Upgrade and Migration Guide

NetIQ® AppManager®

July 2007

Legal Notice

Part Number: 10517-701

Contents

About This Guide

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

Intended Audience

This guide, the AppManager *Upgrade and Migration Guide*, is intended for organizations that have a previous version of AppManager currently installed in their environment and want to upgrade to the latest version. It includes tips and recommendations for ensuring a smooth upgrade of all your 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 are interested in performing a fresh installation instead of an upgrade, see the AppManager *Installation Guide*.

This guide focuses specifically on how to upgrade your 6.0.2 or later AppManager environment to version 7.0.

What's Changed?

AppManager version 7.0 includes significant changes to the installation process. For example, you now install agents on remote computers using a Control Center feature, Remote Deployment. For more information about what's new, see the Release Notes in the AppManager installation kit.

The following installation and upgrade-related Knowledge Scripts for Microsoft Windows agents are obsolete as of version 7.0 of AppManager:

- AMAdmin_AddManagedObject
- AMAdmin_AddManagedObjectProxy
- AMAdmin_AgentInstall
- AMAdmin_AgentInstallProxy
- AMAdmin_AgentPreInstall
- AMAdmin_AgentPreInstallProxy
- AMAdmin_AppManagerUninstall
- AMAdmin_AppManagerUninstallProxy
- AMAdmin_MaintenanceInstall
- AMAdmin_MaintenanceInstallWS.NET
- AMAdmin_RepositoryCleanUp
- AMAdmin_RestartPolicyJobs

These Knowledge Scripts will only run with older versions of AppManager agents and monitoring modules. For AppManager 7.0 and higher, use the Remote Deployment feature of Control Center to install, uninstall, and upgrade.

No changes have been made to the following UNIX installation Knowledge Scripts:

- AMAdmin_UnixAgentInstallProxy
- AMAdminUNIX_AgentInstall
- AMAdminUNIX_AgentUpdate

These Knowledge Scripts will still work with current AppManager components.

For more information about changes in AppManager version 7.0, see "Understanding Changes to the Upgrade Workflow" on page 19.

Conventions

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

Convention	Use
Bold	• Window and menu items • Technical terms, when introduced
Italics	• Book and installation kit titles • Variable names and values • Emphasized words
`Fixed Font`	• File and folder names • Commands and code examples • Text you must type • Text (output) displayed in the command-line interface

Using Online Help

AppManager provides task-based, reference, and context-sensitive online Help.

To access task-based Help or search for Help topics, click **Help Topics** on the Help menu. To view context-sensitive Help in dialog boxes, click **Help** or press **F1**.

You can get help on individual Knowledge Scripts in one of the following ways:

- On the **Values** tab of the Knowledge Script Properties dialog box, click **Help** or press **F1**.
- In the Knowledge Script pane of the Operator Console, highlight a Knowledge Script and press **F1**.
- In the Knowledge Script view of the Control Center Console, double-click a Knowledge Script and press **F1**.

Other Information in the Library

The library provides the following information resources:

- *Installation Guide*: Provides complete information about AppManager pre-installation requirements and step-by-step installation procedures for all AppManager components.
- *Control Center User Guide*: Provides complete information about managing groups of computers, including running jobs, responding to events, creating reports, and working with the Control Center Console. A separate guide is available for the AppManager Operator Console.
- *Administrator Guide*: Provides information about maintaining an AppManager management site, managing security, using scripts to handle AppManager tasks, and leveraging advanced configuration options.
- *Management Guides*: Provide information about installing and monitoring specific applications with AppManager.

The AppManager library is available in Adobe Acrobat (PDF) format and is located in the `\Documentation` folder of the AppManager installation kit.

NetIQ Online Support and Extended Support Web sites provide other resources:

- Downloads, including hotfixes, service packs, and product upgrades.
- Documentation, including white papers and the most current information about version support for the systems and applications monitored by AppManager.

Note You can access NetIQ Support without a password or registration. To access the Extended Support site, you must be a registered AppManager customer.

In addition to the AppManager documentation, you may want to consult the documentation for your Microsoft Windows or UNIX operating system, or other application- or system-specific documentation for reference and conceptual information. This background information can help you get the most out of your AppManager installation.

About Attachmate

Attachmate, owned by an investment group led by Francisco Partners, Golden Gate Capital and Thomas Cressey Equity Partners, enables IT organizations to extend mission critical services and assure they are managed, secure and compliant. Attachmate's leading solutions include host connectivity, systems and security management, and PC lifecycle management. Our goal is to empower IT organizations to deliver trusted applications, manage service levels, and ensure compliance by leveraging knowledge, automation and secured connectivity. For more information, visit www.attachmate.com.

NetIQ Solutions from Attachmate

Attachmate provides a wide selection of systems and security management solutions to help you manage and secure all your essential platforms, including Microsoft Windows, UNIX, Linux, and iSeries. These Knowledge-Based Service Assurance products and solutions include embedded knowledge and tools to implement industry best practices and to better ensure operational integrity, manage service levels and risk, and ensure policy compliance. Our modular, best-of-breed solutions for Performance and Availability Management, Security Management, Configuration and Vulnerability Management, and Operational Change Control integrate through an open, service-oriented architecture allowing for common reporting, analytics and dashboards. Attachmate offers the following systems and security management solutions:

- **Performance and Availability Management** These products offer rapid time-to-value solutions that enable you to align your IT operations with business priorities and optimize the delivery of your IT-based business services. This solution automates the complete IT service management lifecycle: assessment of requirements,

definition of Service Level Agreements, management of day-to-day operations, and review of operational metrics.

- **Security Management** These easy-to-install-and-deploy products provide effective protection from and response to security-related threats. This solution provides powerful features, such as real-time security event monitoring, mapping of threat indicators, policy violation alerts, and expedited incident forensics and resolution. These products reduce the time required to identify and resolve security threats.
- **Configuration and Vulnerability Management** These products allow you to quickly and easily assess vulnerabilities, manage security risks, and assure policy compliance. This powerful solution measures and enforces compliance to configuration baselines based on your corporate policies, regulations, and evolving security threats. You can use the latest security knowledge, which is updated in real time, to resolve compliance and configuration issues.
- **Operational Change Control** These products enable IT organizations to control, manage, and audit operational changes to servers, Active Directory, and Group Policy with unprecedented levels of accountability. NetIQ's Operational Change Control (OCC) solutions enable enterprise customers to meet IT compliance and operational integrity needs in the most cost-effective manner, by delegating access control, managing changes according to policy, and alerting and reporting on change activities and entitlements.

Contacting NetIQ Solutions Support

Please contact us with your questions and comments. We look forward to hearing from you.

Sales Email:	info@netiq.com
Telephone:	1-713-418-5555 (United States) +353 (0) 91-782-677 (Europe, Middle East, and Africa) For other locations, see our Support Contact Information Web site at www.netiq.com/support
Support Web Site:	www.netiq.com/support

Chapter 1

Upgrade Checklist

This chapter provides a reference checklist for performing an upgrade. Depending on how your AppManager components are distributed, several steps may be combined. For example, if your repository and management server are on the same computer, you do not have to stop the `NetIQ Management Service`. Both components are upgraded in a single step instead of two steps.

The following checklists summarize the steps and list the page number where each step is discussed.

Preparing to Upgrade		Page
❑	Identify AppManager components to upgrade and associated information such as versions and accounts.	27
❑	Verify systems requirements for all components you will upgrade.	27
❑	If you have Microsoft Windows agents older than version 5.0.1, upgrade the agents to version 5.0.1 before you upgrade to AppManager 7.0.	27
❑	If you have UNIX agents older than version 6.0.2, upgrade the agents to version 6.0.2 before you upgrade to AppManager 7.0.	27
❑	Set upgrade order and schedule. Plan a schedule for upgrading components and notify users or departments, as necessary, to minimize the impact on your production environment.	32
❑	Back up all AppManager data. If you are using SQL Server mixed-mode security, make a list of the SQL Server and AppManager users used in your repository.	34

Preparing to Upgrade	Page
❑ Upgrade or delete all components, jobs, and Knowledge Scripts that are older than the upgrade supports. The setup program generates a pre-installation check report to help you identify the components you must upgrade or delete.	35
❑ Check out all Knowledge Scripts that are no longer supported.	35

Upgrading	Page
❑ Run the AM`setup.exe` program on the repository server computers to upgrade all components on that computer.	35
❑ Run the AM`setup.exe` setup program on all management server computers that you have not yet upgraded.	44
❑ Run the AM`setup.exe` setup program on all Web management server computers that you have not yet upgraded.	45
❑ Run the AM`setup.exe` setup program on all Control Center and Control Center Console computers that you have not yet upgraded.	47
❑ Run the AM`setup.exe` setup program on all Operator Console computers that you have not yet upgraded.	48
❑ Upgrade all managed clients that you have not yet upgraded.	49
Upgrade all jobs.	57

Chapter 2

Preparing to Upgrade

This chapter describes the steps to take before you upgrade your existing AppManager components. Upgrading AppManager is a straightforward process. However, proper preparation ensures a smooth and successful upgrade with minimal impact on your production environment.

The following topics are covered:

Understanding Changes to the Upgrade Workflow

AppManager version 7.0 provides a set of major enhancements to the installation mechanism for AppManager and Control Center components and agents. Upgrade is easier and quicker, and you will also find it easier to install AppManager components in the future, when a powerful Remote Deployment system will help you install hotfixes, service packs, and module upgrades anywhere in your network.

