleshooting the Module

This chapter describes how to troubleshoot AppManager ResponseTime for Microsoft SQL Server.

4.1 Problems with Installation

Pre-Install Check Failed

Problem: The following message appeared during installation:

```
WARNING: The pre-install check failed for the MO component. This component will not
be installed.
```

Solution: The prerequisites were not met for AppManager ResponseTime for SQL Server to be installed.

If this occurs on a computer that is the repository or Operator Console, some files will be installed, but not the ResponseTime for SQL managed object. This is fine unless you want to run a managed object on this computer.

On the repository computer, the SQL-RT Knowledge Scripts are checked in, and on the Operator Console, the Help files are installed.

Refer to Section 2.1, "System Requirements," on page 15 to check the system prerequisites. Then, run the pre-installation check Knowledge Script on the computer. This script creates an HTML-formatted report that details which requirements were not met. See the *Installation Guide for AppManager* for more information.

Remote Installation Partially Completed

Problem: The event details include the following:

```
WARNING: The pre-install check failed for the MO component. This component will not
be installed.
```

Solution: This event can occur when installing on a computer on which the AppManager Operator Console is installed. This means that the Help files were installed, but the managed object was not installed because the computer did not meet the prerequisites.

Refer to Section 2.1, "System Requirements," on page 15 to check the system prerequisites. Then, run the pre-installation check Knowledge Script on the computer. This script creates an HTML-formatted report that details which requirements were not met. See the *Installation Guide* for AppManager for more information.

4.2   Problems with Discovery

Must Run as "Local System" Account

Problem: Discovery fails with the following message:

```
The "NetIQ AppManager Client Resource Monitor" service (netiqmc.exe) user account
is currently set as "xxxx". It must run as "Local System" account in order to use
this ResponseTime module.
```

Solution: Most ResponseTime Knowledge Script jobs cannot run unless the AppManager Windows agent (netiqmc) is set to run as Local System. In this case, the agent is running under another account.

AppManager ResponseTime for SQL Server runs out-of-process from the AppManager agent. The separate process for the managed object runs as the user ID specified in each Knowledge Script. The agent must have the authority to start a new process as any user ID specified in a Knowledge Script parameter. Therefore, the agent must run with Local System authority.

Update the agent so that it runs as Local System. For more information, see Section 2.4, "Permissions for Running Knowledge Scripts," on page 17.

SQL-RT Is Not Supported

Problem: The following errors are returned after running Discovery:

```
The SQL ResponseTime Managed Object returned SQL-RT is not supported.
```

or:

```
The SQL ResponseTime Managed Object is not installed or not registered. ActiveX
component can't create object.
```

Solution: The ResponseTime for SQL module is not installed on the computer. Ensure the prerequisites are met, and install again.

Refer to Section 2.1, "System Requirements," on page 15 to check the system prerequisites. Then, run the pre-installation check Knowledge Script on the computer. This script creates an HTML-formatted report that details which requirements were not met. See the *Installation Guide for AppManager* for more information.

SQL-RT Is Not Installed

Problem: Discovery failed, with the following message:

```
SQL-RT is not installed.
```

Solution: The AppManager ResponseTime for SQL Server module is not installed. Ensure the prerequisites were met, and try installing again.

Refer to Section 2.1, "System Requirements," on page 15 to check the system prerequisites. Then, run the pre-installation check Knowledge Script on the computer. This script creates an HTML-formatted report that details which requirements were not met. See the *Installation Guide for AppManager* for more information.

Class Not Registered

Problem: Discovery failed, with the following message:

```
SQL-RT cannot work properly. Class not registered.
```

Solution: Some ResponseTime shared components are installed, but AppManager ResponseTime for SQL Server is not installed. Ensure the prerequisites were met, and try installing again.

Refer to Section 2.1, "System Requirements," on page 15 to check the system prerequisites. Then, run the pre-installation check Knowledge Script on the computer. This script creates an HTML-formatted report that details which requirements were not met. See the *Installation Guide for AppManager* for more information.

Parameter Is Incorrect

Problem: Discovery failed with the following message:

```
SQL-RT cannot work properly. The parameter is incorrect.
```

Solution: The `NetIQmc` agent service is running as a specific user on the computer, and a different user is logged on to the computer. Change `NetIQMC` to run as Local System. For more information, see Section 2.4, "Permissions for Running Knowledge Scripts," on page 17.

Backlevel Version of SQL-RT Installed

Problem: Discovery failed with the following message:

```
This is a backlevel version of AppManager ResponseTime for SQL Server. Please
install the latest version of the software.
```

Solution: Install the latest version of the AppManager ResponseTime for Microsoft SQL Server module and re-run Discovery.

4.3 Problems Running Knowledge Scripts

You may experience difficulties when running the SQL-RT Knowledge Scripts. This section covers the following types of AppManager error message:

Networking or Authentication Errors

Invalid Authentication Parameter

Problem: The Knowledge Script job fails with the following error:

```
Invalid or misspelled authentication parameter.
```

Solution: The ResponseTime managed objects validate the *Run As Username/Domain* Knowledge Script parameters, and start the ResponseTime managed object process as that user. If the client cannot access the Domain Controller for the domain listed for the *Run As Domain* parameter, the process cannot be started, and the transaction fails.

