



Configuration Guide for High Availability Distributed System on Microsoft SQL Server

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Configuration Guide for High Availability Distributed System on Microsoft SQL Server
© 2012 Cisco Systems, Inc. All rights reserved.

Contents

| | | |
|----------|--|--------|
| 1 | Getting Started | v |
| | Terminology | v |
| | Overview | v |
| | Architecture | vii |
| | What is Horizontal Scalability? | viii |
| | NCM Internal Monitoring: SQL Server Merge Replication | viii |
| | NCM Internal Monitoring: Inter-NCM Core Communication | ix |
| 2 | Installation, Setup, and Upgrading | xi |
| | System Requirements | xi |
| | Prepare for Initial Replication | xi |
| | Planning | xii |
| | Protocols, Databases, and Ports | xiii |
| | Unpacking the Replication Script Installation Bundle | xiv |
| | Creating a Two NCM Core SQL Server Replication Environment | xiv |
| | Recovering from a Failed Replication Configuration | xx |
| | Removing Replication | xx |
| | System Requirements | xx |
| | Deletion Steps | xxi |
| | Upgrade the NCM Mesh | xxii |
| | Upgrading from NCM 1.3/1.3.01 to NCM 1.4 | xxiii |
| | Upgrading from NCM 1.4x to NCM 1.5 | xxiv |
| | Upgrading from NCM 1.5 to NCM 1.6 | xxv |
| | Upgrading from NCM 1.6 to NCM 1.7 | xxvi |
| | Upgrading from NCM 1.7 to NCM 1.8 | xxvii |
| 3 | System Administration | xxx |
| | Getting Started | xxx |
| | How NCM Removes In-Memory and Database Information | xxx |
| | Suspending Replication for Database Maintenance | xxxii |
| | NCM Generated Events | xxxii |
| | Distributed System - Uniqueness Conflict | xxxii |
| | Distributed System - Time Synchronization Warning | xxxiii |
| | Distributed System - RMI Error | xxxiii |
| | Distributed System - Stopped Merge Agent Job | xxxiv |
| | Distributed System - Data Synchronization Delay Warning | xxxiv |
| | Using the NCM Distributed System Pages | xxxiv |
| | Distributed Monitor Results Page | xxxiv |
| | Distributed Conflict List | xxxv |

| | |
|---|--------------|
| View Distributed Conflict Page | xxxv |
| Partition Reassignment Page | xxxvi |
| List Cores Page | xxxvi |
| Edit Core Page | xxxvii |
| Device Password Rule Priority Reset Page | xxxvii |
| Renew Configuration Options Page | xxxviii |
| 4 Failover and Recovery | xxxix |
| Failover | xxxix |
| Recovery | xxxix |
| Loss of Network Connectivity | xxxix |
| Loss of an NCM Application Server | xl |
| Loss of a Database Server | xl |
| 5 Troubleshooting | xl |
| RMI Errors in the NCM Console | xl |
| Removing In-Memory and Database Information | xl |
| Index | xl |

1 Getting Started

This chapter contains the following topics:

- [Terminology](#) on page v
- [Overview](#) on page v
- [Architecture](#) on page vii
- [What is Horizontal Scalability?](#) on page viii
- [NCM Internal Monitoring: SQL Server Merge Replication](#) on page viii
- [NCM Internal Monitoring: Inter-NCM Core Communication](#) on page ix

Terminology

The following terms are used throughout this guide:

- **NCM Core** — A single NCM Management Engine, associated services (Syslog and TFTP), and a single database. An NCM Core can manage multiple Partitions.
- **Partition** — A set of devices with unique IP addresses. A Partition is managed by one (and only one) NCM Core. Multiple Partitions can be managed by a single NCM Core. For information about segmenting devices, see the *User Guide for CiscoWorks Network Compliance Manager*.
- **NCM Mesh** — Multiple NCM Cores connected via multimaster replication.
- **Publisher** — A SQL Server database that defines what data is replicated and handles transferring changed data to and from the other databases in the NCM Mesh.
- **Subscribers** — Databases in the NCM Mesh that can send and receive replication data, but do not control the replication process.



Both the Publisher and Subscribers will have the same data (subject only to replication time lags).

Overview

The NCM Distributed System on Microsoft® SQL Server is a multimaster system where the data from each NCM Core in an NCM Mesh is accessible to all other NCM Cores. This provides a comprehensive view of your data and allows for redundant data and failover in the event of a problem with a single NCM Core. The multimaster features include both database data and certain file system data, such as software images and device driver packages. Keep in mind that software images and device driver packages are also replicated across the NCM Mesh.

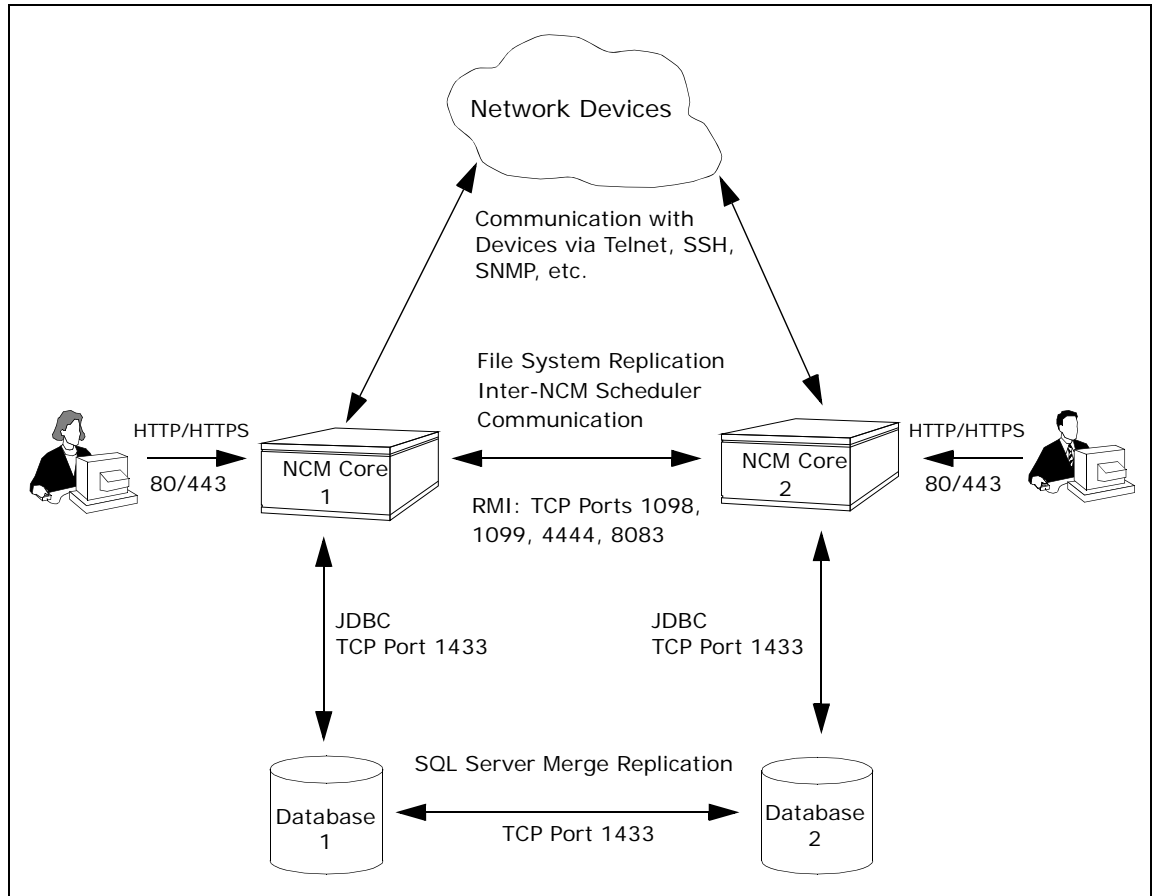
The following comprises the NCM Distributed System:

- A NCM Core and a partition:
 - A device is associated with a single partition.
 - A partition is associated with a single NCM Core.
 - A task is associated with a specific NCM Core.
- Merge replication on SQL Server:
 - Conflict resolution in the event that the same piece of data is modified almost simultaneously on two separate NCM Cores. This conflict is typically resolved using the latest timestamp method.
 - Replication monitoring and conflict notification is built into NCM. You can manage SQL Server replication conflicts and view merge agent job statuses from within the NCM console.
 - The NCM scheduler is multi-core aware. You can schedule group tasks containing devices that are associated with different NCM Cores. The system will run these tasks on the correct NCM Core. You do not have to schedule tasks on the appropriate NCM Core.

System setup requires a thorough understanding of SQL Server and NCM. Installation includes a number of steps that must be performed on the various servers. In addition, certain network changes may be necessary to allow connections between the servers. Once setup is complete, you will need to partition your devices into partitions to ensure proper NCM Core access to devices. For information about segmenting devices, see the *User Guide for CiscoWorks Network Compliance Manager*.

Architecture

The following figure shows a typical two-NCM Core Multimaster Distributed System installation. The installation enables failover from one NCM Core to another, while the remaining NCM Core continues to manage all devices.



Keep in mind that some installations might have two separate network device collections that are inaccessible from the non-managing NCM Core. These installations still provide data redundancy and scaling benefits, however.