With past releases of AppManager, you ran one of the AMAdmin Knowledge Scripts to upgrade or install an agent or monitoring module on a remote computer. To run any of these installation or upgrade Knowledge Scripts, you had to perform several tasks that you no longer need to perform:

- Install at least one agent with the "remote agent installation capability" enabled.
- Ensure that Port 9979 is open on the remote installation-enabled agent in order to use the agent as a "proxy" to install components across a firewall.
- Establish a distribution computer.
- Copy the AppManager agent or module update files to that network share.

None of these tasks is required to remotely deploy managed clients on Microsoft Windows computers. Control Center now uses a Web service and Microsoft IIS to allow for communications between Control Center components, agents, and target computers across firewalls. If you choose not to use Remote Deployment, you can install agent and module upgrades and hotfixes by running the corresponding setup programs locally. The AppManager *Installation Guide* contains a full set of instructions for both installation methods.

Remote Agent Installation No Longer Supported

The AppManager agent version 7.0 can no longer be installed with remote agent installation capability.

The AMAdmin_AgentInstall or AMAdmin_AgentInstallProxy Knowledge Scripts can only perform installation of older agents and monitoring modules. To remotely install agents and modules, you must use Control Center.

If you decide to keep a backlevel agent with remote installation capability installed in your AppManager

management site, you can still use the AMAdmin Knowledge Scripts mentioned above, but only to install backlevel agents and modules.

Understanding Changes to Requirements and Supported Applications

AppManager 7.0 removes support for AppManager components in some configurations and no longer supports earlier versions of the AppManager Microsoft Windows agent.

In addition, monitoring support for some applications has been removed while some applications are supported through an independent software vendor (ISV).

Changes to System Requirements for AppManager Core Components

AppManager 7.0 removes support for AppManager repository database, management server, console applications, Web management server, and the AppManager report agent running on **Microsoft Windows NT 4.0**.

All components now require Microsoft Windows Installer version 3.1 or later.

Check the tables in the "System Requirement Checklists" chapter of the AppManager *Installation Guide* before you upgrade.

AppManager 4.3 and 5.0 No Longer Supported

AppManager 7.0 removes support for AppManager 4.3 and 5.0 agents, jobs, and Knowledge Scripts. Before you upgrade the repository to AppManager 7.0, you must upgrade your version 4.3 and 5.0 Microsoft Windows agents, jobs, and Knowledge Scripts, or delete them. See "Upgrading Older Agents and Jobs" on page 35 for more information.

Changes in UNIX Application Support

Monitoring support for some cross-platform, for example UNIX and Linux, applications is no longer provided with AppManager 7.0 and is not supported with the AppManager 7.0 UNIX agent.

If you are currently monitoring any of the following applications, you can continue to monitor these applications with your existing UNIX agent:

- IBM DB2
- Lotus Domino
- Tivoli Storage Manager

Changes in AppManager Connector Support

The following connectors are no longer provided with AppManager and are not supported with AppManager 7.0:

- Remedy AR System
- Connector for Micromuse Netcool/OMNIbus, versions earlier than 7.0

Changes in Microsoft Management Console Support

The AppManager Microsoft Management Console (MMC) snap-in is no longer supported in AppManager version 7.0. Once you have upgraded to AppManager version 7.0, remove the AppManager MMC snap-in by deleting the `APPMANAGER.msc` file.

Changes in Microsoft Windows Application Support

Monitoring support for the Microsoft Windows applications listed below is no longer provided with AppManager 7.0 and is not supported with the AppManager 7.0 agent for Microsoft Windows.

If you are currently monitoring any of the following applications, you can only continue to monitor these applications with agents older then version 7.0:

- BEA WebLogic Server
- IBM WebSphere Application Server
- IBM WebSphere MQ (MQSeries)
- Legato NetWorker
- SAP
- Sendmail
- ResponseTime for Lotus Domino

ISV Support for Some Microsoft Windows Applications

Support for monitoring the following applications is now available from AK Computer Services Ltd.As a UK-based technology partner of NetIQ, AK Computer Services has been granted a worldwide license to provide ongoing development and support for the following AppManager modules:

- Microsoft Application Center Server 2000
- Microsoft BizTalk Server 2000 and 2002
- Microsoft Commerce Server 2000
- Microsoft Internet Security and Acceleration Server
- Microsoft Message Queue Server
- Microsoft Network Load Balancing
- Microsoft SQL Server 2000 Analysis Services
- Microsoft Systems Management Server
- Microsoft Transaction Server
- Sybari Antigen for Microsoft Exchange
- Trend Micro ScanMail for Microsoft Exchange

These modules are now branded as *Gestio Modules for AppManager*, and will be sold and supported by AK Computer Services and its authorized channel partners. Information about the Gestio modules, product roadmap and partners is available at **www.gestio.co.uk**.

If you are currently monitoring any of the applications listed above, you can continue to do so with a version 5.0.1 or 6.x agent and corresponding module. If you want to monitor the above applications with a version 7.0 AppManager Microsoft Windows agent, you will need to upgrade to the appropriate Gestio module available from AK Computer Services. Monitoring support for these applications is no longer provided directly by NetIQ.

The following modules, which were created and made available since the 6.0.2 release of AppManager, are also supported by a third-party vendor and not by NetIQ:

- Web services running on Microsoft .NET servers
- Web services running on the J2EE platform on UNIX operating systems

These modules are supported by Infravio. For more information, see www.infravio.com/products/management_console.html.

Planning Upgrade Order and Schedule

If your AppManager components are installed on multiple computers, upgrade the components in the following order:

1. AppManager repositories
2. Management servers
3. Web management servers
4. Control Center

5. Console programs
6. Report agents
7. Managed computers

Although AppManager version 7.0 supports AppManager components from earlier versions, all AppManager components on the same computer must be the same version. Upgrade all of your managed clients as soon as possible.

Although the upgrade is not significantly disruptive to users, schedule the component upgrades and notify the appropriate personnel to minimize any impact on your operation.

Determining Which Components to Upgrade

Identify the location and version of all of your AppManager components.

To upgrade to AppManager 7.0, your AppManager components must be at version 6.0 or higher. If you have components earlier than version 6.0, you can either:

- Upgrade to version 6.0, and then upgrade to version 7.0.
- Uninstall older components before installing AppManager version 7.0.

Any agent on the same computer as another AppManager component, such as the repository or management server will be upgraded at the same time. Upgrading the AppManager agent on managed clients is optional unless the report agent is enabled, in which case you must upgrade the agent. Upgrade all components to receive the benefit of new features.

Management servers with versions older than AppManager 6.0.2 do not function with the version 7.0 repository. Once you stop them to upgrade the repository, you cannot restart them until they have also been upgraded.

Control Center supports older versions of the AppManager repository. However, not all features are supported. See "Avoiding Common Problems" on page 71 in the "Troubleshooting" chapter for more information.

Some system requirements have changed since the previous release of AppManager. For more information about AppManager system requirements, see the Release Notes in the AppManager 7.0 installation kit. For information about what has changed, see "Understanding Changes to Requirements and Supported Applications" on page 21.

Knowledge Script Considerations

Knowledge Scripts with a unique name, such as those you created, are migrated without change. If you modified a NetIQ Knowledge Script and saved it with the same name, change the name now or you will lose your modifications when it is updated. Always save modified scripts with a different name.

If you customized any Knowledge Script parameters, these parameters are automatically migrated as you set them, even if there is a new version of the Knowledge Script. If the new version has a new parameter, the default value is used just for the new parameter.

The upgrade provides a report from which you can determine what script code you modified, so you can decide if you need to make the same changes in your new 7.0 Knowledge Scripts.

When you run the setup program, a pre-installation check report is automatically created to provide a list of Knowledge Scripts that must be updated or deleted. For more information, see "Upgrading Older Agents and Jobs" on page 35.

Before you start the upgrade, ensure you have access to the AppManager documentation set. For information about what is new, see the Release Notes in the AppManager installation kit.

Identifying AppManager Component Information

Before you begin to upgrade AppManager, make a list of your AppManager components and the computers they are installed on. Include the following components in your list:

- Control Center
- Repository
- Operator Console
- Chart Console
- Security Manager
- Developer's Utilities
- Management servers
- Web management servers
- Report agents

Include the physical location for each computer. For example, you might create a worksheet similar to this one:

Computer	AppManager Component	Build version	Location
Dynamo	AppManager repository (QDB) and management server	6.0.2.107	Raleigh, NC
Rainbow	AppManager repository (QDB-LA)	6.0.2.123	Los Angeles, CA
Hollywood	Management server	6.0.2.123	Los Angeles, CA
Venice	AppManager Operator Console or Control Center Console	6.0.2.123	Los Angeles, CA
Sunset	AppManager Operator Console and other console programs	6.0.2.123	Los Angeles, CA
Austin	AppManager repository (QDB-TX) and management server	6.0.2.156	Austin, TX
Amarillo	AppManager Operator Console and other console programs	6.0.2.156	Austin, TX

Seeing this information in a worksheet can help you plan the upgrade process in an organized way, so the components for any given management site are upgraded at the same time.

Run the AppManager Component Version and AppManager Component License reports to help collect information about the components installed on all computers in your environment.

There are several ways to get the version information for console programs and management servers. For example, you can find version information using the Registry Editor or from the Microsoft Windows Explorer.

