NetIQ® AppManager®

Upgrade and Migration Guide

May 2020

Legal Notice

Contents

About this Book and the Library

The NetIQ AppManager Suite (AppManager) is a comprehensive solution for managing, diagnosing, and analyzing performance, availability, and server health for a broad spectrum of operating environments, applications, and server hardware.

AppManager provides system and application administrators with a central, easy-to-use console to view critical resources across the enterprise. With AppManager, administrative staff can monitor computer and application resources, check for potential problems, initiate and automate responsive actions, automate routine tasks, and gather performance data for real-time and historical reporting and analysis.

Intended Audience

This guide is intended for organizations that have a previous version of AppManager installed and want to upgrade to the latest version. It includes tips and recommendations for a smooth upgrade of all AppManager components. This guide assumes you are already familiar with AppManager components and the installation process.

If you are new to AppManager, have installed AppManager but never Control Center, or want to perform a fresh installation instead of an upgrade, see the *Installation Guide for AppManager*.

Other Information in the Library

The library provides the following information resources:

Installation Guide

Provides detailed planning and installation information.

Administrator Guide

Provides information about maintaining an AppManager management site, managing security, using scripts to handle AppManager tasks, and leveraging advanced configuration options.

Control Center User Guide

Provides information about managing groups of computers, including running jobs, responding to events, creating reports, and working with the Control Center console.

Operator Console User Guide

Provides information for system and network administrators working with the AppManager Operator Console.

Module management guides

Provide information about installing and monitoring specific applications with AppManager.

NetIQ UNIX Agent documentation

Provides information about installing, upgrading, and configuring the NetIQ UNIX Agent and UNIX Agent Manager.

About NetIQ Corporation

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

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

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

Enabling critical business services, better and faster

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

Our Philosophy

Selling intelligent solutions, not just software

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

Driving your success is our passion

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

Our Solutions

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

Contacting Sales Support

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

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

Contacting Technical Support

For specific product issues, contact our Technical Support team.

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

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ website in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **comment on this topic** at the bottom of any page in the HTML versions of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

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

1 Preparing to Upgrade

This chapter describes items to consider before you upgrade existing AppManager Windows components to version 9.5. For information about upgrading the NetIQ UNIX agent, see the *AppManager for UNIX and Linux Servers Management Guide*, available on the AppManager Modules Documentation page (https://www.netiq.com/documentation/appmanager-modules/).

1.1 Planning Your Upgrade

The following checklist outlines the basic steps for planning an upgrade and provides references to detailed information.

Step		Reference
☐	1. Review the features available with AppManager version 9.5.	Section 1.2, "Understanding New Features," on page 10
☐	2. Decide whether to upgrade components all at once or on an as-needed basis.	Section 1.3, "Understanding Supported Upgrade Scenarios," on page 12
☐	3. Ensure that the components you want to upgrade meet system requirements and upgrade prerequisites.	◆ Section 1.4, "Understanding Changes to Supported Operating Systems and Databases," on page 13 ◆ Section 1.5, "Understanding Upgrade Prerequisites," on page 14 ◆ *Installation Guide for AppManager*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager)
☐	4. Ensure that your existing environment is in a healthy state.	Section 1.6, "Preparing Your Existing Environment for Upgrade," on page 16
☐	5. Schedule components for upgrade in the correct order.	Section 1.7, "Understanding the Recommended Order for Upgrading Components," on page 16
☐	6. Run the setup program to generate a pre-installation check report and resolve issues.	◆ Section 1.8, "Understanding Upgrade Methods," on page 17 ◆ Section 1.9, "Starting an Upgrade and Generating a Pre-Installation Check Report," on page 18

1.2 Understanding New Features

When deciding which components to upgrade, it is important to consider the features available with version 9.5 and the requirements for using those features.

1.2.1 Knowledge Script Propagation Improvements

AppManager 9.2 introduces the concept of **base** and **derived** Knowledge Scripts. A base Knowledge Script is checked in to the AppManager repository (QDB) and holds all of the Knowledge Script settings (the schedule, parameters, and Advanced tab) and the script logic. A derived Knowledge Script is a copy of a base Knowledge Script (a direct copy or a member of a Knowledge Script Group) and shares the logic with the base Knowledge Script but has its own settings. For example, a base Knowledge Script might have a default threshold of 10 for a certain parameter while the derived Knowledge Script has a default threshold of 20. The two Knowledge Scripts share the script logic, but the settings are different.

Because a base Knowledge Script and a derived Knowledge Script share logic, when the logic for a base Knowledge Script changes, AppManager can automatically update the logic for the derived Knowledge Script without updating the settings.

With AppManager 9.2, when you install a new version of a module, AppManager can automatically update the logic of the derived Knowledge Scripts without changing any defaults or settings. For new QDB installations, this is the default behavior. For upgrades, you must enable the automatic propagation feature. To enable the feature, in the **Knowledge Scripts** options in the Control Center console, select **Automatically update Derived KS properties for checked in Base KSs**.

The automatic propagation feature includes monitoring policy jobs but not ad-hoc jobs. For ad-hoc jobs, you must manually propagate changes from the base Knowledge Script to the derived Knowledge Script. With both automatic and manual propagation, the target Knowledge Script or job retains its settings (such as thresholds and schedule). If a base Knowledge Script includes new values or removed values, AppManager also propagates those changes to the derived Knowledge Scripts.

If you do not enable the automatic propagation option, when you log in to the AppManager consoles and base Knowledge Scripts have been updated, AppManager informs you that Knowledge Scripts are pending propagation to running jobs and provides the option to start the Knowledge Script Propagation Wizard to select which Knowledge Scripts to propagate. The wizard includes options to propagate to both derived Knowledge Scripts and to ad-hoc jobs. In Control Center, when you select a Knowledge Script on the **Propagation to Ad Hoc Jobs** tab, you can see the QDBs where the job is running and can choose to propagate to ad-hoc jobs in specific QDBs.

For more information about automatic and manual propagation, see the Control Center User Guide for AppManager.

1.2.2 Improved Efficiency in Diagnosing Remote Deployment Issues

To allow you to more easily diagnose issues with remote deployment, AppManager 9.2 or later includes the ability to view deployment rule processing details in the Control Center console.

After you create and enable a deployment rule, AppManager evaluates the rule for processing and displays the status in the Control Center console. To view the evaluation history, click **Rule History** in the **Deployment** view of the **Navigation** pane.

The **Rule Processing History** portion of the pane indicates whether processing was successful. If AppManager was not able to create a deployment task for the rule, it provides details about the error in the **Task Comment** column.

If the evaluation status changes the next time that AppManager evaluates the rule, AppManager updates the rule history. AppManager does not update the rule history if the evaluation status does not change.

1.2.3 Increased Scalability

Version 9.2 or later incorporates data scalability improvements from version 9.1 into the user interface for improved console performance. In test environments, improved response times were observed in large environments.

1.2.4 Reduced Hardware Costs for Small Environments

To eliminate the requirement for a Microsoft SQL Server license, for small environments with all components on one computer, AppManager 9.2 or later supports hosting the AppManager repository (QDB) and Control Center repository (CCDB) on Microsoft SQL Server Express.

1.2.5 Database Size Control

AppManager 9.2 or later reduces the amount of disk space that the QDB requires by implementing a streamlined method of collecting and storing data.

Previously, AppManager stored collected data for use in short-term charts and graphs in the `Data` table for eight days, and stored data for long-term reporting in the `ArchiveData` table indefinitely (by default). If a Knowledge Script generated data details in addition to data points, AppManager automatically stored the details in the `Data` table.

Now, AppManager stores all of the data that it collects on a given day in `Data_`*`yyyymmdd`* tables for immediate display in real-time charts and graphs for 30 days (by default), and each day removes tables that are more than 30 days old. A new option in the Control Center console, **Remove old data after**, allows you to change the default retention period. For more information, see Global Preferences Options in the *Control Center User Guide for AppManager*. For long-term reporting needs, NetIQ Corporation recommends using NetIQ Analysis Center.

In addition, AppManager no longer automatically collects data details with data points. The **Collect data details with data point** repository preference in the Operator Console allows you to select whether to collect data details by default. If you do collect data details, AppManager stores them in the `Data_`*`yyyymmdd`* tables.

NOTE: When you upgrade, the upgrade does not change the current setting for **Collect data details with data point**. If you previously collected details and no longer want to collect them, you must manually change the setting.

Previously, it was necessary to periodically archive and aggregate data in the QDB to prevent the `ArchiveData` table from becoming too large. With this streamlined method of collecting and storing data, archiving is no longer necessary.

1.2.6 Agent Migration Tool

AppManager 9.2 or later includes a command line tool, `MigrateQ.exe`, that automates migrating existing Windows and UNIX agents to a new QDB and allows you to migrate multiple agents simultaneously, if needed. You can migrate version 7.0.4 and later agents to new QDBs. For more information about using the tool, see Section 3.7, “Migrating an Agent to a New QDB,” on page 32.

1.3 Understanding Supported Upgrade Scenarios

You do not have to upgrade all of your AppManager components to version 9.2 or later at the same time; however, components you do not upgrade might not support new features. This section provides information about the configurations AppManager 9.2 or later supports.

1.3.1 Upgrading Management Servers and Agents in a Single QDB Environment

If you have only one QDB in your environment, you must upgrade the QDB and the primary and secondary management servers that connect to the QDB. When you upgrade a management server, you also upgrade the agent on the management server computer. Otherwise, you can upgrade agents on an as-needed basis.

While a version 9.5 management server can communicate with version 7.x, 8.x, 9.1, and 9.2 agents, version 9.5 agents cannot communicate with earlier versions of the management server.

For more information about upgrading QDBs and management servers, see Chapter 2, “Upgrading Management Site Components,” on page 21.

For more information about upgrading agents, see Chapter 3, “Upgrading and Migrating Agent Components,” on page 29.

1.3.2 Upgrading Components in a Multiple-QDB, Control Center Environment

With Control Center version 9.5, the primary QDB must be the same version as the CCDB. You can either create a new version 9.5 primary QDB or upgrade your existing version 9.1 or 9.2 primary QDB to version 9.5. You cannot upgrade QDBs earlier than version 9.1 to version 9.5. If you create a new primary QDB, in order for the product to function correctly, you must add at least one agent computer to the QDB and discover the computer. With a version 9.5 primary QDB, you can maintain existing non-primary QDBs and agents as long as they meet the requirements described in the following table.

Existing Component	AppManager Version	Microsoft SQL Server Version
Non-primary QDB	9.1 or 9.2	One of the following: ◆ 2014 Standard or Enterprise edition ◆ 2012 Standard or Enterprise edition (32-bit or 64-bit) ◆ 2008 R2 Standard or Enterprise edition (32-bit or 64-bit) ◆ 2008 Standard or Enterprise edition Service Pack 1 or later (32-bit or 64-bit)
Agent	One of the following: ◆ 7.0.4 ◆ 8.0.3 ◆ 8.2 ◆ 9.1 ◆ 9.2	Not applicable

Once you have a version 9.5 primary QDB, you can upgrade non-primary QDBs on an as-needed basis. When you upgrade a non-primary QDB, you must also upgrade the primary and secondary management servers that connect to that QDB. When you upgrade a management server, you also upgrade the agent on the management server computer. Otherwise, you can upgrade agents on an as-needed basis.

For more information about creating a new version 9.5 QDB, see Installing a Management Site in the *Installation Guide for AppManager*. For more information about upgrading existing QDBs and management servers to version 9.5, see Chapter 2, “Upgrading Management Site Components,” on page 21. For more information about upgrading agents, see Chapter 3, “Upgrading and Migrating Agent Components,” on page 29.

For more information about upgrading Control Center to version 9.5, see Chapter 4, “Upgrading Control Center Components,” on page 39.

1.4 Understanding Changes to Supported Operating Systems and Databases

AppManager version 9.5 adds support for the following operating systems and databases:

- For the AppManager repository (QDB) and Control Center repository (CCDB), SQL Server 2017 Standard and Enterprise editions

For more information about supported operating systems, databases, and system requirements, see System Requirements in the *Installation Guide for AppManager*.

1.5 Understanding Upgrade Prerequisites

This section describes prerequisites your existing AppManager components must meet before you upgrade them.