Review the following error message table:

Error Message	Likely Cause
Unable to validate domain user.	Invalid *Run As Username* or *SQL Logon Username.*
	This is often caused by an invalid username. It may also be caused by trust relationship failure between the primary domain and the trusted domain.
The server process could not be started because the configured identity is incorrect.	Server execution failed.
	Invalid *Run As Password* or *SQL Logon Password* parameter.
Check the username and password.	
The xxx ResponseTime Managed Object's RunTransaction method returned –2147024809	Invalid *Run As Domain* or *SQL Logon Domain* parameter
Could not find the domain controller for the domain.	
The specified local group does not exist.	Invalid value for *Administrators group on managed client* parameter

Must Run as "Local System" Account

Problem: The ResponseTime for SQL managed object's `RunTransaction` method returned the following message:

```
The "NetIQ AppManager Client Resource Monitor" service (netiqmc.exe) user account
is currently set as "xyz". It must run as "Local System" account in order to use
this ResponseTime module.
```

Solution: The `netiqmc` service must run as Local System. This is required for most AppManager ResponseTime modules. For more information, see Section 2.4, "Permissions for Running Knowledge Scripts," on page 17.

Then try running the job again.

Could Not Find Domain Controller

Problem: The ResponseTime for SQL managed object's `RunTransaction` method returned the following message:

```
Could not find the domain controller for the domain.
```

Solution: One cause of this problem is that the value you supplied for the *Run As Domain* Knowledge Script parameter is misspelled or does not exist.

The ResponseTime for SQL managed object validates the values supplied for the *Run As Username* and *Domain* parameters, and starts the ResponseTime for SQL managed object process under that user account. If the client cannot access the Domain Controller for the domain listed as the *Run As Domain* parameter, the process cannot be started, and the transaction fails.

Ensure the domain supplied for the *Run As Domain* parameter is valid and is typed properly. Verify that the Domain Controller is active and that there are no networking problems between the client and the Domain Controller. If the problem persists, contact the domain administrator.

Unable to Validate Local Group Member

Problem: The following error occurred while a Knowledge Script was running:

```
Unable to validate the local group member. Problems with operating system.
```

NOTE: It is normal to see this error during network maintenance.

Solution: This message means that the operating system cannot provide information about a local group at that moment. This error may occur during a network outage or system maintenance.

If the problem persists, contact your network administrator.

Unable to Validate Domain User

Problem: The following error occurred while a Knowledge Script was running:

```
Unable to validate Domain User. Problems contacting the domain controller while
validating domain name and user account.
```

NOTE: *Solution*: This error may occur during network outage or system maintenance. It may also occur if the Domain Controller is shut down or reboots during an AppManager ResponseTime for SQL Server operation.

If the problem persists, contact your network administrator.

Member Does Not Exist

The `RunTransaction` method of the ResponseTime for SQL Server module returned the following message:

```
A member could not be added to or removed from the local group because the member
does not exist.
```

Solution: The computer is not part of a domain (it cannot be in a Workgroup), or it is not part of the domain (or a trusted domain) specified for the *Run As Domain* parameter in the Knowledge Script.

Problems with SQL Server or Knowledge Script Configuration

Server Execution Failed

Problem: The ResponseTime for SQL managed object's `RunTransaction` method returned the following:

```
-2146959355 Server execution failed
```

Solution: The transaction may have failed due to network or server problems. Verify that there are no networking problems between the client and SQL Server, and that SQL Server is up and running.

RPC Server Unavailable

Problem: The `RunTransaction` method of the ResponseTime for SQL Server module returned the following message:

```
–2147024809, The RPC server is unavailable.
```

Solution: Increase the schedule time of the SQL-RT jobs you are running, and reduce the number of jobs running on the computer.

Transaction Failures

When the SQL-RT Knowledge Scripts raise events stating that a "Transaction Failure" occurred, the cause is related either to the connection to the network or server or to the transaction that the Knowledge Script is attempting to perform. The SQL Server driver returns the specific error.

Dealing with Any Transaction Failure

With any Transaction Failure event, the event details include information to help you understand why the failure occurred. Use this information to isolate the problem. The event details will point toward one of the following:

- errors returned by the driver.
- values you entered in the Knowledge Script.

Or you may want to look more closely at the "Job progress" section of the event detail message, which denotes the place in the transaction where the failure occurred. For example, the "Job progress" may indicate that the job proceeded no farther than "`Connecting to database server.`"

Transaction Failure

Problem: The job fails with a Transaction Failure. The event details include additional information to help you determine the reason for the failure.

Solution: Use the following information in the event details to isolate the problem:

- errors returned by the SQL driver
- transaction configuration settings you entered in the Knowledge Script
- the Job Progress, which denotes where in the transaction that the failure occurred

Example 1:

```
[Microsoft][ODBC SQL Server Driver][SQL Server]Login failed for user 'test'.
```

Solution: You probably entered an invalid *Run As Username* or *SQL Logon Username,* or an invalid password.