To find the version number of the AppManager management server in Microsoft Windows Explorer:

1 Use Microsoft Windows Explorer to locate the `NetIQms.exe` file. By default, this file is in `Program Files\NetIQ\AppManager\bin.`

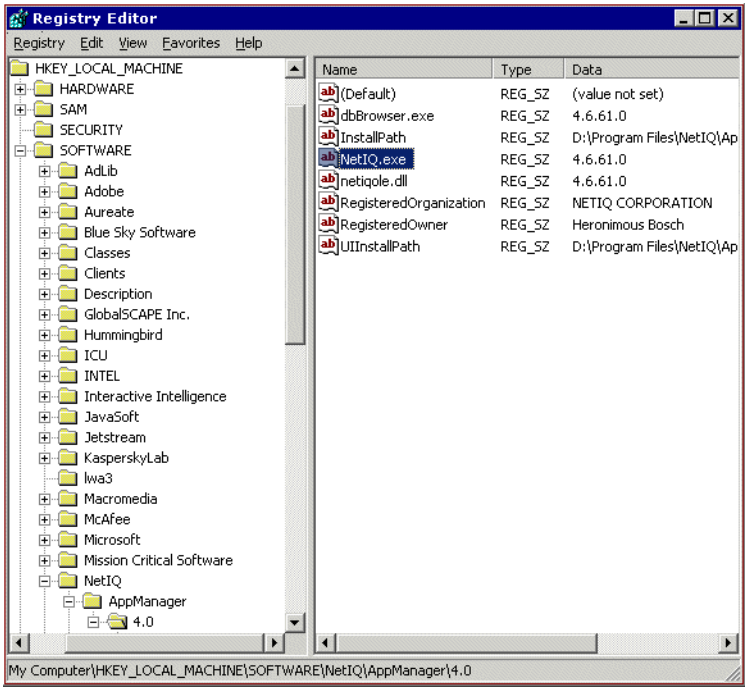
2 Right-click `NetIQms.exe` and select **Properties.**

3 Select the **Version** tab.

This number is not necessarily the same as the NetIQ AppManager Suite version number. The following table relates the AppManager Suite version numbers to the first two digits of component build numbers.

AppManager Suite Version	Agent & Script Build Numbers
5.0.1	4.6....
6.0	6.0...
6.0.2	6.0.2...
7.0	7.0

You can also look in the registry for build numbers. The following figure shows the build number for the Operator Console.



The following table shows how to find AppManager component build number information. .

Component	Steps to Take
AppManager Operator Console	Start the Operator Console, then click **Help > About AppManager Operator Console**. Use Regedit as shown in the example, above.
AppManager repository	In the Repository Browser, select the Version table from the Table list. After the information appears, edit the query statement, and execute the following: `select * from Version where Component = 'Repository'`
AppManager management server	In the Operator Console, right-click on the server Click **Troubleshooter > Management Service Info > Connectivity**. It displays the build number and other information. Use Regedit, or, in the Operator Console, click **Extensions > NetIQCtrl**. From this command line program, enter: `ping <server> netiqms` where: `<server>` is the name of the computer where the management server is installed.
AppManager agent	In the Operator Console, right-click on the computer whose agent you want to check. Click **Troubleshooter > Client Resource Monitor Info > Connectivity**. It displays the version number and other information.
AppManager managed object	In the Operator Console TreeView pane, right-click on the managed computer, then click **Properties**.
Control Center components	Use Regedit to view the following key: `HKEY_LOCAL_MACHINE\SOFTWARE\NetIQ\AppManager\Control Center\1.0` There you will find build versions for the following components: Control Center Console, repository, auto deployment, and Command Queue Service.

Note that in some cases values may be referred to as version numbers when they actually indicate build numbers. Version numbers with more than three decimal attachments (x.x.x.x.x) are build numbers.

Time Considerations

Schedule plenty of time for the repository upgrade. **The upgrade may take a long time**. If you have a very large repository, the upgrade may take more than a couple of hours. As a general guideline, allow at least an hour for every 10 GB.

During the upgrade, the Microsoft Windows Task Manager may indicate that the AppManager setup program is not responding. As long as Microsoft SQL Server is consuming a lot of CPU resources, this indicates that Microsoft SQL Server is upgrading the repository database normally. Do not interrupt the upgrade process.

Encryption Considerations

If you are currently using encryption for agent-management server communications, which is medium security, after you upgrade to AppManager 7.0:

- Existing agents continue to communicate using encryption.
- You can import your existing security key into the repository database and deploy new AppManager 7.0 Microsoft Windows agents using encrypted communication.

For information about encrypting and authenticating communication between the AppManager 7.0 Microsoft Windows and UNIX agents and the management server, see the *Administrator Guide*.

Chapter 3

Upgrading the AppManager Repository Database

This chapter describes how to upgrade the AppManager repository, the first AppManager component you should upgrade. The following topics are covered:

Control Center Support

Because Control Center must communicate with AppManager repositories, you need to consider a few other factors when deciding when and what to upgrade.

Control Center supports AppManager repositories from AppManager version 6.0.2 and later. However, the primary repository must be of an AppManager version greater than or equal to the version of the rest of the AppManager repositories that you are managing with Control Center. If you do not want to upgrade your present primary repository to AppManager version 7.0, you should plan to change the primary repository designation to another repository of version 7.0 after you complete the Control Center and AppManager upgrades.

Preparing for a Repository Upgrade

The following topics offer a series of steps that you should take before you start an AppManager repository upgrade.

Backing Up the AppManager Repository

Back up your current AppManager repository and the NetIQ program directory on each computer on which you have installed an AppManager component. Once you are sure the upgrade has completed successfully, you no longer need the previous version's files, and you may delete them.

For more information about backing up the repository, see the AppManager *Administrator Guide*.

Listing Microsoft SQL Server and AppManager Users

If you are using Microsoft SQL Server mixed security, make a list of all the Microsoft SQL Server and AppManager users. Make sure you record the `netiq` account passwords. This is just a precaution in case a problem develops with the association between the accounts listed in the AppManager repository and the user accounts in Microsoft SQL Server.

Viewing Pre-Installation Check Information

The upgrade process automatically runs a pre-installation check and generates a report that provides detailed information about old agents, jobs, and Knowledge Scripts that must be upgraded or deleted. See "Identifying Old Agents, Jobs, and Knowledge Scripts" on page 40 for a sample report.

For information about the pre-installation check program, see the *Installation Guide*.

Upgrading Older Agents and Jobs

AppManager 7.0 no longer supports version 4.3 or version 5.0 agents, jobs, and Knowledge Scripts. AppManager version 6.0.2 is still supported. Before you can upgrade the repository to AppManager 7.0, you must upgrade or delete the old agents and jobs.

Run the AppManager 7.0 setup program to generate a pre-installation check report that lists version 4.3 agents and jobs that must be upgraded or deleted.

Note The upgrade process automatically deletes old Knowledge Scripts and Knowledge Script Group members.

Upgrading to AppManager 6.0

You must upgrade to AppManager 6.0 before you attempt to upgrade to the current version of AppManager. For information about how to upgrade to AppManager 6.0, see the AppManager version 6.0 *Upgrade and Migration Guide*.

Backing up Your Knowledge Scripts

Depending on how your organization goes about modifying Knowledge Scripts, it's a good practice to check your custom and modified Knowledge Scripts out of the AppManager repository and copy them to a temporary location.

The upgrade program compares the Knowledge Scripts in the repository to the Knowledge Scripts in your installation directory. If you made changes to a script in the installation directory without changing the Knowledge Script name, the upgrade cannot identify the changes because the versions that are being compared are the same. If you do not know how or where customizations were made, it's best to check the Knowledge Scripts out before the upgrade. You can choose the "Upgrade and preserve data" option when you begin the upgrade to save the entire contents of the repository before

upgrading, including management data and Knowledge Scripts. See "Running the Setup Program" on page 37 for more information.

Performing a formal checkout or checkout/check-in operation ensures that your modifications are saved to the AppManager repository and saved on the local disk before an upgrade. Like making a backup of the repository and program files, this is a precautionary step to ensure you can restore customized scripts in the event you run into problems in the upgrade or scripts have been changed through the Operator Console but the changes have not been saved to the local disk copy.

The upgrade process prompts you to automatically delete Knowledge Scripts and Knowledge Script Group members that predate AppManager version 5.0.1 from the repository database. Before upgrading the repository, perform a pre-installation check to see if you have any older Knowledge Scripts that should be updated or backed up.

If you have changed and saved your changes in the local copy of a Knowledge Script file, for example, directly in the file `\netiq\appmanager\qdb\kp\general\General_AsciiLog.qml` rather than by double-clicking and modifying the Ascii Log Properties within the Operator Console, the checkout operation is not required. Your customized copy of the script is saved in the `\netiq\qdb_old\kp_old` folder after the upgrade.

Closing All Connections to the Repository

Before you upgrade, you should close all connections to the AppManager repository database and to the Control Center repository. This task may include connections from the following sources:

- Console applications, such as the Operator Console or Control Center console.
- Management servers.

- Web management servers.
- Any application or component making RPC calls or ODBC calls to the repository.

If you do not close all connections to the database, the installation program displays a list of connections that must be closed.

To ensure all connections are closed:

1. Stop all AppManager console programs.
2. ***If the management server is on a different computer than the repository***, stop the `NetIQ AppManager Management Service`. If you have multiple management servers, be sure to stop the `NetIQ AppManager Management Service` on all of them.
3. Disconnect any additional connections to the repository outside of AppManager. For example, if `isql` or the Query Analyzer is connected to the repository, shut it down. If any current connections are to the master database, stop them for the duration of the upgrade.

Note If the setup program detects any open connections to the repository, it asks you to close each of them before continuing.

Running the Setup Program

Run the AppManager setup program on the computer where the AppManager repository is currently installed.

Running the setup program to upgrade is similar to installing AppManager for the first time. However, the upgrade program asks whether you want to preserve or discard the data that is already stored in your repository.