1.5.1 Understanding Microsoft SQL Server Versions Supported for Upgrade and Migration

Version 9.5 QDBs and CCDBs must be hosted on one of the following versions of Microsoft SQL Server:

- Microsoft SQL Server 2017 Standard or Enterprise edition
- Microsoft SQL Server 2016 Standard or Enterprise edition
- Microsoft SQL Server 2014 Standard or Enterprise edition
- Microsoft SQL Server 2012 Standard or Enterprise edition (32-bit or 64-bit)
- Microsoft SQL Server 2008 R2 Standard or Enterprise edition (32-bit or 64-bit)
- Microsoft SQL Server 2008 Standard or Enterprise edition Service Pack 1 or later (32-bit or 64-bit)

AppManager 9.5 supports migrating version 9.1 or 9.2 QDBs and CCDBs hosted on SQL Server 2008 R2 or later to SQL Server 2016. For more information about upgrading a QDB, see Chapter 2, “Upgrading Management Site Components,” on page 21. For more information about upgrading a CCDB, see Chapter 4, “Upgrading Control Center Components,” on page 39. For more information about migrating an upgraded repository, see Appendix A, “Migrating Repositories,” on page 45.

1.5.2 Understanding AppManager Versions Supported for Upgrade

Before you upgrade AppManager components, ensure that the components meet version prerequisites. The following table lists the components and the AppManager versions supported for upgrade.

Component	Supported AppManager Versions for Upgrade
QDB	9.1 or 9.2
Management server	9.1 or 9.2
Agent	7.0.4 8.0.3 8.2 9.1 9.2

Component	Supported AppManager Versions for Upgrade
CCDB	9.1 or 9.2
Command queue service (CQS)	
Deployment Service	
Deployment Web Service	
Control Center console	
Operator Console	

For more information about obtaining a supported version, contact Technical Support (http://www.netiq.com/support).

If you have a report agent, before you upgrade, you can run the `CompVersion` report Knowledge Script to generate a report detailing the version number of AppManager components installed on all computers in an AppManager site. For more information about running the Knowledge Script, see the *AppManager Knowledge Script Reference Guide*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

If you do not have a report agent, use the methods described in the following table to obtain component version information.

Component	Steps to take
Operator Console	1. Start the Operator Console. 2. Select **Help > About AppManager Operator Console**.

QDB	1. In the Operator Console, select **Extensions > Repository Browser**.
	2. From the **Tables** list, select **Version**.
	3. Run the following query:
	`SELECT * FROM Version`
	4. Scroll through the results until you find the following record:
	`MachineName: computer name`
	`Component: Repository`
	`Version: current QDB version`

Management server	1. In the Operator Console, right-click the server. 2. Select **Troubleshooter > Management Service Info > Connectivity**.
Agent	1. In the Operator Console, right-click the computer whose agent you want to check. 2. Select **Troubleshooter > Client Resource Monitor Info > Connectivity**.

Control Center components	Use the Registry Editor to view the following key:
	`HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager`
	`\Control Center\1.0`

1.6 Preparing Your Existing Environment for Upgrade

This section describes steps that you can take to ensure that your existing environment is in a state that facilitates a smooth upgrade.

To prepare your existing environment for upgrade:

1 Ensure that all agent computers are available and communicating with their designated management server:

1a In the Control Center console, ensure that there are no open events from the AppManager for Self Monitoring module (AMHealth) that indicate a problem with an agent computer. If you find issues, resolve them.

1b Use Control Center to create a **Server** view that shows agent computers where the agent status is offline and resolve any issues with offline computers. For more information about creating a **Server** view, see the *Control Center User Guide for AppManager*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

1c Use the `netiqsync.exe` utility to check for differences in job status between agents and QDBs and, optionally, stop orphaned jobs. The utility is available in the `Extras\Agent Migration\Utilities\Appmanager Tools` folder in the location where you saved the installation package. For more information about using the utility, see the `readme.txt` file.

2 Back up the CCDB and all QDBs. For more information about creating a backup copy, see Section A.2.1, “Creating Backup Copies of the Repositories,” on page 49.

3 Ensure that the management server is in a healthy state:

3a From a command prompt on the management server, change directory to `Program Files (x86)\NetIQ\AppManager\bin` and enter `netiqctrl.exe`.

3b To identify any pending jobs, enter `stat` *`Server_Name`* `netiqms`, where *Server_Name* is the name of the management server.

3c (Conditional) If the output shows jobs in a constantly pending state, cold start the management server to force a refresh of the pending jobs.

4 (Conditional) If you have configured custom maintenance jobs (for example, database maintenance jobs), disable them until the upgrade is complete.

1.7 Understanding the Recommended Order for Upgrading Components

NetIQ Corporation recommends upgrading AppManager components in the following order:

1 Primary QDB

2 Management server

3 Windows agent

4 Task Scheduler service

5 Modules

6 CCDB

Before you upgrade the CCDB, ensure that it does not contain any QDBs that are earlier than version 9.1. If it does, either remove the QDBs or upgrade them to version 9.1.

7 Control Center and Deployment services

8 Control Center console

9 Non-primary QDBs

10 Operator Console

11 Jobs

    When you start the Control Center console, if derived Knowledge Scripts or ad hoc jobs are pending propagation, you are prompted to start the Knowledge Script Propagation wizard. The wizard allows you to select the scripts and ad hoc jobs to process. For more information about using the wizard, see Manually Propagating a base Knowledge Script to Derived Knowledge Scripts and Ad Hoc Jobs in the *Control Center User Guide for AppManager*.

For the computer on which you run the setup program, the program automatically detects the components available for upgrade and upgrades those components in the recommended order. If you are upgrading components on multiple computers, schedule the upgrades according to the recommended order. For example, if you are upgrading the management server and agent on different computers, run the setup program on the management server before you run it on the agent computer.

If you do not upgrade Control Center components in the recommended order, the Deployment Service will not start after the upgrade.

1.8 Understanding Upgrade Methods

You can upgrade AppManager components interactively or silently from a command prompt. For information about silently upgrading components, see Performing a Silent Installation in the *Installation Guide for AppManager*. Regardless of whether you choose to upgrade components interactively or silently, ensure that the installation path contains only ASCII characters. If a component is already installed in a path that contains non-ASCII characters, uninstall the component, and then install it in a supported path.

To interactively upgrade components, NetIQ Corporation recommends using the AppManager setup program. When you run `Setup.exe`, the setup program runs a pre-installation check script to verify system requirements and then runs individual Windows Installer packages for the components you selected to upgrade. During the upgrade process, the setup program automatically installs the runtime libraries required for the selected components. For more information about running `Setup.exe` to generate a pre-installation check report and interactively upgrade components, see Section 1.9, “Starting an Upgrade and Generating a Pre-Installation Check Report,” on page 18.

You can run Windows Installer packages for individual components instead of using the AppManager setup program. If you use this method to upgrade components, you must complete the following tasks before you run the Windows Installer packages:

- Manually install the required runtime libraries for the components you are upgrading.

  For more information about installing the runtime libraries, see Installing Runtime Libraries in the *Installation Guide for AppManager*.

- (Conditional) If the computer from which you will run the Windows Installer package has the User Access Control (UAC) feature enabled, ensure that the user who will perform the installation is authorized to run the package.

  For more information about ensuring the user is authorized, see Running Windows Installer Packages When UAC is Enabled in the *Installation Guide for AppManager*.

If you choose to upgrade the Control Center services by running the Windows Installer package instead of the AppManager setup program, after the upgrade, the NetIQ AppManager Client Resource Monitor (`NetIQmc`) and NetIQ AppManager Client Communication Manager (`NetIQccm`) services will not start until you upgrade the agent. For more information about upgrading Control Center components, see Chapter 4, “Upgrading Control Center Components,” on page 39.

1.9 Starting an Upgrade and Generating a Pre-Installation Check Report

The upgrade steps in this section are common to all AppManager components, except the UNIX agent. For information about upgrading the UNIX agent, see the *AppManager for UNIX and Linux Servers Management Guide*, available on the AppManager Modules Documentation page (https://www.netiq.com/documentation/appmanager-modules/).

The upgrade process does not change the settings from your previous AppManager installation. If you want to change the installation settings, uninstall the components and then perform a new installation. For more information about performing a new installation, see the *Installation Guide for AppManager*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

To start an upgrade and generate a pre-installation check report:

1. (Conditional) If deployment tasks are in the Waiting for Approval state, approve or reject the tasks before you start the upgrade. Otherwise, the tasks will fail.
2. (Conditional) If you are upgrading only a QDB or CCDB, run `Setup.exe` on the computer from which you want to perform the upgrade.
3. (Conditional) If you are upgrading components other than the QDB or CCDB, run `Setup.exe` on the computer where the components you want to upgrade are installed.
4. On the Welcome window, view the components available for upgrade on the computer.
5. (Conditional) If you want to upgrade a QDB or CCDB, select the appropriate option.
6. Click **Next**.
7. On the Confirmation window, click the link to view the pre-installation check report.

   The AppManager pre-installation check script verifies system requirements and generates a report that summarizes the results. For each requirement, the report provides information about how your environment meets or does not meet the requirement and the check result. The following results are possible:

   - Passed - Your environment passed the check.
   - Warning - Your environment passed the check, but configuration issues exist.
   - Failed - Your environment failed the check.
8. (Conditional) If your environment passed all requirements, click **Next**.

   The AppManager setup program launches the individual component setup programs.
9. (Conditional) If your environment did not pass all requirements, resolve issues and re-generate the pre-installation check report.
10. Complete the upgrade steps for the components you want to upgrade.

The following table provides references to detailed information about upgrading AppManager components.

For more information about upgrading the...	See...
QDB	Chapter 2, “Upgrading Management Site Components,” on page 21
Management server	Chapter 2, “Upgrading Management Site Components,” on page 21
Windows agent	Chapter 3, “Upgrading and Migrating Agent Components,” on page 29
CCDB	Chapter 4, “Upgrading Control Center Components,” on page 39
Control Center and Deployment services	Chapter 4, “Upgrading Control Center Components,” on page 39
Control Center console	Chapter 4, “Upgrading Control Center Components,” on page 39

2 Upgrading Management Site Components

This chapter describes how to upgrade a management site. A management site comprises one QDB, one or more management servers, and the Task Scheduler service.

2.1 Understanding Upgrade Prerequisites

This section describes prerequisites your environment must meet before you upgrade management site components.

2.1.1 Understanding Component Versions Supported for Upgrade

Before you upgrade management site components, the components must be version 9.1 or 9.2. If a component does not meet the prerequisite, you can either uninstall it and install a new version 9.5 component, or you can first upgrade it to version 9.1 or 9.2.

For more information about installing a new version 9.5 QDB or management server, see Installing a Management Site in the *Installation Guide for AppManager*.

When you upgrade a management server, you also upgrade the agent on the management server computer. For more information about upgrading agents, see Chapter 3, “Upgrading and Migrating Agent Components,” on page 29.

For more information about obtaining a version of AppManager that is supported for upgrade, contact Technical Support (http://www.netiq.com/support).

2.1.2 Understanding Microsoft SQL Server Versions Supported for Upgrade

Version 9.5 QDBs must be hosted on one of the following versions of Microsoft SQL Server:

- Microsoft SQL Server 2017 Standard or Enterprise edition
- Microsoft SQL Server 2016 Standard or Enterprise edition
- Microsoft SQL Server 2014 Standard or Enterprise edition
- Microsoft SQL Server 2012 Standard or Enterprise edition (32-bit or 64-bit)
- Microsoft SQL Server 2008 R2 Standard or Enterprise edition (32-bit or 64-bit)
- Microsoft SQL Server 2008 Standard or Enterprise edition Service Pack 1 or later (32-bit or 64-bit)

For small environments with all components installed on the same computer, AppManager 9.5 also supports hosting QDBs on SQL Server Express.

AppManager 9.5 supports migrating version 9.1 or 9.2 QDBs hosted on SQL Server 2008 R2 or later to SQL Server 2016. For more information about migrating an upgraded QDB, see Appendix A, “Migrating Repositories,” on page 45.

2.2 Understanding Control Center Support for QDBs

A version 9.5 CCDB only supports a primary QDB of the same version. You can either create a new version 9.5 primary QDB or upgrade your existing version 9.1 or 9.2 primary QDB to version 9.5. If you create a new primary QDB, in order for the product to function correctly, you must add at least one agent computer to the QDB and discover the computer. For more information about creating a new version 9.5 primary QDB, see Installing a Management Site in the *Installation Guide for AppManager*.

You can attach version 9.1 or 9.2 QDBs to a version 9.5 CCDB as non-primary QDBs.

When you upgrade the CCDB to version 9.5, if the setup program detects a primary QDB that is not the same version or a non-primary QDB that is not version 9.1 or higher, the CCDB upgrade cannot continue.