Note that the database data transport uses standard SQL Server Merge Replication. For details on how replication works, see [System Administration](#) on page xxxi.

The communication between the NCM Cores is done using Java's Remote Method Invocation (RMI). RMI is used to ensure:

- Certain file system objects, such as software images and driver packages, are synchronized.
- NCM tasks are scheduled and run on the correct NCM Core.

NCM does not replicate NCM Core specific options, including:

- TFTP server information
- TACACS server information
- Scripting language settings
- Hop Box definitions

- Driver directory
- Install directory
- Local Gateway information
- SWIM proxy server
- Active Directory settings
- Scheduler (concurrent task limits)

What is Horizontal Scalability?

Horizontal Scalability is the ability to combine multiple NCM Cores with a single database so that they work as a single logical unit to improve the performance of the overall system. NCM Horizontal Scalability can be configured in different ways, depending on how you use your system and where bottlenecks are apt to occur. For installation and configuration information, see the *Horizontal Scalability User Guide for CiscoWorks Network Compliance Manager*.

NCM Internal Monitoring: SQL Server Merge Replication

SQL Server Merge Replication can be configured to automatically handle certain types of conflicts. Where possible, NCM uses SQL Server's built-in conflict handling software. For example, for most tables with *update/update* conflicts, NCM installs the default SQL Server "latest timestamp wins" handler.

When it is not possible to use SQL Server's default handler, NCM typically has additional code inside the application with logic that attempts to correct the error. If the error is uncorrected through that logic, NCM alerts the NCM administrator to the issue using NCM's standard event notification emailing functionality. For information about configuring email notification, see the *User Guide for CiscoWorks Network Compliance Manager*.



When a second NCM Core is used as a standby for failover and not for normal operation, you should not encounter *update/update* conflicts.

For detailed information on NCM monitors, see [Distributed Monitor Results Page](#) on page xxxiv.

In addition, in SQL Server, NCM monitors on a regular schedule the following on each NCM Core for the database transport (i.e., replication):

- Merge agent status — If the check fails and the merge agent is found to have stopped, NCM will attempt to restart it.
- Merge agent delays — If the merge agent synchronization falls behind, NCM will alert the NCM Administrator via email notification.
- Unrecoverable uniqueness conflicts — NCM alerts the NCM Administrator in the event of unrecoverable uniqueness conflicts. Typically, there is no way to automatically correct these types of issues.



For information about configuring email notification, see the *User Guide for CiscoWorks Network Compliance Manager*.

NCM Internal Monitoring: Inter-NCM Core Communication

In addition to replication monitoring, NCM also monitors the following between each NCM Core:

- RMI connectivity
- NCM server timestamp differences

These monitors generate events during error conditions. The events can be emailed to the NCM administrator or SQL Server DBA using a standard event rule in NCM. For information about configuring email notification, see the *User Guide for CiscoWorks Network Compliance Manager*. For detailed information about NCM system events, see [NCM Generated Events](#) on page xxxii.



NCM includes an example event rule that can be updated as necessary.

2 Installation, Setup, and Upgrading

The chapter contains the following topics:

- [System Requirements](#) on page xi
- [Prepare for Initial Replication](#) on page xi
- [Creating a Two NCM Core SQL Server Replication Environment](#) on page xiv
- [Removing Replication](#) on page xx
- [Upgrade the NCM Mesh](#) on page xxii

For information about troubleshooting an NCM Mesh, see [Troubleshooting](#) on page xliii.

System Requirements

For information about database version and server hardware requirements for the NCM Multimaster Distributed System on SQL Server software, see the *Installation and Upgrade Guide for CiscoWorks Network Compliance Manager*.

Prepare for Initial Replication

To properly install the Distributed System software, you must first complete:

- Device partitioning planning across NCM Cores
- Network configuration planning for connectivity between NCM servers and devices. For example, what network connectivity is required to support failover for device access? The ability of an NCM Mesh to failover for device access depends in part on proper network setup to ensure access to devices. In some cases, you might not want to have failover work for complete device access, but instead have it ensure access to data while corrective action is taken to restore the network connectivity to the affected NCM Core.
- Network configuration planning for connectivity and bandwidth between the different servers (NCM and database) that comprise the NCM Mesh and between the NCM Management Engines and devices. The NCM Cores in the NCM Mesh will also need bandwidth between them equal to the bandwidth provided between a single NCM server and its database in a single NCM Core. For more information, see [Architecture](#) on page vii.
- Network configuration planning for bandwidth usage by the NCM Mesh. The bandwidth required between an NCM Management Engine and its database depends on the size of the deployment, including the number of devices, average configuration size, and the number of concurrent users. The bandwidth available between the databases should be as large as between the NCM Management Engine and the database. For a deployment of 14,000 devices with an average 10K configurations that change once a week, the bandwidth requirements could conservatively be on the order of 1Mb/s. Peak bandwidth requirements could be higher, depending on usage.

- SQL Server setup planning. Access to SQL Server Management Studio and the `sqlcmd` command line tool are needed for initial database creation.

Keep in mind that during replication setup, a snapshot of the initial database is transferred to each database in the NCM Mesh. This requires ample time, disk space, and bandwidth.

You can estimate the time it will take to copy data from server to server given the bandwidth between the servers. You can also calculate the disk space requirements for the export (and subsequent import) operations by knowing the size of your database. If you want to export or import data from the same server as the database, the disk space requirement is twice the size of your database.



Estimating time for the import and export operations could be difficult. You should allocate a lengthy time frame to complete this work. In addition, the NCM application servers must be offline during the export and import steps. A sufficiently long maintenance window should be planned for.

You will also have to ensure:

- Time synchronization setup for the NCM servers in the NCM Mesh
- Users are instructed to login to their “closest” NCM Core
- Access to a SQL Server DBA to support the NCM Distributed System installation

To assist in planning, please note the following limitations and suggestions concerning the NCM Distributed System:

- NCM only supports two NCM Cores in a Distributed System on SQL Server and can support no more than 6,500 nodes.
- NCM currently does not support joining multiple existing NCM installations into an NCM Mesh. You can only create an NCM Mesh from a single existing NCM server, adding a new NCM Core as appropriate. You can also create an NCM Mesh from scratch.
- Users should not share logon accounts. Due to the replication system used to share data across NCM Cores, two users should never use the same logon name to connect to two different NCM Cores at the same time. If they do so, the system will likely require additional work by the system administrator to ensure that the affected user’s profile is properly synchronized across the NCM Mesh.
- NCM currently assumes that all application and database servers in the Mesh share a single time zone.
- Future NCM upgrades will take longer and require more downtime due to the need to both update the replication setup and to update all servers in the NCM Mesh. NCM does not support rolling upgrades where one part of the NCM Mesh is running a version of NCM while the rest of the NCM Mesh is running a different version.
- The `SQLServerReplicationScript.sql` script updates the `RN_CORE` table. When the replication script runs, the `RN_CORE` table changes. There is no need for it to `UPDATE` and `INSERT` into the `RN_CORE` table on both Cores because replication is already running. The database changes will be pushed to the second database by using replication. The contents of the `RN_CORE` tables on both databases should match whatever is set up in the `SQLServerReplicationScript.sql` file.
- To successfully recover from the loss of the Publisher and Subscriber server, you must reconfigure replication by using existing NCM data. Refer to [Loss of a Database Server](#) on page xl for information on restoring databases.

Planning

This section includes information for initial replication setup on a Publisher and a Subscriber. NCM will be offline during initial replication. It is recommended that you carefully read the following sections so as to streamline the replication process. A new database instance on the Subscriber is manually created when running initial replication setup.

When initially setting up replication, complete the following table before starting. Be sure to save this information. You will need it each time you upgrade in the future.

Table 1 Distributed System Server Information

| Information to Collect | NCM Core 1 | NCM Core 2 |
|--------------------------------|--------------------------------------|---------------------------------------|
| Time Zone Offset from UTC | | Same for all servers in the NCM mesh |
| Application Server Data | NCM Application Server 1 | NCM Application Server 2 |
| NCM App Server FQDN | | |
| NCM App Server RMI Port | | |
| Database Server Data | Database Server 1 (Publisher) | Database Server 2 (Subscriber) |
| Database Server FQDN | | |
| SQL Server Listening Port | | |
| Database Name | | |
| Database User | | Same as for publisher |
| Database Password | | Same as for publisher |

Protocols, Databases, and Ports

NCM communicates with devices using a combination of the following protocols, databases, and ports. If you use a given protocol, NCM requires access to the corresponding port. Specifically, if NCM communicates with devices protected by firewalls, these ports must be open. For more information, see [Architecture](#) on page vii.

| Protocol/Database/Port | From/To |
|---|---|
| NCM server (running the Management Engine, Syslog, TFTP) and network devices | |
| Telnet (port 23) | From the NCM server to network devices. |
| SSH (port 22) | From the NCM server to network devices. |
| TFTP (port 69/udp) | From network devices to the NCM server. |
| Syslog (port 514/udp) | From network devices to the NCM server. |
| SNMP (port 161/udp) | From the NCM server to network devices. |
| Between the database servers | |
| SQL Server (port 1433) | In a Distributed System configuration, the SQL Server processes connect to each other on port 1433. |
| Between the NCM servers | |
| JNDI (ports 1098, 1099) | NCM server to NCM server. You can change this by editing the NCM configuration files. Please contact Customer Support for assistance. |

| Protocol/Database/Port | From/To |
|---|---|
| RMI (ports 4444 and 8083) | NCM server to NCM server. You can change this by editing the NCM configuration files. Please contact Customer Support for assistance. |
| Between the NCM server and the database server | |
| SQL Server (port 1433) | From the NCM server to a SQL Server database server. |
| NCM server and NCM users | |
| HTTPS (port 443) | From the NCM server to NCM users. You can change this by editing the NCM configuration files. Please contact Customer Support for assistance. |
| Telnet (port 23 - Windows or 8023 - Solaris/Linux) | From the NCM client to the NCM server. This can be changed from the Administrative Settings option. |
| SSH (port 22 - Windows or 8022 - Solaris/Linux) | From the NCM client to the NCM server. This can be changed from the Administrative Settings option. |

Unpacking the Replication Script Installation Bundle

The setup files for the NCM Multimaster Distributed System are the standard setup files for a normal NCM installation, with the addition of the `SQLServerReplicationScriptToolBundle.zip` file. This file should be unzipped onto a standard NCM application server. The setup files include a Java program that you run to generate SQL scripts for replication setup. This program uses the Java runtime that is available on the NCM application server.