- **Upgrade and preserve data**: The repository is upgraded to version 7.0, and all existing management data contained in the backlevel repository is preserved and transferred to the new repository.
- **Upgrade and discard data**: The repository is upgraded to version 7.0, and any existing management data contained in the backlevel repository is deleted. The new repository is empty—with the exception of Knowledge Scripts—once the upgrade is complete. Any parameter setting you had made in your Knowledge Scripts are lost unless you backed up your Knowledge Scripts ahead of time. See "Backing up Your Knowledge Scripts" on page 35

A repository upgrade takes a lot longer to complete if you choose to preserve existing management data. The more data you have in the backlevel repository, the longer it can take. However, preserving your management data is recommended in order to maintain consistent monitoring and historical data.

You do not have to select components to upgrade. The upgrade program identifies all AppManager components on that computer.

If you have any questions about specific steps in the upgrade procedure, click a **Help** button or see the *Installation Guide*. This section provides more information about specific steps in the upgrade procedure for an AppManager repository.

Specifying the Name of the Repository to Upgrade

The installation program prompts you for the name of the Microsoft SQL Server computer, the Microsoft SQL Server instance, and the repository database to upgrade. By default, the repository name is selected for you. The setup program allows you to select another Microsoft SQL Server instance and repository and the type of authentication being used by this instance.

The installer must be able to log in to the SQL Server to upgrade the AppManager repository, so you must choose the type of login authentication that the installer should use to access the SQL Server. If you select SQL Authentication, supply a valid SQL Server user name and password.

Select either Microsoft Windows Authentication or Microsoft SQL Server Authentication and click **Next**.

If you also have repositories on this computer that you do not want to upgrade, they will be preserved unless you select them here.

Any other AppManager components that are installed on this computer must also be upgraded.

The Netiq User Password

When you installed the repository that is now being upgraded, you were required to supply a unique password for the "`netiq`" user account. This Microsoft SQL Server login account, created by the AppManager setup program, acts as repository owner (`db_owner`) and is assigned the name `netiq`. During the installation, you supplied a password for this account.

Supply that same password here.

For more information about the `netiq` user account, and for a full explanation of the repository installation procedure, see the "Installing the Repository" chapter of the AppManager *Installation Guide*.

Unless you have backlevel agents and jobs in your repository from significantly older AppManager installations (pre-dating version 5.0.1), the repository upgrade can now proceed without further input from you. See the following section for information about any system requirements check failure notifications you might see at this point.

Depending on whether you chose to “Upgrade and preserve” or “Upgrade and discard” the data that’s already in the repository, the repository upgrade can take some time to complete. Discarding existing management data takes less time than upgrading the repository while preserving that data. But we recommend that you preserve your management data to maintain continuity in historical for trend analyses and save yourself the time and effort associated with re-creating the jobs you had previously configured.

Identifying Old Agents, Jobs, and Knowledge Scripts

If your repository contains agents, jobs, Knowledge Scripts, or Knowledge Script Group members older than version 4.3, the installation program may tell you these items cannot be upgraded.

If this happens, you cannot upgrade the repository until you manually update or delete the old agents and jobs.

For more information about upgrading old agents, jobs, and Knowledge Scripts, see “Upgrading Older Agents and Jobs” on page 35.

Checking the Upgrade Log Files

When you upgrade the repository, several log files track the progress of the upgrade. You can find these log files in the `\NetIQ\AppManager\Temp\NetIQ_Debug\<computer>` folder.

You can review these files to verify the completion of a successful upgrade. For more information about log files, see Chapter 7, “Troubleshooting an Upgrade.”

If your AppManager components are installed on separate computers, verify that the repository upgrade is successful before upgrading any other components. Then immediately back up the upgraded repository. This backup should be separate from the backup you made before upgrading. When you are sure the repository upgrade has succeeded, upgrade the management server; see "Upgrading Management Servers" on page 44 for more information.

Restarting Connections to the Repository

After the repository upgrade completes, you can restore any connections to the repository database. However, you must upgrade the other AppManager components before they can communicate with the new repository.

Chapter 4

Upgrading Management Servers, Control Center, and Consoles

After you have upgraded the AppManager repository database, you are ready to upgrade the management server, console programs, and Web management server. This chapter describes how to upgrade these components. The following topics are covered:

Preparing to Upgrade Components

The AppManager upgrade program will upgrade all AppManager components on the computer where you run the program. Upgrade component computers in the following order:

1. Repositories computers
2. Management servers
3. Web management servers
4. Control Center computers
5. Console computers
6. Report agents computers

7 Managed clients

To complete a successful upgrade, you must upgrade the repository, management servers, Operator Console, Control Center, and Web management server. All AppManager components on the same computer must be of the same version.

Upgrading Management Servers

If your AppManager environment is configured to use a single management server, run the AppManager setup program to upgrade the management server.

If you have configured multiple management server support for failover or static load-balancing, run the setup program to upgrade each management server. After you upgrade, configure the management servers in your environment so that a single management server performs agent installation-related tasks.

When you are ready to upgrade, run the setup program as described in the *Installation Guide*. During the upgrade process, additional agent-related upgrade prompts are displayed. For more information, see Chapter 5, "Upgrading Managed Clients."

When the setup program prompts you to select the components to install, select the management server and the management agent. The AppManager agent is always installed on your management server and must be upgraded when you upgrade the management server.

After you complete the setup, the `NetIQ Management Service` will be stopped and restarted.

Upgrading Web Management Servers

Exit any Operator Web Consoles that are currently connected to the AppManager Web management server. The setup program will ask you whether it's okay to stop the IIS-related services temporarily. The IIS-related services are restarted automatically when the upgrade is done.

Depending on your environment, after you upgrade the management servers in your site, you may be required to perform additional tasks.

Converting an Old Security Key for Encrypted Communications

If your AppManager site is configured to use encrypted communications, which is medium security, after you upgrade the management server, the management server continues to communicate with existing agents.

However, after you upgrade the management servers in your site, you should use the `NQKeyGenWindows.exe` utility with the `-convert` option to convert the older key file that was generated using the NetIQ Encryption Utility `rpckey.exe` to the new key format. After converting an old key file, use the `-change` option to check the key information into the repository, set the security level to **1** with the `-seclev` option, and restart your management servers.

Storing your existing key file in the repository provides the following benefits:

- The management server can use encryption to communicate with existing, before 7.0, and new, 7.0, agent installations.
- When you are ready to add another management server, the management server will be able to find the security key file in the repository.

For more information about using the `nqkeygenwindows.exe` utility, see the *Administrator Guide*.

Changing the Security Level

After you upgrade, if you want to change the security level for an AppManager site from encrypted communication, which is medium security to cleartext communications, which is no security, you must run a Knowledge Script.

To change the security level for a management site:

1. Run the AMAdmin_AgentConfigSecurityLevel Knowledge Script to change the security level for the agents within your management site.

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

3. ***If you have not configured the repository 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 repository to store the security key information, see "Converting an Old Security Key for Encrypted Communications" on page 45.

4. ***If you used the `NQKeyGenWindows.exe` utility to store security information in the repository***, use that utility again with the `-seclev` option to set the security level to **0**, and restart your management servers.

For more information about setting or changing the security level of an AppManager management site, see the AppManager *Administrator Guide*.

Upgrading Control Center

When you upgrade Control Center, you can select from the following Control Center components to install on each computer:

- Deployment service
- Deployment Web service
- Control Center database
- Command queue service
- Console

For more information about each component, see the *Control Center User Guide*.

To upgrade Control Center:

1. Ensure that you have upgraded all repositories managed by Control Center. If you upgrade Control Center without upgrading the repositories first several problems might occur, for example, AppManager will not be able to synchronize modes properly.

2. Back up your current AppManager Control Center database and the NetIQ program directory on each computer on which you have installed Control Center. For more information about backing up the Control Center database, see the *Administrator Guide*.

3. ***If you are using SQL Server authentication for your repository while using Windows authentication for your separate Control Center computer***, ensure that Kerberos delegation is properly configured for you environment. To configure this delegation, in AD Users and Computers, right-click on your computer and select **Properties**, then select **Trust computer for delegation**.

4. Ensure there are no commands in the Control Center queue.

5 Close all Control Center Consoles on the computer where you are upgrading. Close all consoles connected to the Control Center database you are upgrading.

6 To upgrade, run the setup program on the computer where Control Center is installed. If you have any other AppManager components installed on the computer, you must upgrade those components as well.

7 After you upgrade, reconfigure your settings. Settings are not maintained during the upgrade process. For more information, see the *Control Center User Guide*.

8 ***If you modified any permissions during the upgrade process***, configure your permissions. For more information about how to assign permissions, see the *Control Center User Guide*.

9 ***If you previously had any permissions assigned at any level other than a user group***, configure your permissions to be at a user group level. Control Center no longer allows you to assign permissions at any level other than user group. For more information about how to assign permissions, see the *Control Center User Guide*.

Upgrading Console Programs

Close any open console programs on the computer you are upgrading. For example, if you have a computer that has the Operator Console or Chart Console, ensure that it is closed before you run the setup program.

When you are ready to upgrade, run the setup program. When the setup program prompts you to select the console programs to install, select all of the console programs installed on the computer you are upgrading. If you have any other AppManager components installed on the computer, you must upgrade those components as well.

Chapter 5

Upgrading Managed Clients

The most significant part of a complete upgrade is upgrading your AppManager agents and managed objects. Typically, this is done in a phased process with some managed clients upgraded immediately and others upgraded over time. Before you upgrade managed Microsoft Windows and UNIX clients, make sure you have already upgraded the repository, management server, Control Center, console programs, and Web management server. For more information about the order in which to upgrade AppManager components, see Chapter 1, "Upgrade Checklist."