2.3 Understanding the Order for Upgrading Management Site Components

If you have management site components installed on multiple computers, ensure that you upgrade components in the correct order. On the computer where you run the setup program, the program ensures that you upgrade components in the correct order. However, when you have components installed on multiple computers, ensure that you upgrade the computers in order. For example, if the QDB and the management server are on different computers, upgrade the QDB before you upgrade the management server. For more information about the correct order for upgrading AppManager components, see Section 1.7, “Understanding the Recommended Order for Upgrading Components,” on page 16.

2.4 Discovering Upgraded Management Site Components for Health Monitoring

Once the setup program successfully upgrades a management server, if an upgraded agent is already present, the setup program automatically runs the Discovery_AMHealth Knowledge Script to prepare the management site components for health monitoring in Control Center. Otherwise, the setup program runs the Knowledge Script after the agent upgrade. For more information about using Control Center to monitor the health of your AppManager components, see the *Control Center User Guide for AppManager*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

2.5 Upgrading the QDB

This section describes the steps required to upgrade a QDB. You can upgrade QDBs on remote SQL Servers. You do not have to run the setup program on the SQL Server.

To upgrade the QDB:

1. Close connections to the QDB.

   For more information about the connections to close, see Section A.2, “Preparing to Migrate the Repositories,” on page 45.

2. Create a QDB backup.

For more information about creating a QDB backup, see Section A.2.1, “Creating Backup Copies of the Repositories,” on page 49. Since you are not migrating the QDB to a different computer, it is not necessary to copy the backup files to a different location.

3. Start the upgrade and generate a pre-installation check report.

   For more information about generating the report, see Section 1.9, “Starting an Upgrade and Generating a Pre-Installation Check Report,” on page 18.

4. When you reach the Target SQL Server and Repository Name window, provide the following information and click **Next**:

   - Name of the SQL Server and, if applicable, instance that hosts the QDB you are upgrading. To specify a SQL Server instance, use the format *Server_Name\instance*.
   - Name of the QDB you are upgrading.
   - Account that can log in to the SQL Server for the upgrade. Ensure that the account is a member of the `sysadmin` SQL Server role.

   The setup program prepares the existing QDB data for upgrade and checks in current Knowledge Scripts.

2.6 Upgrading Management Servers

This section describes the steps required to upgrade the management server on the same computer. For information about moving the management server to a new computer, see Section 2.6.5, “Moving the Management Server to a New Computer,” on page 24.

2.6.1 Understanding Version Requirements for Connected Components

If you have only one QDB in your environment, you must upgrade the QDB and the primary and secondary management servers that connect to the QDB. When you upgrade a management server, you also upgrade the agent on the management server computer. Otherwise, you can upgrade agents on an as-needed basis.

While a version 9.5 management server can communicate with version 7.0.4, 8.x, 9.1, or 9.2 agents, version 9.5 agents cannot communicate with earlier management server versions.

For more information about supported management server versions in a multiple-QDB, Control Center environment, see Section 1.3.2, “Upgrading Components in a Multiple-QDB, Control Center Environment,” on page 12.

2.6.2 Understanding Changes to Encryption Algorithms and Effects on Communications with UNIX Agents

AppManager 9.2 includes an update of the OpenSSL version from 1-0-1m to 1-0-2j and replaces the DES encryption algorithm, which is not FIPS-compliant, with AES128, which is FIPS compliant.

With previous versions of AppManager, in environments using the **authentication and encrypted communications** security level, the management server used the DES encryption algorithm to encrypt and decrypt public key data that it shared with UNIX agents. In environments where FIPS is enabled, OpenSSL 1-0-2j does not allow using the DES encryption algorithm. However, when you upgrade a management server to version 9.2, the management server must decrypt public key data that was encrypted using the DES algorithm. To allow a version 9.2 management server to continue working with existing UNIX agents without requiring you to rekey the agents, when the management server starts, AppManager temporarily disables FIPS mode so that it can use the DES algorithm to decrypt the public key data and then restores FIPS mode when the decryption is complete.

2.6.3 Reviewing Additional Upgrade Recommendations

If you choose to upgrade by running the Windows Installer package for the management server instead of the AppManager setup program, after the upgrade, the NetIQ AppManager Client Resource Monitor (`NetIQmc`) and NetIQ AppManager Client Communication Manager (`NetIQccm`) services will not start until you upgrade the agent.

NetIQ Corporation does not recommend clustering the management server. Instead, you can install multiple management servers and designate them as primary and secondary to provide failover support. For more information about installing additional management servers and designating primary and secondary management servers, see the *Installation Guide for AppManager* and the *Administrator Guide for AppManager*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager). If you have a requirement to install a management server on Microsoft Cluster Service (MSCS), contact Technical Support (http://www.netiq.com/support).

2.6.4 Performing the Upgrade

To upgrade the management server on the same computer:

1. Ensure that the upgrade of the QDB to which the management server connects completed successfully.
2. Start the upgrade and generate a pre-installation check report.

   For more information about generating the report, see Section 1.9, “Starting an Upgrade and Generating a Pre-Installation Check Report,” on page 18.
3. Complete the management server setup program.

Depending on your environment, you might need to update existing encryption keys after you upgrade the management server. For more information about updating encryption keys, see Section 2.6.6, “Using Existing Security Keys for Encrypted Communications,” on page 26.

You can also change the security level after you upgrade. For more information about changing the security level, see Section 2.6.7, “Changing the Security Level,” on page 27.

2.6.5 Moving the Management Server to a New Computer

This section describes how to move the management server to a new computer. You might want to move the management server if the current computer does not support upgrading to an operating system that AppManager 9.5 supports.

Moving the management server to a new computer requires ensuring that it will be able to communicate with the QDB and agents after the move, installing a new version 9.5 management server on the new computer, and uninstalling the management server from the old computer.

To move the management server:

1. Upgrade the QDB with which the management server communicates to version 9.5.

   For more information about upgrading the QDB, see Section 2.5, “Upgrading the QDB,” on page 22.

2 To allow agent computers that communicate with the management server to temporarily allow communication with anonymous management servers, run the AMAdmin_SetAllowMS Knowledge Script and set the **New hostname(s) for AllowMS** parameter to an asterisk (*).

For more information about the Knowledge Script, see the *AppManager Knowledge Script Reference Guide*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

3 (Conditional) If the following registry keys on the agent computers are not set to an asterisk (*), use the NTAdmin_RegistrySet Knowledge Script to add the name of the new management server computer to the key values:

```
HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQmc\Security\AllowDosCmd
HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQmc\Security\AllowMS
HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQmc\Security\AllowReboot
```

Otherwise, certain actions will not be allowed after you move the management server. For example, the `AllowReboot` registry key will no longer allow Action_RebootSystem. For information about the parameters to specify in the Knowledge Script, see the *AppManager Knowledge Script Reference Guide*.

4 Install a new version 9.5 management server on the new computer.

For more information about installing a new version 9.5 management server, see Installing a Management Site in the *Installation Guide for AppManager*.

5 For each agent that communicates with the management server, run the AMAdmin_SetAllowMS Knowledge Script and update the **New hostname(s) for AllowMS** parameter with the name of the new management server computer.

For more information about the Knowledge Script, see the *AppManager Knowledge Script Reference Guide*.

6 For each agent that communicates with the management server, run the AMAdmin_SetPrimaryMS Knowledge Script to update the management server name.

Depending on whether the management server is primary or secondary for the agent, update the primary or secondary management server name.

For more information about the Knowledge Script, see the *AppManager Knowledge Script Reference Guide*.

7 Uninstall the management server from the old computer.

8 In the Operator Console tree view, select the old computer and press **Alt+F8**. Note the **ObjID** of the old management server computer.

9 In Microsoft SQL Server Management Studio, right-click the QDB with which the management server communicates and select **New Query**.

10 To change the status of the old management server computer to an agent computer so that you can remove it from the Operator Console, in the query window, type the following SQL statement and click **Execute**:

```
UPDATE dbo.Object
SET     Status = Status ^ 0x00000002
WHERE   ObjID = ObjID_of_old_management_server
        AND Status & 0x00000002 != 0
```

where *ObjID_of_old_management_server* is the **ObjID** you noted in Step 8 on page 25.

11 In the Operator Console, delete the old management server computer.

12 (Conditional) If the agent is still installed on the old computer, use Control Center to add the computer and rediscover it to establish a new **ObjID** for the computer.

13 (Conditional) If you edited registry keys in Step 3 on page 25, use the NTAdmin_RegistrySet Knowledge Script to remove the name of the old management server computer from the key values.

    For information about the parameters to specify in the Knowledge Script, see the *AppManager Knowledge Script Reference Guide*.

Depending on your environment, you might need to update existing encryption keys after you upgrade the management server. For more information about updating encryption keys, see Section 2.6.6, “Using Existing Security Keys for Encrypted Communications,” on page 26.

You can also change the security level after you upgrade. For more information about changing the security level, see Section 2.6.7, “Changing the Security Level,” on page 27.

2.6.6 Using Existing Security Keys for Encrypted Communications

If your AppManager environment uses encrypted communications between the management server and agents, the upgraded management server uses existing encryption keys to communicate with existing and upgraded agents. For an upgraded management server to use an existing encryption key to communicate with a new version 9.5 agent, you must use the `NQKeyGenWindows.exe` utility to export the key from the upgraded QDB and import it to the new agent. If the existing encryption key was generated using the NetIQ Encryption Utility, `rpckey.exe`, you must use the `NQKeyGenWindows.exe` utility to convert the older key file to the new key format before you import it to the new agent. The `NQKeyGenWindows.exe` utility is located in the `NetIQ\AppManager\bin` folder. For more information about the utility, see the *Administrator Guide for AppManager*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

To convert a key generated using the NetIQ Encryption Utility to the new key format:

1. To convert the older key file to the new key format, run the following command on the management server:

   ```
   NQKeyGenWindows -convert old_key_location new_key_location
   ```

2. To check the key information into the QDB, run the following command on the management server:

   ```
   NQKeyGenWindows -db QDB_name:user_name:SQL_Server_name\instance -change
   new_key_location
   ```

3. To set the desired security level, run the following command on the management server:

   ```
   NQKeyGenWindows -db QDB_name:user_name:SQL_Server_name\instance -seclev level
   ```

4. Restart the management server.

To import an existing encryption key to a new agent:

1. To extract the agent portion of the key from the QDB, run the following command on the management server:

   ```
   NQKeyGenWindows -db QDB_name:user_name:SQL_Server_name\instance -ckey
   agent_key_file_location
   ```

2. To import the key to the new agent, run the following command on the agent computer:

   ```
   NQKeyGenWindows -agentchange agent_key_file_location
   ```

2.6.7 Changing the Security Level

After you upgrade, if you want to change the security level for a management site from encrypted communications to cleartext communications, run the AMAdmin_AgentConfigSecurityLevel Knowledge Script.

To change the security level for a management site:

1. To change the security level for the agents within your management site, run the AMAdmin_AgentConfigSecurityLevel Knowledge Script.

   For more information about the Knowledge Script, see the *AppManager Knowledge Script Reference Guide*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

2. Run the AMAdmin_AgentConfigSecurityLevel Knowledge Script again on each management server.

3. (Conditional) If you have not configured the QDB to store the security key information on each management server computer, edit the following Microsoft Windows registry key:

   ```
   \HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQMS\Config\
   RPC Encryption
   ```

   and change its value from `1` to `0`. You must restart the management server to apply your changes. For more information about how to configure the QDB to store the security key information, see Section 2.6.6, “Using Existing Security Keys for Encrypted Communications,” on page 26.

   WARNING: Be careful when editing your Windows registry. If there is an error in your registry, your computer might become nonfunctional. If an error occurs, you can restore the registry to its state when you last successfully started your computer. For more information, see the Help for the Windows Registry Editor

4. (Conditional) If you used the `NQKeyGenWindows.exe` utility to store security information in the QDB, use that utility again and set the `-seclev` option to `0`.

5. Restart your management servers.

For more information about setting or changing the security level for an AppManager management site, see the *Administrator Guide for AppManager*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

3 Upgrading and Migrating Agent Components

This chapter describes how to upgrade Windows agent components and how to migrate a version 7.0.4 or 8.x, 9.1, 9.2 Windows or UNIX agent to a new version 9.5 AppManager repository (QDB). For information about upgrading the UNIX agent, see the *AppManager for UNIX and Linux Servers Management Guide*, available on the AppManager Modules Documentation page (https://www.netiq.com/documentation/appmanager-modules/).