Creating a Two NCM Core SQL Server Replication Environment

NCM can already be installed on one application server and database server (a single NCM core). Alternatively, you can complete the first NCM core installation as part of setting up initial replication.

This procedure uses the following identifiers:

- NCM application servers: `App1` and `App2`
- SQL Server database names:
 - `NCM1` on database server 1, the publisher
 - `NCM2` on database server 2, the subscriber
- NCM database user for both databases: `nauser`
- Password for the NCM database user: `napass`

To use a different identifier, substitute the actual value for each instance of the example value within the procedure.

To set up initial replication between two NCM cores, complete the following tasks:

- [Task 1: Verify Prerequisites](#)
- [Task 2: \(New Installations Only\) Create the NCM Schema on Database Server 1](#)

- [Task 3: Prepare for Replication](#)
- [Task 4: Configure Replication](#)

Task 1: [Verify Prerequisites](#)

Prepare two database servers and two application servers. For best performance, each application server-database server pair should be co-located. Additionally, the application server-application server link and the database server-database server link should each be a high-speed connection.

Verify that the following prerequisites have been met:

- The two database servers are running a supported version of SQL Server. Both database servers must use the same version of the database software.
 - Identify the user name (for example, `sa`) and password of a SQL Server user account with the `sysadmin` role on both database servers.
 - Ensure `sqlcmd` access to both database servers.

You will need to supply credentials for a database user account that is a member of the `sysadmin` role when running the scripts.
- The SQL Server agent service is running on database server 1, the publisher.
 - Identify the user name and password of a Windows user account under which SQL Server agents can run.
- The two application servers that will run NCM have working hostnames. Note the following:
 - Each application server should have a high-speed connection to one of the database servers.
 - For application server hardware and operating system requirements, see the *Installation and Upgrade Guide for CiscoWorks Network Compliance Manager*.
- The host names of both database servers and both application servers are in the `hosts` file on each application server. This file is located as follows:
 - *Windows*: `<Drive>:\Windows\System32\drivers\etc\hosts`
 - *UNIX*[®]: `/etc/hosts`
- All database servers and application servers are set to use the same time and time zone.

Task 2: [\(New Installations Only\) Create the NCM Schema on Database Server 1](#)

If NCM has not yet been installed, create the NCM schema on database server 1 by following these steps:

- 1 Install NCM on the first application server (`App1`).

During installation, supply the following database credentials:

 - For the database IP address or hostname, enter the fully-qualified domain name of database server 1.
 - For the database port, enter the port on database server 1 that NCM will use to communicate with the database.
 - For the database name, enter the name of the NCM database on the database server 1 (for example, `NCM1`).
 - For the database user name and password, enter a name and password for accessing the NCM database, for example, `nauser` and `napass`.
- 2 After NCM installation is complete, log on to NCM (as user `nauser`) to ensure that it works as expected.

If NCM does not work correctly, follow the steps in [Recovering from a Failed Replication Configuration](#) on page xx.

- 3 Stop the NCM management engine on App1:


Task 3: Prepare for Replication

To configure replication between the publisher and the subscriber databases, follow these steps:

- 1 Stop the NCM management engine on App1:
 - **Windows:** Open the **Services** control panel. In the list of services, right-click **TrueControl ManagementEngine**, and then click **Stop**.
 - **UNIX:** Run the following command:

```
/etc/init.d/truecontrol stop
```
- 2 If the NCM database contains working data (is not a new installation), back up the database.

Alternatively, if you are recovering from a failed replication configuration and had working data before starting configuration, restore the database from backup at this time.
- 3 On database server 1, create a shared directory (for example, C:\NCM_Share).
 - Set the sharing permissions so that both database servers have read-write access to this directory.
 - Also set the sharing permissions so that the administrative user on App1 (or other system) who will run the Java script in [step 2 of Task 4](#) has read-write access to this directory.
 - Ensure that the directory has at least 50MB available disk space. If the NCM database contains working data, this requirement might increase.
- 4 On database server 2, create an empty database. Note the following:
 - Use a default database instance. (NCM replication does not support a named instance.)
 - Use both Windows and SQL Server authentication.
 - Enable full-text indexing.
 - The name of this database must be *different* from the name of the first NCM database, for example, NCM2.
 - In all other aspects, such as privileges and owner, this database configuration must be the same as the configuration of the first database.
- 5 On database server 1, unpack the `SQLServerReplicationScriptToolBundle.zip` file to a known location with low security, for example, C:\tmp or /tmp.

- 6 From the known location of [step 5](#), customize the `SQLServerReplicationScriptTool.properties` file for your environment:
 - a In a text editor such as WordPad or vi, open the `SQLServerReplicationScriptTool.properties` file.
 - b Search for and replace the text `README` with the appropriate value for that variable.
 - Use short hostnames to identify the database servers.
 -  Verify DNS resolution of the database server hostnames to IP addresses.
 - Use fully-qualified hostnames to identify the application servers and the replication directory.
 - Set `windows.username.0` to either `<machineName>\\<administrator>` or `<domain>\\<domainUser>`. If you use the first format, ensure that the administrator can access the shared directory created in [step 3](#) of this task.
 - For example:


```
db.server.0=db1
db.name.0=NCM1
db.username.0=nauser
db.password.0=napass
windows.username.0=db1\\Administrator
windows.password.0=adminpass
db.tableowner.0=dbo
db.port.0=1433
app.server.0=App1.com
app.rmiport.0=1099
app.corename.0=Core1

replication.data.dir=\\db1.example.cisco.com\\NCM_Share

db.server.1=db2
db.name.1=NCM2
db.port.1=1433
app.server.1=App2.example.com
app.rmiport.1=1099
app.corename.1=Core2
```
 - c Set `timezone_offset` to the value that matches the time zone setting for all servers in the distributed system.
 - d Verify that `mode` is set to `initial`.

Task 4: Configure Replication

- 1 From the known location of [step 5](#) of [Task 3](#), run `java -version` to verify that Java is version 1.6 or later.
- 2 From the known location of [step 5](#) of [Task 3](#), run the `ReplicationScriptTool`:


```
java -classpath . SQLServerReplicationScriptTool
```


This command creates two scripts for starting replication. By default, the scripts are named `SQLServerPreSnapshotScript.sql` and `SQLServerReplicationScript.sql`.



The following error message can mean that Java does not have permission to run the replication tool:

```
Could not find the main class: ReplicationScriptTool.
```

Move all files from the replication bundle to a directory with lower security, and then run the `java` command from that directory.

- 3 Copy the `SQLServerPreSnapshotScript.sql` script to the shared directory created in [step 3 of Task 3](#).
- 4 Copy the `SQLServerReplicationScript.sql` script to a known location on database server 1, for example, `C:\tmp`.
 - Set the sharing permissions so that the SQL Server `sysadmin` user account has read-write access to this directory.
 - Verify that `sqlcmd` is accessible from this directory.

- 5 From the known location on database server 1 ([step 5](#) of this task), run the `SQLServerReplicationScript.sql` script by using the `sqlcmd` command:

```
sqlcmd -S <Publisher> -U <User> -P <Password>
-i SQLServerReplicationScript.sql -o log.txt
```

For `<Publisher>`, use the short hostname of database server 1, for example `db1`.

For `<User>` and `<Password>`, use a SQL Server user account with the `sysadmin` role, for example, `sa`.

This command creates a directory named `UNC` in the shared directory created in [step 3 of Task 3](#). It also populates the `UNC` directory with a snapshot of the NCM database on database server 1.

- 6 Verify that the new `UNC` directory contains the NCM schema.

If the `UNC` directory does not contain the NCM schema, follow the steps in [Recovering from a Failed Replication Configuration](#) on page xx.

- 7 Finish configuring replication by entering the following Transact-SQL queries on the appropriate database servers:



If you copy the select statement from this PDF and paste it to the `sqlcmd` line, re-type the single quotation mark characters (`'`) on the command line.