This chapter describes what you should consider in upgrading your managed clients, and how to upgrade managed clients locally using the setup program and remotely with Knowledge Scripts. The following topics are covered:

- Why You Should Upgrade
- Upgrade Options
- Preparing to Upgrade Managed Clients
- Selectively Upgrading Managed Objects on a Microsoft Windows Computer
- Upgrading the Local Repository for Managed Clients
- Automatically Discovering Resources after the Upgrade
- Upgrading Remote Managed Clients

Why You Should Upgrade

To get all of the new features and performance enhancements in this new version of AppManager, you must upgrade all of your AppManager components, including the AppManager agent on all of your managed client computers.

Although we recommend upgrading all components, some organizations cannot to do so all at once. For example, you may have components distributed across a wide area network and an upgrade may require the coordination of several departments.

At a minimum, you must upgrade the AppManager repository, management server, console programs, Control Center, Web management server, and report agents to AppManager 7.0.

You can continue to use AppManager 5.0.1 UNIX agents with this release of AppManager. However, the UNIX agent has been significantly revised, starting with the 6.5 UNIX agent release. It now has vastly improved stability and functionality.

You can have version 6.0.x Microsoft Windows agents with AppManager 7.0 core components. You should, however, plan to upgrade all of your managed clients over time, as part of a staged roll-out, for example, to ensure the best performance and functionality.

If you are planning to implement secure communications using management server authentication and encryption, you must upgrade all of your Microsoft Windows or UNIX agents to AppManager 7.0.

Upgrade Options

This release of AppManager allows you to upgrade Microsoft Windows managed clients from version 5.0.1 or later and UNIX managed clients from version 6.0.2 or later to version 7.0. The upgrade process does not allow you to

upgrade clients that are earlier than these versions. For more information about how to upgrade previous versions of AppManager clients, see "Upgrading Older Agents and Jobs" on page 35.

If you are planning to upgrade your management site to use management server authentication and encryption, you must upgrade all of your Microsoft Windows or UNIX agents to version 7.0. If you are currently using the encryption security setting, you can continue to do so without upgrading your existing agents.

To upgrade the managed Microsoft Windows and UNIX clients, you have several options:

- Run the main AppManager setup program locally on the managed computer as you would if installing AppManager for the first time. Select the **Agent** option on the main installation screen.
- For Microsoft Windows computers, run the agent setup program on the managed computer. For UNIX computers, run the UNIX agent installation script from the `root` user account on the local computer.
- For Microsoft Windows computers, using Control Center to upgrade the agent on remote managed clients. For more information, see "Upgrading Remote Managed Clients" on page 55.

The upgrade process does **not** change the security settings on the Microsoft Windows agent. However, if you run the installation script locally to upgrade the UNIX agent, you can change its security level setting from Cleartext to Encryption or Authentication and Encryption. For more information about agent configuration options, see the *Installation Guide*.

We recommend that you configure site security after you upgrade the agents in your environment. See the AppManager *Administrator Guide* for more information.

Note You should install the deployment web service before upgrading the managed clients. However, if you must upgrade before you install the deployment web service, ensure you run the AMAdmin_SetDeploymentWebService on your upgraded agents after you install the deployment web service.

When you upgrade the AppManager agent:

- The `NetIQ Client Resource Monitor` and `NetIQ Client Communication Manager` services on the managed Microsoft Windows computer, or the `nqmagt` executable and supporting libraries on UNIX, are replaced with new versions.
- The local repository is either updated and keeps the existing jobs, data, and events, or the repository is replaced so no existing information is retained, and you must start new versions of your jobs after the upgrade. You can manually start the UNIX agent after the upgrade to force the agent to use the current settings in the configuration file.

Agent installation used to include options to install or upgrade managed objects. With AppManager version 7.0, managed object installation is a separate step.

Preparing to Upgrade Managed Clients

If the managed client computer you want to upgrade is configured to monitor an application that is not supported by the latest version of AppManager, the installation program displays a warning

- Click **No** to cancel the upgrade and continue to monitor the currently discovered applications with an earlier, supported version of the agent.
- Click **Yes** to upgrade the agent. We recommend that you stop and delete existing jobs that monitor applications not supported by the AppManager 7.0 agent.

For the latest information about supported products, visit the NetIQ Web Site at www.netiq.com/support/am/supportedproducts.asp.

Selectively Upgrading Managed Objects on a Microsoft Windows Computer

When you run the setup program on a Microsoft Windows computer where managed objects are installed, you can choose the managed objects to upgrade on a managed computer, and you can install additional managed objects. For example, if you are running the setup program on a Dell server with Microsoft SQL Server and IIS installed, the setup program lists the managed objects for Dell OpenManage, Microsoft SQL Server, and Microsoft IIS, as well as other managed objects that can additionally be installed on that computer.

Note The AMAdminUNIX_AgentUpdate Knowledge Script, which upgrades UNIX managed objects, does not install new managed objects.

To avoid discovery errors after upgrade, upgrade all of your existing managed objects around the same time. If you want to continue monitoring an application that is no longer supported by AppManager 7.0, do not upgrade the agent. You can continue to use the existing Knowledge Scripts on a version 6.0.2 agent. For more information about unsupported applications, see Chapter 2, “Preparing to Upgrade.”

Upgrading the Local Repository for Managed Clients

To minimize the impact of upgrading the agent on your production environment, you can either upgrade or replace the local agent repository, which contains job information.

- If you upgrade the local agent repository, the upgrade preserves current Knowledge Script properties. They cannot take advantage of any new functionality, but continue to run exactly as they did before the upgrade.
- If you replace the local repository, you install a new local repository, existing jobs are deleted, and you must start new jobs after the upgrade.

Automatically Discovering Resources after the Upgrade

You can configure the setup program and remote upgrade Knowledge Scripts to automatically discover resources for all managed objects that are currently installed. If you have not upgraded a managed object and you configure the setup program to discover resources, an event will be raised. For more information, see "Selectively Upgrading Managed Objects on a Microsoft Windows Computer" on page 53.

Upgrading the AppManager report agent changes the default output file from `..\NetIQ\AppManager\Web\Report` to `..\NetIQ\Common\Report`.

The upgrade process does not rediscover the AppManager report agent. If you are upgrading an AppManager report agent, you must manually rediscover the AppManager report agent to run new reports.

Upgrading Remote Managed Clients

After you have upgraded the AppManager repository, management server, Control Center, and console programs, use one of the following methods to remotely upgrade managed clients with the new agent and managed objects:

- For Microsoft Windows computers, deploy the managed clients remotely using Control Center. Remote Deployment pushes any agent and module packages that you have "checked in" to the Deployment Web Service out to the computers you specify, using a set of deployment rules and a schedule you create. Agents and managed objects can also be installed across a firewall.
  Install the deployment web service before upgrading the managed clients. However, if you must upgrade before you install the deployment web service, ensure you run the AMAdmin_SetDeploymentWebService on your upgraded agents after you install the deployment web service.
- For UNIX computers where the UNIX agent runs as a root user, deploy the managed clients remotely using the AMAdminUNIX_AgentUpdate Knowledge Script that uses the user account under which the `nqmdaemon` runs to run the setup program and upgrade the UNIX agent. To use this Knowledge Script, make sure the `nqmdaemon` is configured to run as `root`. For UNIX computers where the UNIX agent does not run as a root user, you must run the `netiq_agent_install` script on the local computer.

For more information about upgrading agents, see the AppManager for UNIX or AppManager for Microsoft Windows *Management Guide*.

Chapter 6

Upgrading Jobs

This chapter describes how to upgrade all child Knowledge Script jobs for a selected parent job. Upgraded jobs will use the latest Knowledge Script. Any changes in the Knowledge Script parameters between AppManager versions are taken into account. The following topics are covered:

Overview

Use the AMAdmin_UpgradeJobs Knowledge Script to upgrade all child jobs for one or more parent jobs.

After you upgrade the AppManager agent, existing jobs on the managed client computer are not automatically upgraded to use the latest Knowledge Script functionality.

Upgrading jobs to use the latest Knowledge Script version allows the jobs to take advantage of the latest Knowledge Script logic while maintaining existing parameter values for the job. Any associated graph data and event information are also retained if they have not changed. If the latest version of a Knowledge Script has been modified to have new parameters, for example, to create different events or data streams, the default values for the new parameters in the latest Knowledge Script are used.

Warning The functionality provided in the latest version of the Knowledge Script logic may not be supported by older agents with older managed objects. Upgrade your managed Microsoft Windows clients to the latest version of the AppManager agent before you upgrade jobs running on those agents.

Preparing to Upgrade Jobs

The AMAdmin_UpgradeJobs Knowledge Script uses an AppManager 7.0 agent that is configured to run under a Microsoft Windows user account to access the repository database:

- The Microsoft Windows user account being used to run this utility must belong to the AppManager **Administrator** role.
- The Log On As account for the agent services, `NetIQ Client Resource Monitor` and `NetIQ Client Communication Manager`, must belong to the AppManager **Administrator** role.

To verify that your Microsoft Windows user account belongs to the AppManager **Administrator** role, in AppManager Security Manager, expand **AppManager Roles** in the TreeView and click **Administrator** to see a list of valid AppManager administrators.

AMAdmin_UpgradeJobs

The AMAdmin_UpgradeJobs Knowledge Script upgrades all child jobs for one or more parent jobs. You can select the parent jobs you want to upgrade based on:

- **Knowledge Script**—Select this option to upgrade all ad hoc jobs started by the specified Knowledge Script. This option upgrades ad hoc jobs started by a particular Knowledge Script and ad hoc jobs started by a Knowledge

Script Group member. This option does not upgrade policy-based jobs.