For Windows computers, the agent consists of the following components:

- NetIQ AppManager Client Resource Monitor (`NetIQmc`) Windows service
- NetIQ AppManager Client Communication Manager (`NetIQccm`) Windows service
- Local repository
- AppManager for Microsoft Windows module

The setup program upgrades the agent services and local repository first, and then upgrades the AppManager for Microsoft Windows module.

3.1 Understanding Agent Versions Supported for Upgrade

You can upgrade the following versions of the Windows agent to version 9.5:

- 7.0.4
- 8.0.3
- 8.2
- 9.1
- 9.2

3.2 Understanding When to Upgrade

You do not have to upgrade all of your agents to version 9.5 at the same time. When you upgrade a management server, you also upgrade the agent on the management server computer. Otherwise, you can upgrade agents on an as-needed basis.

If you installed a new version 9.5 QDB and want to maintain existing agents, you must migrate the agents to the new QDB. For more information about migrating an existing agent to a new QDB, see Section 3.7, “Migrating an Agent to a New QDB,” on page 32.

3.3 Understanding the Order for Upgrading Components

Before you upgrade an agent, upgrade the QDB and management servers with which the agent communicates. While a version 9.5 management server can communicate with version 7.0.4 and 8.x, 9.1, 9.2 agents, version 9.5 agents cannot communicate with earlier management server versions. For more information about upgrading QDBs and management servers, see Chapter 2, “Upgrading Management Site Components,” on page 21.

3.4 Understanding Upgrade Methods

You can upgrade Windows agents by running the setup program locally on the agent computer, or you can use Control Center to upgrade agents on remote computers. For more information about using the setup program to upgrade agents, see Section 3.5, “Upgrading Agent Components on the Local Computer,” on page 30. For more information about using Control Center to upgrade agents, see Section 3.6, “Using Control Center to Upgrade Agents on Remote Computers,” on page 31.

The upgrade process does not change the settings from your previous agent installation and retains existing jobs, data, and events in the local repository.

3.5 Upgrading Agent Components on the Local Computer

This section describes the steps required to upgrade agent components on the local computer.

To upgrade agent components on the local computer:

1. Ensure that the upgrade of the QDB and management servers with which the agent communicates completed successfully.
2. Start the upgrade and generate a pre-installation check report.

   For more information about generating the report, see Section 1.9, “Starting an Upgrade and Generating a Pre-Installation Check Report,” on page 18.
3. Complete the agent setup program.

After you successfully upgrade agent components, to ensure that you have the latest AM Health Knowledge Scripts for health monitoring in Control Center, install the latest AppManager for Self Monitoring (AM Health) module on the primary QDB computer. For more information about upgrading the module, see Section 3.5.1, “Upgrading the AM Health Module,” on page 30. For more information about using Control Center to monitor the health of your AppManager components, see the *Control Center User Guide for AppManager*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

AppManager 9.2 includes an update to the OpenSSL library version and location change for the OpenSSL `.dll` files. For information about removing older versions of the `.dll` files from the agent computer, see “Removing Older Versions of OpenSSL Library .dll files” on page 31.

3.5.1 Upgrading the AM Health Module

After you successfully upgrade agent components, upgrade the AM Health module on the primary QDB computer and then run the Discovery_AMHealth Knowledge Script to prepare AppManager components for health monitoring in Control Center.

To upgrade the AM Health module and prepare components for health monitoring:

1. From the location where you extracted the installation package, navigate to the `Setup\Setup Files` folder and run `AM70-AMHealth-8.2.`*`x`*`.0.msi`.
2. Select **Install Knowledge Scripts** to install the repository components, including the Knowledge Scripts.
3. Specify the SQL Server name of the server hosting the primary QDB, as well as the case-sensitive QDB name.
4. After Control Center replicates the Knowledge Scripts to non-primary QDBs, run the Discovery_AMHealth Knowledge Script on all agent computers.

3.5.2 Removing Older Versions of OpenSSL Library .dll files

AppManager 9.2 includes an update of the OpenSSL library from version 1-0-1m to version 1-0-2j. In previous versions of AppManager, the agent installation program installed the OpenSSL `.dll` files (`libeay32.dll` and `ssleay32.dll`) in the `Program Files\NetIQ\AppManager\bin` directory on the agent computer. Starting with version 9.2, the management server installation program installs the files on the management server. Although the agent installation program no longer installs the `.dll` files, it does not remove older versions of the files. You must manually remove older versions of the `.dll` files from the agent computer.

3.6 Using Control Center to Upgrade Agents on Remote Computers

After you upgrade the QDB, management server, and Control Center components, you can use Control Center to upgrade agents on remote computers.

In Control Center, you will specify a user account to run the agent installation package. Ensure that the account has the required Group Policy object (GPO) setting. The account must be a member of the `Replace a process level token` policy, which determines the user accounts that can call the `CreateProcessAsUser()` application programming interface (API) so that one service can start another. By default, only local system accounts are members of the policy. You can edit the policy in the default domain controller GPO and in the local security policy of workstations and servers. The policy is located in the following path in the Microsoft Management Console:

```
Computer Configuration\Windows Settings\Security Settings\Local Policies\User
Rights Assignment
```

For more information about upgrading agents on remote Windows computers, see Deploying AppManager to Agent Computers in the *Control Center User Guide for AppManager*. For information about upgrading agents remotely on UNIX or Linux computers, see the *AppManager for UNIX and Linux Servers Management Guide*, available on the AppManager Modules Documentation page (https://www.netiq.com/documentation/appmanager-modules/).

3.7 Migrating an Agent to a New QDB

If you installed a new QDB and want to maintain existing agents for that QDB, you must migrate the agents to the new QDB. You can migrate version 7.0.4, 8.x, 9.1, 9.2 agents to new QDBs. AppManager 9.5 includes a command line tool, `MigrateQ.exe`, that automates the migration process and allows you to migrate multiple agents simultaneously, if needed. The tool migrates the following agent attributes to the new QDB:

- Security Manager settings
- Custom properties
- Running discovery jobs
- Running ad hoc jobs
- Running jobs that a custom Knowledge Script started

In addition, the tool sets the primary management server for the agent in the new QDB and performs the following clean-up activities in the source QDB following a successful migration:

- Stops all running jobs for the agent
- Optionally, deletes the agent

NOTE: The option to leave the source agent in the source QDB is not available when migrating a UNIX agent. You must delete the agent from the source QDB.

The tool does not migrate data and events to the new QDB.

The tool will not perform a migration in the following situations:

- The agent is offline.
- The agent is in maintenance mode.
- The agent is already present in the destination QDB.
- The destination QDB is an earlier version than the source QDB.
- You are using **Authentication and encrypted communications**, and the source QDB and the destination QDB have different encryption keys. The encryption keys must be identical for successful migration.

The `MigrateAgents.vbs` script that is included with `MigrateQ.exe` allows you to select multiple agents for migration based on server or management group membership or a custom property. Server groups must be part of the **Master** management group.

For example, you could create a management group called `Migration` and use the script to select all of the agents in the management group for migration, or you could define a custom property called `Migrate` and use the script to select all of the agent computers with the `Migrate` custom property defined.

To migrate an individual agent to a new QDB:

1. Ensure that the agent meets the version requirements for migration.
2. Ensure that the computer from which you will run the tool meets the following requirements:
   - The computer is running Microsoft .NET Framework 3.5 Service Pack 1
   - The computer can connect to both the source QDB and the destination QDB.

- The computer has a version 8.*x* or 9.*x* Windows agent installed. Even if you are migrating a version 7.0.4 agent, the computer from which you run the migration tool must meet this requirement.
- If you are migrating a UNIX agent, the computer can connect to the UNIX agent computer through Secure Shell (SSH).

3. (Conditional) If necessary, from the computer where you saved the AppManager 9.5 installation files, copy the contents of the `Agent Migration` folder to the computer where you will run the tool.
4. Open a command prompt and change directory to the location of the `Agent Migration` folder.
5. (Conditional) If you are migrating a Windows agent, type the following command, and then press **Enter**:

   ```
   MigrateQ -agent:agent_display_name -source:QDB_computer_name#QDB_name
   -dest:QDB_computer_name#QDB_name -
   MS:primary_management_server_name[,secondary_management_server_name]
   -sauth:{0|1}#[user_name$password] -dauth:{0|1}#[user_name$password] -
   delete:{0|1} -suffix:suffix
   ```

   Sample commands are included at the end of this section.
6. (Conditional) If you are migrating a UNIX agent, type the following command, and then press **Enter**:

   ```
   MigrateQ -agent:agent_display_name -source:QDB_computer_name#QDB_name
   -dest:QDB_computer_name#QDB_name -
   MS:primary_management_server_name[,secondary_management_server_name]
   -sauth:{0|1}#[user_name$password] -dauth:{0|1}#[user_name$password] -delete:1
   -uxuser:user_name -uxpassword:password -port:ssh_port_number
   ```

   Sample commands are included at the end of this section.

The command parameters are as follows:

Parameter	Description
`agent`	Name of the agent you want to migrate, as displayed in the AppManager consoles To specify multiple agents, use a comma to separate the names of the agents.
`source`	Name of the computer where the source QDB is installed and the name of the QDB, separated by `#`
`dest`	Name of the computer where the destination QDB is installed and the name of the QDB, separated by `#`
`MS`	Name of the primary management server for the agent, to be set in the destination QDB Optionally, you can specify a secondary management server for the agent. Use a comma to separate the primary and secondary management servers. For example, `-MS:MyPrimaryMS,MySecondaryMS`.
`sauth`	Whether the source QDB uses Windows or SQL Server authentication For Windows authentication, specify `0`. You do not need to provide credentials for Windows authentication. For SQL Server authentication, specify `1` and provide the credentials. For example, `-sauth:1#sa$Control123`.

Parameter	Description
dauth	Whether the destination QDB uses Windows or SQL Server authentication For Windows authentication, specify 0. You do not need to provide credentials for Windows authentication. For SQL Server authentication, specify 1 and provide the credentials. For example, -sauth:1#sa$Control123.
delete	Whether to delete the agent from the source QDB after successful migration To delete the agent, specify 1. To keep the agent, specify 0. **NOTE:** 0 is not available when migrating UNIX agents. You must specify 1 when migrating UNIX agents.
uxuser	User name for connecting to the remote UNIX agent Ensure that the user has sudo access.
uxpassword	Password for connecting to the remote UNIX agent
port	If the SSH server is not using port 22, SSH port number to use to connect to the remote UNIX agent
suffix	provide the suffix to be added with the current agent name.

Following are sample commands:

For this situation...	Use this command...
The source and destination QDBs use SQL Server authentication and you do not want to delete the agent from the source QDB **NOTE:** If you are migrating a UNIX agent, you must delete the agent from the source QDB.	MigrateQ -agent:MyAgent -source:MySourceQDB#QDB_1 -dest:MyDestQDB\MyDestInstance#QDB_2 -MS:MyManagementServer –sauth:1#sa$Control123 –dauth:1#sa$Control123 –delete:0
The source QDB uses SQL Server authentication, the destination QDB uses Windows authentication, and you want to delete the agent from the source QDB	MigrateQ -agent:MyAgent -source:MySourceQDB#QDB_1 -dest:MyDestQDB\MyDestInstance#QDB_2 -MS:MyManagementServer –sauth:1#sa$Control123 –dauth:0 –delete:1
The source and destination QDBs use Windows authentication and you want to delete the agent from the source QDB	MigrateQ -agent:MyAgent -source:MySourceQDB#QDB_1 -dest:MyDestQDB\MyDestInstance#QDB_2 -MS:MyManagementServerPrimary,MyManagementServerSecondary –sauth:0 –dauth:0 –delete:1
You want to migrate multiple agents in one session	MigrateQ -agent:MyAgent1,MyAgent2,MyAgent3 -source:MySourceQDB#QDB_1 -dest:MyDestQDB\MyDestInstance#QDB_2 -MS:MyManagementServer –sauth:0 –dauth:0 –delete:1

For this situation...	Use this command...

The value for a parameter contains special characters (enclose the parameter in quotes)	`MigrateQ -agent:MyAgent "-source:MySourceQDB#QDB123&%"`
	`-dest:MyDestQDB\MyDestInstance#QDB_2 -`
	`MS:MyManagementServer -sauth:1#sa$Control123 -`
	`dauth:1#sa$Control123 -delete:0`

You are migrating a UNIX agent	`MigrateQ -agent:MyAgent -source:MySourceQDB#QDB_1`
	`-dest:MyDestQDB\MyDestInstance#QDB_2 -`
	`MS:MyManagementServerPrimary,MyManagementServerSecondar`
	`y`
	`-sauth:1#sa$Control123 -dauth:1#sa$Control123 -delete:1`
	`-uxuser:sudouser -uxpassword:mypasswd -port:1234`

You want to migrate an agent with a suffix name	`MigrateQ -agent:MyAgent -source:MySourceQDB#QDB_1`
	`-dest:MyDestQDB\MyDestInstance#QDB_2 -`
	`MS:MyManagementServer –sauth:1#sa$Control123 –dauth:0 –`
	`delete:1 -suffix:dom.lab`

To select multiple agents for migration:

1. Ensure that the agents meet the version requirements for migration.
2. Ensure that the computer from which you will run the tool meets the following requirements:
   - The computer is running Microsoft .NET Framework 3.5 Service Pack 1.
   - The computer can connect to both the source QDB and the destination QDB.
   - The computer has a version 8.*x* or 9.*x* Windows agent installed. Even if you are migrating a version 7.0.4 agent, the computer from which you run the migration tool must meet this requirement.
   - If you are migrating a UNIX agent, the computer can connect to the UNIX agent computer through Secure Shell (SSH).
3. Ensure that you are using the `cscript` scripting host to run the script. From a command prompt, either:
   - Set `cscript` as the default. For example, `c:\> cscript /h:cscript`.
   - Explicitly run the script with `cscript`. For example, `c:\> cscript MigrateAgents.vbs`.
4. Ensure that `MigrateQ.exe` is in the same folder as `MigrateAgents.vbs`.
5. From a command prompt, change directory to the folder that contains `MigrateAgents.vbs`.
6. (Conditional) To select multiple Windows agents, type the following command, and then press **Enter**:

   ```
   c:\> cscript MigrateAgents.vbs /
   s_server:"Source_SQL_Server\SQL_Server_Instance" /s_db:"Source_QDB_Name" /
   s_uid:"Source_SQL_Server_Username" /s_pwd:"Source_SQL_Server_Password" /
   d_server:"Destination_SQL_Server\SQL_Server_Instance"
   d_db:"Destination_QDB_Name" /d_uid:"Destination_SQL_Server_Username" /
   d_pwd:"Destination_SQL_Server_Password" /ms:"Management_Server_For_Agent" /
   delete:"0|1" /sg:"Server_Group_Name" /mgname:"Management_Group_Name" /
   cp_name:"Custom_Property_Name" /cp_val:"Custom_Property_Value"
   ```

   Parameter descriptions are provided at the end of this section.

   The script creates a batch file, `RunMigration.bat`, in the current directory.
7. (Conditional) To select multiple UNIX agents, type the following command, and then press **Enter**:

```
c:\> cscript MigrateAgents.vbs /
s_server:"Source_SQL_Server\SQL_Server_Instance" /s_db:"Source_QDB_Name" /
s_uid:"Source_SQL_Server_Username" /s_pwd:"Source_SQL_Server_Password" /
uxuser:"UNIX_User_Name" /uxpassword:"UNIX_Password" /port:"Port_Number" /
d_server:"Destination_SQL_Server\SQL_Server_Instance"
d_db:"Destination_QDB_Name" /d_uid:"Destination_SQL_Server_Username" /
d_pwd:"Destination_SQL_Server_Password" /ms:"Management_Server_For_Agent" /
delete:"0|1" /sg:"Server_Group_Name" /mgname:"Management_Group_Name" /
cp_name:"Custom_Property_Name" /cp_val:"Custom_Property_Value"
```

Parameter descriptions are provided at the end of this section.

The script creates a batch file, `RunMigration.bat`, in the current directory.

8. Run the batch file, `RunMigration.bat`, to migrate the agents.

The command parameters are as follows:

Parameter	Description
`/s_server:`	Name of the SQL Server that hosts the QDB from which you want to migrate agents. If the SQL Server is on the computer where you are running the script, this parameter is not required.
`/s_db:`	Name of the QDB from which you want to migrate agents. If you do not specify this parameter, `QDB` is used.
`/s_uid:`	If the source SQL Server uses SQL authentication, the user name for connecting to the source SQL Server. This parameter is not required if the source SQL Server uses Windows authentication.
`/s_pwd:`	If the source SQL Server uses SQL authentication, the password for connecting to the source SQL Server. This parameter is not required if the source SQL Server uses Windows authentication.
`/d_server:`	Name of the SQL Server that hosts the QDB to which you want to migrate agents. If the SQL Server is on the computer where you are running the script, this parameter is not required.
`/d_db:`	Name of the QDB to which you want to migrate agents. If you do not specify this parameter, `QDB` is used.
`/d_uid:`	If the destination SQL Server uses SQL authentication, the user name for connecting to the destination SQL Server. This parameter is not required if the destination SQL Server uses Windows authentication.
`/d_pwd:`	If the destination SQL Server uses SQL authentication, the password for connecting to the destination SQL Server. This parameter is not required if the destination SQL Server uses Windows authentication.
`/ms:`	The primary and, optionally, secondary management server for the agent. To specify a secondary management server, use a comma to separate the primary and secondary management servers. For example, `/ms:"MyPrimaryMS,MySecondaryMS"`.
`delete`	Whether to delete the agent from the source QDB after successful migration To delete the agent, specify `1`. To keep the agent, specify `0`. **NOTE:** `0` is not available when migrating UNIX agents. You must specify `1` when migrating UNIX agents.

Parameter	Description
`/uxuser:`	User name for connecting to the UNIX agent. Ensure that the user has `sudo` access.
`/uxpassword:`	Password for connecting to the UNIX agent.
`/port:`	If the SSH server is not using port 22, SSH port number to use to connect to the UNIX agent.
`/sg:`	Name of the server group that contains the agents to migrate. The server group must be within the **Master** management group. When selecting the agents to migrate, any child server groups will be included.
`/mgname:`	Name of the management group that contains the agents to migrate.
`/cp_name:`	Name of the custom property that is assigned to the agents you want to migrate.
`/cp_value:`	Value of the custom property that is assigned to the agents you want to migrate.

3.7.1 MigrateQ Configuration

The MigrateQ.exe xml configuration file, which is included with MigrateQ tool, allows you to enable specific functionalities or to configure timeout values for the utility.

Note that you are allowed to modify MigrateQ.exe xml configuration file based on your environment requirement, otherwise you can continue with the default values.

Key	Description
`DiscoveryWaitTimeOut`	After the agent is added to the destination QDB, discovery job is run on the migrated agent to discover the modules associated with it. DiscoveryWaitTimeOut is the timeout value for the discovery job to complete. Default value is 300 sec. However, if your environment requires more time for the discovery to be successful, you can modify the value.
`AddAgentWaitTimeOut`	Migration utility waits for some time for the agent to be successfully added to the destination QDB. AddAgentWaitTimeOut is the timeout value for the add agent job to complete. Default value is 1000 sec. You can modify this value if you are migrating multiple agents at the same time.
`MigrateToLowQDB`	The agent migration is supported only from lower version of QDB to higher or same version of QDB. To allow migration from higher to lower version QDB, set the value of MigrateToLowQDB as 1. Default value is 0

Key	Description
`SaveJobstoXML`	During the migration of agents, all the running jobs associated with the agent is migrated to the destination QDB. In case any error occurs during migration, enabling SaveJobstoXML helps by saving the running job details to an XML file in the source machine. To save running jobs to XML file, set the value of SaveJobstoXML as 1. Default value is 0.
`CheckParentJobExist`	This flag only affects ad-hoc jobs. MigrateQ utility migrates agents and creates the agents' jobs one by one. While migrating ad-hoc jobs, instead of creating a new parent job for the already running KS, new job can be appended to the existing parent job ID based on KS name. This helps to avoid creating 'n' number of parent and child jobs for the same KS in the target QDB. (for example, if you are migrating 10 different agents, and if the source QDB had one parent job for same KS with 10 child jobs on 10 different agents, by default the target QDB ends up with 10 parents and one child job each. To retain the same one parent each with 10 child jobs, you can use this flag.) To append migrated job to the already existing parent job based on KS name, set the value of CheckParentJobExist as 1. Default value is 0.

4 Upgrading Control Center Components

This chapter describes how to upgrade Control Center components. Control Center consists of the following components:

- CCDB
- Control Center and Deployment services, which includes the following services:
  - Command queue service
  - Deployment Service
  - Deployment Web Service
- Control Center console

4.1 Understanding Upgrade Prerequisites

This section describes prerequisites your environment must meet before you upgrade Control Center components.

4.1.1 Understanding Control Center Versions Supported for Upgrade

Before you upgrade Control Center components, the components must be version 9.1 or 9.2. If a component does not meet the prerequisite, you can either uninstall it and install a new version 9.5 component, or you can first upgrade it to version 9.1 or 9.2.

For more information about installing new version 9.2 Control Center components, see Installing Control Center Components in the *Installation Guide for AppManager*.

For more information about obtaining a version of AppManager that is supported for upgrade, contact Technical Support (https://www.netiq.com/support/).

4.1.2 Understanding Microsoft SQL Server Versions Supported for Upgrade

Version 9.5 CCDBs must be hosted on one of the following versions of Microsoft SQL Server:

- Microsoft SQL Server 2017 Standard or Enterprise edition
- Microsoft SQL Server 2016 Standard or Enterprise edition
- Microsoft SQL Server 2014 Standard or Enterprise edition
- Microsoft SQL Server 2012 Standard or Enterprise edition (32-bit or 64-bit)
- Microsoft SQL Server 2008 R2 Standard or Enterprise edition (32-bit or 64-bit)
- Microsoft SQL Server 2008 Standard or Enterprise edition Service Pack 1 or later (32-bit or 64-bit)

For small environments with all components installed on the same computer, AppManager 9.5 also supports hosting CCDBs on SQL Server Express.

AppManager 9.5 supports migrating version 9.1, 9.2 CCDBs hosted on SQL Server 2008 R2 or later to SQL Server 2016. For more information about migrating an upgraded CCDB, see Appendix A, “Migrating Repositories,” on page 45.

4.2 Understanding QDB Requirements

A version 9.5 CCDB only supports a primary QDB of the same version. You can either create a new version 9.5 primary QDB or upgrade your existing version 9.1 or 9.2 primary QDB to version 9.5. If you create a new primary QDB, in order for the product to function correctly, you must add at least one agent computer to the QDB and discover the computer. For more information about creating a new version 9.5 primary QDB, see Installing a Management Site in the *Installation Guide for AppManager*. For more information about upgrading a QDB, see Chapter 2, “Upgrading Management Site Components,” on page 21.

You can attach version 9.1 or 9.2 QDBs to a version 9.5 CCDB as non-primary QDBs.

Before you upgrade the CCDB, ensure that it does not contain any QDBs that are earlier than version 9.1 or 9.2. If it does, either remove the QDBs or upgrade them to version 9.1 or 9.2. When you upgrade the CCDB to version 9.5, if the setup program detects a primary QDB that is not the same version or a non-primary QDB that is not version 9.1 or higher, the CCDB upgrade cannot continue.

4.3 Understanding Agent Requirements

Because version 9.5 Control Center and Deployment services require an agent, if the setup program detects that the services are available for upgrade but no agent is present, it automatically installs a version 9.5 agent after it upgrades the services.

4.4 Understanding the Order for Upgrading Control Center Components

If you have Control Center components installed on multiple computers, ensure that you upgrade components in the correct order. On the computer where you run the setup program, the program ensures you upgrade components in the correct order. However, when you have components installed on multiple computers, ensure that you upgrade the computers in order. For example, if you are upgrading the CCDB and the command queue service and they reside on different computers, run the setup program on the CCDB computer before you run it on the command queue service computer.

Upgrade Control Center components in the following order:

1. CCDB
2. Command queue service
3. Deployment Service
4. Deployment Web Service
5. Control Center console

If you do not upgrade components in the recommended order, the Deployment Service will not start after the upgrade.

4.5 Discovering Upgraded Control Center Components for Health Monitoring

Once the setup program successfully upgrades the command queue service, if an upgraded agent is already present, the setup program automatically runs the Discovery_AMHealth Knowledge Script to prepare Control Center components for health monitoring in Control Center. Otherwise, the setup program runs the Knowledge Script after the agent upgrade. For more information about using Control Center to monitor the health of your AppManager components, see the *Control Center User Guide for AppManager*, available on the AppManager Documentation page (http://www.netiq.com/documentation/appmanager).