Near the bottom of the event details, the "Job progress" section indicates that the job failed while "Connecting to database server."

Example 2:

```
Microsoft OLE DB Provider for SQL Server: Invalid object name 'xxxxx'.
```

Solution: You probably entered an invalid name in your SQL statement.

Near the bottom of the event details, the "Job progress" section indicates that the job failed while "Execut[ing] SQL statement."

Transaction Initialization Failures

Transaction initialization failures indicate that a failure occurred in the network authentication procedure.

The ResponseTime for SQL Server module validates the values you enter for the *SQL Logon* and *Run As User ID/Domain* Knowledge Script parameters. The managed object then connects to the SQL Server Database server to run the test as that "run as" user. If the client cannot access the SQL Server server using the information provided, the process cannot be started, and the transaction fails. Event details vary depending on the cause of the failure.

SQL-RT cannot be initialized

Problem: The job fails with a Transaction Initialization Failure. The following are examples of what the event details might include:

Example 1:

```
Knowledge Script Error 0x803CF007: SQL-RT cannot be initialized
Error Code: 0x80070057
Error Message: Unable to validate Domain User. The UserName is not a valid user on
the domain.
```

Solution: This error was a "transaction initialization failure" because the transaction itself was never performed. As you can see from the event details, an invalid username, domain, or password was supplied for one of the *SQL Logon* parameters in the Knowledge Script. The transaction was never initialized because the user could not be validated by the domain controller.

If you ran the job as "interactive user," ensure a user was logged onto the computer.

Example 2:

```
Knowledge Script Error 0x803CF007: SQL-RT cannot be initialized
Error Code: 0x80080005
Error Message: Server execution failed
```

Solution: You may see this error if you run many simultaneous SQL-RT Knowledge Script jobs. However, the real problem is not the number of jobs you are running. If you see this error, upgrade your AppManager agents the latest version. For information about the latest version, see the NetIQ Web site: www.netiq.com/support/am/extended/upgrade/default.asp.

Example 3:

```
Event: Transaction initialization failure
Event Details:
Knowledge Script Error 0x803CF007: SQL-RT cannot
be initialized
Error Code: 0x803CF004
Error Message: The module requested has been uninstalled or is outdated. Please
install the latest version of the module.
```

Solution: The latest versions of the SQL-RT Knowledge Scripts are installed on the repository, but you are attempting to run them on a backlevel version of the AppManager ResponseTime for SQL Server module. You may have installed the most recent version of another AppManager ResponseTime module, but you are missing the latest engines needed to run this Knowledge Script job.

Install the latest version of AppManager ResponseTime for SQL Server on the client computer, re-run the Discovery Knowledge Script, and then restart this job. For more information about the latest version of the module, see the NetIQ Web site: www.netiq.com/support/am/extended/modules.asp.

SQL-RT cannot be initialized

Problem: The job fails with a transaction initialization failure. The event details include the following:

```
Knowledge Script Error 0x803CF007: SQL-RT cannot be initialized.
Error Code: 0x8000401A
Error Message: The server process could not be started because the configured
identity is incorrect. Check the username and password.
```

Solution: Verify that the *Run As Username, Password*, and *Domain* are correct. If running as "Interactive User", ensure a user is logged onto the computer.

The Knowledge Script's ConfigJob method failed

Problem: The Knowledge Script job failed with the following error message:

```
The Knowledge Script's ConfigJob method failed
Knowledge Script Error 0x803CF003: ConfigJob failed unexpectedly. The Knowledge
Script may have been modified manually.
Error Message: The remote server machine does not exist or is unavailable.
```

The error code associated with this failure is one of the following: `0x1CE`, `0x800706BE`, or `0x800706BF`.

Solution: You may see this error if you run many simultaneous SQL-RT Knowledge Script jobs. However, the real problem is not the number of jobs you are running. If you see this error, upgrade your AppManager agents the latest version. For information about the latest version, see the NetIQ Web site: www.netiq.com/support/am/extended/upgrade/default.asp.

The configured identity is incorrect

Problem: When you ran a Knowledge Script job, you received the following error message:

```
Error Code: 0x8000401A
Error Message: The server process could not be started because the configured
identity is incorrect. Check the username and password.
```

Solution: This error indicates that you entered incorrect security credentials for the Knowledge Script parameters.

Check the values you entered for the *SQL Logon Username, Password*, and *Domain* parameters. One of them is probably invalid or mistyped.

SQL-RT is not installed

Problem: The Knowledge Script job failed with the following message:

```
Knowledge Script Error 0x803CF004: SQL-RT module has been uninstalled or is
outdated. Please install the latest version of the module.
Error Code: 0x1AD
Error Message: ActiveX component can't create object
```

`Solution`: The latest versions of the SQL-RT Knowledge Scripts are installed on the repository, but you are attempting to run them on a backlevel version of AppManager ResponseTime for SQL Server.

Install the latest version of AppManager ResponseTime for SQL Server on the client computer, re-run the Discovery Knowledge Script, and then restart this job. For more information about the latest version of the module, see the NetIQ Web site: www.netiq.com/support/am/extended/modules.asp.