- **Knowledge Script Category**—Select this option to upgrade all ad hoc jobs started by the specified Knowledge Script category. This option does not upgrade policy-based jobs.
- **Parent Job Identifier**—Select this option to upgrade all ad hoc child jobs that belong to the specified Parent Job ID. This option does not upgrade policy-based jobs.
- **Monitoring Policy**—All policy-based jobs started by the specified Knowledge Script Group are upgraded. If you are using a Knowledge Script Group in one or more monitoring policies, all affected monitoring policies are updated. This option does not upgrade ad hoc jobs started by a Knowledge Script Group.

Performing an Instant-Check Query before Upgrading Jobs

Before you attempt to upgrade jobs using this Knowledge Script, you should identify jobs that have not yet been upgraded by performing an **instant-check query**.

The instant-check query provides a list of jobs to upgrade and jobs that have already been upgraded. You should use the instant-check query as a starting point to identify the jobs to upgrade and develop a strategy for upgrading existing jobs.

The instant-check query identifies jobs by AppManager version and displays both Microsoft Windows and UNIX jobs. Use the name of the Knowledge Script category to identify Microsoft Windows or UNIX jobs.

The query results for each job also include the version of the AppManager agent. You cannot upgrade UNIX jobs on a backlevel UNIX agent. You must upgrade the backlevel UNIX

agent to the latest version before you can upgrade the jobs on that agent.

To perform an instant-check query, select the query you want:

- **Out-of-date parent jobs**—Parent jobs with a Knowledge Script that is not the latest. Run this query to get a list of parent IDs that you should upgrade. Note that some parent jobs may contain two different versions of a Knowledge Script. If that is the case, and either one of them is not the latest, the KS Build ID field says `multiple build IDs`.
- **Up-to-date parent jobs**—Parent jobs whose scripts are the latest. Run this query to get a list of parent job IDs that are presently using the latest Knowledge Script in the repository and cannot be updated.
- **Old parent jobs with no upgrade**—Jobs with an old Knowledge Script but for which there is no newer version in the repository. If this query returns any parent job IDs, it means the Knowledge Script has either been discontinued in later versions of AppManager, or it is a Knowledge Script you created or customized under a new name and for which you have yet to create a new version in the repository. When this query returns no values it means you have no parent jobs using out-of-date Knowledge Scripts. No further upgrading is required.
- **Old Knowledge Scripts with no upgrade**—Old scripts for which there is no newer version in the repository. Run this query for a list of Knowledge Scripts in the repository that might be custom scripts in need of a newer version. Unlike the other queries, this is not a list of jobs.

  You can also use this query to verify that scripts you recreated as a new version were done correctly, and no longer appear.

- **Child jobs on v5.0.1 and v6.x agents**—Queries for child jobs running on version 5.0.1 or 6.x agents. These queries list child jobs running on the specified agent version.

  You cannot upgrade ad hoc or policy-based jobs on a backlevel UNIX agent. You must upgrade the backlevel UNIX agent to the latest version to upgrade jobs. If you attempt to upgrade child jobs running on a mix of version 6.0 UNIX agents, the upgrade job appears to runs successfully, but the jobs on 6.0 UNIX agents are not upgraded.

- **Agent build IDs**—This query lists the agent build number on each computer. You can use this list to identify agents that you may want to upgrade.
- **Monitoring-policy jobs**—This query lists all of the jobs that are currently part of a monitoring policy. The jobs are listed according to the view or server group associated with the monitoring policy and then sorted by Knowledge Script group. The Knowledge Script group names as shown in the **KSG Name** field have the prefix `KSG_`. If you specify a Knowledge Script group to upgrade, add this prefix to the group name.

  You cannot upgrade backlevel UNIX jobs that are policy-based. After you upgrade the backlevel UNIX agent to the latest version, remove the existing backlevel policy-based jobs and create new policy-based jobs.

Getting a Preview before You Upgrade Jobs

Before you upgrade jobs, you should use the **Generate Report** option to identify Knowledge Scripts with new parameters.

This option provides detailed information about the changes to the actual script, including a list of new or changed parameters. If the latest Knowledge Script has new or changed

parameters, you can preview the default values for these parameters before they are applied when you upgrade.

Each time you run the Knowledge Script, AppManager reports job upgrade information. New reports are saved in `Program Files\NetIQ\Temp\NetIQ_Debug\[Computername]\jobupgrade\Out of date parent jobs\Out of date parent jobs.html`. For more information, see "Viewing Sample Job Upgrade Reports" on page 68.

Upgrading Jobs Created by a Custom Knowledge Script

If you have written a custom Knowledge Script, you do not need to upgrade existing jobs created by that Knowledge Script unless you have made changes to the Knowledge Script. In most cases, custom Knowledge Scripts can be run successfully on AppManager 7.0 agents.

If you have copied a standard AppManager Knowledge Script, see "Upgrading Jobs Created by a Copy of a Standard AppManager Knowledge Script" on page 62 for more information.

Upgrading Jobs Created by a Copy of a Standard AppManager Knowledge Script

Before you can upgrade jobs created by a copy of a Knowledge Script, you must update the copy of the Knowledge Script in the AppManager repository.

To update a copy of an AppManager Knowledge Script:

1. On the repository computer, use Microsoft Windows Explorer to open the `\NetIQ\AppManager\qdb\kp` folder and click the folder that contains the new version of the original Knowledge Script upon which the copy is based.

2. Copy the Knowledge Script and rename it to use the same name as the Knowledge Script copy.
3. Check the updated Knowledge Script copy into the repository. You are now ready to upgrade existing jobs.

Verifying that a Job Has Been Upgraded

To verify that a job has been upgraded, you need to know its Knowledge Script version. In the Operator Console, customize the **Jobs** tab of the List pane to display Knowledge Script version information.

To verify that a Knowledge Script job has been upgraded:

1. In the Operator Console, right-click the **Job** column in the **List** pane and click **Customize**.
2. Click **KS Version** in the Available Columns list and click **Add**.
3. Click **OK**.
4. On the **Jobs** tab, the Knowledge Script version for each job appears in the **KS Version** column.
5. Ensure that all jobs that were running before upgrade have been restarted.

Resetting Password Information for Upgraded Jobs

In some rare cases, running the AMAdmin_UpgradeJobs Knowledge Script replaces the existing password you have specified for your environment with the default password specified in the original Knowledge Script properties. After these jobs are upgraded, they no longer run because the

password is incorrect. This problem occurs for the following Knowledge Scripts:

- NTADMIN_AddUser
- NTADMIN_ChangePassword
- SQL_Bcp

If you upgrade any of these Knowledge Script jobs, update the job Properties to restore the correct password information.

Using the UpgradeJobs70.exe Utility

The `upgradejobs70.exe` utility upgrades all child jobs for one or more parent jobs. Installed by default in your `Program Files\NetIQ\AppManager\bin` directory, this utility can also be used to generate a report that lists the jobs to be upgraded. The utility does not support the Instant Check Query feature. You must use the AMAdmin_UpgradeJobs Knowledge Script to perform an instant-check query.

The `upgradejobs` utility uses Microsoft Windows authentication to access the repository database. The Microsoft Windows user that you are logged in as when you run this utility must belong to the AppManager **Administrator** role. To verify that your Microsoft Windows user account belongs to the AppManager **Administrator** role, in AppManager Security Manager, expand **AppManager Roles** in the TreeView and click **Administrator** to see a list of valid AppManager administrators.

To run the `UpgradeJobs` utility:

1. On the computer where the `upgradejobs70.exe` file has been saved, open a command prompt.
2. CD to the directory where the utility has been saved. The default directory is saved to the `c:\Program Files\NetIQ\AppManager\bin.`

3 Enter the `upgradejobs` command followed by the options to specify the AppManager repository you want, the parent jobs to upgrade, and whether to upgrade the jobs or generate a report.

Use the following format to enter the information:

```
upgradejobs
        -s <SQLserver>:<dbname>

          -j <list> |
          -f <filename> |
          -k <ksname> |
          -t <kscategory> |
          -a all |
          -g <ksgroup>
                                  } Pick one.

        -m <mode>

        [-r reportonly]
        [-w txt]
        [-q <integerID>]
```

Notation:

- **[optional]** – You do not need to provide options in brackets.
- **a|b|c** – The vertical bar means "or." Pick **one** of these options.
- **<information>** – Less Than and Greater Than symbols enclose information you must supply. Do not actually use the symbols.
- Separate elements with a space.
- Order is not important.
- Use lower case for the option letters, but the option arguments are not case-sensitive unless Microsoft SQL Server is.

Use the `upgradejobs` command-line arguments as needed.

Option	Description
`-s server: dbname`	Specifies repository logon information, where *server* is the repository server name, followed by a colon (:) and *dbname* is the repository database name. To run this program, the Microsoft Windows user you are currently logged in as must belong to the **AppManager Administrator** role.
`-j <list>`	Specifies a list of parent job IDs in a comma-delimited list. For example: `-j 11,15,25` (no spaces between items) If the list contains invalid characters, such as non-numeric characters, the upgrade will fail.
`-f <filename>`	Specifies the name of a text file containing a list of parent job IDs. The list in the file should be a comma-delimited list with no spaces, as used in the -j option, above. For example: `-f parentjobs.txt`
`-k <ksname>`	Specifies a single Knowledge Script name to upgrade. The command upgrades all jobs using Knowledge Scripts with this name. For example: `-k NT_CpuLoaded`
`-t <kscategory>`	Specifies the name of a Knowledge Script category. These are listed on the tabs in the AppManager Console. The command upgrades all jobs using scripts from this category. For example: `-t NT`
`-a all`	Upgrades all the jobs that can be upgraded based on the selected mode (either `Force` or `Restricted`).