4.6 Configuring Kerberos Delegation for a Distributed Control Center Environment

NetIQ Corporation recommends distributing Control Center components across computers to improve performance. If you plan to use Windows authentication to authenticate users between Control Center and the QDBs it manages in a distributed Control Center environment, configure Kerberos constrained delegation to ensure successful communication between Control Center components and QDBs. If Kerberos constrained delegation is not properly configured, connections between Control Center components and QDBs will fail with the following error:

```
Login failed for user 'NT AUTHORITY\ANONYMOUS LOGON'
```

To avoid this error, complete the following tasks:

- Prepare each QDB computer and the CCDB computer to authenticate using Kerberos.
- Configure the SQL Server service for each QDB computer and the CCDB computer to be trusted for delegation.
- Configure the CCDB computer to impersonate the SQL Server service for each QDB computer that connects to Control Center.

To prepare the QDB and CCDB computers to authenticate using Kerberos:

1. Set TCP/IP and Named Pipes as the preferred client protocols on the SQL Server and ensure that TCP/IP is listed first.
2. Determine the TCP dynamic port number the SQL Server service uses and verify that it is not blocked by a firewall.
3. Ensure that the SQL Server service is running under a domain account.
4. (Conditional) If you are running Microsoft Windows Server 2008, 2008 R2, or 2012, run the following commands to create the required service principal names:

   ```
   setspn -A MSSQLSvc/{fully-
   qualified_domain_name_of_the_QDB_or_CCDB_computer}:{SQL_Server_name\instance}
   {domain_account_name_under_which_the_SQL_Server_service_runs}
   ```

   ```
   setspn -A MSSQLSvc/{fully-
   qualified_domain_name_of_the_QDB_or_CCDB_computer}:{port_on_which_the_SQL_
   Server_service_runs}{domain_account_name_under_which_the_SQL_Server_service_ru
   ns}
   ```

```
setspn -A MSSQLSvc/
{NETBIOS_name_of_the_QDB_or_CCDB_computer}:{SQL_Server_name\instance}
{domain_account_name_under_which_the_SQL_Server_service_runs}
```

```
setspn -A MSSQLSvc/
{NETBIOS_name_of_the_QDB_or_CCDB_computer}:{port_on_which_the_SQL_
Server_service_runs}
{domain_account_name_under_which_the_SQL_Server_service_runs}
```

To configure the SQL Server service to be trusted for delegation:

1 On the domain controller, in Active Directory Users and Computers, right-click the domain account under which the SQL Server service runs and select **Properties**.

2 On the **Delegation** tab, select the following options:

- **Trust this user for delegation to specified services only**
- **Use Kerberos only**

3 Click **Add**.

4 Click **Users and Computers**.

5 Enter the name of the domain account under which the SQL Server service runs and click **OK**.

6 Select the `MSSQLSvc` entries associated with the QDB or CCDB computer and click **OK.**

7 (Conditional) If the SQL Server service will connect to Control Center across a firewall, run the following commands to register the required service principal names:

```
setspn -A MSSQLSvc/{fully-
qualified_domain_name_of_the_QDB_or_CCDB_computer}:{DNS_service_port}
{domain_account_name_under_which_the_SQL_Server_service_runs}
```

```
setspn -A MSSQLSvc/
{NETBIOS_name_of_the_QDB_or_CCDB_computer}:{DNS_service_port}
{domain_account_name_under_which_the_SQL_Server_service_runs}
```

```
setspn -A MSSQLSvc/{fully-
qualified_domain_name_of_the_QDB_or_CCDB_computer}:{kerberos_ticket_granting_
service_port} {domain_account_name_under_which_the_SQL_Server_service_runs}
```

```
setspn -A MSSQLSvc/
{NETBIOS_name_of_the_QDB_or_CCDB_computer}:{kerberos_ticket_granting_service_
port} {domain_account_name_under_which_the_SQL_Server_service_runs}
```

```
setspn -A MSSQLSvc/{fully-
qualified_domain_name_of_the_QDB_or_CCDB_computer}:{time_service_port}
{domain_account_name_under_which_the_SQL_Server_service_runs}
```

```
setspn -A MSSQLSvc/
{NETBIOS_name_of_the_QDB_or_CCDB_computer}:{time_service_port}
{domain_account_name_under_which_the_SQL_Server_service_runs}
```

8 Restart the SQL Server service on the QDB or CCDB computer.

To configure the CCDB computer to impersonate the SQL Server service for connected QDB computers:

1 In the Local Security Policy application of Administrative Tools, select **Local Policies** > **User Rights Assignment**.

2 Right-click **Impersonate a client after authentication** and select **Properties**.

3 Click **Add User or Group**.

4 For each QDB computer that connects to Control Center, enter the name of the domain account under which the SQL Server service runs and click **OK**.

To verify that components are using Kerberos delegation:

1 On the command queue service and QDB computers, run the following command:

```
osql -E -S {CCDB_SQL_Server_name\instance}
```

2 From the osql command prompt, run the following query:

```
select net_transport, auth_scheme from sys.dm_exec_connections where
session_id=@@spid
GO
```

The query should return the values `TCP` and `KERBEROS`.

3 On the command queue service and CCDB computers, run the following command:

```
osql -E -S {QDB_SQL_Server_name\instance}
```

4 Repeat Step 2 on page 43.

4.7 Upgrading the CCDB

This section describes the steps required to upgrade a CCDB.

You can upgrade CCDBs on remote SQL Servers. You do not have to run the setup program on the SQL Server.

On a SQL Server or instance hosting multiple CCDBs, you must upgrade each CCDB to version 9.5. If you have a version 9.1 or 9.2 CCDB on the same SQL Server or instance as a version 9.5 CCDB, the Control Center console for the version 9.1 or 9.2 CCDB might reflect the version 9.5 command queue service in the status pane.

To upgrade the CCDB:

1 Close connections to the CCDB.

For more information about the connections to close, see Section A.2, “Preparing to Migrate the Repositories,” on page 45.

2 Create a CCDB backup.

For more information about creating a CCDB backup, see Section A.2.1, “Creating Backup Copies of the Repositories,” on page 49. Since you are not migrating the CCDB to a different computer, it is not necessary to copy the backup files to a different location.

3 Ensure that your primary QDB meets requirements.

For more information about QDB requirements, see Section 4.2, “Understanding QDB Requirements,” on page 40.

4 Ensure that any existing components you plan to maintain in your version 9.5 Control Center environment meet requirements.

For more information about maintaining existing components, see Section 1.3.2, “Upgrading Components in a Multiple-QDB, Control Center Environment,” on page 12.

5 Ensure that there are no commands in the Control Center queue.

6 Start the upgrade and generate a pre-installation check report.

For more information about generating the report, see Section 1.9, “Starting an Upgrade and Generating a Pre-Installation Check Report,” on page 18.

7. When you reach the Target SQL Server and Repository Name window, provide the following information and click **Next**:
   - Name of the SQL Server and, if applicable, instance that hosts the CCDB you are upgrading. To specify a SQL Server instance, use the format *server_name\instance*.
   - Name of the CCDB you are upgrading.
   - Account that can log in to the SQL Server for the upgrade. Ensure that the account is a member of the `sysadmin` SQL Server role.

   The setup program prepares the existing CCDB data for upgrade.

4.8 Upgrading Control Center and Deployment Services

This section describes the steps required to upgrade the command queue service, Deployment Service, and Deployment Web Service.

To upgrade the Control Center and Deployment services:

1. Ensure that the upgrade of the CCDB with which the services communicate completed successfully.
2. Start the upgrade and generate a pre-installation check report.

   For more information about generating the report, see Section 1.9, “Starting an Upgrade and Generating a Pre-Installation Check Report,” on page 18.
3. Complete the Control Center setup program.

The setup program stops the command queue service and the Deployment Service to perform the upgrade.

4.9 Upgrading the Control Center Console

This section describes the steps required to upgrade the Control Center console.

To upgrade the Control Center console:

1. Ensure that the CCDB and Control Center and Deployment services upgrades completed successfully.
2. Start the upgrade and generate a pre-installation check report.

   For more information about generating the report, see Section 1.9, “Starting an Upgrade and Generating a Pre-Installation Check Report,” on page 18.
3. Complete the Control Center console setup program.

A Migrating Repositories

This appendix describes how to migrate a QDB or CCDB to a new SQL Server including migrating to a new computer.

A.1 Understanding the Migration

The recommended method for migrating a repository is to install a new, empty version repository on the new SQL Server, close connections to the repository you will migrate, create a backup copy, and restore the backup copy over the empty repository on the new SQL Server. After you restore the repository, additional tasks are necessary to ensure proper operation.

If you migrate a repository from a 32-bit computer to a 64-bit computer, you must disconnect the 32-bit repository.

The following checklist outlines the migration process and provides references to detailed information.

Step		Reference
□	1. Prepare for repository migration.	Section A.2, "Preparing to Migrate the Repositories," on page 45
□	2. Restore the backup copy over the new, empty repository on the new SQL Server.	Section A.3, "Restoring Repositories on the New SQL Server," on page 50
□	3. Configure the restored repository on the new SQL Server.	Section A.4, "Configuring Restored Repositories," on page 52
□	4. Update components and services that connect to the repository.	Section A.5, "Updating Connected Components and Services," on page 55
□	5. Verify that the migration was successful.	Section A.6, "Verifying Successful Migration," on page 59

A.2 Preparing to Migrate the Repositories

Before you migrate a repository to a new version of Microsoft SQL Server, install a new, empty version 9.5 repository on the new SQL Server, record information about the repository you will migrate, close connections to it, and create a backup copy.

To prepare for repository migration:

1 On the new SQL Server, install a new QDB or CCDB. During installation, when you specify the repository name, specify the same name as the repository you will migrate.

For more information about installing a new repository, see Installing a Management Site in the *Installation Guide for AppManager*.

2 (Conditional) If you are migrating a QDB and Control Center manages it, ensure that Control Center uses Windows authentication to connect to the QDB for the migration.

If Control Center currently uses SQL Server authentication to connect to the QDB, use Control Center to change the authentication method. You can change back to SQL Server authentication after the migration. For information about changing the authentication method Control Center uses to connect to a QDB, see Performing QDB Management Tasks in the *Control Center User Guide for AppManager*.

3 On each computer, to ensure that the Distributed Transaction Coordinator (DTC) security settings are the same, complete the following steps:

WARNING: If the settings are not the same, the migration will fail.

3a In the Component Services application in Administrative Tools, expand `Component Services\Computers\My Computer\Distributed Transaction Coordinator`.

3b Right-click **Local DTC** and select **Properties**.

3c On the **Security** tab, note the settings.

3d (Conditional) If the settings on the new computer do not match the settings on the old computer, adjust the settings on the new computer and restart it.

4 On the old SQL Server, to note the properties for SQL Server logins with access to the repository, complete the following steps:

4a In Microsoft SQL Server Management Studio, expand `*SQL_Server_Name*\Databases`.

4b Right-click the repository and select **New Query**.

4c In the query window, type the following command, and then click **Execute**:

```
SELECT  name,
        CASE WHEN type = 'S' THEN 'SQL'
             ELSE 'Windows'
        END AS 'Type'
FROM    sys.database_principals
WHERE   type IN ( 'S', 'U' )
        AND name != 'dbo'
        AND default_schema_name IS NOT NULL
        AND default_schema_name != 'guest'
ORDER BY name ASC
```

4d Expand `Databases\*Repository_Name*\Security\Users` and compare the accounts listed in the results table for the query to the accounts in the `Users` folder.

4e For each account that appears in both the results table and the `Users` folder, right-click the user in the `Users` folder and select **Properties**.

4f On the **General** page, note the **Login name** and **Database role membership**.

4g On the **Securables** page, note the **Explicit permissions**.

You will recreate the SQL Server logins after you restore the repository on the new SQL Server.

5 (Conditional) If you changed the schedule for any NetIQ SQL Server jobs, use the Task Scheduler Configuration Utility to view the job schedules and note the settings for each modified job:

5a In the utility, select the repository that contains the modified job.

5b In the job grid, select the job and then click **Change Schedule**.

5c Note the settings, and then click **Cancel**.

6 (Conditional) If you will migrate a CCDB that manages remote QDBs, on the SQL Server that hosts the CCDB you will migrate, complete the following steps to note the linked server properties for the remote QDBs:

6a In Microsoft SQL Server Management Studio, expand `Server Objects\Linked Servers`.