- Publisher:

```
sqlcmd -S <Publisher> -U <User> -P <Password>
use <DatabaseName>
Select count(*) from Information_Schema.Tables where Table_Type =
'Base Table' and Table_Name like 'RN_%'
go
```

For `<Publisher>`, use the short hostname of database server 1, for example `db1`.

For `<User>` and `<Password>`, use a SQL Server user account with the `sysadmin` role, for example, `sa`.

`<DatabaseName>` is the name of the NCM database on database server 1, for example `NCM1`.

- Subscriber:

```
sqlcmd -S <Subscriber> -U <User> -P <Password>
use <DatabaseName>
Select count(*) from Information_Schema.Tables where Table_Type =
'Base Table' and Table_Name like 'RN_%'
go
```

For <Subscriber>, use the short hostname of database server 2, for example db2.

For <User> and <Password>, use a SQL Server user account with the sysadmin role, for example, sa.

<DatabaseName> is the name of the NCM database on database server 2, for example NCM2.

- Verify that the previous step completed correctly.

The query results should be the same on both database servers. For example:

```
Changed database context to 'NCM'.
-----
          79
(1 rows affected)
```

If the query results for both database servers are not identical, follow the steps in [Recovering from a Failed Replication Configuration](#) on page xx.

- Verify that the replication is working correctly. Make sure:

- The log file generated by running the replication SQL script contains no errors.
- In Microsoft SQL Management Studio, examine the NCM databases:
 - Under Databases, look for the database name.
 - Verify that the table count is the same for each database.
 - Verify that the RN_CORE table on each database contains the appropriate list of servers in the NCM mesh.
 - Verify that the replication of data is working correctly by adding a comment (for example, Test from db1 to db2) to the CoreID=1 row of the RN_CORE table for the publisher database. After a minute, this comment should be visible for the same table row of the subscriber database.

If replication is not working correctly, follow the steps in [Recovering from a Failed Replication Configuration](#) on page xx.

- Install NCM on the second application server (App2).


During installation, connect NCM to the database on Master 2. When prompted for the database credentials, choose: use existing database.

- Stop the NCM management engine on App2:

- Windows:** Open the **Services** control panel. In the list of services, right-click **TrueControl ManagementEngine**, and then click **Stop**.
- UNIX:** Run the following command:


```
/etc/init.d/truecontrol stop
```

- 12 Copy the `distributed.rcx` file from the known location of [step 5](#) to the following directory on both application servers, App1 and App2:
 - *Windows*: `<NCM_HOME>\jre`
 - *UNIX*: `<NCM_HOME>/jre`
- 13 Start the NCM management engine on both application servers, App1 and App2:
 - *Windows*: Open the **Services** control panel. In the list of services, right-click **TrueControl ManagementEngine**, and then click **Start**.
 - *UNIX*: Run the following command:

```
/etc/init.d/truecontrol restart
```
- 14 Connect to the NCM console on both application servers, App1 and App2, (as user `nauser`), and ensure that there are no issues.
 -  An RMI error message at this point can mean that NCM is unable to identify the localhost. For more information, see [RMI Errors in the NCM Console](#) on page xliii.
- 15 Verify the NCM installation:
 - a Edit an object on one NCM server (for example, a Comments field for a device).
 - b Wait a minute and then verify that the updated comment exists on the second server.
 - c Check the status of the Oracle Distributed Monitor in the NCM console to ensure that no problems are being reported. (For information, see [Distributed Monitor Results Page](#) on page xxxiv.) It could take up to five minutes for this monitor to initially run.
- 16 *Optional*. Delete the shared directory created in [Task 3: Prepare for Replication](#) on page xvi (for example, `C:\NCM_data`).

Recovering from a Failed Replication Configuration

If you see errors during replication setup, follow these steps:

- 1 Uninstall NCM from both application servers.
- 2 Delete the NCM databases from both database servers.
- 3 Redo the replication configuration process starting with [Task 2: \(New Installations Only\) Create the NCM Schema on Database Server 1](#) on page xv.

Removing Replication

When deleting a subscriber from a NCM Mesh, NCM Core 1 is the Publisher and NCM Core 2 is the Subscriber to remove.

System Requirements

You must have the following items configured before deleting a Subscriber from a NCM Mesh:

- `sqlcmd` access to both database servers in the mesh.
- The `SQLServerReplicationScriptTool` application installed on a Java-capable system.

Deletion Steps

To delete the Subscriber from an NCM Mesh:

1 Collect the following information:

- Login name and password of a SQL Server login that is a member of sysadmin on both database servers in the mesh.
- Database name, NCM server hostname, NCM server RMI listening port, database hostname, and database listening port for NCM Core 1 and NCM Core 2.

2 Ensure that all devices in NCM belong to partitions on NCM Cores that are not going to be removed.

3 Modify all partitions to point to the publisher. Alternatively, remove those partitions.

4 Stop the NCM management engine on the subscriber application server:

- *Windows*: Open the **Services** control panel. In the list of services, right-click **TrueControl ManagementEngine**, and then click **Stop**.
- *UNIX*: Run the following command:

```
/etc/init.d/truecontrol stop
```

5 Delete the RN_CORE entry for the subscriber using the following commands using sqlcmd:

```
UPDATE RN_SITE SET OwningCoreID = 1 WHERE OwningCoreID = <coreID>;
UPDATE RN_SITE SET ManagingCoreID = 1 WHERE ManagingCoreID = <coreID>;
UPDATE RN_SCHEDULE_TASK SET CoreID = 1 WHERE CoreID = <coreID>;
DELETE FROM RN_CORE WHERE CoreID = <coreID>;
```



To identify the Core ID of the NCM Core to delete, enter:

```
SELECT * FROM RN_CORE;
```

6 Update the variables for NCM Core 2 in the `SQLServerReplicationScriptTool.properties` file.

- Make sure that the variables for NCM Core 1 are correct.
- Set mode to `delete_server`.

These properties are described in detail in the file.

7 From the replication bundle directory, run the `ReplicationScriptTool`:

```
java -classpath . SQLServerReplicationScriptTool
```

8 Copy the `SQLServerReplicationScript.sql` script to a known location on the publisher database server, for example, `C:\tmp`.

- Set the sharing permissions so that the SQL Server sysadmin user account has read-write access to this directory.
- Verify that `sqlcmd` is accessible from this directory.

9 From the known location on database server 1 (step 8 of this task), run the `SQLServerReplicationScript.sql` script by using the `sqlcmd` command:

```
sqlcmd -S <Publisher> -U <User> -P <Password>
-i SQLServerReplicationScript.sql -o log.txt
```

For `<Publisher>`, use the short hostname of database server 1, for example `db1`.

For `<User>` and `<Password>`, use a SQL Server user account with the sysadmin role, for example, `sa`.

- 10 Remove the `distributed.rcx` file from each NCM application server.
- 11 Start the NCM management engine on NCM Core 1:
 - *Windows*: Open the **Services** control panel. In the list of services, right-click **TrueControl ManagementEngine**, and then click **Start**.
 - *UNIX*: Run the following command:

```
/etc/init.d/truecontrol restart
```
- 12 On the publisher, create a script with the following contents:

```
use [master]
exec sp_dropdistributor @no_checks = 1
GO
```

Upgrade the NCM Mesh

During the NCM upgrade process, the NCM Cores must be completely offline. You will be upgrading the:

- NCM application servers
- NCM database schema
- Replication between the database cores

Note the following:

- The NCM mesh upgrade path is as follows:

(NCM 1.3 or NCM 1.3.01) to NCM 1.4 to NCM 1.5 to NCM 1.6 to NCM 1.7 to NCM 1.8

Enter this path at your current version of NCM.
- At each step on the NCM mesh upgrade path, both NCM cores in the NCM mesh before moving to the next step in the upgrade path.
- For instructions for upgrading from NCM 1.2 to NCM 1.3, see the *Configuration Guide for High Availability Distributed System on Oracle*.


For upgrade instructions, see the following topics:

- [Upgrading from NCM 1.3/1.3.01 to NCM 1.4](#) on page xxiii
- [Upgrading from NCM 1.4x to NCM 1.5](#) on page xxiv
- [Upgrading from NCM 1.5 to NCM 1.6](#) on page xxv
- [Upgrading from NCM 1.6 to NCM 1.7](#) on page xxvi
- [Upgrading from NCM 1.7 to NCM 1.8](#) on page xxvii

Upgrading from NCM 1.3/1.3.01 to NCM 1.4

To upgrade from NCM 1.3/1.3.01 to NCM 1.4, be sure that:

- You have `sqlcmd` access to NCM Core 1 and NCM Core 2.
 - The *ReplicationScriptTool* bundle files for NCM 1.4 are installed on a Java-capable system (on one of the NCM systems).
- 1 Shut down both NCM Management Engines in the NCM Mesh.
 - 2 Collect the following information:
 - Login name and password of a SQL Server login that is a member of `sysadmin` on NCM Core 1 and NCM Core 2.
 - Database name, NCM server hostname, NCM server RMI listening port, database hostname, and database listening port for NCM Core 1 and NCM Core 2.
 - 3 Update the variables for the publisher and subscriber in the `SQLServerReplicationScriptTool.properties` file. Ensure that you modify all “REPLACEME” text entries in the `SQLServerReplicationScriptTool.properties` file with the appropriate values for your environment.

 All application and database servers should be listed in the `SQLServerReplicationScriptTool.properties` file. Make sure the `mode` property is set to the appropriate value for the upgrade you want to perform. For example, if you are upgrading from NCM 1.3 to NCM 1.4, this value should be set to “`upgrade_from_1_3`”. If you are upgrading from NCM 1.3.01 to NCM 1.4, this value should be set to “`upgrade_from_1_3_01`”.

- 4 Run the *ReplicationScriptTool*. Enter:

```
java -classpath . SQLServerReplicationScriptTool
```

- 5 Run the first output file using `sqlcmd` with a login that is a member of the `sysadmin` role on NCM Core 1. Enter:

```
sqlcmd -S <PublisherServer> -U <user> -P <password> -i
SQLServerReplicationScript.sql -o log.txt
```

Where `<PublisherServer>` is the server of Publisher, `<user>` and `<password>` were collected in [step 2](#).

- 6 Verify that the replication is working correctly. Make sure:


- The log file generated by running the replication SQL script contains no errors.
- In Microsoft SQL Management Studio, examine the NCM databases:
 - Under Databases, look for the database name.
 - Verify that the table count is the same for each database.
 - Verify that the `RN_CORE` table on each database contains the appropriate list of servers in the NCM mesh.
 - Verify that the replication of data is working correctly by adding a comment (for example, `Test from db1 to db2`) to the `CoreID=1` row of the `RN_CORE` table for the publisher database. After a minute, this comment should be visible for the same table row of the subscriber database.

- 7 Upgrade each NCM server using the NCM 1.4 Service Pack Installer. For information about running the NCM 1.4 Service Pack Installer, see the *Installation and Upgrade Guide for CiscoWorks Network Compliance Manager 1.7*.
- 8 Restart the NCM servers in the NCM Mesh.

Upgrading from NCM 1.4x to NCM 1.5

To upgrade from 1.4x to NCM 1.5, be sure that:

- You have `sqlcmd` access to NCM Core 1 and NCM Core 2.
 - The *ReplicationScriptTool* bundle files for NCM 1.5 are installed on a Java-capable system (on one of the NCM systems).
- 1 Shut down both NCM Management Engines in the NCM Mesh.
 - 2 Collect the following information:
 - Login name and password of a SQL Server login that is a member of `sysadmin` on NCM Core 1 and NCM Core 2.
 - Database name, NCM server hostname, NCM server RMI listening port, database hostname, and database listening port for NCM Core 1 and NCM Core 2.
 - 3 Update the variables for the publisher and subscriber in the `SQLServerReplicationScriptTool.properties` file. Ensure that you modify all “REPLACEME” text entries in the `SQLServerReplicationScriptTool.properties` file with the appropriate values for your environment.

 All servers should be listed in the `SQLServerReplicationScriptTool.properties` file. Set the mode property to “`upgrade_from_1_4`”.

- 4 Run the `ReplicationScriptTool`. Note that you must run this on a system that has Java 1.6.x installed. Enter:
- 5 Run the first output file using `sqlcmd` with a login that is a member of the `sysadmin` role on NCM Core 1. Enter:

```
java -classpath . SQLServerReplicationScriptTool
```

```
sqlcmd -S <PublisherServer> -U <user> -P <password> -i  
SQLServerReplicationScript.sql -o log.txt
```

Where `<PublisherServer>` is the server of Publisher, `<user>` and `<password>` were collected in [step 2](#).

- 6 Verify that the replication is working correctly. Make sure:
 - The log file generated by running the replication SQL script contains no errors.
 - In Microsoft SQL Management Studio, examine the NCM databases:
 - Under Databases, look for the database name.
 - Verify that the table count is the same for each database.
 - Verify that the `RN_CORE` table on each database contains the appropriate list of servers in the NCM mesh.
 - Verify that the replication of data is working correctly by adding a comment (for example, `Test from db1 to db2`) to the `CoreID=1` row of the `RN_CORE` table for the publisher database. After a minute, this comment should be visible for the same table row of the subscriber database.


- 7 Upgrade each NCM server using the NCM 1.5 Service Pack Installer. For information about running the NCM 1.5 Service Pack Installer, see the *Installation and Upgrade Guide for CiscoWorks Network Compliance Manager 1.7*.
- 8 Restart the NCM servers in the NCM Mesh.

Upgrading from NCM 1.5 to NCM 1.6

To upgrade from 1.5 to NCM 1.6, be sure that:

- You have `sqlcmd` access to NCM Core 1 and NCM Core 2.
 - The *ReplicationScriptTool* bundle files for NCM 1.6 are installed on a Java-capable system (on one of the NCM systems).
- 1 Shut down both NCM Management Engines in the NCM Mesh.
 - 2 Collect the following information:
 - Login name and password of a SQL Server login that is a member of `sysadmin` on NCM Core 1 and NCM Core 2.
 - Database name, NCM server hostname, NCM server RMI listening port, database hostname, and database listening port for NCM Core 1 and NCM Core 2.

- 3 Update the variables for the publisher and subscriber in the `SQLServerReplicationScriptTool.properties` file. Ensure that you modify all “REPLACEME” text entries in the `SQLServerReplicationScriptTool.properties` file with the appropriate values for your environment.

 All servers should be listed in the `SQLServerReplicationScriptTool.properties` file. Set the mode property to “`upgrade_from_1_5`”.

- 4 Run the `ReplicationScriptTool`. Note that you must run this on a system that has Java 1.6.x installed. Enter:

```
java -classpath . SQLServerReplicationScriptTool
```

- 5 Run the first output file using `sqlcmd` with a login that is a member of the `sysadmin` role on NCM Core 1. Enter:

```
sqlcmd -S <PublisherServer> -U <user> -P <password> -i
SQLServerReplicationScript.sql -o log.txt
```


Where `<PublisherServer>` is the server of Publisher, `<user>` and `<password>` were collected in [step 2](#).

- 6 Verify that the replication is working correctly. Make sure:
 - The log file generated by running the replication SQL script contains no errors.
 - In Microsoft SQL Management Studio, examine the NCM databases:
 - Under Databases, look for the database name.
 - Verify that the table count is the same for each database.
 - Verify that the `RN_CORE` table on each database contains the appropriate list of servers in the NCM mesh.
 - Verify that the replication of data is working correctly by adding a comment (for example, `Test from db1 to db2`) to the `CoreID=1` row of the `RN_CORE` table for the publisher database. After a minute, this comment should be visible for the same table row of the subscriber database.

- 7 Upgrade each NCM server using the NCM 1.6 Service Pack Installer. For information about running the NCM 1.6 Service Pack Installer, see the *Installation and Upgrade Guide for CiscoWorks Network Compliance Manager 1.7*.
- 8 Restart the NCM servers in the NCM Mesh.

Upgrading from NCM 1.6 to NCM 1.7

To upgrade from NCM 1.6 to NCM 1.7, be sure that:

- You have `sqlcmd` access to NCM Core 1 and NCM Core 2.
 - The *ReplicationScriptTool* bundle files for NCM 1.7 are installed on a Java-capable system (on one of the NCM systems).
- 1 Shut down both NCM Management Engines in the NCM Mesh.
 - 2 Collect the following information:
 - Login name and password of a SQL Server login that is a member of `sysadmin` on NCM Core 1 and NCM Core 2.
 - Database name, NCM server hostname, NCM server RMI listening port, database hostname, and database listening port for NCM Core 1 and NCM Core 2.
 - 3 Update the variables for the publisher and subscriber in the `SQLServerReplicationScriptTool.properties` file. Ensure that you modify all “REPLACEME” text entries in the `SQLServerReplicationScriptTool.properties` file with the appropriate values for your environment.
 -  All servers should be listed in the `SQLServerReplicationScriptTool.properties` file. Set the mode property to “`upgrade_from_1_6`”.
 - 4 Run the `ReplicationScriptTool`. Note that you must run this on a system that has Java 1.6.x installed. Enter:

```
java -classpath . SQLServerReplicationScriptTool
```

- 5 Run the first output file using `sqlcmd` with a login that is a member of the `sysadmin` role on NCM Core 1. Enter:


```
sqlcmd -S <PublisherServer> -U <user> -P <password> -i  
SQLServerReplicationScript.sql -o log.txt
```

Where `<PublisherServer>` is the server of Publisher, `<user>` and `<password>` were collected in [step 2](#).

- 6 Verify that the replication is working correctly. Make sure:
 - The log file generated by running the replication SQL script contains no errors.
 - In Microsoft SQL Management Studio, examine the NCM databases:
 - Under Databases, look for the database name.
 - Verify that the table count is the same for each database.
 - Verify that the RN_CORE table on each database contains the appropriate list of servers in the NCM mesh.
 - Verify that the replication of data is working correctly by adding a comment (for example, Test from db1 to db2) to the CoreID=1 row of the RN_CORE table for the publisher database. After a minute, this comment should be visible for the same table row of the subscriber database.
- 7 Upgrade each NCM server using the NCM 1.7 Service Pack Installer. For information about running the NCM 1.7 Service Pack Installer, see the *Installation and Upgrade Guide for CiscoWorks Network Compliance Manager 1.7*.
- 8 Restart the NCM servers in the NCM Mesh.

Upgrading from NCM 1.7 to NCM 1.8

To upgrade from NCM 1.7 to NCM 1.8, be sure that:

- You have `sqlcmd` access to NCM Core 1 and NCM Core 2.
 - The *ReplicationScriptTool* bundle files for NCM 1.8 are installed on a Java-capable system (on one of the NCM systems).
- 1 Shut down both NCM Management Engines in the NCM Mesh.
 - 2 Collect the following information:
 - Login name and password of a SQL Server login that is a member of `sysadmin` on NCM Core 1 and NCM Core 2.
 - Database name, NCM server hostname, NCM server RMI listening port, database hostname, and database listening port for NCM Core 1 and NCM Core 2.
 - 3 Update the variables for the publisher and subscriber in the `SQLServerReplicationScriptTool.properties` file. Ensure that you modify all “REPLACEME” text entries in the `SQLServerReplicationScriptTool.properties` file with the appropriate values for your environment.
 -  All servers should be listed in the `SQLServerReplicationScriptTool.properties` file. Set the `mode` property to “`upgrade_from_1_7`”.
 - 4 Run the `ReplicationScriptTool`. Note that you must run this on a system that has Java 1.6.x installed. Enter:


```
java -classpath . SQLServerReplicationScriptTool
```

- 5 Run the first output file using `sqlcmd` with a login that is a member of the `sysadmin` role on NCM Core 1. Enter:

```
sqlcmd -S <PublisherServer> -U <user> -P <password> -i
SQLServerReplicationScript.sql -o log.txt
```

Where `<PublisherServer>` is the server of Publisher, `<user>` and `<password>` were collected in [step 2](#).

- 6 Verify that the replication is working correctly. Make sure:
 - The log file generated by running the replication SQL script contains no errors.
 - In Microsoft SQL Management Studio, examine the NCM databases:
 - Under Databases, look for the database name.
 - Verify that the table count is the same for each database.
 - Verify that the `RN_CORE` table on each database contains the appropriate list of servers in the NCM mesh.
 - Verify that the replication of data is working correctly by adding a comment (for example, `Test from db1 to db2`) to the `CoreID=1` row of the `RN_CORE` table for the publisher database. After a minute, this comment should be visible for the same table row of the subscriber database.
- 7 Upgrade each NCM application server as described in the “Upgrading to NCM 1.8 from a Different System” chapter or the “Upgrading to NCM 1.8 on the Same System” chapter of the *Installation and Upgrade Guide for CiscoWorks Network Compliance Manager*, as appropriate.

Run the upgrade procedures in parallel. That is, complete step 1 on each NCM application server before initiating step 2 on any NCM application server, and so forth. Follow the referenced procedure through the step to run the NCM 1.8 Service Pack Installer.

Note the following:

- To prevent database access, on each NCM application server, after the NCM 1.8 Service Pack Installer runs, stop all NCM services.
 - *Windows*: Open the **Services** control panel. In the list of services, right-click each of the following services, and then click **Stop**:
 - TrueControl ManagementEngine**
 - TrueControl FTP Server**
 - TrueControl SWIM Server**
 - TrueControl Syslog Server**
 - TrueControl TFTP Server**
 - *UNIX*: Run the following command:

```
/etc/init.d/truecontrol stop
```
- The NCM database upgrade has already been handled by running the replication script in [step 4](#).
- Do *not* upgrade SQL Server while upgrading NCM.
- If you are moving NCM to different application servers, copy the NCM-specific files (the NCM directory and, on UNIX, the startup files) from one existing NCM application server to only one new NCM application server.

For example, consider a two-core Distributed System environment. The existing NCM application servers are Server X1 and Server X2. The new NCM application servers are Server Y1 and Server Y2. Do the following:

- Copy the NCM directory from Server X1 to Server Y1.
Retain the permissions on the files.
On UNIX systems, also copy the `/etc/truecontrol` and `/etc/init.d/truecontrol` files from Server X1 to Server Y1.
 - Copy the NCM directory from Server X2 to Server Y2.
Retain the permissions on the files.
On UNIX systems, also copy the `/etc/truecontrol` and `/etc/init.d/truecontrol` files from Server X2 to Server Y2.
- 8 If you moved NCM to different application servers in [step 7](#), in Microsoft SQL Management Studio update the `RN_CORE` table on one database with the fully-qualified domain names of the new NCM application servers.
 - 9 Restart the NCM servers in the NCM Mesh.
 - 10 Complete the upgrade procedure from the step to run the Data Pruning task.

3 System Administration

This chapter contains the following topics:

- [Getting Started](#) on page xxxi
- [How NCM Removes In-Memory and Database Information](#) on page xxxi
- [Suspending Replication for Database Maintenance](#) on page xxxii
- [NCM Generated Events](#) on page xxxii
- [Using the NCM Distributed System Pages](#) on page xxxiv

Getting Started

In general, an NCM application server that is part of a Distributed System NCM Mesh should be transparent to users. However, there are a number of operations that the system administrator might need to do to keep the Distributed System NCM Mesh functioning properly.

How NCM Removes In-Memory and Database Information

There could be a delay in removing or cancelling a task because NCM must communicate with the appropriate owning NCM Core to perform this operation. To ensure proper removal of all in-memory and database information, and to avoid replication conflicts due to matching timestamps, NCM active tasks are automatically deleted on the NCM Core with which they are associated. An active task is any task that does not have the “Succeeded,” “Failed,” “Duplicate”, “Skipped”, or “Warning” status.

Keep in mind that deleted tasks could be displayed in task lists for a few moments while the replication process pushes the deletes to other NCM Cores in the system. In addition, if the NCM Core from which the task originated is not accessible, the delete will fail.

Suspending Replication for Database Maintenance

To suspend replication because you are bringing down one or both database servers for maintenance, complete the following procedure:

- 1 Shut down the NCM management engine on both NCM application servers:
 - **Windows:** Open the **Services** control panel. In the list of services, right-click **TrueControl ManagementEngine**, and then click **Stop**.
 - **UNIX:** Run the following command:

```
/etc/init.d/truecontrol stop
```
- 2 Disable database replication by using the Disable Publishing and Distribution Wizard in SQL Server Management Studio. For information, see the Microsoft TechNet Library:
 - SQL Server 2005: *How to Disable SQL Server Database Replication* at <http://technet.microsoft.com/en-us/library/bb693954.aspx>
 - SQL Server 2008: *How to: Disable Publishing and Distribution (Replication Transact-SQL Programming)* at <http://technet.microsoft.com/en-us/library/ms147921.aspx>

NCM Generated Events

By default, NCM generates system events. Event rules can alert you to certain error conditions requiring attention. Each event is listed below, along with an explanation and required action to be taken. You should examine the default “Distributed System” event rule to ensure all of the events are included in the event rule and that the event rule is configured to send the email notification to the appropriate administrator.



For information about configuring email notification, see the *User Guide for CiscoWorks Network Compliance Manager*.

Distributed System - Uniqueness Conflict

Event format:

```
rowguid: <the guid of the database row that had the conflict>
origin_datasource: <database server>.<database name>
reason_text: <a description of why the conflict occurred>
conflict_type: <type of conflict according to SQLServer>
reason_code: <error message from SQLServer, depends on the type of
conflict>
repl_create_time: <time the conflict was generated by SQLServer>
```

```
conflict_table: <where SQLServer stores the conflicting data>
dataTable: <NCM table that contains the conflicting data>
SQLServerConflictID: <ID of the conflict recorded by NCM>
status: <status>
```

Conflicting Data: <the columns that are conflicting>

Refer to the *SQL Server Replication* documentation for instructions on correcting this conflict.

This event is sent when NCM detects a conflict in a uniqueness constraint. You will receive an event per NCM Core, since the conflicts are local to each NCM Core. To correct a naming conflict, go to one NCM Core and update the names for the affected objects. Both the renamed <NAME>.<SID> and <NAME> should be edited to force an update on the other NCM Cores.

To correct a rule priority conflict, go to the Device Password Rule Priority Reset page click the Reset Priority button. See [Device Password Rule Priority Reset Page](#) on page xxxvii.

If this does not solve the problem, you will need to manually edit the rules on each NCM Core, setting the priority order correctly and verifying existence of correct rules. When finished, return to the Device Password Rule Priority Reset page and click the Reset Priority button.

Distributed System - Time Synchronization Warning

Event format:

```
Time difference: <N> seconds
Local Core: <hostname>
Remote Core: <hostname>
```

NCM replication conflict resolution depends on a latest timestamp method. To work correctly, this requires different NCM servers' clocks to differ by only a small amount. To correct this problem, make sure that the time is synchronized on the NCM server systems across the NCM Mesh.

Distributed System - RMI Error

Event format:

```
Local Core: <hostname>
Remote Core: <hostname>
Error: <Exception text>
```

This error typically occurs when there are network problems between the NCM servers. To troubleshoot this problem, make sure:

- 1 The host that the server cannot connect to is up and running.
- 2 The NCM instance on that host is running.
- 3 From a command line, enter `ping <host>` to ensure that network connectivity exists between servers.
- 4 From a command line, enter `telnet <host> to port 1099` (or whatever your RMI listen port is set to) to ensure that RMI connections are being accepted. If working correctly, you should get back some data that includes the text string "java.rmi.MarshalledObject".

Failures of any of these steps will point to corrective actions needed, such as updating the RMI port being used in the Edit NCM Core page, or restarting NCM to make sure that the RMI port has been bound correctly and is not being used by another application.

Distributed System - Stopped Merge Agent Job

Event format:

```
Merge Agent Job Stopped
<Job Details>
[Successfully attempted to restart the job.]
```

NCM monitors the replication merge agent and sends the event if it determines that the agent has stopped. If the “Successfully attempted to restart the job” message is displayed, NCM restarted the agent. The agent’s status can be reviewed in SQL Server Management Studio.

Distributed System - Data Synchronization Delay Warning

Event format:

```
Publisher '<DatabaseName>' has been unable to push changes
to subscriber '<DatabaseName2>' since <tim>. If you wish to
avoid potential data loss, please rectify this as soon as
possible.
```

```
Subscriber '<DatabaseName2>' has been unable to push changes
to Publisher '<DatabaseName>' since <tim>. If you wish to
avoid potential data loss, please rectify this as soon as
possible.
```

NCM has detected that the replication system is having trouble keeping the data and the databases in sync. Examine the databases, merge agents, and network connections between the databases for issues.

Using the NCM Distributed System Pages

When you install the Distributed System software, the NCM user interface includes specific Distributed System pages to help you monitor and administer the system.

Distributed Monitor Results Page

The Distributed Monitor Results page displays the overall health of the Distributed System. By default, the Distributed monitor runs every five minutes.

To open the Distributed Monitor Results page, on the menu bar under Admin select Distributed and click Monitor Results. The Distributed Monitor Results page opens.

NCM monitor several properties necessary for proper functioning of the Distributed System, including:

- **RMI Connections** — RMI (Remote Method Invocation) is Java's remote procedure call protocol. The distributed system makes RMI calls between NCM servers in the NCM Mesh to transfer information about scheduled tasks, system settings, software images, and so on.
- **Uniqueness Conflicts** — Certain NCM database constraints restrict columns to unique values. In a distributed environment, these constraints can be violated when updates are made on two different NCM Cores where the unique column is set to the same value. These conditions are captured by the Replication Conflict Resolution System and logged. NCM cannot automatically resolve these conflicts. They must be resolved manually.

- **Merge Agents** — Merge Agents are the processes at the Publisher that handle transferring replicated data. NCM monitors the SQL Server jobs that schedule these processes. If for some reason the process stops, NCM reports that here. Stopped Merge Agent jobs should be restarted as soon as possible.
- **Local NCM Core Definition** — The local NCM Core must be able to determine which entry in the RN_CORE table it is. If the “The *local core for this system is undefined.*” error message is displayed, the CoreHostname property needs to be updated for the NCM Core. This can be done using the Edit Core page. For information, see [Edit Core Page](#) on page xxxvii.

▶ When this condition occurs, the NCM Management Engine’s log file will contain the following text: “Fatal error - could not assign local core.”

The CoreHostname value can be either the DNS, *etc/hosts* value, or an IP address. If you are using an NCM server with multiple IP addresses, you might need to tell NCM which IP address to use. This is done by adding the following setting to the *distributed.rcx* file:

```
<option name="distributed/NCM_server_local_ip">A.B.C.D</option>
```

▶ The *distributed.rcx* file is located in the same location as the *appserver.rcx* file.

The value A.B.C.D should be replaced with the appropriate NAT IP address for the NCM server and should match the RN_CORE table’s CoreHostname value for that NCM Core.

Distributed Conflict List

The Distributed Conflict List page displays the uniqueness constraint conflict list. This provides information about uniqueness conflicts that will need to be manually corrected to ensure that the databases in the NCM Mesh are in sync.

To open the Distributed Conflict List, on the menu bar under Admin select Distributed and click Conflict List. The Distributed Conflict List opens.

Distributed Conflict List Page Fields

| Field | Description |
|-------------------|--|
| origin_datasource | The database on which the conflict occurred. |
| Table | The table on which the conflict occurred. |
| rowguid | The guid of the row on which the conflict occurred. |
| Status | Status is “event_generated” if the system has sent an alert that this conflict exists. |
| Actions | You can select the following options: <ul style="list-style-type: none"> • Detail — Opens the View Distributed Conflict page, where you can view details on an individual uniqueness constraint. See View Distributed Conflict Page on page xxxv. • Delete — Deletes the conflict from the database. |

View Distributed Conflict Page

The View Distributed Conflict page provides details on a specific uniqueness constraint.

To open the View Distributed Conflict page:

- 1 On the menu bar under Admin select Distributed and click Conflict List. The Distributed Conflict List opens.
- 2 In the Actions column, click the Detail option. The View Distributed Conflict page opens.

View Distributed Conflict Page Fields

| Field | Description |
|---------------------|--|
| origin_datasource | The database on which the conflict occurred. |
| Table | The table on which the conflict occurred. |
| Conflicting Data | The columns and their values that are causing the conflict. |
| reason_text | A description of why the conflict occurred. |
| rowguid | The guid of the row on which the conflict occurred. |
| conflict type | The type of conflict according to the SQL Server. |
| reason_code | Depending on the type of conflict, the error message from the SQL Server. |
| MS_repl_create_time | The time the conflict was generated by the SQL Server. |
| conflict_table | The location where the SQL Server stores the conflicting rows. |
| Status | Status is "event_generated" if the system has sent an alert that this conflict exists. |

Partition Reassignment Page

The Partition Reassignment page enables the partition-to-NCM Core mapping to be modified. This is useful for failover of partitions from one NCM Core to another and for restoring partitions back to their original NCM Core.

To open the Partition Reassignment page, on the menu bar under Admin select Distributed and click Partition Reassignment. The Partition Reassignment page opens. You can select NCM Cores from the drop-down menu.

List Cores Page

The List Cores page lists all NCM Cores in the NCM Mesh. This page provides information to properly manage the Distributed System.

To open the List Cores page, on the menu bar under Admin select Distributed and click Core List. The List Cores page opens.

List Cores Page Fields

| Field | Description |
|-------|----------------------|
| Name | The NCM Core's name. |

| Field | Description |
|-----------------|---|
| Core Hostname | The hostname of the NCM Core's NCM server. |
| Is Master? | Is the NCM Core the Publisher? (Yes or No) |
| Timezone Offset | The timezone offset of the actual NCM Core server. |
| Status | Currently, there is only Normal status. |
| Realm | The default Realm for the NCM Core. |
| Actions | You can select the following option: <ul style="list-style-type: none"> • Edit — Open the Edit Core page. See Edit Core Page on page xxxvii. |

Edit Core Page

The Edit Core page enables you to edit the NCM Core definition.

To open the Edit Core page:

- 1 On the menu bar under Admin select Distributed and click Core List. The List Cores page opens.
- 2 In the Actions column, click the Edit option. The Edit Core page opens.

You can complete the following fields:

- Name — Enter the NCM Core name.
- Database Identifier — Enter the database name. This information is needed to make connections to a particular SQL Server instance on a server.
- Core Hostname - Enter the hostname of this NCM Core's server.
- RMI Port — Enter the RMI port. RMI (Remote Method Invocation) is Java's remote procedure call protocol. The distributed system makes RMI calls between NCM servers in the NCM Mesh to transfer information about scheduled tasks, system settings, software images, and so on.
- Database hostname — Enter the Database hostname.
- Database Port — Enter the port on the database server with which NCM communicates with the database.
- Timezone Offset — Select a Timezone offset from the drop-down menu.
- Replication Admin User — Enter the name of the Replication Admin user. The Replication Admin user is created and used by the Oracle database to manage replication.
- Replication Password — Not applicable for SQL Server.
- Confirm Replication Password — Not applicable for SQL Server.
- Comments — Add any comments about the NCM Core.
- Realm Name — Enter the Realm in which the NCM Core resides. For information about segmenting devices, see the *User Guide for CiscoWorks Network Compliance Manager*.

Device Password Rule Priority Reset Page

The Device Password Rule Priority Reset page enables you to reset device password rule priorities in the event that a uniqueness constraint conflict occurs for those objects.

To open the Reset Password Priority page, on the menu bar under Admin select Distributed and click Device Password Rule Priority Reset. The Device Password Rule Priority Reset page opens.

Click the Reset Priority button to reset the device password rule priorities.

Renew Configuration Options Page

The Renew Configuration Options page enables you to reset the configuration options when the configuration options on an NCM Core become out-of-sync with other servers in the NCM Mesh.

To open the Renew Configuration Options page, on the menu bar under Admin select Distributed and click Renew Configuration Options. The Renew Configurations Options page opens.

Click the Renew Config Options button to ensure that all options on the NCM Core are in sync with the rest of the NCM Mesh.

4 Failover and Recovery

This chapter contains the following topics:

- [Failover](#) on page xxxix
- [Recovery](#) on page xxxix

Failover

When the network has been configured to failover, if an NCM Core fails, users can continue to access all data in the system using a different NCM Core. All partitions that were originally managed by the failed NCM Core can be pointed to a new NCM Core using the Partition Reassignment page. For information, see [Partition Reassignment Page](#) on page xxxvi. Procedures for system recovery will vary depending on how the remote server failed.

Recovery

There are three basic recovery scenarios:

- Loss of network connectivity
- Loss of an NCM application server
- Loss of a Database server

Loss of Network Connectivity

In the case of lost network connectivity, failover occurred due to network issues. No problems occurred with the NCM application server or with the SQL Server database server. Recovery consists of the following steps:

- 1 Resolve the network issues.
- 2 Reset partitions that had been reassigned back to their original NCM Core. This can be accomplished in NCM. See [Partition Reassignment Page](#) on page xxxvi.
- 3 If any drivers have been added to the system during the outage, click the “Reload Drivers” button on the Start/Stop Services page. This action reloads the driver files and pushes them across to other NCM Cores in the NCM Mesh. This action should be performed on the NCM server where the drivers were added. For information, see the *User Guide for CiscoWorks Network Compliance Manager*.
- 4 If any system settings have been modified during the outage, use the “Renew Config Options” page to make sure options are synced across the NCM Mesh. See [Renew Configuration Options Page](#) on page xxxviii.

- 5 If any NCM Cores have lost connectivity for a long period of time, restart the NCM Core server that lost connectivity after data sync so as to reload certain partition data and avoid exception errors due to obsolete data.

After the network issues are resolved, the system should recover as replication synchronizes data between the databases.

Loss of an NCM Application Server

In cases where the NCM application server suffers a failure that requires re-installation of the NCM application, recovery consists of the following steps:

- 1 During NCM installation, select the “Use existing database” option. In addition, the database selected should be the one the failed server was previously using.
- 2 Add the *distributed.rcx* file from the *ReplicationScriptToolBundle* to the directory where the *appserver.rcx* file resides.
- 3 Re-add any NCM Core-specific special case options for patches and support issues.
- 4 Restart NCM.
- 5 Reset partitions that had been re-assigned back to their original NCM Core. This can be accomplished in NCM. See [Partition Reassignment Page](#) on page xxxvi.
- 6 If any drivers have been added to the system during the outage, click the “Reload Drivers” button on the Start/Stop Services page. This action reloads the driver files and pushes them across to other NCM Cores in the NCM Mesh. This action should be performed on the NCM server where the drivers were added. For information, see the *User Guide for CiscoWorks Network Compliance Manager*.
- 7 If any system settings have been modified during the outage, use the “Renew Config Options” page to make sure options are synced across the NCM Mesh. See [Renew Configuration Options Page](#) on page xxxviii.
- 8 Edit the original NCM Core to modify any parameters that may be different (perhaps the installation happened on a new server with a different hostname).
- 9 Copy the software images repository from a good NCM Core to the recovered NCM Core.

Loss of a Database Server

In the case of a lost database server, the NCM application server is still running but cannot access the database. The database will need to be rebuilt and replication set up again on the database. Recovery consists of the following steps:

- 1 Pause or delete any tasks that appear to be waiting, pending, or not running because they are associated with the lost database server. You can perform this action on the other NCM application server in the NCM Mesh.
- 2 If the lost database server is the Subscriber, replication needs to be removed from the NCM Mesh. For information, see [Removing Replication](#) on page xx. Keep in mind that Replication can be reconfigured after removal. For information, see [Creating a Two NCM Core SQL Server Replication Environment](#) on page xiv.
- 3 If the lost database server is the Publisher, replication needs to be reinstalled.
 - a Export all RN_% tables from the Subscriber database.
 - b Create a new database on the NCM Core 1 database server with data exported from the Subscriber database.

- c Reinstall NCM on the NCM Core 1 Server. Be sure NCM points to the existing database on NCM Core 1.
 - d Edit the RN_CORE entry. See [Deletion Steps](#) on page xxi (step 5).
 - e Install replication. For information, see [Creating a Two NCM Core SQL Server Replication Environment](#) on page xiv.
- 4 Make sure the SQL Server Agent is running on the restored database server. Open a command prompt on the restored database server and enter: **net start**. You should see `SQL Server Agent (MSSQLSERVER)` in the output. If you do not, enter: **net start SQL Server Agent <MSSQLSERVER>**.
 - 5 Reset partitions that had been reassigned back to their original NCM Core. This can be accomplished in NCM. See [Partition Reassignment Page](#) on page xxxvi.
 - 6 In NCM, edit the NCM Core that was recovered to make sure all information is correct for the new setup. For information, see the *User Guide for CiscoWorks Network Compliance Manager*.

5 Troubleshooting

This chapter contains the following topic:

- [RMI Errors in the NCM Console](#) on page xliii
- [Removing In-Memory and Database Information](#) on page xliv

RMI Errors in the NCM Console

An RMI error that prevents NCM console logon or inhibits use of the NCM console can mean that NCM is unable to identify the localhost.

RMI Error

Event format:

Local Core: <hostname>

Remote Core: <hostname>

Error: <Exception text>

This error typically occurs when there are network problems between the NCM servers. To troubleshoot this problem, make sure:

- 1 The host that the server cannot connect to is up and running.
- 2 The NCM instance on that host is running.
- 3 From a command line, enter `ping <host>` to ensure that network connectivity exists between servers.
- 4 From a command line, enter `telnet <host> to port 1099` (or whatever your RMI listen port is set to) to ensure that RMI connections are being accepted. If working correctly, you should get back some data that includes the text string `java.rmi.MarshalledObject`.

Failures of any of these steps will point to corrective actions needed, such as updating the RMI port being used in the Edit NCM Core page, or restarting NCM to make sure that the RMI port has been bound correctly and is not being used by another application.

To correct the problem, update the localhost section of the `hosts` file on each NCM application server as follows:



This solution is for static IP environments only.

- 1 In a text editor such as WordPad or vi, open the following file:
 - *Windows:* <Drive>:\Windows\System32\drivers\etc\hosts
 - *UNIX:* /etc/hosts
- 2 Set the localhost line to read:


```
127.0.0.1 localhost
```


- 3 For each NCM application server, add a line in the following format:

```
<xx.xx.xx.xx> <NCM.example.com> <NCM>
```

 - Replace `<xx.xx.xx.xx>` with the IP address of the NCM application server.
 - Replace `<NCM.example.com>` with the fully-qualified domain name of the NCM application server.
 - Replace `<NCM>` with the short hostname of the NCM application server.
- 4 Repeat [step 3](#) until the `hosts` file includes all NCM application servers in the distributed system environment.
- 5 To use the updated hosts information, restart the NCM application server.

Removing In-Memory and Database Information

To ensure proper removal of all in-memory and database information, and to avoid replication conflicts due to matching timestamps, NCM active tasks are automatically deleted on the NCM Core with which they are associated. An active task is any task that does not have the SUCCEEDED, FAILED, DUPLICATE, SKIPPED, or WARNING status.

Keep in mind that deleted tasks could be displayed in task lists for a few moments while the replication process pushes the deletions to other NCM Core in the system. In addition, if the NCM Core from which the task originated is not accessible, the task deletion fails.

Index

B

- Buttons
 - Renew Config Options xxxviii
 - Reset Priority xxxvii

C

- Conflict table xxxvi

D

- Deleting a Subscriber xx
- Deleting a subscriber xx
- Device Password Rule Reset page xxxvii
- Distributed Conflict List xxxv
- Distributed Monitor Results page xxxiv
- Distributed systems
 - Failover xxxix
 - Installation xi
 - Overview v
 - Recovery xxxix
 - Terminology v
 - Troubleshooting xliii

E

- Edit Core page xxxvii
- Events
 - RMI error xxxiii
 - Status xxxv, xxxvi
 - Time synchronization warnings xxxiii

F

- Failover
 - Loss of a database server xl
 - Loss of a NCM server xl
- Files
 - distributed.rcx xx
 - ReplicationScriptTool xvii, xl

H

- Horizontal scalability viii

I

- Installation
 - Creating two NCM Cores xiv
 - Planning xi
 - SQL Server setup xi
 - Unpacking the Replication script xiv

L

- List Cores page xxxvi

M

- Monitor results xxxiv

N

- NCM
 - Renew Configuration Options xxxviii
- NCM
 - Core v
 - Device Password Rule Reset page xxxvii
 - Distributed Conflict List xxxv
 - Distributed Monitor Results page xxxiv
 - Edit Core page xxxvii
 - List Cores page xxxvi
 - Mesh v
 - Site Reassignment page xxxvi
 - View Distributed Conflict page xxxv

- NCM scheduler vi

P

- Partition Reassignment page xxxvi
- Ports
 - JNDI xiii
 - SNMP xiii
 - Telnet xiii

R

- Recovery
 - Loss of network connectivity xxxix
 - Replication data xxxix
- Removing database information xxxi, xliv
- Removing replication xx

Renew Configuration Options xxxviii

Replication

- Monitoring vi
- Script tool xviii
- Two NCM Cores xiv
- Unpacking xiv

RMI connections xxxiv

Rowguid xxxv, xxxvi

S

SQL Server

- Deleting a subscriber xx
- Multi-master replication vi
- Publisher xiv
- Server setup xi
- Subscriber xiv

sqlcmd xviii

Subscriber

- Definition v
- Deleting xx

System administration

- List Cores xxxvi
- Monitor results xxxiv
- NCM generated events xxxii
- NCM UI pages xxxiii, xxxiv
- Partition reassignment xxxvi
- Uniqueness conflicts xxxv

T

Terminology v

Timezone offset xxxvii

Troubleshooting

- Removing information xliv

U

Uniqueness conflicts xxxii, xxxiv

V

View Distributed Conflict page xxxv