Option	Description

`-g <ksgroup>`	Specifies the Knowledge Script Group you want. All policy-based jobs started by the specified Knowledge Script Group are upgraded. If you are using a Knowledge Script Group in one or more monitoring policies, all monitoring policies are updated. This option does not upgrade ad hoc jobs started by a Knowledge Script Group.
	Take care when entering these names, most of the Knowledge Script group names are prefixed by `KSG_`, but the console shows only the Knowledge Script group name without this prefix. Please make sure that the correct full name is entered. Some of the KS group names contain a space. In this case, quotes must be used in order to enter the correct name. For example, `KSG_My 1 2 3 group`.
	Force mode is required for the KSG upgrade option. If you use this option, the `-m` option must be set to **`force`**.
	For example:
	`-g KSG_mygroup -m force`
	`-g ”KSG_my group” -m force`

`-m <mode>`	Specifies the mode you want to use for the upgrade: • **Force** Upgrades all child jobs for the selected parent job regardless of the agent version to the latest version (if there is a newer Knowledge Script. This Knowledge Script does not upgrade jobs on a version 4.3 or 5.0 UNIX agent. • **Restricted** Upgrades all child jobs for the selected parent job if all of the agents that are running the job are version 7.0. This is the default mode.
`-r reportonly`	Generates a report of the jobs it would upgrade, but has not yet upgraded. Lets you check to make sure the upgrade decisions are correct before you actually run the upgrade. **Note** The command-line default behavior is to actually upgrade the jobs. That is different than the AMAdmin_UpgradeJobs script, which runs the report by default. By default the report goes to the following directory: `\netiq\temp\netiq_debug\<machine>\jobupgrade` on the machine on which you ran the `upgradejobs` command.

Option	Description
-w txt	Sends the command output to a text file instead of to the screen. The file will be called upgradejob.txt unless you use the -q option. The command places this file in the directory you were in when you ran the command. By default the output goes to the command line. With this option it goes to following directory on the computer where you ran the job upgrade utility: \netiq\temp\netiq_debug\<machine>\jobupgrade
-q <integerID>	Appends _<integerID> to the filename of the output file (if you specified the -w option), the log file, and the report file: For example, if you say -q 12 you get a file called upgradejob_12.txt.

Viewing Sample Job Upgrade Reports

Each time you run the UpgradeJobs Knowledge Script, job upgrade reports are created under:
\NetIQ\Temp\Netiq_debug\<*computername*>\jobupgrade

where <computername> is the computer where you ran the report. The following reports are always generated regardless of whether you configure this job to generate a report or upgrade jobs:

- Upgradejob_<id>.txt, where <id> is the UpgradeJobs ID, provides information about which jobs are upgraded.
- Upgradejob_<id>.rpt, where <id> is the UpgradeJobs job ID, provides detailed information about each job.

Upgradejob_<id>.log, where <id> is the UpgradeJobs ID, lists the Job IDs that are upgraded and references the corresponding .rpt file and .log files for more information.

If the child of a specified parent job is running on an agent that has not been upgraded to the current version, and you specified the **Restricted** upgrade option, the

`UpgradeJob_<id>.txt` file displays information similar to the following example:

```
Connected to SQL Server : RACKR14 repository QDB.

Time stamp: 03/03/04 14:20:47

  [Child Job] [Parent Job] [Build ID]  [Computer\KS]

2 5.0.1 agent(s) found.

2 6.0.2 agent(s) found.

Parent job 436 is skipped because under restricted mode, there cannot be
any non-7.0 agents.

Upgrade is finished.


Please check upgradejob_1343.rpt and upgradejob_1343.log located in

D:\NetIQ\Temp\NetIQ_Debug\RACKR14\jobupgrade.

Time stamp: 03/03/06 14:20:47
```

If the child of a specified parent job can be upgraded with parameter changes, the `UpgradeJob_<id>.rpt` file displays information similar to the following:

```
Connected to SQL Server : RACKR14 repository QDB.
Time stamp: 03/03/06 15:14:30
*************************************************************
Parent job 54 can be upgraded under force mode.
2 5.0.1 agent(s) found.
2 6.0.2 agent(s) found.
1)
Child job ID = 55
Parent job ID = 54
KS name = NT_CpuLoaded
Machine name = RACKN08
Version = 5.0
Job 55 can be upgraded.
The following parameters in the existing job are not found in
the new version of the KS:
1) Event? (y/n)
Existing value is y.
2) Collect Data? (y/n)
Existing value is y.
3) Overall Load? (y/n)
```

```
Existing value is y.
4) Cpu Threshold >
Existing value is 0
. . .
14) Threshold - Processor queue length
Default value is 0
Check for OldParameter tag
1) Create event if total system CPU is high?
Default value is y
OldParameter tag value = ?DO_EVENT="y" ((AND))
DO_OVERALL="y":"y":"n".
New StringValue = "y"
. . .
4) Severity - Individual CPU
Default value is 15
OldParameter tag value = ?DO_EVENT="y" ((AND))
DO_OVERALL="n":Severity:$default$.
No matching value, will keep original.
. . .
10) Threshold - Individual CPU
Default value is 98
OldParameter tag value = ?DO_OVERALL="n":TH_UTIL:$default$.
No matching value, will keep original.
```

If the child of a specified parent job cannot be upgraded because the agent on which it is running is from a version no longer supported (such as 5.0), the entry looks like this:

```
Parent job 1536 cannot be upgraded under restricted mode.
29 5.0 agents are found.
Please upgrade these agents and restart the upgrade process.
```

In this case, you would need to upgrade the agent or use the **Force** option to upgrade the jobs on the older agent.

Chapter 7

Troubleshooting an Upgrade

In most cases, you will not encounter problems in the upgrade process. If you are not sure whether the upgrade has been successful or think errors may have occurred, this chapter provides suggestions for where to look for information in troubleshooting the upgrade.

Avoiding Common Problems

Problems occurring when upgrading the agent or repository may stem from not having proper access permissions for the Deployment Web Service or the user account executing the remote setup on the target computer.

To resolve problems updating the agent:

1. Verify the user account access rights running the agent services, `NetIQ Client Resource Monitor` and `NetIQ Client Communication Manager`, on the management server.
2. Verify the access rights for the user account running setup on the target computer.
3. Verify network communication between the management server and the target computers.

4 ***If you see the following event message when attempting to run a job on a managed client computer***, you must manually configure the agent's list of authorized management servers by updating the `AllowMS` registry key:

```
Communication is not authorized.
```

```
Start job request is not authorized to run on the agent
machine <hostname>. Check the security settings on the
agent machine.
```

For more information about registry keys, see the *Administrator Guide.*

5 ***If your site is configured to use the secure communication, either Encryption or Encryption and Authentication, and you see the following event message***, verify that the security level on the agent matches the security level in the repository database:

```
This job is not supported on the managed client.
```

```
The security level set for the agent machine does not
match the security level set for the site. You may use
NQKeyGenWindows.exe to modify security level settings.
Please see the Help for further information on the use
of this utility.
```

If the security level on the agent matches the security level in the repository, restart the management server.

Control Center and Repository Upgrades

If Knowledge Script jobs from Control Center do not start or do not performing as expected, verify the primary repository version is 7.0. While Control Center partially supports AppManager repositories from AppManager version 6.0.2, the primary repository must be of an AppManager version greater than or equal to the version of the rest of the AppManager repositories you are managing with Control Center. If you do not want to upgrade your present primary repository to AppManager version 7.0, change the primary repository designation to another AppManager repository that is at least

version 7.0 after you complete the Control Center and AppManager upgrades.

When you attempt to run jobs from Control Center, you may find that you:

- Cannot create or modify new jobs on 6.0.2 or earlier repositories.
- Cannot create or modify custom properties on 6.0.2 or earlier repositories.
- See commands in your Control Center repository that have an error status, but have actually completed successfully.

Custom properties are not supported on AppManager repositories older than version 7.0. If you try to add, edit, or delete a custom property for a backlevel repository, you'll see a message indicating that the changes are not reflected on repositories prior to version 7.0.

You should not experience any difficulties viewing and manipulating data from those repositories, but you should plan to upgrade all repositories eventually to avoid unexpected problems with Control Center.

Where to Look for Help

AppManager includes numerous log files and diagnostic tools that track the progress and outcome of the upgrade as well as AppManager's ongoing operation. Logs are located in the `\NetIQ\Temp\NetIQ_Debug\<computer>` directory. The following

table provides a quick reference to these sources of information.

Computer	Filename	Information Provided
Repository server	kscheckin.log	Tracks the progress of the check-in process for Knowledge Scripts and schedules (the files that define intervals such as Hourly).
	qdbinstall.log	Tracks the installation and configuration of the repository database.
	qdbupgrade60.log	Tracks the installation and configuration of a repository database during an upgrade. Knowledge Scripts that may require manual migration are indicated by “Failed to prepare modified KS.”
	rplib.log	Records communication between the Operator Console and the repository and between the management server and the repository, including any error messages that result from running the repository’s stored procedures.