6b Right-click a linked QDB and select **Properties**.

6c On the **General** page, note the linked server name and server type.

6d On the **Security** page, note each local login defined and how Control Center makes the connection.

A login can:

- **Be made without using a security context.** If this option is selected, Control Center connects without using any login and password.
- **Be made using the login's current security context**. If this option is selected, Control Center uses the Log On As account for the SQL Server Agent service to log in to the remote QDB.
- **Be made using this security context**. If this option is selected, it implies you checked the **Use SQL Server authentication** option when you added the QDB to Control Center. When you restore the SQL Server link on the new CCDB computer, provide the same SQL Server user name and password you provided when you added the QDB to Control Center.

6e On the **Server Options** page, note the **RPC** and **RPC Out** values.

You will restore the SQL Server links after you restore the CCDB on the new computer.

7 On each computer, to ensure that the SQL Server collation order, sort order, and character set are the same, complete the following steps:

WARNING: If the settings are not the same, the migration will fail.

7a In Microsoft SQL Server Management Studio, right-click the SQL Server instance and select **Properties**.

7b On the **General** page, note the **Server Collation** setting, and then click **OK** to close the Properties window.

7c Right-click the SQL Server instance and select **New Query**.

7d In the query window, type the following command, and then click **Execute**:

```
sp_helpsort
```

The sort order and character set is displayed in the results table.

When you install SQL Server, the collation order is set by default according to the locale of the operating system. You can use advanced installation options to change the collation order. If the collation order is not the same, re-install SQL Server on the new computer and set the collation order to be the same as the collation order on the old computer.

8 (Conditional) If you will migrate the QDB, complete the following steps to close connected services:

8a Click **Start** > **Administrative Tools** > **Services**.

8b For each of the following services, right-click the service and select **Stop**:

- On the SQL Server that hosts the QDB, SQL Server Agent service
- On primary and secondary management servers that connect to the QDB, NetIQ AppManager Management Service

- (Conditional) If you are running NetIQ Advanced Analytics, on the computer where you installed the services, NetIQ Advanced Analytics Configuration Service and NetIQ Advanced Analytics Service
- On primary and secondary management servers that connect to the QDB, NetIQ AppManager Client Communication Manager and NetIQ AppManager Client Resource Monitor services
- On the computer where you installed the Task Scheduler service, NetIQ AppManager Task Scheduler Service
- (Conditional) If Control Center manages the QDB, on the command queue service computer, NetIQ AppManager Control Center Command Queue Service
- (Conditional) If Control Center manages the QDB, on the SQL Server that hosts the CCDB, SQL Server Agent service
- (Conditional) If report agents connect to the QDB, on the agent computers, NetIQ AppManager Client Communication Manager and NetIQ AppManager Client Resource Monitor services

If a service is set to automatically restart when it stops, disable the service.

9 (Conditional) If you will migrate the CCDB, complete the following steps to close connected services:

9a Click **Start** > **Administrative Tools** > **Services**.

9b For each of the following services, right-click the service and select **Stop**:

- On the computer where you installed the Task Scheduler service, NetIQ AppManager Task Scheduler Service
- On the command queue service computer, NetIQ AppManager Control Center Command Queue Service
- On the SQL Server that hosts the CCDB, SQL Server Agent service
- On the Deployment Service computer, NetIQ AppManager Deployment Service
- On the Deployment Web Service computer, World Wide Web Publishing Service that manages the Deployment Web Service and the Web Depot virtual directories

10 (Conditional) If you will migrate the QDB, stop AppManager Connectors that connect directly to it, such as the AppManager Connector for Micromuse Netcool/OMNIbus or AppManager Connector for Security Manager.

11 Close any AppManager consoles, such as the Control Center console and Operator Console, that connect to the repository.

12 (Conditional) If you will migrate the QDB and it is a data source for Analysis Center, complete the following steps on the Data Mart computer to stop the Analysis Center ETL job:

12a In Microsoft SQL Server Management Studio, expand `*SQL_Server_Name*\SQL Server Agent\Jobs`.

12b Navigate to the ETL job.

12c Right-click the job and select **Disable**.

13 To verify that there are no open connections to the repository you will migrate, complete the following steps:

13a On the repository computer, in Microsoft SQL Server Management Studio, expand *SQL_Server_Name*`\Databases`.

13b Right-click the repository and select **New Query.**

13c In the query window, type the following command, and then click **Execute**:

```
USE master
GO
Exec sp_who2
GO
```

13d In the results table, check the **DBName** column for the repository name. The column should not contain entries for the repository.

14 (Conditional) If the **DBName** column contains entries for the repository, complete the following steps for each entry:

14a In the **SPID** column for the row in which the repository name appears, note the SPID number.

14b Right-click the repository and select **New Query**.

14c In the query window, type the following command, and then click **Execute**:

```
kill SPID_Number
```

15 Repeat Step 13 on page 48 and Step 14 on page 49 until the **DBName** column does not contain entries for the repository.

16 Create a backup copy of the repository you will migrate.

For more information about creating backup copies, see Section A.2.1, “Creating Backup Copies of the Repositories,” on page 49.

After you create a backup copy of the repository you will migrate, you can restore the backup copy over the new, empty repository. For more information about restoring the repository, see Section A.3, “Restoring Repositories on the New SQL Server,” on page 50.

A.2.1 Creating Backup Copies of the Repositories

This section describes how to use Microsoft SQL Server Management Studio to create backup copies of the QDB and CCDB before you migrate them to the new computer.

Prior to create a backup copy of the QDB or CCDB, make sure to:

- Remove the QDB or CCDB from the old Task Scheduler, if already added.
- Manually remove the QDB or CCDB from the old Task Scheduler, if the repository is off-line.

To remove a repository from the service:

1. In the **Tasks** pane, click **Remove**.
2. Select the repository you want to remove, and then click **OK**.

To create a backup copy of the QDB or CCDB:

1. In Microsoft SQL Server Management Studio, expand `SQL_Server_Name\Databases`.
2. Right-click the repository and select **Tasks** > **Back Up**.
3. On the **General** page, complete the following steps:

3a From the **Database** list, select the repository.

3b Note the setting in the **Recovery model** field.

3c From the **Backup type** list, select **Full**.

3d For **Backup component**, select the **Database** radio button.

4 On the Options page, complete the following steps:

4a Select the **Back up to the existing media set** radio button.

4b (Conditional) If you want to add this backup to existing backups, select the **Append to the existing backup set** radio button.

4c (Conditional) If you want to discard existing backups, select the **Overwrite all existing backup sets** radio button.

5 Click **OK** to start the backup.

6 (Conditional) If the Recovery model is set to FULL on the General page, after the backup completes, complete the following steps to add a backup device for the transaction log and to back up the transaction log:

6a Right-click the repository and select **New Query**.

6b In the query window, type the following command, and then click **Execute**:

```
USE master
EXEC sp_addumpdevice 'disk', 'dump_device_log',
'C: \repository_nameBACKUP\repository_name_Log.bak'
GO
BACKUP LOG repository_name TO dump_device_log
GO
sp_dropdevice 'dump_device_log'
GO
```

where *repository_name* is the name of the QDB or CCDB you are backing up and *dump_device_log* is the name of the backup device or file

7 Copy the backup file to the computer where you will restore the QDB or CCDB.

After you copy the backup file to the new computer, you can restore the backup copy over the new, empty repository on the new computer. For more information about restoring the repository, see Section A.3, “Restoring Repositories on the New SQL Server,” on page 50.

A.3 Restoring Repositories on the New SQL Server

After installing a new version 9.5 repository on the new SQL Server and creating a backup copy of the repository you will migrate, use Microsoft SQL Server Management Studio to restore the backup copy over the new, empty repository on the new SQL Server.

If you are migrating both the QDB and the CCDB, restore the QDB first.

To restore repositories on the new computer:

1 On the new computer, stop the SQL Server Agent service.

2 To ensure that the repository you are restoring is not the default database for the account you are using to perform the restore, complete the following steps:

2a In Microsoft SQL Server Management Studio, expand `SQL_Server_name\Security\Logins`.

2b Right-click the account you are using and select **Properties**.

2c On the General page, note the selection in the **Default database** list.

3 (Conditional) If the default database for the account you are using is the repository you are restoring, either change the default database for the account, or log in to SQL Server Management Studio with a different account.

4 To ensure that no users are connected to the new repository, complete the following steps:

4a Expand `SQL_Server_name\Databases`.

4b Right-click the repository and select **New Query.**

4c In the query window, type the following command, and then click **Execute**:

```
USE master
GO
Exec sp_who2
GO
```

4d In the results table, check the **DBName** column for the repository name. The column should not contain entries for the repository.

5 (Conditional) If the **DBName** column contains entries for the repository, to close the open connections, complete the following steps for each connection:

5a In the **SPID** column for the row in which the repository name appears, note the SPID number.

5b Right-click the repository and select **New Query**.

5c In the query window, type the following command, and then click **Execute**:

```
kill SPID_number
```

6 Repeat Step 4 on page 51 and Step 5 on page 51 until the **DBName** column does not contain entries for the repository.

7 Right-click the repository and select **Tasks** > **Restore** > **Database**.

8 On the **General** page, complete the following steps:

8a Select the **From device** radio button and click the button to specify the backup device.

8b On the Specify Backup window, select **File** from the **Backup media** list, and then click **Add**.

8c On the Locate Backup File window, browse to the location where you saved the backup copy, select the backup file, and then click **OK**.

8d Click **OK** to return to the **General** page.

8e Select the backup set to restore.

9 On the **Options** page, complete the following steps:

9a Under **Restore options**, select the **Overwrite the existing database (WITH REPLACE)** check box.

9b Under **Recovery state**, select the **RESTORE WITH RECOVERY** radio button.

9c Click **OK** to restore the repository.

10 After the restore completes, restart the SQL Server Agent service.

After you restore the repository, additional configuration is required. For more information about the configuration tasks, see Section A.4, “Configuring Restored Repositories,” on page 52.

A.4 Configuring Restored Repositories

After you restore the repository on the new SQL Server, additional configuration is required to ensure proper operation.

To configure the restored repository:

1 To verify that the compatibility level of the restored repository is set to the appropriate version of SQL Server, complete the following steps in Microsoft SQL Server Management Studio:

1a Expand `SQL_Server_name\Databases`.

1b Right-click the restored repository and select **Properties**.

1c On the **Options** page, ensure that the compatibility level is set to SQL Server 2016 (130).

2 To identify the SQL Server user accounts you must recreate for the repository, complete the following steps:

2a Expand `SQL_Server_name\Databases`.

2b Right-click the repository and select **New Query**.

2c In the query window, type the following command, and then click **Execute**:

```
SELECT  name,
        CASE WHEN type = 'S' THEN 'SQL'
             ELSE 'Windows'
        END AS 'Type'
FROM    sys.database_principals
WHERE   type IN ( 'S', 'U' )
        AND name != 'dbo'
        AND default_schema_name IS NOT NULL
        AND default_schema_name != 'guest'
ORDER BY name ASC
```

2d Expand `repository_name\Security\Users` and compare the accounts listed in the results table for the query to the accounts in the `Users` folder. Note the accounts that appear in both locations.

3 To recreate the SQL Server logins for the repository, complete the following steps for each account you noted in Step 2 on page 52:

3a Right-click the repository and select **New Query**.

3b In the query window, type the following command, and then click **Execute**:

```
sp_dropuser 'user_name'
```

3c Expand `SQL_Server_name\Security\Logins`.

3d With the exception of the `probe` account, for each account you removed in Step 3b on page 52, right-click **Logins** and select **New Login**.

3e Configure the login with the properties you noted in Step 4 on page 46 of Section A.2, "Preparing to Migrate the Repositories," on page 45.

3f (Conditional) If a repository account you removed already exists as a SQL Server account, use AppManager Security Manager (for the QDB) or the Control Center console (for the CCDB) to assign the accounts to the repository.

4 To verify the repository owner, complete the following steps:

4a Right-click the restored repository and select **New Query**.

4b In the query window, type the following command, and then click **Execute**:

```
sp_helpdb 'repository_name'
```

5 (Conditional) If the repository owner is not correct, to change the owner, type the following command, and then click **Execute**:

```
sp_changedbowner 'repository_owner'
```

6 (Conditional) If you restored the repository on a different computer or with a different name, use the Task Scheduler Configuration Utility to remove the old repository from the Task Scheduler Service and add the restored repository to the service. For information about removing and adding repositories, see Configuring the Task Scheduler Service and SQL Server Jobs in the *Administrator Guide for AppManager*.

7 (Conditional) If you are migrating the QDB, complete the following steps to update the `ComponentCurrentVersion` and `DataSource` tables with the new SQL Server information:

7a Right-click the QDB and select **New Query**.

7b In the query window, to obtain the `idDataSource` value for the QDB you are migrating, type the following command, and then click **Execute**:

```
select idDataSource from dbo.DataSource
where DataSourceName = 'old_SQL_Server_name\instance:old_QDB_name'
```

7c In the results table, note the `idDataSource` value. You will need this value for Step 7e on page 53.

7d Right-click the QDB and select **New Query**.

7e In the query window, type the following commands, and then click **Execute**:

```
update dbo.ComponentCurrentVersion
set MachineName = 'New_SQL_Server_name\instance'
where ComponentName = 'NetIQ AppManager Repository'
update dbo.DataSource
set DataSourceName = 'New_SQL_Server_name\instance:restored_QDB_name',
ServerName = 'new_SQL_Server_name\instance',
DatabaseName = 'restored_QDB_name'
where idDataSource = 'idDataSource_value_from_Step 7c on page 53'
```

8 (Conditional) If you are migrating the QDB and restored it on a different computer, complete the following steps to update the `Version` table with the new computer name:

8a Right-click the QDB and select **New Query**.

8b In the query window, type the following command, and then click **Execute**:

```
update dbo.Version
set MachineName = 'new_computer_name'
where Component = 'Repository'
```

9 (Conditional) If you are migrating the QDB and previously created charts for use in the Chart Console, complete the following steps to enable the charts for use with the restored QDB:

9a Right-click the QDB and select **New Query**.

9b In the query window, type the following command, and then click **Execute**:

```
update dbo.blob set comment = replace(comment,
'_old_SQL_Server_name\instance\', '_new_SQL_Server_name\instance\')
from dbo.blob
where charindex('_old_SQL_Server_name\instance\', comment) > 0
```

10 (Conditional) If you are migrating the CCDB and restored it on a different computer, complete the following steps to update the `Version` table with the new computer name:

10a Right-click the restored CCDB and select **New Query**.

10b In the query window, type the following command, and then click **Execute**:

```
update dbo.Version
set MachineName = 'new_computer_name'
where Component = 'CCDB'
```

11 (Conditional) If you are migrating the CCDB and it manages remote QDBs, complete the following steps to restore SQL Server links to the remote QDBs:

11a Expand *`SQL_Server_name`*`\Server Objects`.

11b Right-click the `Linked Servers` folder and select **New Linked Server**.

11c On the **General** page, in the **Linked server** field, specify the name and instance, if applicable, of the SQL Server that hosts the QDB for which you are restoring the link.

11d On the **General** page, for **Server type**, select the **SQL Server** radio button.

11e On the **Security** page, to add local logins defined before you migrated the CCDB, click **Add**.

11f On the **Server Options** page, set the **RPC** and **RPC Out** values to **True**.

11g Click **OK** to restore the SQL Server link.

After you configure the restored repository, update components and services that connect to it. For more information about the components and services to update, see Section A.5, "Updating Connected Components and Services," on page 55.

A.5 Updating Connected Components and Services

After you configure the restored repository, update components and services that connect to it.

To update components and services:

1 (Conditional) If you are migrating the QDB, complete the following steps to update the primary management server and each secondary management server that connects to it:

1a (Conditional) If you customized any management server port or persistent IOC settings, use the Windows Registry Editor to back up the following registry keys on the management server computer:

On this type of operating system...	Back up...
32-bit	◆ `HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQms\NetIQmc Port` ◆ `HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQms\Port` ◆ `HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQms\Unix Port` ◆ `HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQms\Config\Persistent IOC` ◆ `HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQms\Config\PIOC Map File Path`
64-bit	◆ `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NetIQ\AppManager\4.0\NetIQms\NetIQmc Port` ◆ `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NetIQ\AppManager\4.0\NetIQms\Port` ◆ `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NetIQ\AppManager\4.0\NetIQms\Unix Port` ◆ `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NetIQ\AppManager\4.0\NetIQms\Config\Persistent IOC` ◆ `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NetIQ\AppManager\4.0\NetIQms\Config\PIOC Map File Path`

1b Restore persistent IOC settings.

For more information about restoring persistent IOC settings, see Section A.5.1, “Restoring Persistent IOC Settings,” on page 57.

1c (Conditional) If you have UNIX agents, restore the port setting that defines where the management server listens for communications from UNIX agents.

For more information about restoring the port setting, see Section A.5.2, “Restoring the UNIX Port Setting,” on page 58.

1d Restore any registry keys you backed up.

1e Start the NetIQ AppManager Management Server service (`NetIQms`).

2 (Conditional) If you are migrating the CCDB, on each QDB Control Center manages, complete the following steps to enable the QDB to connect to the restored CCDB:

2a In Microsoft SQL Server Management Studio, right-click the QDB and select **New Query**.

2b In the query window, type the following command, and then click **Execute**:

```
UPDATE dbo.CC_CacheManager SET Name = 'new_CCDB_SQL_Server_name\instance'
WHERE Name = 'old_CCDB_SQL_Server_name\instance'
```

3 (Conditional) If you are migrating the QDB and Control Center manages it, complete the following steps to update the QDB connection information in the CCDB:

3a (Conditional) If Control Center uses SQL authentication to communicate with the QDB, configure the QDB with the same SQL Server user account and permissions.

3b Log on to the Control Center console with an account that is a member of the Administrator group and has the `db_owner` database role for the QDB.

3c On the **Global Tasks** tab of the ribbon, click **Manage Repositories.**

3d Select the QDB, and then click **Modify**.

3e Provide the following information, and then click **OK**:

- Name of the SQL Server and instance, if applicable, that hosts the QDB
- Name of the QDB
- Whether to use Windows or SQL Server authentication
- (Conditional) If you select SQL Server authentication, SQL Server account information

3f Click **Close** to close the Manage Repositories window.

4 (Conditional) If you are migrating the CCDB, complete the following steps to update the command queue service:

4a Use the Control Center console to add the Windows user account for the command queue service as a Control Center administrator.

4b In Microsoft SQL Server Management Studio, right-click the CCDB and select **New Query**.

4c In the query window, to clear the previous command queue service settings from the CCDB, type the following command, and then click **Execute**:

```
delete from Property where Scope = 'cqs'
```

4d On the command queue service computer, from the `AppManager\Control Center\bin folder`, open the `NQCQS.exe.config` file in a text editor.

4e Under `<appSettings>`, change the value of the `ServerName` parameter to specify the SQL Server and instance that hosts the restored CCDB, and change the value of the `DBName` parameter to specify the restored CCDB name. For example:

```
<appSettings>
  <add key="ServerName" value="MYSQLSERVER\INSTANCE1" />
  <add key="DBName" value="CCDB_name" />
```

4f Restart the command queue service to apply the changes.

5 (Conditional) If you are migrating the CCDB, complete the following steps to update the Deployment Service:

5a (Conditional) If the Deployment Service will use different credentials or a different account to log on to the migrated CCDB, from the `AppManager\Control Center\bin` folder on the Deployment Service computer, issue the following command to change the account:

```
deploymentservice -setwindowsauth domain\user_name password
```

5b Use the Control Center console to add the Windows user account for the Deployment Service as a Control Center administrator.

5c On the Deployment Service computer, from the `AppManager\Control Center\bin folder`, open the `DeploymentService.exe.config` file in a text editor.

5d Under `<appSettings>`, change the value of the `ServerName` parameter to specify the SQL Server and instance that hosts the restored CCDB, and change the value of the `DBName` parameter to specify the restored CCDB name. For example:

```
<appSettings>
  <add key="ServerName" value="MYSQLSERVER\INSTANCE1" />
  <add key="DBName" value="CCDB_name" />
```

5e Restart the Deployment Service to apply the change.

6 (Conditional) If you are migrating the CCDB, complete the following steps to update the Deployment Web Service:

6a Use the Control Center console to add the Windows user account for the Deployment Web Service as a Control Center administrator.

6b On the Deployment Web Service computer, from the `AppManager\Control Center\web` folder, open the `Web.config` file in a text editor.

6c Under `<appSettings>`, change the value of the `ServerName` parameter to specify the SQL Server and instance that hosts the restored CCDB, and change the value of the `DBName` parameter to specify the restored CCDB name. For example:

```
<appSettings>
    <add key="ServerName" value="MYSQLSERVER\INSTANCE1">
<add key="DBName" value="CCDB_name" />
```

6d Restart the World Wide Web Publishing Service to apply the change.

After you update the connected components and services, verify successful migration. For more information about verifying successful migration, see Section A.6, “Verifying Successful Migration,” on page 59.

A.5.1 Restoring Persistent IOC Settings

Re-registering the management server service disables persistent IOC settings in the registry. This section describes how to restore the settings.

WARNING: Be careful when editing your Windows registry. If there is an error in your registry, your computer might become nonfunctional. If an error occurs, you can restore the registry to its state when you last successfully started your computer. For more information, see the Help for the Windows Registry Editor.

To restore persistent IOC settings:

1 Click **Start** > **Run**.

2 In the **Open** field, type `regedit`, and then click **OK**.

3 (Conditional) If the management server is installed on a 32-bit operating system, in the left pane of the Registry Editor, navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQms\Config`.

4 (Conditional) If the management server is installed on a 64-bit operating system, in the left pane of the Registry Editor, navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NetIQ\AppManager\4.0\NetIQms\Config`.

5 In the right pane, double-click **Persistent IOC**.

6 In the **Value data** field, set the value to **1**, and then click **OK**.

7 In the right pane of the Registry Editor, double-click **PIOC Map File Path**.

8 In the **Value data** field, set the value to the location of your persistent IOC files, and then click **OK**.

Typically, the location is `Program Files\NetIQ\AppManager\dat\pioc`.

After you restore the persistent IOC settings, return to Step 1 on page 55 of Section A.5, "Updating Connected Components and Services," on page 55.

A.5.2 Restoring the UNIX Port Setting

Re-registering the management server service resets the port setting that defines where the management server listens for communications from UNIX agents. This section describes how to restore the setting.

WARNING: Be careful when editing your Windows registry. If there is an error in your registry, your computer might become nonfunctional. If an error occurs, you can restore the registry to its state when you last successfully started your computer. For more information, see the Help for the Windows Registry Editor.

To restore the UNIX port setting:

1 Click **Start** > **Run**.

2 In the **Open** field, type `regedit`, and then click **OK**.

3 (Conditional) If the management server is installed on a 32-bit operating system, in the left pane of the Registry Editor, navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\4.0\NetIQms`.

4 (Conditional) If the management server is installed on a 64-bit operating system, in the left pane of the Registry Editor, navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NetIQ\AppManager\4.0\NetIQms`.

5 In the right pane, double-click **Unix Port**.

6 For **Base**, select **Decimal**.

7 In the **Value Data** field, set the value to the port number you specified when you installed AppManager, and then click **OK**.

The default port where the management server listens for communications from UNIX agents is 9001.

After you restore the UNIX port setting, return to Step 1 on page 55 of Section A.5, "Updating Connected Components and Services," on page 55.

A.6 Verifying Successful Migration

After you update components and services that connect to the migrated repository, verify successful migration.

To verify successful repository migration:

1. (Conditional) If you migrated a QDB, on the primary management server and each secondary management server that connects to the QDB, ensure that the following services are running:
   - NetIQ AppManager Management Service (`NetIQms`)
   - NetIQ AppManager Client Resource Monitor (`NetIQmc`)
   - NetIQ AppManager Client Communication Manager (`NetIQccm`)
2. (Conditional) If you migrated a CCDB, on the command queue service and Deployment Service computers, ensure that the following services are running:
   - NetIQ AppManager Control Center Command Queue Service
   - NetIQ AppManager Deployment Service
3. Log on to the Operator Console and verify that the primary management server appears in the tree view and is not disabled.
4. (Conditional) If the primary management server does not appear in the tree view, re-register the management server.

   For more information about re-registering the management server, see Step 1 on page 55 of Section A.5, “Updating Connected Components and Services,” on page 55.
5. Use the Operator Console and the Control Center console to start some jobs.

   For example, start an NT_Discovery job. When that job completes, start an NT_CpuLoaded job and wait for it to complete.
6. (Conditional) If the jobs do not complete successfully, contact Technical Support (https://www.netiq.com/support/).