Computer	Filename	Information Provided
	kscustom60.log	Lists Knowledge Scripts with changes to Knowledge Script properties and indicates whether the properties were migrated: • Successfully migrated Knowledge Scripts have a status of “Customized.” This status indicates that changes to the default script properties in the existing Knowledge Script have been migrated to the new 6.0 Knowledge Script. • Unsuccessfully migrated Knowledge Scripts have a status of “Not customized because parameters of 5.0.1 or 6.0 and 6.7 KSs do not match.” This status indicates that there were changes in the number, name, or data type of the parameters in the script logic. In this case, you can review the changes and manually update the new 7.0 Knowledge Script to use the default values you want. **Note** Unsuccessfully migrated changes to the KPP section can be related to changes in the actual script logic of the KPS section, which this log file does not identify.
Management server	ms.log	Traces the internal operations performed by the NetIQ Management Service.
Managed client	ccmtrace.log	Traces the internal operations performed by the NetIQ Client Communication Manager service.
	ioc.log	Traces internal pipe communication between the NetIQ Client Resource Monitor and NetIQ Client Communication Manager services.
	mctrace.log	Traces the internal operations performed by the NetIQ Client Resource Monitor service.
	mo.log	Traces the internal operations performed by managed object(s) during Knowledge Script execution.

Appendix A

Upgrading in a Clustered Environment

This chapter outlines the differences in AppManager support for monitoring clustered applications that are introduced with this release. It offers advice for maintaining your monitoring environment during and after the upgrade and for making the necessary modifications to your AppManager management site to take advantage of new support. This chapter contains the following sections:

About AppManager Support for Monitoring Clusters

AppManager only supports monitoring servers that are clustered using Microsoft Cluster Server (MSCS).

For some applications, such as Microsoft Exchange Server, Microsoft SQL Server, and Oracle RDBMS, you can run application-specific Knowledge Scripts on the physical nodes that make up the cluster to monitor the virtual server. For these applications, monitoring is handled through the application-specific managed object, such as the Microsoft Exchange managed object or the Microsoft SQL Server managed object.

For cluster resources that are not linked to a particular application, monitoring is handled through the Microsoft Cluster Server managed object and the MSCS Knowledge Scripts. Not all Microsoft Windows server applications are cluster-enabled, however. In general, you can use

AppManager to monitor Microsoft Exchange Server, Microsoft SQL Server, and Oracle RDBMS running as virtual servers in a cluster and cluster resources such as logical disk drives through the Microsoft Cluster Server managed objects.

To perform any cluster monitoring, the AppManager agent must be installed on the local, not shared, disk of each node in a cluster.

Changes to Discovery of Clustered Resources

With previous releases of AppManager, discovery of the disk resources of the virtual server could only take place when the virtual server was active on the computer where you were installing the agent and managed object. To allow failover, you had to moving the active virtual server to each node before installing the managed object. *This procedure is no longer necessary*. Instead, use the new Discovery_Cluster Knowledge Script to discover clustered server resources.

Appendix B

Migrating to Microsoft SQL Server 2005

This appendix covers backing up your Microsoft SQL Server AppManager repository database and migrating your repository to Microsoft SQL Server 2005. This Appendix contains the following topics:

Backing up Your Repository

Back up your AppManager repository before and after you attempt to upgrade. You should also create a backup of the AppManager repository before you migrate your repository from Microsoft SQL Server 2000 to Microsoft SQL Server 2005. To minimize the impact on your environment and have the repository as up-to-date as possible, you should back up the repository immediately before starting the upgrade.

For more detailed information on backing up and restoring databases, see the Microsoft SQL Server documentation.

This appendix describes two common ways to back up the AppManager repository database:

- Using SQL Enterprise Manager
- Running the SQL_RunSql Knowledge Script

Using SQL Enterprise Manager to Create a Backup

Before you attempt an upgrade, you must ensure that your data is saved.

To create a backup using SQL Enterprise Manager:

1. In SQL Enterprise Manager, select the server on which the AppManager repository resides.
2. Click **Tools > Backup Database**.
3. Select the name of the AppManager repository, for example, the default repository name is **QDB**, in the **Database** list.
4. Select **Database - complete**.
5. Select an item in the **Backup to** list. If you do not have a backup device designated, you must create one. If this is the first time you are backing up the repository or you need to create a new backup device, see "Creating a New Backup Device" on page 80.
6. ***If you want to discard existing backups on the selected backup device***, select **Overwrite existing media**. If you leave **Append to media** selected, the backup is appended to the backup device. If you choose **Overwrite existing media**, click the **Options** tab, select **Initialize and label media** and then specify a backup name and description, then click **OK**.
7. When the backup procedure finishes, click **OK**.

Creating a New Backup Device

If you have not previously established a backup device, you must create one.

To create a backup device:

1. In the Microsoft SQL Server Backup window, click **Add**.

2 Click **Backup Device** and then select **<New Backup Device>**.

3 Type a name for the backup device and select a location, then click **OK**.

4 Ensure the backup device is selected, then click **OK**.

5 Your new backup device is displayed. You can then select an Overwrite option and click **OK** to begin backing up the repository database.

Using SQL_RunSql to Create a Backup

To automate and simplify backing up the AppManager repository, you may want to use the SQL_RunSQL Knowledge Script. This Knowledge Script enables you to create a job that backs up the repository at scheduled intervals.

To create a backup using SQL_RunSql:

1 In the AppManager Operator Console or Operator Web Console, start a SQL_RunSql Knowledge Script job on the SQL Server where the repository to be backed up is located. For information on starting jobs, see the *User Guide*.

2 In the Values tab in the Knowledge Script Properties window, either:

- Specify a SQL script that contains all the necessary commands for backing up the AppManager repository, msdb, and master databases by setting **Load SQL Script** to **y** and, for **SQL Script File**, enter the full path to the script file.
- Specify a SQL statement by entering the statement for each **SQL Statement**.

An example of a SQL statement for backing up the data in one database is:

```
backup database <QDB> to <backup_device>
```

where `<QDB>` is the name of the AppManager repository and `<backup_device>` is the name of the backup device.

Use the `backup` statement be used instead of the `dump` statement. In a future version of SQL Server, `dump` will not be supported. For information about SQL scripts and SQL statements, see your Microsoft SQL Server documentation.

For more information about using the SQL_RunSql Knowledge Script, see the Knowledge Script Help.

Backing Up Tasks, Login Accounts, and Stored Procedures

In addition to the repository database, the AppManager repository includes Microsoft SQL Server **tasks**, **login accounts**, and **stored procedures**. Microsoft SQL Server tasks are used to perform routine maintenance and updates at scheduled intervals. A login account establishes a connection to Microsoft SQL Server. (With Microsoft Windows Authentication, you do not need to maintain a separate login ID for Microsoft SQL Server; you can use your Microsoft Windows user account.) Stored procedures are used to perform internal AppManager operations and in generating AppManager reports.

In addition to the AppManager database (for example, `QDB`), you should periodically back up your SQL tasks, login accounts, and stored procedures.

- To back up SQL tasks, follow the steps described in "Using SQL Enterprise Manager to Create a Backup" on page 80, except select **msdb** in Step 3, then continue with Step 4 through Step 7.

- To back up Microsoft SQL Server login accounts and extended SQL stored procedures, follow the steps described in "Using SQL Enterprise Manager to Create a Backup" on page 80, except select **master** in Step 3, then continue with Step 4 through Step 7.

Restoring your Environment from a Backup

When you are ready to bring your data back to your database, restore the database from the back up.

To restore your AppManager repository from the backup:

1. Close any open console programs, including connections from remote computers, connected to the repository you want to restore.
2. On the computer where the management server is running, open the Services Control Panel and stop the `NetIQ Management Service` that communicates with the repository server. You may need to wait a few moments to allow the service to stop and for connections to the repository to be terminated.
3. If any other programs or users are connected to the repository, close those connections. The restore operation requires exclusive use of the system.
4. Open Microsoft SQL Enterprise Manager.
5. In the Server Manager, expand the repository server and the Databases folder to view the AppManager repository database, for example, `QDB`.
6. Right-click the AppManager repository database and select **Properties** from the right-click menu.

7 Click the **Options** tab and select the **Single user** option, then click **OK**. The Single user option ensures the System Administrator, `sa`, user has exclusive use of the database while the repository is restored.

8 Right-click on AppManager repository database and select **All Tasks > Restore Database** from the right-click menu.

9 Verify that the AppManager repository database you want to restore appears in the **Restore as database** list, then click **OK**.

10 After the database is restored, verify that the database has been returned to a working state by confirming that the **Single User** database option is deselected:

- Right-click on the AppManager repository database and select **Properties** from the right-click menu.
- Select the **Options** tab and clear the **Single User** option.

11 Repeat Steps 6 through 10 to restore the Microsoft SQL Server tasks in the `msdb` database and login accounts and stored procedures in the `master` database.

Restarting Connections after Restoring a Database

Because the Restore operation requires exclusive access to the repository, after you finish restoring a database you need to restart the management server and any console programs.

To restart the connections:

1 On the computer where the management server is installed, open the Services Control Panel and start the `NetIQ Management Service` that communicates with the repository server.

2 Log on to the AppManager repository with any console program or utility.

Migrating to a New Version of Microsoft SQL Server

If your current AppManager repository is on Microsoft SQL Server 2000, migrating it to Microsoft SQL Server 2005 is optional.

To migrate your repository to Microsoft SQL Server 2005:

1. Back up your existing repository.
2. Upgrade the repository to version 7.0.
3. Back up the version Microsoft SQL Server 2000 repository.
4. Use the Microsoft SQL Server Upgrade Wizard to upgrade the version Microsoft SQL Server 2000 repository to Microsoft SQL Server 2005.
5. Copy `nqsql.dll` and `nqsqlext.dll` from `\Program Files\NetIQ\AppManager\qdb` to `\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn`