This documentation (the "Documentation") and related computer software program (the "Software") (hereinafter collectively referred to as the "Product") is for the end user's informational purposes only and is subject to change or withdrawal by CA at any time.

This Product may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Product is confidential and proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of the Documentation for their own internal use, and may make one copy of the Software as reasonably required for back-up and disaster recovery purposes, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the provisions of the license for the Software are permitted to have access to such copies.

The right to print copies of the Documentation and to make a copy of the Software is limited to the period during which the license for the Product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to certify in writing to CA that all copies and partial copies of the Product have been returned to CA or destroyed.

EXCEPT AS OTHERWISE STATED IN THE APPLICABLE LICENSE AGREEMENT, TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS PRODUCT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS PRODUCT, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

The use of this Product and any product referenced in the Documentation is governed by the end user's applicable license agreement.

The manufacturer of this Product is CA.

This Product is provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7013(c)(1)(ii), as applicable, or their successors.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Copyright © 2006 CA. All rights reserved.
# Table of Contents

## Preface

- Audience ......................................................... 7
- About This Guide ............................................. 7
- Reading Path .................................................. 7
- Revision Information ........................................... 8
- Documentation Conventions .................................. 8
- Technical Support ............................................. 9

## Chapter 1

### Distributed eHealth Concepts and Overview

- eHealth Site Configuration Overview ....................... 11
  - Standalone eHealth Systems ............................... 11
  - Remote Polling Sites ...................................... 12
  - Distributed eHealth Sites ................................. 13
- Distributed eHealth Concepts ................................. 14
  - Distributed eHealth Systems .............................. 14
  - Distributed eHealth Consoles ............................. 15
  - Trusted and Non- Trusted Systems ....................... 16
- Self-Monitoring Using eHealth SystemEDGE ................. 16
- Cluster Concepts ............................................. 16
- Replication Overview ........................................ 19
- Synchronization Overview ................................... 21
- Log Files ....................................................... 23
- Work-in-Progress Indications ................................. 23
- Distributed Reports ........................................... 23
  - Running Distributed Reports .............................. 24
- Unsupported eHealth Features at the Distributed eHealth Console .... 24
- Administration Tasks Overview ............................. 24
  - Distributed Commands ..................................... 25
  - eHealth Commands ........................................ 26
# Chapter 2

**Distributed eHealth Site Planning**

- Site Planning Considerations ........................................... 29
- Identifying Your Resource Inventory .................................. 29
  - Understanding Your Internet Infrastructure ......................... 30
- Creating a Site Plan ...................................................... 30
- System Requirements ................................................... 31
- eHealth SystemEDGE ..................................................... 33
- License Requirements .................................................. 33
- User Accounts and Access ............................................. 34
- Connection Requirements ............................................. 35
- Platform/Workstation Requirements .................................. 36
- System Locations ...................................................... 36
- Time Synchronization .................................................. 36
- Naming Rules, Duplicates, and Uniqueness ........................... 36
- Creating a Site Layout ................................................ 38
  - Single-Network/Local Sites .......................................... 38
  - Distributed eHealth Consoles Outside a Firewall .................. 39
  - Multi-Network/Dispersed Sites .................................... 41
  - Security Considerations ........................................... 41
  - Using the Inventory to Assign Elements to Distributed eHealth Systems .................................................. 42
  - Other Site Planning Options ....................................... 42
- Converting Existing eHealth Systems to a Cluster ..................... 42
  - Clustering Existing Standalone Systems ............................. 43
  - Converting a Remote Polling Site to a Cluster ..................... 43

# Chapter 3

**Creating the Distributed eHealth Cluster**

- Installing eHealth on Each Cluster System .......................... 45
- Before You Create the Cluster ....................................... 46
  - Time Synchronization Among Systems ............................. 46
  - Configuring the TCP Ports ......................................... 51
- Adding Systems to the Cluster ..................................... 51
  - Adding the First Member of a Cluster .............................. 52
  - Adding Additional Members to the Cluster ....................... 53
  - Testing Cluster Status ............................................ 54
- Listing the Cluster Members ....................................... 54
  - Listing the Cluster Members of a Specific System ............. 55
  - Listing the Cluster Members of All Systems ...................... 55
  - Listing the Removed Cluster Members ............................. 56
  - Listing Only the Cluster Member Names ........................... 56
- Modifying a Cluster Member ....................................... 56
  - Modifying the Information on One Cluster Member ............ 57
  - Modifying the Information on All Other Cluster Members .... 57
  - Modifying the Information on All Cluster Members ............ 57
- Removing a Cluster Member ...................................... 57
  - Removing a System from All Cluster Members ........................ 58
  - Restoring a Removed System .................................... 58
  - Restoring a System in All Cluster Member Tables ............... 58
Chapter 4  Managing the Distributed eHealth Cluster  59

Managing the Cluster Membership ........................................59
Monitoring Cluster Status .................................................59
Starting Cluster Status .....................................................59
Monitoring the Processes and Log Files ..............................60
  Monitoring Logs .........................................................60
  Monitoring Processes ..................................................62
  Configuring SystemEDGE on a Cluster Member .................63
Monitoring the Health of the Distributed eHealth Cluster ........64
  Discovering the eHealth Systems ......................................64
  Creating a Group of the Cluster Members for Reporting ..........65
Managing the Distributed eHealth Console .......................65
  Managing Web User Accounts ........................................65
  Monitoring Disk Space and System Performance .................65
  Adding Another Distributed eHealth Console ....................66
Managing the Distributed eHealth Systems .......................66
  Monitoring Disk Space ...............................................66
  Monitoring Reports and Data Analysis .............................66
Recovering a Cluster System ............................................67
Moving a Database to a New Cluster System ......................68
Managing the Objects in the Cluster ................................70
  Finding Duplicate Objects in the Cluster ...........................70
  Identifying Where an Element is “Homed” ........................71
Managing Security for the Cluster ....................................73
Upgrading the Cluster Systems .........................................74
Installing Service Packs and Certification Releases .............74

Chapter 5  Managing Synchronization and Replication  75

About Replication and Synchronization ...............................75
  About Immediate Replication .........................................76
  About Scheduled Synchronization ....................................77
  About Manual Replication .............................................78
Using Replication Log Information ....................................80
Finding Differences in Object Information .........................80
  Reviewing the Scheduled Job Log Files ...........................81
Replication and Soft-Deletion ..........................................83
  Configuring the Replication Timeout ...............................83
## Chapter 6 Managing Distributed Reports

Managing Distributed Reports

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generating Distributed Reports</td>
<td>85</td>
</tr>
<tr>
<td>Running Distributed Reports On-Demand</td>
<td>85</td>
</tr>
<tr>
<td>Scheduling Distributed Reports</td>
<td>87</td>
</tr>
<tr>
<td>Using the Large-Scale Reports</td>
<td>88</td>
</tr>
<tr>
<td>The Heartbeat Timer</td>
<td>89</td>
</tr>
<tr>
<td>The Impact of Missing Elements on Reports</td>
<td>89</td>
</tr>
<tr>
<td>Important Information about Distributed Reports</td>
<td>90</td>
</tr>
<tr>
<td>Understanding Distributed Report Performance</td>
<td>90</td>
</tr>
<tr>
<td>About Time Zones</td>
<td>91</td>
</tr>
<tr>
<td>Generating Reports from the Distributed eHealth Console</td>
<td>91</td>
</tr>
<tr>
<td>Distributed Health Report Changes</td>
<td>92</td>
</tr>
<tr>
<td>About Service Profiles</td>
<td>92</td>
</tr>
<tr>
<td>Troubleshooting Report Failures</td>
<td>93</td>
</tr>
<tr>
<td>Rerunning Failed Reports</td>
<td>93</td>
</tr>
<tr>
<td>Troubleshooting Missing Information</td>
<td>94</td>
</tr>
<tr>
<td>Configuring Distributed Report Timeouts and Controls</td>
<td>94</td>
</tr>
<tr>
<td>Distributed eHealth Console Variables</td>
<td>96</td>
</tr>
</tbody>
</table>

## Appendix A Troubleshooting Cluster Administration

Troubleshooting Cluster Administration

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Error: Object Has Been Marked for Deletion</td>
<td>99</td>
</tr>
<tr>
<td>Troubleshooting Replication Problems and Failures</td>
<td>99</td>
</tr>
<tr>
<td>Detecting Inconsistencies</td>
<td>99</td>
</tr>
<tr>
<td>Recovering Failed Replications</td>
<td>99</td>
</tr>
<tr>
<td>Recovering from Conflicting Edits</td>
<td>100</td>
</tr>
<tr>
<td>Timeout Errors</td>
<td>100</td>
</tr>
<tr>
<td>Cluster Administration Command Timeout Errors</td>
<td>100</td>
</tr>
<tr>
<td>nhRunCommand Timeout Errors</td>
<td>100</td>
</tr>
<tr>
<td>Replication Timeout Errors</td>
<td>101</td>
</tr>
<tr>
<td>RCS Timeout Errors</td>
<td>101</td>
</tr>
<tr>
<td>nhJoinCluster Timeout Errors</td>
<td>103</td>
</tr>
<tr>
<td>Unable to Connect to Port and Host</td>
<td>103</td>
</tr>
<tr>
<td>nhListClusterMembers Returns Tables That are Not the Same</td>
<td>104</td>
</tr>
<tr>
<td>Object Changes Are Lost</td>
<td>104</td>
</tr>
<tr>
<td>Recovering Soft-Deleted Objects</td>
<td>104</td>
</tr>
<tr>
<td>Problems after Hostname Changes</td>
<td>104</td>
</tr>
<tr>
<td>Redirecting Users When a Distributed Console Fails</td>
<td>105</td>
</tr>
<tr>
<td>Troubleshooting Distributed Report Messages</td>
<td>105</td>
</tr>
<tr>
<td>Missing Data Notes in Reports</td>
<td>105</td>
</tr>
<tr>
<td>Report Failed</td>
<td>106</td>
</tr>
<tr>
<td>Application Response Reports Fail</td>
<td>106</td>
</tr>
<tr>
<td>Troubleshooting Cluster Communication Problems</td>
<td>106</td>
</tr>
<tr>
<td>Connection Refused Errors</td>
<td>106</td>
</tr>
</tbody>
</table>

## Index

Index

Distributed eHealth Administration Guide
This guide describes how to plan, install, and manage a Distributed eHealth® site. This guide supports eHealth Release 6.0 and later.

**Audience**

This guide is intended for eHealth administrators and personnel who will install and configure a Distributed eHealth cluster of systems. Before you read this guide, you should be familiar with the general concepts described in the *Introduction to eHealth* guide and the basic administration tasks described in the *eHealth Administration Overview Guide*.

**NOTE**

This guide does not describe the specifics of database administration and maintenance. For a description of how to install eHealth and Oracle, refer to the *eHealth Installation Guide* for your platform. For a description of how to manage the eHealth database, refer to the *eHealth Database Management Guide*.

In addition, you should be familiar with windows-based operating systems, command line tools, and how to edit text files using a tool such as vi, Emacs, or Notepad.

**About This Guide**

This section describes the possible reading paths that you should follow, depending on your needs. It also includes the documentation conventions used in this guide.

**Reading Path**

You should read the sections of this guide in the following order:

1. Review the concepts described in Chapter 1, “Distributed eHealth Concepts and Overview,” to familiarize yourself with the Distributed eHealth concepts and terms.
2. Work with your CA sales representative to plan your site following the instructions in Chapter 2, “Distributed eHealth Site Planning.” Thoroughly plan the layout and organization of your site to ensure that you obtain the proper system resources and to speed the installation and setup process.
3. After your Distributed eHealth site is installed and configured, review the following:
   • Chapter 3 to create a cluster
   • Chapter 4 to manage the cluster systems and the elements in a cluster
   • Chapter 5 to manage the replication and synchronization processes
4. Finally, refer to Chapter 6 for details on how to run and manage distributed reports.

Revision Information
This revision includes the following changes:
• In Chapter 1, “Distributed eHealth Concepts and Overview,” added a new section on setting up planned downtime in a cluster.
• In Chapter 4, “Managing the Distributed eHealth Cluster,” expanded the instructions for moving a database to a new cluster system.
• Revised Chapter 5, “Managing Synchronization and Replication,” to incorporate an expanded discussion of replicating groups.
• Throughout the guide, removed references to technology-specific groups. With Release 6.0, you now use the new Group Editor available in the OneClick for eHealth console to create groups of elements. eHealth no longer requires you to use the eHealth console to organize your elements based on a specific technology type.
• Removed Appendix B, “Command Reference.” For details on any eHealth command, refer to the Web Help.

Documentation Conventions
Table 1 lists the conventions used in this document.

Table 1. Documentation Conventions (Page 1 of 2)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File or Directory Name</td>
<td>Text that refers to file or directory names.</td>
</tr>
<tr>
<td>code</td>
<td>Text that refers to system, code, or operating system command line examples.</td>
</tr>
<tr>
<td>emphasis</td>
<td>Text that refers to guide titles or text that is emphasized.</td>
</tr>
<tr>
<td>enter</td>
<td>Text that you must type exactly as shown.</td>
</tr>
<tr>
<td>Name</td>
<td>Text that refers to menus, fields in dialogs, or keyboard keys.</td>
</tr>
<tr>
<td>New Term</td>
<td>Text that refers to a new term, that is, one that is being introduced.</td>
</tr>
<tr>
<td>Variable</td>
<td>Text that refers to variable values that you substitute.</td>
</tr>
<tr>
<td>→</td>
<td>A sequence of menus or menu options. For example, File → Exit means “Choose Exit from the File menu.”</td>
</tr>
</tbody>
</table>
Table 1. Documentation Conventions  (Page 2 of 2)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NOTE</strong></td>
<td>Important information, tips, or other noteworthy details.</td>
</tr>
<tr>
<td><strong>CAUTION</strong></td>
<td>Information that helps you avoid data corruption or system failures.</td>
</tr>
<tr>
<td><strong>WARNING</strong></td>
<td>Information that helps you avoid personal physical danger.</td>
</tr>
</tbody>
</table>

**Technical Support**

If you have a Support Contract ID and password, you can access our Support Express knowledgebase at the following URL: http://search.support.concord.com.

If you have a software maintenance contract, you can obtain assistance with eHealth. For online technical assistance and a complete list of primary service hours and telephone numbers, contact Technical Support at http://support.concord.com.
This chapter describes the Distributed eHealth® product and important concepts. It also describes the tasks required to manage a Distributed eHealth site.

**eHealth Site Configuration Overview**

When you purchase eHealth, you can choose one of the following site configurations:

- Standalone eHealth system
- Remote polling site
- Distributed eHealth site

The following sections describe these different configuration options. You should familiarize yourself with these options so that you can choose the option that best meets your needs.

**Standalone eHealth Systems**

A standalone eHealth system is a single workstation on which you have installed the eHealth software. It identifies and collects data for all elements that you are monitoring, and it stores the database of performance information. It also provides web server access for users who want to view or run reports, as well as to use the Live Health applications. Figure 1 shows a standalone eHealth system.

![Figure 1. Standalone eHealth System](image)

The number of elements that a standalone eHealth system can support varies depending on the system configuration, types of elements, polling intervals, and the types of reports that you run.
A standalone eHealth system is the typical site choice for most small to medium networks (up to 50,000 monitored elements using 5-minute polling, or up to 80,000 elements at 15-minute polling). The monitored elements are typically located within the network and can respond within the configured polling intervals. Most or all web users access the eHealth web server from systems that reside within the monitored network, although users may also access the eHealth system through a firewall.

Remote Polling Sites

A remote polling site (also referred to as a distributed polling site) contains several eHealth systems that share the polling workload and send their data and element information to a central eHealth system. Figure 2 shows a remote polling site. The central site is typically a high-performance system with a large amount of disk space because it collects all of the polled data from each polling system. It must support the large database and have the processing power to run reports for all monitored elements. The central system typically does not poll any elements of its own; instead, it reserves its processing resources for generating reports. The central site can typically support up to 80,000 elements depending on the system configuration and the scheduled and on-demand reports that you run.

![Figure 2. eHealth Remote Polling Site]

The remote poller sites are typically mid-to-high performance systems capable of discovering and polling their unique sets of elements and sending the data back to the central site. You usually do not run reports from the remote sites, as they do not retain long-term, historical data needed for reports or baselines. If a remote site retains polled data, you can run reports for the elements that it manages, just like any standalone eHealth system.

Users typically choose a remote polling site when it is critical that the polling systems do not know about each other or share information. For example, the remote sites could each manage separate customers of a service provider, or separate business units within a large organization. These remote poller sites could be geographically dispersed sites, as well as local sites separated by firewalls or other barriers over which polling could be delayed or impossible.

**NOTE**

For detailed information about remote polling and how to configure a remote polling site, refer to the *eHealth Remote Poller Guide*. If you want to convert a remote polling site to a Distributed eHealth cluster, contact CA Technology Services.
**Distributed eHealth Sites**

A **Distributed eHealth site** contains several eHealth systems that are connected in a configuration referred to as a **cluster**. The eHealth systems in a cluster can have one of the following roles:

- A **Distributed eHealth System** identifies, collects, and stores data for a number of unique elements. It shares information about its global objects such as elements, groups, and group lists with the other cluster members. It also processes requests to run reports from the Distributed eHealth Console.

- A **Distributed eHealth Console** allows users to view and run reports for any or all of the elements monitored by the Distributed eHealth Systems. (The Distributed eHealth Console cannot discover or collect information for any elements; it serves only as a reporting front-end from which users can access their distributed reports.)

Figure 3 illustrates a Distributed eHealth site. The figure shows the Distributed eHealth Consoles “outside” the firewall, but you can place them anywhere in your infrastructure.

![Figure 3. Distributed eHealth Site](image_url)

**A firewall is a security application or system that connects your infrastructure to the Internet. It prevents unauthorized access to your resources, and it can also restrict the Internet sites that your local users can access.**

The eHealth systems communicate using interfaces already present in your infrastructure. The interfaces that connect your eHealth systems must support the communications traffic between cluster members. For detailed communication requirements, refer to “Connection Requirements” on page 35.

The Distributed eHealth Systems essentially act as standalone eHealth systems, but they share their information with the other members of the cluster. You can locate the Distributed eHealth Systems in the same area of the network, or disperse them geographically throughout your infrastructure to place them closer to the elements that they monitor. You can have several Distributed eHealth Systems in the same area of the infrastructure if you have a large number (more than 50,000) of elements to monitor in that area. Like standalone eHealth systems, each Distributed eHealth System can monitor up to 50,000 elements at 5-minute polls, or 80,000 elements at 15-minute polls (based on system performance and report load).
Locating the Distributed eHealth Consoles

Users typically locate the Distributed eHealth Consoles outside their firewalls to allow remote users to access eHealth reports without accessing the protected areas of their infrastructure. Users at the Distributed eHealth Console can run reports for any element monitored by any of the Distributed eHealth Systems, as well as for groups or group lists defined within the cluster. To a report user, the cluster of eHealth systems appears to be one virtual system.

Running Distributed Reports. Unlike an eHealth remote polling environment, in which the remote pollers send the actual polled data to the central site, a Distributed eHealth Console does not collect the polled data. Instead, it collects the element, group, and group list information from each Distributed eHealth System. When a user runs a report, the Distributed eHealth Console sends the report request to the Distributed eHealth Systems to process the report, which in turn send the report information back to the Distributed eHealth Console. The Distributed eHealth Console collates the report data and displays the final report.

For reports on subjects that span more than one Distributed eHealth System, eHealth divides the workload among the Distributed eHealth Systems. Thus, report data generation occurs in parallel. The Distributed eHealth Systems respond with report data, and the Distributed eHealth Console collects the data and generates the final report. Thus, reports run from the Distributed eHealth Console are distributed reports.

You would typically choose a Distributed eHealth site when you want to run reports for more elements than a standalone eHealth system can support. You might also choose a Distributed eHealth site if you want to place an eHealth web server system outside the firewall and insulate the Distributed eHealth Systems within the firewall of your infrastructure. Depending on the number of Distributed eHealth Systems that you have, and the system performance of the Distributed eHealth Console, a Distributed eHealth site could support reports for up to one million elements.

Distributed eHealth Concepts

This section describes concepts and terms used throughout the documentation relating to Distributed eHealth processes and procedures.

Distributed eHealth Systems

A Distributed eHealth System is an eHealth system that is a member of a cluster and that discovers, polls, and stores data for elements in your infrastructure. A Distributed eHealth System performs all of the same functions of a standalone eHealth system, and you must administer and manage it the same way. Because it is a member of a cluster, it shares (replicates) certain information with the other cluster systems so that users can run distributed reports.

A Distributed eHealth System requires an eHealth – Distributed license, which enables it to run cluster commands and participate as a full polling system in the cluster. It also requires other eHealth licenses to collect data for elements of various types as well as to run optional eHealth applications and reports, if you purchase those features.

On each Distributed eHealth System in your cluster, you identify the elements that it monitors and for which it collects data. You can also create groups and group lists to organize your elements on each Distributed eHealth System, just as you can create them on a standalone eHealth system. When you create a group or group list on a Distributed eHealth System, eHealth replicates that information to the other
Distributed eHealth Systems so that you can add elements managed by other Distributed eHealth Systems to those groups. The elements, groups, and group lists that you create on a Distributed eHealth System are owned by that system. It shares its information with the other cluster members, but eHealth always retains information about the original owner.

Distributed eHealth Systems provide all of the features of a standalone eHealth system; you can access the Web interface of the Distributed eHealth System to run reports for the locally-defined elements and groups, as well as to use applications like Traffic Accountant and Live Health. In addition, you must manage database backups and maintenance jobs to maintain the Distributed eHealth System, just like any standalone eHealth system.

**NOTE**

Traffic Accountant and Application Response are not distributed. You must access each Distributed eHealth System to run Traffic Accountant or Application Response for the elements monitored by that system. For other restrictions, refer to “Unsupported eHealth Features at the Distributed eHealth Console” on page 24. You cannot configure distributed paths (response or Frame Relay) where the endpoints are monitored by different eHealth systems in the cluster. The endpoint elements and their paths must be monitored by the same eHealth system.

The Distributed eHealth Systems send their element, group, and group list information, as well as information about the members of those groups and group lists, to the Distributed eHealth Consoles in the cluster. They send their group, group list, and group list content information to the other Distributed eHealth Systems in the cluster. They do not share their actual polled data; the data remains on each Distributed eHealth System. For a detailed description of the replication process and a summary of the information that is replicated within the cluster, refer to “Replication Overview” on page 19.

In addition, Distributed eHealth Systems synchronize their information with information from the other cluster systems to maintain the latest configuration information. For a detailed description of the synchronization process, refer to “Synchronization Overview” on page 21.

**Distributed eHealth Consoles**

A Distributed eHealth Console allows users to view and run distributed reports for any or all of the elements monitored by the Distributed eHealth Systems in the cluster. It collects information about the elements, groups, and group lists that exist in the cluster and provides the list as the possible subjects for reports that users can run. The Distributed eHealth Console cannot discover or collect data for any elements, nor can it manage element configuration information, groups, or group lists. Those features are disabled on the eHealth console interface of a Distributed eHealth Console.

A Distributed eHealth Console system requires an eHealth – Distributed license; the license enables the system to run cluster commands and obtain information from the other systems in the cluster. In addition to the Distributed license, the system also requires other eHealth licenses to run reports for various types of elements and to run optional eHealth applications, if you purchase those features.

Web users access the eHealth Web interface of a Distributed eHealth Console to generate and view their reports. As with a standalone eHealth system, an eHealth web administrator creates web user accounts and manages access to the elements, groups, group lists, and reports. Traffic Accountant and several Live Health applications are not distributed. You must access each Distributed eHealth System to run Traffic Accountant, Live Trend, or Live Status applications for the elements monitored by that system. For other restrictions, refer to “Unsupported eHealth Features at the Distributed eHealth Console” on page 24.
One Distributed eHealth Console can support approximately 250 web users running reports simultaneously. It can support more web users in general, as long as they do not view large-element reports simultaneously. For details on how to create and manage web account access for an eHealth system, refer to the eHealth Administration Web Help.

**Trusted and Non-Trusted Systems**

Since eHealth systems in the cluster could reside inside as well as outside the firewalls in your infrastructure, eHealth allows you to identify the trusted systems in your cluster. A **trusted system** can issue commands (such as nhRunCommand and nhPutFile) which can modify the configuration and operation of other systems in the cluster. A **non-trusted system** cannot issue these eHealth commands; the other cluster members will not accept these commands from a non-trusted system. You also cannot add new cluster members by joining them to a non-trusted member of the cluster.

![Firewall](image)

Typically, if you have an eHealth cluster member that resides outside the firewall—such as the Distributed eHealth Console—you configure it as a non-trusted system to ensure that outside users cannot run eHealth commands that could alter your cluster configuration or introduce any viruses or other damaging programs into the cluster.

**Self-Monitoring Using eHealth SystemEDGE**

You must install eHealth SystemEDGE on each eHealth system in the cluster and configure it to monitor the critical processes and log files for Distributed eHealth. You can also use eHealth SystemEDGE to restart critical processes that have stopped. For details on these tasks, refer to “Configuring SystemEDGE on a Cluster Member” on page 63. For instructions on using eHealth to monitor your cluster members, refer to “Monitoring the Health of the Distributed eHealth Cluster” on page 64.

**NOTE**

Deploy AdvantEDGE View on eHealth Distributed Systems only. Do not deploy AdvantEDGE View on a Distributed eHealth Console (the reporting front-end system).

**Cluster Concepts**

A **cluster** is a loosely coupled set of eHealth systems that share information. Your cluster can contain several Distributed eHealth Systems and Distributed eHealth Consoles to meet your organizational needs. To support 1,000,000 elements, CA recommends that your cluster include up to 25 Distributed eHealth Systems and up to four Distributed eHealth Consoles.
If one of the systems in the cluster fails, the other systems in the cluster continue to operate.

- If a Distributed eHealth Console fails, users will not be able to access their distributed reports on that system until it is recovered. (However, you can have multiple Distributed eHealth Consoles in your cluster to ensure that users have a backup access point to run distributed reports.)

- If a Distributed eHealth System fails, the elements managed by that system will not be polled during that downtime, and any distributed reports that include those elements will not include data for the missing elements.

Creating a Cluster

Distributed eHealth has a simple cluster configuration process; you install eHealth on two or more systems following the standard installation process. When you install the eHealth software, you specify whether the system is a Distributed eHealth Console or a standalone system (which you can then add to a cluster as a Distributed eHealth System).

Once eHealth is installed and running, you create a cluster by adding the first system using the nhJoinCluster -firstMember command. You then add the other systems using nhJoinCluster and specifying the name of a trusted system already in the cluster. Chapter 3 provides the detailed steps for creating a cluster.

**NOTE**

Although you could create several clusters of eHealth systems within your infrastructure, an eHealth system can belong to only one cluster.

Also, once a Distributed eHealth System joins a cluster, you cannot remove it to join a different cluster. To move an eHealth system to a new cluster, or to merge several clusters into one, contact Support for assistance.

Cluster Password

The cluster systems use a cluster password to confirm membership in the cluster. When you use the nhJoinCluster command to add a system to the cluster, you must specify the cluster password. Each system in a cluster must have the cluster password configured to communicate with the other systems. Each cluster member must use the same password.

For security reasons, eHealth does not display the password as you type it, and it does the following:

- Encrypts the password when it saves the password and when it transmits the password to other cluster members.

- By default, does not install the nhGetClusterPasswd command; you must obtain it from your Systems Engineer (SE) or Technical Support.

You can display and set the cluster password using the nhGetClusterPasswd and nhSetClusterPasswd commands. Any system that does not have the correct password configured cannot run cluster commands and communicate with the other systems.
Uniquely Identifying Cluster Members

eHealth uses a value called the member ID to uniquely identify each eHealth system in the cluster. You can specify a member identifier (ID) for an eHealth system when you add it to the cluster. If you do not specify a member ID, eHealth obtains a default ID based on the IP address of the eHealth system. (The ID is not the IP address itself; it is a mathematical conversion of part of the IP address.) You cannot change the member ID of an eHealth system, and it remains the same for a cluster member even if you change the IP address of the system.

In addition, eHealth uses a value called the member name to identify systems in the cluster. When you add a system to a cluster, eHealth uses the hostname of the system as the member name value. If you change the hostname of a cluster system, you must modify the member name using the nhModifyClusterMember command. The cluster members use the member name to communicate with each other; if the member names are not the hostnames, or if the systems cannot resolve the hostnames to IP addresses, your cluster members will not be able to communicate.

The Member Table

Each system in a cluster has an internal table of information about all of the cluster members. This internal table, the member table, identifies all cluster members and contains information that the local system needs to communicate with them. When you add an eHealth system to a cluster, Distributed eHealth immediately updates the member table on each system with the new information. Likewise, when you remove an eHealth system from the cluster, eHealth updates the member table on each cluster member system to show that the system is no longer an active member of the cluster. The local cluster system is the eHealth cluster system to which you are logged in and from which you run a command.

**Note**

When you remove a system from a cluster, eHealth does not delete the information for that system from the table. Instead, it updates the information to reflect that the system is no longer an active member. That is, it does not allow the system to run eHealth cluster commands, and it does not replicate its information to or obtain information from the other systems. In the future, you could restore the system to the cluster (which would enable the processes). eHealth would then update the table information to reflect the active status.

How Cluster Systems Communicate

Each eHealth system in the cluster has special services and processes that manage communication and information transferred between cluster members. Although these processes are transparent to eHealth administrators, you should become familiar with the following terms:

- The **Remote Communication Service (RCS)** is a set of processes that allows eHealth systems in the cluster to communicate with each other. The cluster uses RCS to perform both file transfers and inter-process communications across the cluster.

- The **File Staging Area (FSA)** is an internal directory within the eHealth installation directory. The FSA contains temporary files that were sent by the local system to the other cluster members, as well as files that were received from other cluster members. eHealth runs a scheduled job that deletes old files in the FSA on a regular basis to clean up the directory and save disk space. (You cannot modify the scheduled job.)
Monitoring Cluster System Status

The Cluster Status application on the eHealth Web interface allows you to monitor the status of the cluster members. You can display information about the active members and their connection status at any time. For more information on using the Cluster Status application, refer to “Starting Cluster Status” on page 59.

Replication Overview

eHealth uses replication to “push” information about the objects defined within the eHealth system in the cluster to one or all other eHealth systems in the cluster. The following sections describe objects and how eHealth replicates object information.

Objects

An object is a specific entity such as an element, group, group list, or service profile. Objects also include the contents (members) of groups and group lists, and a scheduled data analysis job record. When you create an object (by discovering an element or creating a group list, for example), the eHealth system on which you create the object assigns a unique ID to the object. This ID, referred to as a database ID or a handle, is a unique value within the eHealth system. eHealth uses the combination of the eHealth system member ID and the object database ID to uniquely identify an object within the cluster.

While element names must be unique within the eHealth system that discovers them, object names are not required to be unique within the cluster. However, you should name objects uniquely within the cluster to make identification easier.

For example, when you create a new group for a Distributed eHealth System, replication processes send information about that group to the other cluster systems. Administrators at other Distributed eHealth Systems can add their own elements to the group, and users at the Distributed eHealth Consoles can generate reports for that group. eHealth will not allow another administrator to create a group that has the same name as an existing group in the cluster.

Information Automatically Replicated

eHealth replicates information when you create, modify, rename, or delete any of the following objects:

- Groups
- Group lists
- Group list members (groups assigned to the group list)
- Service profiles
- Data analysis records (which result when you add or modify a scheduled report on a Distributed eHealth Console)

Groups, group lists, group list members, and service profiles are global objects because their information is shared with all eHealth systems in the cluster. When you save the changes to one of these objects, eHealth immediately replicates the information for these objects to the other eHealth systems in the cluster.

eHealth does not immediately replicate information for elements and group members (elements assigned to groups) when you add, modify, rename, or delete them. Since elements and group members are known only to the eHealth system on which they are created and the Distributed eHealth Consoles, these objects are not considered global objects.
After you discover new elements, modify or delete elements, or change group memberships, you save the changes locally (on the Distributed eHealth System that you are using). Every four hours on Distributed eHealth Consoles, eHealth runs a scheduled job called Element Synchronize that collects new, modified, or deleted element information and group membership information from the Distributed eHealth Systems in the cluster. (The job collects the element and group membership changes since the last Element Synchronize job, and updates the Distributed eHealth Console accordingly.)

In addition, on the Distributed eHealth Consoles, eHealth runs a scheduled Synchronize job each night to update the Console with the latest element, group membership, and any other new or modified object information from the Distributed eHealth Systems.

The following figures illustrate the information that eHealth replicates between the eHealth systems in a cluster. Figure 4 shows the information replicated from a Distributed eHealth Console to the Distributed eHealth Systems.

![Figure 4. Data Replicated from a Distributed eHealth Console](image)

Figure 5 shows the information replicated from Distributed eHealth Systems to a Distributed eHealth Console.

![Figure 5. Data Replicated from Distributed eHealth Systems to a Distributed eHealth Console](image)

Figure 6 shows the data that Distributed eHealth Systems replicate to each other.
Information Manually Replicated

eHealth replicates information for the global objects immediately, and for some objects during the scheduled Synchronize and Element Synchronize jobs. If you make object changes for elements or group members and you want to update the Distributed eHealth Consoles immediately, or if you are fixing inconsistent object information in the cluster, you can also use the nhReplicate command to replicate information on demand.

You might want to replicate other types of eHealth information from one eHealth system in the cluster to other systems in the cluster. For example, you could create a custom report or a custom icon for the Web interface that you want to replicate to all systems in the cluster. You can use the nhPutFile command to send files to other cluster members. Cluster systems accept these requests only from trusted cluster members (as described in “Trusted and Non-Trusted Systems” on page 16).

Synchronization Overview

While replication “pushes” information to other cluster members, synchronization “pulls” information from other cluster members to update the local eHealth system. Most synchronization occurs during scheduled jobs that run automatically at regular intervals. You can also use the nhSynchronize command to synchronize information on demand. The following sections describe how eHealth synchronizes information.

NOTE

When one or more eHealth systems have different information for the same object, the cluster is unsynchronized (“out-of-sync”). For example, if two eHealth administrators simultaneously change the information for the same service profile, the profile becomes out-of-sync. For a description of how to correct and avoid these problems, refer to “Troubleshooting Replication Problems and Failures” on page 99.

Information Automatically Synchronized

The following figures illustrate the information that eHealth synchronizes between the systems in a cluster. Each system pulls only the information that it needs from the other systems. Figure 7 shows the information synchronized by Distributed eHealth Consoles.
Figure 7. Data Synchronized by a Distributed eHealth Console

Figure 8 shows the information synchronized by Distributed eHealth Systems from Distributed eHealth Consoles.

Figure 8. Data Synchronized by Distributed eHealth Systems

Figure 9 shows the information synchronized between Distributed eHealth Systems.

Figure 9. Data Synchronized between Distributed eHealth Systems

**Information Manually Synchronized**

You can use the nhSynchronize command to update the local system with the latest cluster information on-demand. For example, you can use nhSynchronize to update a Distributed eHealth Console with changes to elements and groups at Distributed eHealth Systems. Also, you can use nhSynchronize to fix inconsistent object information in the cluster.
You can manually synchronize other types of eHealth information between eHealth systems in the cluster. For example, you could create a custom report on one system, and use nhGetFile on another system to obtain that file.

**Log Files**

eHealth creates log files for many critical tasks and processes, such as installation, polling, discovery, report generation, scheduled jobs, replication, and synchronization. Log files can help you to determine the status of a job as well as troubleshoot failed processes or errors.

For eHealth systems in a cluster, you must configure eHealth SystemEDGE to monitor critical eHealth processes and log files. For more information, refer to “Monitoring the Processes and Log Files” on page 60.

**Work-in-Progress Indications**

eHealth provides work-in-progress (WIP) indications to inform you of the status and the completion of various critical processes such as report generation, object creation, replication, synchronization, and others. If one of these processes fails, the work-in-progress windows display important information that can help you to diagnose the problem.

**Distributed Reports**

The distributed version of an eHealth report is identical in appearance and function to a standalone-system version of the report. The charts, layout, names, and presentation are identical, as are the drill-down options and capabilities. The difference is that the subject of the report (such as an element, group, or group list) can reside on any Distributed eHealth System in the cluster.

Groups and group lists can contain elements and groups from one or more Distributed eHealth Systems in the cluster. The person who runs or views the report does not need to know which Distributed eHealth System owns the elements, or whether the group contains elements monitored by several Distributed eHealth Systems.

When you run a distributed report, the Distributed eHealth Console sends messages to each Distributed eHealth System that owns any of the elements or groups included in the report. Since each Distributed eHealth System is responsible for generating the data for the elements that it owns, the report process workload is spread across several Distributed eHealth Systems; they work in parallel to create the report data and return it to the Distributed eHealth Console. The Distributed eHealth Systems do not participate in the distributed report processing unless they own elements included in the report subject.

**NOTE**

If you run a report for a large number of elements that are all owned by the same Distributed eHealth System, it takes approximately the same amount of time to produce as a large-element report run on a standalone eHealth system.
Running Distributed Reports

From the Distributed eHealth Console, you can run all types of distributed reports: Trend, Top N, At-a-Glance, What-If, MyHealth, Health, and Service Level. eHealth administrators can run these reports from the eHealth console or the Web interface, as they can run them for a standalone eHealth system. Web users must access the web server on the Distributed eHealth Console to run or view these reports.

Unsupported eHealth Features at the Distributed eHealth Console

From the Distributed eHealth Console, you cannot use the following features:

- Traffic Accountant reports
- Live Trend and Live Status applications
- Application Response Management interface

You must run Traffic Accountant reports, Live Trend, Live Status, and the Application Response Management interface locally from the Distributed eHealth System that is monitoring the specific elements that you want to view.

The following features are also not supported:

- Distributed eHealth Systems cannot participate as remote pollers in a remote polling environment; however, a central site can serve as a Distributed eHealth System in a cluster.
- You cannot configure distributed paths (response or Frame Relay) where the endpoints are monitored by different eHealth systems in the cluster. The endpoint elements and their paths must be monitored by the same eHealth system.

Administration Tasks Overview

This guide describes how to perform the following tasks relating to a Distributed eHealth site:

1. Plan the cluster site and the number of eHealth systems in it.
2. Create the eHealth cluster and test communication between cluster members.
3. Identify the elements that will be owned by each Distributed eHealth System in your cluster. (You should distribute the elements as evenly as possible among the Distributed eHealth Systems for best performance.)
4. Define your distributed groups and group lists, and add members to them from each Distributed eHealth System as applicable.
5. Manage each cluster member system to ensure that critical processes are running, take actions on events, and manage the database backups and recovery for the eHealth systems in your cluster.
6. Manage distributed report operations, including generating and scheduling reports.
7. Troubleshoot the cluster configuration and distributed reports that fail or that contain incomplete data.
Distributed Commands

Table 2 presents several commands that eHealth administrators can access to perform tasks within the cluster in a Distributed eHealth environment:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>nhAddSysEdgeMonEntries</td>
<td>Configures eHealth SystemEDGE to monitor the critical log files and processes on an eHealth cluster system.</td>
</tr>
<tr>
<td>nhFindClusterDiffs</td>
<td>Displays a list of object definition differences between two or more cluster systems.</td>
</tr>
<tr>
<td>nhFindDupObjects</td>
<td>Displays a list of duplicate objects on one or more cluster systems.</td>
</tr>
<tr>
<td>nhFindElement</td>
<td>Locates the Distributed eHealth System that owns an element.</td>
</tr>
<tr>
<td>nhGetClusterPasswd</td>
<td>Displays the cluster password. (You can obtain this command from your SE or Technical Support.)</td>
</tr>
<tr>
<td>nhGetFile</td>
<td>Obtains a copy of a file from one or more cluster systems.</td>
</tr>
<tr>
<td>nhJoinCluster</td>
<td>Adds an eHealth system to a cluster.</td>
</tr>
<tr>
<td>nhListClusterMembers</td>
<td>Displays information about the cluster members defined for one or more eHealth systems.</td>
</tr>
<tr>
<td>nhModifyClusterMember</td>
<td>Changes the cluster configuration information for a cluster system.</td>
</tr>
<tr>
<td>nhPutFile</td>
<td>Sends a copy of a file to one or more cluster systems.</td>
</tr>
<tr>
<td>nhRemoveClusterMember</td>
<td>Removes an eHealth system from the cluster member table of one or more systems.</td>
</tr>
<tr>
<td>nhReplicate</td>
<td>Sends object information to one or more cluster systems.</td>
</tr>
<tr>
<td>nhRestoreClusterMember</td>
<td>Returns a system to the cluster.</td>
</tr>
<tr>
<td>nhRunCommand</td>
<td>Runs a specified command on one or more cluster systems.</td>
</tr>
<tr>
<td>nhSetClusterPasswd</td>
<td>Specifies the cluster password.</td>
</tr>
<tr>
<td>nhSynchronize</td>
<td>Updates the local system with the object information from one or all other cluster systems.</td>
</tr>
<tr>
<td>nhUpdateClusterTable</td>
<td>Updates the cluster member table information on one or all other cluster systems.</td>
</tr>
<tr>
<td>nhUpdateDatabaseIds</td>
<td>Updates the database ID table on one or more cluster systems.</td>
</tr>
<tr>
<td>nhUpgradeCluster</td>
<td>Allows you to manage eHealth release upgrades across a cluster.</td>
</tr>
</tbody>
</table>
**eHealth Commands**

Many eHealth commands include arguments that allow you to execute them on a specific cluster system (-host), on the other remote systems in the cluster (-cluster), or on all systems in the cluster (-all). These commands include the following:

- nhConfig
- nhDbStatus
- nhDelete Analyzed Data
- nhElementStatus
- nhExportConfig
- nhExportData
- nhImportData
- nhSchedule
- nhUpdate Db Protocol

These commands support one or more of the three cluster options.

**NOTE**

The nhConfig, nhExportConfig, nhExportData, and nhImportData commands are not supported on a Distributed eHealth Console since it does not contain element information or polled data.

---

**Setting Up Planned Downtime in a Cluster**

You can set up planned downtime on individual Distributed eHealth Systems in a cluster using the nhImport Downtime and nhExport Downtime commands; however, these commands are not “cluster aware.” Each Distributed eHealth System must have a unique .pdi file to report the planned downtime for the elements being polled on that system. Because each system monitors different elements, you should not replicate planned downtime files to other Distributed eHealth Systems.

**NOTE**

If you move the .pdi files to another system, the load will fail because the elements do not exist on the other system. You will also lose the existing planned downtime records at that system.

The -cluster argument is not useful for exporting downtime from or importing downtime to Distributed eHealth Systems. Running nhExportDowntime -cluster from a Distributed eHealth Console and then running nhImportDowntime -cluster does not merge the planned downtime to all Distributed eHealth Systems in the cluster, and it is not recommended. The -cluster argument is only useful if you launch the commands from a central location in the cluster to take effect at remote systems.
To set up planned downtime in a cluster:

1. Define outages within a .pdi file for each element for the given Distributed eHealth System. For instructions, refer to the section on accounting for planned downtime in the eHealth Report Management Guide.

2. Import the downtime schedule for the system into the eHealth database by entering the following:
   
   ```
   nhImportDowntime filename.pdi -host distributedeHealthSystemName
   ```

   **NOTE**
   
   If you are running this command from the actual Distributed eHealth System, you can omit the -host argument; otherwise, you must specify it.

3. Repeat Steps 1 and 2 for each Distributed eHealth System in the cluster.

To modify the planned downtime schedule for a Distributed eHealth System:

1. Enter the following command:
   
   ```
   nhExportDowntime filename.pdi -host distributedeHealthSystemName
   ```

2. Open the file and review the summary; then do the following:
   
   a. Resolve discrepancies among existing planned outages and actual events.
   b. Correct errors.
   c. Add new outages.
   d. Delete outages that did not occur.
   e. Save the file.

3. Reimport the data into the eHealth database:
   
   ```
   nhImportDowntime filename.pdi -host distributedeHealthSystemName
   ```

   **NOTE**
   
   If you are running this command from the actual Distributed eHealth System, you can omit the -host argument; otherwise, you must specify it.
Distributed eHealth Site Planning

This chapter describes how to plan a Distributed eHealth site. It describes system requirements, site options, and various management concepts that you should consider before you install the software on the cluster member systems. Review this chapter to familiarize yourself with the planning issues, but work with your Sales representative to develop the final plan.

Site Planning Considerations

Before you install eHealth Release 6.0 in a distributed environment, consider the following points:

- **Resource Inventory.** How many resources do you want to monitor, where are they located, and what types of resources do you have?
- **Number of eHealth systems.** How many Distributed eHealth Systems do you need to monitor all of your resources? How many Distributed eHealth Consoles do you need to provide access to your eHealth report users?
- **Licenses.** What types of resources do you want to monitor, and what types of eHealth reports and tools do you want to use?
- **Users.** How many eHealth administrators will be working in the Distributed eHealth site? How many users require access to the eHealth reports and tools?

Identifying Your Resource Inventory

The first major task in the site planning process is to identify the resources that you want to monitor. You might have systems, applications, LAN/WAN interfaces, routers, and many other types of resources that you want to manage using eHealth. Using an inventory list or map, you should work with your Sales representatives to identify the following:

- Types of resources that you want to monitor
- Location of the resources (geographically or within a campus or business park configuration)
- Number of each type of resource at any given location
- IP addresses for the monitored devices, and possibly, any special kinds of management agents that might be used or required
- Any network or infrastructure issues such as firewalls, network connections, internet policies, security concerns, or other issues
- Number of users (and their locations) who require access to eHealth reports and tools
With this information, you can create a site plan that identifies the locations and types of resources that you want to manage and monitor with eHealth. This information can help you to determine the number of Distributed eHealth Systems that you need to monitor your elements, the number of Distributed eHealth Consoles that you need for user access, where the systems should be located, the eHealth license requirements for the applications that you will use, and the system requirements for the eHealth systems.

**Understanding Your Internet Infrastructure**

Identifying the resources in your internet infrastructure can be a very difficult task. Most organizations have a diverse and sometimes very large number of resources, and the inventory can change frequently. For example, your organization might merge with another—or divide into several smaller ones—and resources often move with those organizational changes. You might change to a new business application to provide better services to your users or customers. You might replace older network interfaces with newer, higher-speed routers and interfaces to improve your network performance.

If you do not already have an inventory of your resources, you should develop one. It can help you to more quickly resolve problems with those resources, as well as identify when changes occur. It is very helpful for planning your eHealth monitoring requirements.

You can develop resource inventories from a variety of sources. You might have a tool that provides a topological or graphical map of your resources. You might have a network management system (NMS) that can output information for the resources that it has identified. For example, many router and switch vendors provide NMS tools that can output resource information for their hardware inventory.

**Creating a Site Plan**

As you work with (or develop) your resource inventory, you should also identify the location and type of resources. For example, Figure 10 shows a map of a sample internet infrastructure for a geographically dispersed organization. The map identifies the locations of each major office, and summarizes the number of manageable resources (elements) at each location.

![Figure 10. A Sample Site Planning Diagram](image-url)
If you want to manage all of these elements in a Distributed eHealth cluster, you need to obtain more details about the elements at each location and how you want to manage them. Table 3 shows a sample worksheet that identifies more information about each site.

Table 3. Sample Resource Inventory Table

<table>
<thead>
<tr>
<th>Location</th>
<th>Total Elements</th>
<th>Types of Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>250,000</td>
<td>100,000 LAN/WAN interfaces 100,000 Systems 50,000 Applications</td>
</tr>
<tr>
<td>London</td>
<td>50,000</td>
<td>40,000 Frame Relay circuits 10,000 Router/Switch elements</td>
</tr>
<tr>
<td>Rio de Janeiro</td>
<td>25,000</td>
<td>10,000 Applications 15,000 Response</td>
</tr>
<tr>
<td>Sydney</td>
<td>100,000</td>
<td>50,000 Systems 10,000 Applications 40,000 Response</td>
</tr>
</tbody>
</table>

**System Requirements**

After you develop your resource inventory, you can begin to plan your eHealth system and license requirements. The following sections describe the important planning considerations for your eHealth monitoring requirements.

**Distributed eHealth Systems**

Each Distributed eHealth System in your cluster should meet or exceed the system requirements specified in the eHealth Sizing Wizard. A Solaris-based Distributed eHealth System can support up to 50,000 elements at five-minute polling intervals, or up to 80,000 elements at fifteen-minute polling intervals. HP-UX- and Windows-based Distributed eHealth Systems can support up to 40,000 elements at five-minute polling intervals, or up to 80,000 elements at fifteen-minute polling intervals. As you determine the number of elements that each Distributed eHealth System will monitor, you can tune the system, disk space, and performance requirements with your Sales representative.

Using the resource numbers in the sample site shown in Figure 10 on page 30, you can estimate the minimum number of Distributed eHealth Systems that you need to monitor these elements. Table 4 shows the sample resource inventory table with estimates for the Distributed eHealth Systems. (This example assumes that the eHealth systems are Solaris-based platforms.)
Distributed eHealth Consoles

Each Distributed eHealth Console system in your cluster should meet the system requirements specified in the eHealth Sizing Wizard. You must have one Distributed eHealth Console for every 250,000 elements in your Distributed eHealth site. As you plan the number of users who will access the console and their report generation and access requirements, you can tailor CPU and performance resources for their needs.

Since the eHealth database on a Distributed eHealth Console does not contain polled data, the disk space requirements are less than that of a Distributed eHealth System. However, you must plan the system performance to meet the needs of the number of elements that the console supports, the number of web users, and the number of on-demand and drilldown reports that might occur during a typical day.

Using the resource numbers in the sample site shown in Figure 10 on page 30, you can estimate the minimum number of Distributed eHealth Consoles that you need for your site. Consider the following:

- Since there are 250,000 elements in New York, you need one Distributed eHealth Console for the New York elements. Likewise, you might want another Distributed eHealth Console for the 100,000 elements in Sydney.
- The London and Rio de Janeiro locations have 50,000 and 25,000 elements, respectively. You could have one Distributed eHealth Console to support both. If access time or performance is an issue, consider having two Distributed eHealth Consoles (one for each location).
- In addition to the number of elements at each location, consider the number of web accounts for the users who will access the Distributed eHealth Console as well as the potential number of reports that they might run on demand. These additional factors might require you to add more Distributed eHealth Consoles in a location or use more powerful systems to support the user needs.

Table 5 shows the sample resource inventory table with estimates for the Distributed eHealth Consoles.

---

**Table 4. Sample Resource Inventory Table**

<table>
<thead>
<tr>
<th>Location</th>
<th>Total Elements</th>
<th>Types of Elements</th>
<th>Distributed eHealth Systems¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>250,000</td>
<td>100,000 LAN/WAN interfaces 100,000 Systems 50,000 Applications</td>
<td>5</td>
</tr>
<tr>
<td>London</td>
<td>50,000</td>
<td>40,000 Frame Relay circuits 10,000 Router/Switch elements</td>
<td>1</td>
</tr>
<tr>
<td>Rio de Janeiro</td>
<td>25,000</td>
<td>10,000 Applications 15,000 Response</td>
<td>1</td>
</tr>
<tr>
<td>Sydney</td>
<td>100,000</td>
<td>50,000 Systems 10,000 Applications 40,000 Response</td>
<td>2</td>
</tr>
</tbody>
</table>

¹ One Solaris-based Distributed eHealth System can support up to 50,000 elements with 5-minute polling, or up to 80,000 elements with 15-minute polling. (HP-UX- and Windows-based Distributed eHealth Systems can support up to 40,000 elements with 5-minute polling.)
You must install *eHealth SystemEDGE* on each *eHealth* system in your cluster to monitor processes and logs. *eHealth* provides a series of tools that allows you to configure SystemEDGE to monitor log files and processes and alert you when problems occur. The *eHealth SystemEDGE User Guide* provides a complete description of the installation process and the agent features. For a description of how to configure the agent to monitor your Distributed *eHealth* processes and logs, refer to “Monitoring the Processes and Log Files” on page 60.

**License Requirements**

You typically install similar *eHealth* licenses on each *eHealth* system in the cluster. The following sections describe the requirements for each system.

**Distributed *eHealth* System License Requirements**

Each Distributed *eHealth* System in your cluster must have the following licenses:

- *eHealth* – Distributed
- Poller licenses (to support the number of polled elements on this system)
- One or more licenses for each technology monitored by the Distributed *eHealth* System (such as LAN/WAN, System, Router/Switch, and so on)
- Optionally, licenses for *eHealth* integration module components if you plan to collect data from elements managed by another NMS

**Table 5. Sample Resource Inventory Table**

<table>
<thead>
<tr>
<th>Location</th>
<th>Total Elements</th>
<th>Types of Elements</th>
<th>Distributed eHealth Systems(^1)</th>
<th>Distributed eHealth Consoles(^2)</th>
<th>Number of Web Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>250,000</td>
<td>100,000 LAN/WAN interfaces, 100,000 Systems, 50,000 Applications</td>
<td>5</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>London</td>
<td>50,000</td>
<td>40,000 Frame Relay circuits, 10,000 Router/Switch elements</td>
<td>1</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>Rio de Janeiro</td>
<td>25,000</td>
<td>10,000 Applications, 15,000 Response</td>
<td>1</td>
<td>0 or 1</td>
<td>25</td>
</tr>
<tr>
<td>Sydney</td>
<td>100,000</td>
<td>50,000 Systems, 10,000 Applications, 40,000 Response</td>
<td>2</td>
<td>1</td>
<td>50</td>
</tr>
</tbody>
</table>

1. One Solaris-based Distributed *eHealth* System can support up to 50,000 elements with 5-minute polling, or up to 80,000 elements with 15-minute polling. (HP-UX- and Windows-based Distributed *eHealth* Systems can support up to 40,000 elements with 5-minute polling.)

2. You must have at least one Distributed *eHealth* Console for every 250,000 elements.
Each Distributed eHealth System in your cluster should have the same set of technology and reporting licenses to run distributed reports for those elements. However, since Traffic Accountant, Live Trend, and Live Status applications are not distributed, you could have different license arrangements for those applications on each Distributed eHealth System.

**Distributed eHealth Console License Requirements**

Each Distributed eHealth Console in your cluster must have the following licenses:

- eHealth – Distributed
- One or more licenses for each technology that will be included in reports run on the system (such as LAN/WAN, System, Router/Switch, and so on)
- Licenses for any reports such as Health and Service Level reports that you use

Typically, each Distributed eHealth Console in your cluster has the same set of technology and reporting licenses as the Distributed eHealth Systems to run distributed reports for those elements.

**User Accounts and Access**

On a typical eHealth system, you have the following types of user accounts:

- eHealth administrators
- eHealth database users
- eHealth web accounts
  - The admin eHealth web administrator account
  - Other web accounts

The following sections describe the role of these accounts in a Distributed eHealth site and how to manage them.

**Planning for eHealth Administrator Accounts**

When you install the eHealth software on a system, you must specify the user name of the eHealth administrator. The eHealth administrator is the account that you use to log in to the eHealth system and from which you can run the eHealth console, run commands, and manage the operations of the eHealth system.

**Primary eHealth Administrator Account.** You must have at least one eHealth administrator account on each eHealth system. The primary account is the account that you specified during the eHealth software installation.

If your Distributed eHealth Systems are securely protected within your network infrastructure, you could use the same account name and password for the eHealth administrator account on each system. It might be easier to remember a common name and password when you need to log in remotely to each Distributed eHealth System to perform administrative tasks.

However, since your Distributed eHealth Consoles typically reside outside a firewall, you should use a unique account name and password for the eHealth administrator account on those systems. This can help to reduce the chance that a user might obtain the eHealth administrator account name and use it to access your Distributed eHealth Systems.
Supplemental eHealth Administrator Accounts. You can have additional eHealth administrator accounts if your system has multiple users who perform eHealth administrative tasks such as discovery, managing groups, and so on.

If you have more than one eHealth administrator for your cluster systems, consider creating policies or guidelines to help them understand their roles and functions in a Distributed eHealth environment. For example, if each administrator manages the elements for a particular customer or business organization, consider implementing naming conventions for elements, groups, group lists, and any custom reports to help each administrator to clearly identify the objects for which he or she is responsible.

eHealth Web Accounts
When you install the eHealth software on a system, the installation program creates two web user accounts (admin and ehealth) that you and other users can use to access the eHealth Web interface. The admin account is the account name for the eHealth web administrator, the user who can manage the site and other web user accounts. The ehealth account is a generic account for other, non-administrative, web users.

In standalone eHealth system environments, you typically log in to the Web interface as the admin user to create each web account. You can create and manage as many accounts as you need to support the number of users who access your eHealth system.

In a Distributed eHealth site, you create one or more web user accounts on each Distributed eHealth Console to identify the users who can log in and use eHealth. If you have more than one Distributed eHealth Console, you can do one of the following:

- Create the same accounts on each Distributed eHealth Console to provide users with multiple access points to the eHealth reports and tools.
- Create unique accounts on each Distributed eHealth Console to force users to access a specific Distributed eHealth Console.

You can also create web accounts on the Distributed eHealth Systems to allow certain users to access applications such as Live Trend, Live Status, and Traffic Accountant, or to run reports for those elements homed on a specific Distributed eHealth System. For a complete description of how to create web accounts in a Distributed eHealth environment, refer to “Managing Web User Accounts” on page 65.

Connection Requirements
For best performance, the eHealth systems in your cluster should be connected by links capable of at least T1-level speeds [1.544 million bits per second (Mbps)]. For an example, refer to “Multi-Network/Dispersed Sites” on page 41.

To ensure secure communication between the eHealth systems in your cluster, you can configure a virtual private network (VPN) to connect the eHealth systems. CA strongly recommends that you use VPN connections if your eHealth cluster systems reside within several firewall-separated networks.

NOTE
Your cluster members must have static (unchanging) IP addresses. eHealth does not support connection configurations in which the cluster systems have dynamic IP addresses.
Platform/Workstation Requirements

eHealth currently supports installation on Solaris, HP-UX, and Windows 2000. A Distributed eHealth cluster can contain systems of different platform types. For details on the supported operating systems, refer to the eHealth Installation Guide for that platform.

System Locations

As you plan your cluster site, you need to plan the location of the Distributed eHealth Systems and the Distributed eHealth Consoles. You should place your Distributed eHealth Systems as close as possible to the elements that each will monitor, and you should identify the elements owned by each Distributed eHealth System as part of your resource inventory.

Time Synchronization

You must synchronize each cluster member in the eHealth cluster to the same system time. The Distributed eHealth Consoles collate information from the Distributed eHealth Systems based on hour, day, week, month, or quarter samples. eHealth systems in a cluster can use different time zones to represent the displayed time, but their internal or system time must be synchronized with each other.

If each Distributed eHealth System is set to a different system time, the Distributed eHealth Console incorrectly aggregates data from different times into the same time period. In addition, replication uses time stamps to obtain the latest (most recent) information for objects in the cluster. If the systems have different time settings, eHealth might overwrite the most recent changes with older changes that happen to have a later time stamp on that cluster member.

Several time synchronization software products are available. For a description of the time synchronization administration tasks, refer to “Time Synchronization Among Systems” on page 46.

Naming Rules, Duplicates, and Uniqueness

This section describes some naming and planning issues that you should consider for elements, groups, and group lists.

Avoiding Duplicate Elements in the Cluster

As you plan your Distributed eHealth site, plan your resources to ensure that each resource is monitored (polled) by only one Distributed eHealth System. If more than one Distributed eHealth System polls the same resource, you will have duplicate entries in a distributed report that includes both elements.

Duplicate elements usually result when you discover the same resources from more than one Distributed eHealth System, or you import the same elements into more than one Distributed eHealth System. The result is that more than one Distributed eHealth System is collecting the data for the same element.

NOTE

You may want to have multiple Distributed eHealth Systems monitor the same elements; however, you should avoid including duplicate elements as the subject of distributed reports that aggregate (or total) the data for the subject, such as Trend elements for groups, Health reports, or Service Level reports.
Finding Duplicate Objects

You can periodically check for duplicate objects (elements, groups, group lists, and service profiles) within the cluster using the nhFindDupObjects command. eHealth defines duplicate groups, group lists, and service profiles as those that have the same name. eHealth defines duplicate elements as those that have two out of three matching values for system name (sysName), hardware ID, and IP address. If two of the three values match, eHealth also checks the Simple Network Management Protocol (SNMP) index values (index1, index2, index3, and index4) to find matches.

**NOTE**

If two or more elements have the same name but different system names, hardware IDs, or IP addresses, eHealth does not consider them duplicate elements. The nhFindDupObjects command will not display different elements that happen to have the same name.

Finding Duplicate Element Names

Within your cluster, you may encounter a situation in which you have elements that have the same name, but they are *not* the same element (that is, they are not duplicate elements). For example, you might have two different system elements named NewYorkSales-SH. Duplicate element names usually result when administrators change the element names to be more meaningful than the element names created by the eHealth discover process, but they could result for other reasons.

eHealth replicates element names to the Distributed eHealth Consoles in the cluster. When eHealth encounters duplicate element names, it appends the member ID of the Distributed eHealth System that owns each element to the element name; for example, NewYorkSales-SH: memberId.

Duplicate element names can confuse users at the Distributed eHealth Console. If two or more elements have the same name, but they are different elements, users may not know which one to select for reports that they run. (The member ID alone might not be sufficient to identify each element to the general report users.) When possible, you should look for duplicate element names and change them to make the elements clearly unique and identifiable.

Avoiding Duplicate Names for Groups and Group Lists

When you create a group or group list on a Distributed eHealth System, eHealth immediately replicates that new group to the other eHealth systems in the cluster. eHealth prevents you from adding another group or group list that has the same name and type. Duplicate group and group lists names can also result when two administrators simultaneously create objects with the same name on different cluster systems.

However, if you create groups and group lists on standalone eHealth systems that you later add as members of a cluster, you could encounter duplicate names for different groups and group lists. For example, if you create a group of system elements named Chicago_Sales on two different standalone eHealth systems that you later add to a cluster, you will have two groups that have the same name.

eHealth replicates both group names to the other systems in the cluster. To distinguish the groups, it appends the member ID of the Distributed eHealth Systems where they originated to the group name; for example, Chicago_Sales: memberId.
To resolve any confusion with these two different groups, do one of the following:

- If the two groups actually are different groups with different purposes, rename one to a more unique name. For example, rename one of the groups Chicago_SalesSupport if that is more appropriate.
- If the two groups should be merged into one group for distributed reports, you can do one of the following:
  - Create a new group of systems, and on each Distributed eHealth System, move the elements from each Chicago_Sales group to the new group. You can then delete the empty Chicago_Sales groups and optionally rename the new group.
  - On one of the Distributed eHealth Systems, move the elements in Chicago_Sales: memberId1 to Chicago_Sales: memberId2; then delete Chicago_Sales: memberId1.

Once you make the group and group list names unique again, eHealth does not display the member ID with the names in the user interfaces.

Using the nhPopulateGroup Command

In a Distributed eHealth environment, if you use the nhPopulateGroup command to add elements to a group that already exists, you must identify the member ID of the cluster system that owns the group. Otherwise, the command creates a new group with a duplicate name on the local system. For example, if you run the command:

```
nhPopulateGroup Sales -inFile FileName
```

This command creates a group named Sales on the local system. If a group named Sales already exists somewhere in the cluster, the cluster now has two Sales groups.

To add elements to a group that already exists in the cluster, specify the member ID after the group name, as follows:

```
nhPopulateGroup groupname@memberId -inFile FileName
```

Creating a Site Layout

After you determine the number of Distributed eHealth Systems and Distributed eHealth Consoles that you need to monitor and manage your resources, you should plan the appropriate site layout for the eHealth systems. A site layout describes the location of the eHealth systems, how they will be connected, and also any issues for security, high availability, and other configuration concerns that you might require. The following section describes some typical site layouts and characteristics.

Single-Network/Local Sites

If the resources that you plan to monitor and manage using eHealth are all located within the security of a single infrastructure or network, your Distributed eHealth site will likely contain eHealth systems that are also locally arranged.

For example, if you plan to monitor 500,000 elements located in your organization office in Chicago, you would likely have at least 10 Distributed eHealth Systems and two Distributed eHealth Consoles in a site layout similar to that shown in Figure 11.
Creating a Site Layout

If all of your users (administrators and web users) are already located within the security of your network, you can avoid installing firewalls between your Distributed eHealth Consoles and your Distributed eHealth Systems. You can leverage the existing security of your network to protect your Distributed eHealth systems.

Your network could also be a geographically dispersed network, with resources that span large areas; however, if you have a secure (private) WAN connecting the systems, you will likely have a similar layout to the one shown in Figure 11.

**Distributed eHealth Consoles Outside a Firewall**

If your Distributed eHealth Systems are locally arranged, but some or all of your web users access the eHealth reports from outside the security of your infrastructure, you should consider a site layout that places one or more Distributed eHealth Consoles outside your private network. For example, Figure 12 shows a sample layout for the same Chicago site in which both Distributed eHealth Consoles reside outside a firewall.

If you choose a site layout in which one or more eHealth systems of your cluster reside outside a firewall, you are responsible for configuring the firewall router and connections. You will also have to open several ports for TCP traffic, as described in “Opening Ports in the Firewall” on page 40.

---

**Figure 11. A Local, Self-Contained Site Layout**

**Figure 12. Distributed eHealth Consoles Outside a Firewall**
Distributed eHealth Consoles in a DMZ

If your Distributed eHealth Consoles reside outside the primary firewall protection of your infrastructure, you might consider configuring a DMZ with a “double” firewall of protection. A DMZ provides an added layer of access protection to your system from users who access your Distributed eHealth Consoles (as well as other sites) from the Internet. For example, Figure 13 shows a site layout that places the Distributed eHealth Consoles within a DMZ.

Figure 13. Distributed eHealth Consoles in a DMZ

A number of software and hardware vendors provide DMZ support features.

Opening Ports in the Firewall

The eHealth RCS processes on each cluster system use four ports to communicate with the other cluster systems. By default, eHealth systems use the following four ports:

• 5050 – Incoming messages port
• 5051 – Outgoing messages port
• 5052 – Incoming file transfer port
• 5053 – Outgoing file transfer port

If another application in your environment already uses these ports, you can configure the eHealth system to use different ports. For more information, refer to “Configuring the TCP Ports” on page 51.

NOTE

If you change the ports for one eHealth system in the cluster, you must configure all eHealth systems to use the same ports. Do not open the ports for all types of traffic, only for TCP. Also, configure your firewall to allow access only by the eHealth systems in the cluster.
If you place one or more eHealth cluster systems outside a firewall of your private network, you must configure the firewall to allow TCP traffic on ports 5050 and 5052 for the eHealth systems in your cluster.

**CAUTION**

Distributed eHealth cannot operate in a firewall configuration in which the firewall allows responses to outgoing requests only on the same port that the outgoing request used. That is, outgoing requests on port 5051 receive responses on port 5050, and your firewall must support this communication.

**Multi-Network/Dispersed Sites**

If the resources that you plan to monitor and manage using eHealth are located in a variety of networks/infrastructures, such as the example shown in Figure 10 on page 30, your Distributed eHealth site will likely contain eHealth systems that are separated by multiple firewalls and that are geographically dispersed, as shown in Figure 14.

![Figure 14. Multi-Network/Dispersed Site Layouts](image)

In a dispersed site, you should connect your eHealth cluster systems (Distributed eHealth Systems and Distributed eHealth Consoles) using VPN connections to provide an increased level of secure communications among systems. In Figure 14, note that the web users could access any of the three Distributed eHealth Consoles to obtain reports for the elements in the cluster. You should instruct your web users to access a specific Distributed eHealth Console to obtain their reports. This enables you to balance the load on the Consoles by dividing the users among them.

**Security Considerations**

In general, as you plan your site layout, consider the following security characteristics:

- If you place a Distributed eHealth Console in a DMZ, consider shutting off all traffic other than HTTP to reduce the possibility of outside users accessing the system using FTP, telnet, or other protocols.
- If you create a double firewall with a DMZ, consider placing the firewall routers and the Distributed eHealth Consoles on different subnets. This can help to limit or reduce unauthorized network access because the traffic has to be routed over the different subnets.
Using the Inventory to Assign Elements to Distributed eHealth Systems

You can use your inventory of manageable resources to identify the elements that each Distributed eHealth System in your cluster will poll and monitor. This can help you to avoid accidentally monitoring the same elements with different Distributed eHealth Systems (duplicate elements), and it can help you to plan for growth in your infrastructure. As you add more resources, you can “assign” them to different Distributed eHealth Systems to ensure best monitoring. You can also plan for additional Distributed eHealth Systems when one reaches the maximum number of elements that it can support.

Other Site Planning Options

As you plan your Distributed eHealth site, you might consider using these additional site planning options:

- High-speed connections between cluster systems
- VPN connections between eHealth systems
- High-availability solutions for Distributed eHealth Consoles and Distributed eHealth Systems

The following sections describe these options.

High-Speed Connections for Cluster Systems

For best performance, CA recommends that the links that connect your eHealth cluster systems support at least T1-level speeds. If you connect your systems using slower lines and notice that your commands and processes are “timing out” (that is, they return an error that they could not complete due to timeouts), consider increasing the speed of the links that connect them or changing the timeouts for the processes. For a description of the various timeout controls, refer to “Timeout Errors” on page 100.

VPN Connections between eHealth Systems

To provide secure connectivity between your eHealth systems in your cluster, you should configure VPN connections between them. You must configure your VPN connections to assign static IP addresses to the cluster members, as described in “Connection Requirements” on page 35.

High-Availability Distributed eHealth Consoles

If you would like to provide a high-availability solution for your web user access to the Distributed eHealth Console, CA Technology Services can help you to design a solution. If you have many web users and several Distributed eHealth Consoles, a high-availability solution can help to ensure that your web users have full-time access to their web reports, and can help to load-balance web access to the Distributed eHealth Consoles.

Converting Existing eHealth Systems to a Cluster

If you already have one or more eHealth systems monitoring your resources, you can convert them to a cluster of systems to run distributed reports for all of the resources. The following sections describe how to convert one or more standalone systems to a cluster. For a description of site types, refer to “eHealth Site Configuration Overview” on page 11.
Clustering Existing Standalone Systems
If you have one or more standalone eHealth systems that you want to cluster, you must plan for the following:

- You must upgrade each system to eHealth Release 6.0.
- You must install one or more Distributed eHealth Consoles to run the distributed reports for the cluster. You cannot upgrade an existing eHealth system to be a Distributed eHealth System.
- The existing standalone eHealth systems must meet the system hardware requirements for eHealth Release 6.0. If they do not, you must upgrade the systems.
- You should review your eHealth license support on each system. Each eHealth system in a cluster requires an eHealth – Distributed license. You should also review the existing licenses on each system that you plan to cluster to ensure that they meet your monitoring and distributed reporting needs.
- If you have groups or group lists that have the same name on different eHealth systems, you might want to rename the groups and group lists to clearly identify them before you cluster the systems.

For instructions on upgrading eHealth systems to Release 6.0, refer to the eHealth Installation Guide for your platform.

Converting a Remote Polling Site to a Cluster
If you have a remote polling site of eHealth systems that you want to convert to a Distributed eHealth site, contact CA Technology Services for assistance.
Creating the Distributed eHealth Cluster

This chapter describes how to create a Distributed eHealth cluster. To create a cluster, you perform the following tasks:

- Install eHealth on each system that serves as a Distributed eHealth System.
- Install eHealth on each system that serves as a Distributed eHealth Console.
- Start eHealth on each eHealth system in the cluster and enter the license information.
- Create the cluster by adding the first eHealth system to the cluster.
- Join the additional eHealth systems to the first system in the cluster.

This chapter also describes additional tasks for managing the membership of a cluster such as listing, modifying, removing, and restoring cluster members.

Installing eHealth on Each Cluster System

This section provides an overview of the process that you should follow to install eHealth on your systems in the cluster. For details, refer to the eHealth Installation Guide.

Installing a Distributed eHealth Console

When you install eHealth on a system that will be a member of a cluster, follow the standard installation process described in the eHealth Installation Guide. For a Distributed eHealth Console, note the following:

- You must install eHealth Release 6.0 on a clean system; that is, you cannot install eHealth on a system that already has eHealth installed. You cannot upgrade an existing eHealth system to be a Distributed eHealth Console.
- The eHealth installation displays a prompt that asks you if the eHealth system is a Distributed eHealth Console. (A sample prompt from the UNIX installation script follows.)

```
eHealth System Type.
(1) Standard eHealth System
(2) Traffic Accountant (TA) System
(3) Distributed eHealth System (back-end/polling system)
(4) Distributed eHealth Console (front-end/reporting system)
Please enter your selection [1]:
```
To configure the software correctly, you must enter 4 when you install eHealth on a Distributed eHealth Console.
Installing a Distributed eHealth System

When you install eHealth on a system that will be a member of a cluster, you follow the standard installation process described in the eHealth Installation Guide. For a system that will serve as a Distributed eHealth System in a cluster, note that the eHealth installation displays a prompt that asks you for the type of the eHealth system. (A sample prompt from the UNIX installation script follows.)

```
eHealth System Type.

(1) Standard eHealth System
(2) Traffic Accountant (TA) System
(3) Distributed eHealth System (back-end/polling system)
(4) Distributed eHealth Console (front-end/reporting system)
Please enter your selection [1]:
```

To configure the software correctly, you must enter 3 when you install eHealth on a Distributed eHealth System.

Clustering Existing Standalone Systems

If you already have one or more eHealth systems monitoring your infrastructure, you can join them as a cluster to provide distributed reports for your report consumers. The existing systems act as the Distributed eHealth Systems in your cluster; you must install one or more Distributed eHealth Consoles on new systems to provide the distributed reporting capabilities.

For example, if you have separate eHealth systems that monitor elements in your network, you first ensure that each system is upgraded to eHealth Release 6.0 or later. You must add the eHealth – Distributed license to provide the cluster support, and then create the cluster by following the steps in “Adding Systems to the Cluster” on page 51.

Before You Create the Cluster

Before you create your Distributed eHealth cluster, do the following:

- Synchronize the time on each eHealth system that will be a member of a cluster.
- Ensure that the systems use the same TCP ports for communication.
- Make sure that the cluster systems have fixed IP addresses; they cannot have changing IP addresses due to the use of Dynamic Host Configuration Protocol (DHCP).

Time Synchronization Among Systems

You must synchronize each cluster member in the eHealth cluster to the same system time. System time is not the same as the timezone setting. The system time is the system clock, which shows time in Coordinated Universal Time (UTC). A timezone is a setting that shows the UTC time for a specific region of the world. (You can change the timezone to change the display, but your system time does not change.)
The Distributed eHealth Consoles collate information from the Distributed eHealth Systems based on hour, day, week, month, or quarter samples. If each Distributed eHealth System is set to a different system time, the Distributed eHealth Console incorrectly aggregates data from different times into the same time period. In addition, replication uses time stamps to obtain the latest (most recent) information for objects in the cluster. If the systems have different time settings, eHealth might overwrite the most recent changes with older changes that happen to have a later time stamp.

As the eHealth administrator, you must implement time synchronization among the eHealth cluster members. The time synchronization process differs for Solaris, Windows, and HP-UX platforms. The following sections describe the necessary steps for each platform.

NOTE
Although you can use several methods and software packages to synchronize time on systems, you should use the following methods for a Distributed eHealth environment.

Synchronizing the Time on Solaris Systems

Solaris systems use the /etc/inet/ntp.conf file to maintain time synchronization. Your system could have an ntp.conf file, or you may need to create one.

A sample ntp.conf file follows.

```bash
# more ntp.conf
# Solaris /etc/ntp.conf
#
#--------------------------------------------------
server timeSystem
precision -18       # clock reading precision (10 msec)
driftfile /etc/ntp.drift      # path for drift file
```

You need to replace the value timeSystem with the hostname of a system in your network that regularly checks an atomic clock and synchronizes to its time.

NOTE
The following examples refer to the public Network Time Protocol (NTP) server tick.ucla.edu. For a list of public NTP servers, refer to the WWW Time page at http://www.eecis.udel.edu/~ntp.

To synchronize your Solaris system:

1. Log in as root to a Solaris system in your cluster.
2. Using any text editor, create an /etc/inet/ntp.conf file similar to the previous example.
3. Verify that an *xntpd file exists in the /etc/rc2.d directory. (Note that the file name typically begins with a series of characters such as S74.) Enter these commands:
   ```bash
cd /etc/rc2.d
ls
```

NOTE
You should see a series of files including a file named similar to S74xntpd. If you do not have this file, contact your UNIX system administrator.
4. Repeat these steps for each Solaris system in your cluster.

5. On each cluster member, while logged in as root, run the following commands to confirm that the NTP processes are running:

   ```
   ps -ef | grep -i ntp
   ```

   The command should display information similar to the following:

   ```
   root  185     1    0 14:31:55 ?        0:00 /sbin/sh /etc/rc2.d/S74xntpd start
   root  186   185  0 14:31:55 ?        0:00 /usr/sbin/ntpdate -s -w timeSystem
   ```

   ```
   ntpq -p
   ```

   The command should display information similar to the following:

   ```
   remote refid          st t when poll reach   delay   offset    disp
   ===========================================================================
   *timeSystem tick.ucla.edu 3 u   55   64   377     0.92    -21.485  2.61
   ```

6. If you do not see the above two lines or output from ntpq, do one of the following:

   - Reboot the Solaris system.
   - `sync; sync; reboot`

   - Enter the following command:

     ```
     /sbin/sh /etc/rc2.d/S74xntpd start
     ```

   - If you need to stop the ntpd process, while logged in as root, enter the following command:
     ```
     /sbin/sh /etc/rc2.d/S74xntpd stop
     ```

**Synchronizing the Time on Windows Systems**

Windows systems can use the AT command utility to maintain time synchronization. Note that the Task Scheduler service must be running on the Windows system to use this command.

You can create a .bat (batch) file to add the necessary AT commands to your Windows system. A sample `time_sync.bat` file follows.

```bash
# Time Sync for Windows

@echo off

set /a sec=0
set /a sec=%sec%+1
set /a min=0
set /a min=%min%+1
set /a hour=0
set /a hour=%hour%+1

time /t

time /t

time /t

time /t

```

`echo %sec% %min% %hour% | ntpdate -s...`
Sync_time.bat
@rem The AT command schedules commands and programs to run on a computer at
@rem a specified time and date. The Schedule service must be running to use
@rem the AT command.
@rem
@rem Example of command syntax and how the AT command is used.
@rem AT \[\computername\] [ [id] [/DELETE] | /DELETE [/YES]]
@rem AT [\computername] time [/INTERACTIVE]
@rem       [ /EVERY:date[,]... | /NEXT:date[,]...]] "command"
@rem
@rem To list the tasks scheduled issue AT from a dos cmd window.
@rem C: \> at
@rem Status ID   Day                                 Time          Command Line
@rem
------------------------------------------------------------------------------
@rem             1   Each M T W Th F S Su    9:00 AM       testtime.bat
@rem
@rem To delete the task scheduled issue the following with the ID for the task
@rem C:\> at 1 /delete
@rem NOTE: If there are no tasks scheduled after issuing the AT command then
@rem verify that the task scheduler service is started.
@rem
@rem C:\> at
@rem There are no entries in the list.
@rem
@rem Verify that the Task Scheduler service is started.
@rem
@rem ***********************  COMMAND ***********************
@rem
@at 09:00 /every:su,m,t,w,th,f,s "net time \timeSystem \set /y"

In the file above, replace the value \timeSystem with the hostname of a system in your network that
regularly checks an atomic clock and synchronizes to its time.

If you create a batch file similar to the preceding example, save it in the C:\ directory and run it to schedule
the AT commands to synchronize the time. You can then copy the batch file to each Windows system in
your cluster to synchronize the time.

**Synchronizing the Time on HP-UX Systems**

HP-UX systems use the /etc/ntp.conf file to maintain time synchronization. Your system could have an
ntp.conf file, or you may need to create one using the System Administration Manager (SAM) utility.

A sample ntp.conf file follows. (The actual file is very long; the example below is abbreviated.)

```
# Sample XNTP Configurations File
#
# Use "peer", "server" and "broadcast" statements to specify various time
# server to be used and/or time services to be provided.
# File truncated for readability...

server timeSystem
```

Replace the value \timeSystem with the hostname of a system that regularly checks an atomic clock and
synchronizes to its time.
To synchronize your HP-UX system:

1. Log in as root to an HP-UX system in your cluster.
2. Enter the following command:
   ```bash
   sam
   ```
   If the ntp.conf file is missing or empty, SAM displays the following message:
   ```bash
   ntp.conf does not exist
   ```
4. If your system has an /etc/ntp.conf file, make a backup copy by entering the following:
   ```bash
   cp /etc/ntp.conf /etc/ntp.conf.backup
   ```
5. Using any text editor, create an /etc/ntp.conf file or edit the existing file to match the previous example.
6. Verify that an *xntpd file exists in the /sbin/rc2.d directory. (Note that the file name typically begins with a series of characters such as S660.) Enter the following:
   ```bash
   cd /sbin/rc2.d
   ls
   ```
   **NOTE**
   You should see a series of files including a file named similar to S660xntpd. If you do not have this file, contact your UNIX system administrator.
7. Repeat these steps for each HP-UX system in your cluster.
8. On each cluster member, while logged in as root, run the following commands to confirm that the NTP processes are running:
   ```bash
   ntpq -p
   ```
   The command should display information similar to the following:
   ```bash
   remote refid st t when poll reach delay offset disp
   =========================================================================
   *timeSystem tick.ucla.edu 3 u 55 64 377 0.92 -21.485 2.61
   ```
9. If you do not see the output from ntpq, do one of the following:
   - Reboot the system by entering the following:
     ```bash
     sync; reboot
     ```
   - Enter the following command:
     ```bash
     /sbin/sh /sbin/rc2.d/S660xntpd start
     ```
   - If you need to stop the ntpd process, while logged in as root, enter the following command:
     ```bash
     /sbin/sh /sbin/rc2.d/S660xntpd stop
     ```
Configuring the TCP Ports

The eHealth systems in the cluster communicate with each other using a set of four ports that carry TCP traffic. Two of the ports carry basic messages that pass between systems, and two carry files that pass between systems.

eHealth provides a set of environment variables that specify the numbers that the systems use for each port. If another application or process uses any of the default ports, you can change the port number for all eHealth systems in the cluster. Table 6 describes the ports and the associated environment variables.

Table 6. TCP Ports

<table>
<thead>
<tr>
<th>Port</th>
<th>Description</th>
<th>Environment Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>5050</td>
<td>Remote Message Transfer (In)</td>
<td>NH_RMT_IN_PORT</td>
</tr>
<tr>
<td>5051</td>
<td>Remote Message Transfer (Out)</td>
<td>NH_RMT_OUT_PORT</td>
</tr>
<tr>
<td>5052</td>
<td>Remote File Transfer (In)</td>
<td>NH_RFT_IN_PORT</td>
</tr>
<tr>
<td>5053</td>
<td>Remote File Transfer (Out)</td>
<td>NH_RFT_OUT_PORT</td>
</tr>
</tbody>
</table>

To change the numbers for one or more ports, define the related environment variable for the new port number.

**Caution**

Distributed eHealth cannot operate in a firewall configuration in which the firewall allows responses to outgoing requests only on the same port that the outgoing request used. That is, outgoing requests on port 5051 receive responses on port 5050, and your firewall must support this communication.

Adding Systems to the Cluster

After installing, starting, and licensing the eHealth systems that will be the members of your cluster, you then create the cluster as follows:

1. Add the first member to the cluster.
2. Add the subsequent members to the cluster.
Adding the First Member of a Cluster

The first eHealth system that you add to the cluster must be a trusted system.

To add the first system to the cluster:

1. Log in to the eHealth system as the eHealth administrator.
2. Ensure that all primary and secondary eHealth consoles are closed on the eHealth system.
3. If the eHealth system is a UNIX system, do the following:
   a. Change to the eHealth home directory, where /ehealth represents the pathname of the directory where eHealth is installed:
      ```
      cd /ehealth
      ```
   b. Source the nethealthrc file for your shell environment using one of the commands listed in Table 7.

<table>
<thead>
<tr>
<th>Shell</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bourne</td>
<td>. nethealthrc.sh</td>
</tr>
<tr>
<td>C</td>
<td>source nethealthrc.csh</td>
</tr>
<tr>
<td>Korn</td>
<td>. nethealthrc.ksh</td>
</tr>
</tbody>
</table>

**NOTE**

If you do not source the eHealth resource file, change to the /ehealth/bin directory or specify that directory in your eHealth commands.

4. Enter the following command:

   `nhJoinCluster -firstMember`

**NOTE**

You can specify additional optional arguments. Before you run the command, close the eHealth console, any secondary eHealth consoles, and any open Live Health applications. This ensures that eHealth updates the object IDs on the local system with the cluster information. You can restart these applications after you join the system to the cluster. For the complete syntax of the nhJoinCluster command, refer to the Web Help.

5. At the following prompt, enter the password for the cluster. The password can have from 1 to 8 characters. If you specify more than 8, the command displays an error.

   Enter New Cluster Password:

**NOTE**

As a security precaution, the command never displays the password as you type it. Also, eHealth saves the password in an encrypted format on the eHealth system. When eHealth systems exchange the password, they transfer it in an encrypted format.
6. Confirm the cluster password by entering it again at the following prompt:

   Re-enter New Cluster Password:

   The command then displays the following information:

   Join Cluster initiated:
   Updating local database ...... OK
   Sending machine ID change notification. OK
   Enable scheduled job for cluster admin. OK

   The machine ID change notification is a broadcast message to notify other cluster members of a new
   member and its member ID. Since this is the first cluster member, this message indicates that the local
   system has updated its cluster member table with the new member ID (machine ID).

   **NOTE**
   If you see the message “Sending machine ID change notification. FAILED,” stop and restart the eHealth
   server to ensure that the member table is updated.

   7. At this point, you now have a Distributed eHealth cluster, although it has only one member. Proceed to
   the next section to add additional members to the cluster.

**Adding Additional Members to the Cluster**

After you add the first member to a cluster, you can add additional eHealth systems.

**NOTE**
Always add one eHealth system at a time. To keep the cluster member tables synchronized, allow the
nhJoinCluster command to finish before you run it on another system that you want to add to the cluster.

To add additional systems to the cluster:

1. Log in to the eHealth system that you are adding to the cluster as the eHealth administrator.
2. If the eHealth system is a UNIX system, repeat Step 3 on page 52 to source the nethealthrc resource file.
3. Close the eHealth console, any secondary eHealth consoles, and any open Live Health applications on the
   local system. This ensures that eHealth updates the object IDs with the cluster information. You can
   restart these applications after you join the system to the cluster.
4. Enter the following command, where `hostname` is the hostname of a system that is already a member of
   the cluster that you want to join:

   ```
   nhJoinCluster -host hostname -useStdDac yes
   ```

   The -useStdDac argument specifies whether the system that you are adding should replicate its Standard
   service profile settings to the other cluster members (yes value), or whether it should replace its Standard
   profile with the Standard profile used by the cluster (no value). Within the cluster, the Standard service
   profile *must* have the same values on all cluster members to ensure that the reports from each Distributed
   eHealth System use the same thresholds for data analysis.

   **NOTE**
   You can specify additional optional arguments; for the complete syntax, refer to the Web Help.
5. At the following prompt, enter the cluster password:

   Enter Cluster Password:

The command displays messages that inform you of the status of the join process. When you add a Distributed eHealth Console to a cluster, the messages are similar to the following:

   Join cluster initiated:
   Validating supplied host .. OK
   Adding local machine to cluster .. OK
   Updating local database ...... OK
   Sending machine ID change notification .. OK
   Synchronizing objects of type element .. OK
   Synchronizing objects of type group .. OK
   Synchronizing objects of type groupSection .. OK
   Synchronizing objects of type groupList .. OK
   Synchronizing objects of type groupListContents .. OK
   Synchronizing objects of type serviceProfile.. OK
   Replicating objects of type serviceProfile .. OK
   Replicating objects of type dataAnalysisRec .. OK

These messages inform you that the join process first synchronizes the local (joining) system with the information on the other cluster members. It then replicates the information for the objects on the local system with the other cluster members. For a Distributed eHealth Console, the synchronized information includes elements, groups, group contents, group lists, group list contents, and service profiles. The replication process includes service profile and data analysis record objects. For a description of the information that each type of system replicates, refer to “Replication Overview” on page 19 and “Synchronization Overview” on page 21.

The machine ID change notification is a broadcast message to notify other cluster members of a new member and its member ID.

**NOTE**

If you see the message “Sending machine ID change notification. FAILED,” stop and restart the eHealth server to ensure that the member table is updated.

**Testing Cluster Status**

As you add eHealth systems to the cluster, you can use the Cluster Status utility to check the status of the cluster systems. For more information, refer to “Starting Cluster Status” on page 59.

**Listing the Cluster Members**

You use the nhListClusterMembers command to obtain information about the cluster members. This command displays the member table for one or more eHealth systems in the cluster. The **member table** contains information about each cluster system known to each eHealth system in the cluster.

The command helps you to locate differences in the cluster member tables, and also provides summary information about the cluster members in the table. The following sections provide several examples of the command and its use; for a complete description of the syntax, refer to the Web Help.
Listing the Cluster Members of a Specific System

To display the member table of a specific cluster member, use the -host argument, as follows:

```
nhListClusterMembers -host dracut
Cluster List Members initiated:
Querying cluster .. OK

Cluster Members for host: dracut

MemberId    Name     MemberType  Trusted  IpAddress
+--------+--+-------------+--+---------+--+-----+--+-------------+
739247905   DRACUT    server       Yes  17.1.7.33
Contacts:
1996820652   CHICAGO  console      No   17.1.5.119
```

Listing the Cluster Members of All Systems

To display the member table of all eHealth systems in the cluster, use the -all argument, as follows:

```
nhListClusterMembers -all
Cluster List Members initiated:
Querying cluster .. OK

Cluster Members for host: DRACUT

MemberId    Name     MemberType  Trusted  IpAddress
+--------+--+-------------+--+---------+--+-----+--+-------------+
739247905   DRACUT    server       Yes  17.1.7.33
Contacts:
1996820652   CHICAGO  console      No   17.1.5.119
```

Cluster Members for host: CHICAGO

```
MemberId    Name     MemberType  Trusted  IpAddress
+--------+--+-------------+--+---------+--+-----+--+-------------+
739247905   DRACUT    server       Yes  17.1.7.33
Contacts:
1996820652   CHICAGO  console      No   17.1.5.119
```
Listing the Removed Cluster Members

To display the member table (including removed members) of all eHealth systems in the cluster, use the `nhListClusterMembers` command with the `-all` and `-includeRemoved` arguments, as follows:

```
_nhListClusterMembers -all -includeRemoved
Cluster List Members initiated:
Querying cluster .. OK

Cluster Members for host: DRACUT

<table>
<thead>
<tr>
<th>MemberId</th>
<th>Name</th>
<th>MemberType</th>
<th>Trusted</th>
<th>IpAddress</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>739247905</td>
<td>DRACUT</td>
<td>server</td>
<td>Yes</td>
<td>17.1.7.33</td>
<td>No</td>
</tr>
</tbody>
</table>

Contacts:

<table>
<thead>
<tr>
<th>MemberId</th>
<th>Name</th>
<th>IP Address</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996820652</td>
<td>CHICAGO</td>
<td>17.1.5.119</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Cluster Members for host: CHICAGO

<table>
<thead>
<tr>
<th>MemberId</th>
<th>Name</th>
<th>MemberType</th>
<th>Trusted</th>
<th>IpAddress</th>
</tr>
</thead>
<tbody>
<tr>
<td>739247905</td>
<td>DRACUT</td>
<td>server</td>
<td>Yes</td>
<td>17.1.7.33</td>
</tr>
</tbody>
</table>

Contacts:

<table>
<thead>
<tr>
<th>MemberId</th>
<th>Name</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996820652</td>
<td>CHICAGO</td>
<td>17.1.5.1192</td>
</tr>
</tbody>
</table>

Listing Only the Cluster Member Names

To display only the names of the systems in the member table (not any additional details), use the `-nameOnly` argument, as follows:

```
_nhListClusterMembers -all -includeRemoved -nameOnly
Cluster List Members initiated:
Querying cluster .. OK

Cluster Members for host: aladdin

aladdin
chicago (removed)

Modifying a Cluster Member

You can modify the following types of information about one or more cluster members using the `nhModifyClusterMember` command:

- **Member name.** Modify a member name if the hostname of the eHealth system changes.
- **Member IP address.** Only modify the member IP if the IP address of the eHealth system changes.
- **Contact information.** Specify information about the owners or persons responsible for managing the cluster system.
- **Trusted setting.** Modify a cluster member’s status as either a trusted or non-trusted member of the cluster.
You must run the command on a trusted member of the cluster to change information about any other cluster members. If you run it on an untrusted cluster member, you can change the information for only the local (untrusted) cluster member.

**NOTE**

You cannot change the member ID of a cluster member.

When you modify member information, you change the information in the member table on one or more cluster members. Note that if you change member table information on one system, eHealth propagates those changes to the member tables of all other active (non-removed) members in the cluster.

**Modifying the Information on One Cluster Member**

To modify the information about a cluster member in the member table on a specific cluster member, use the following command:

```
nhModifyClusterMember -host aladdin -name chicago -trusted yes
Cluster Modify Member initiated:
Sending modifications . OK
```

This example changes the trusted setting to yes for the cluster member chicago in the member table of the system named aladdin.

**Modifying the Information on All Other Cluster Members**

To modify the information about a cluster member in the member table on all other cluster members (except the one on which you run the command), use the following command:

```
nhModifyClusterMember -cluster -name chicago -trusted yes
Cluster Modify Member initiated:
Sending modifications . OK
```

**Modifying the Information on All Cluster Members**

To modify the information about a cluster member in the member table on all cluster members (including the one on which you run the command), use the following command:

```
nhModifyClusterMember -all -name chicago -trusted yes
Cluster Modify Member initiated:
Sending modifications . OK
```

**Removing a Cluster Member**

You can remove an eHealth system from the member table of one or all cluster members. A removal is not actually a deletion; it is a change in status. When you remove an eHealth system, the removed system will not share information with any cluster member that does not show it as an active member in its member table. If you remove a system from the member table of all systems in a cluster, you essentially remove the system as a participating member of the cluster.
Note the following impacts when you remove a cluster member:

- **When you remove a Distributed eHealth Console,** eHealth updates all scheduled data analysis jobs in the Distributed eHealth Systems in the cluster. The data analysis jobs will no longer analyze data for the scheduled reports on the removed system. The time that the data analysis jobs run may also change, if the earliest scheduled reports were on the removed system.

- **When you remove a Distributed eHealth System,** eHealth removes all elements owned by that system from all Distributed eHealth Consoles in the cluster. In addition, eHealth updates the data analysis job on the removed system to coincide with the earliest scheduled report on the removed system. The removal process does not update the member table of the cluster member that you are removing.

Although you can remove an eHealth system from the member table on one cluster member, you should remove the system from all cluster members to keep the cluster member information synchronized. If you remove a system from only one cluster member’s table, that cluster member does not share information with the removed system; however, the other cluster members continue to replicate and synchronize information.

**NOTE**

You can restore a removed system using the nhRestoreClusterMember command.

**Removing a System from All Cluster Members**

To remove the system named detroit from the member tables of all systems in the cluster (including the one on which you run the command), use the following command:

```
nhRemoveClusterMember -all -name detroit
```

This is the recommended removal procedure. When you remove a cluster member, remove it from all cluster members.

**Restoring a Removed System**

If you remove a system from a member table of one or all systems in the cluster, you can restore it at any time. Although you can restore an eHealth system to the member table on one cluster member, you should restore the system to all cluster members to keep the cluster member information synchronized.

**NOTE**

After you restore a system, you should also run nhSynchronize and then nhReplicate to update the restored system with the cluster information.

**Restoring a System in All Cluster Member Tables**

To restore the system named detroit in the member tables of all the systems in the cluster (including the one on which you run the command), use the following command:

```
nhRestoreClusterMember -all -name detroit
```

This is the recommended restore procedure. When you restore a cluster member, restore it to all cluster members. If you restore a system such as a Distributed eHealth Console or a system that has been removed for some time, the restore process could cause the system to resynchronize with the cluster.
Managing the Distributed eHealth Cluster

This chapter describes how to manage the eHealth systems in a cluster as well as the elements monitored by the Distributed eHealth Systems in the cluster.

Managing the Cluster Membership

Over time, your cluster membership could change with the addition of new members and the removal of old (no longer used) members. For a complete description of cluster membership tasks, refer to Chapter 3, “Creating the Distributed eHealth Cluster.”

Monitoring Cluster Status

You use the eHealth Cluster Status utility to monitor the status of the eHealth systems in your cluster. The Cluster Status utility is a tool that you access from the eHealth Web interface Admin page. It provides the status and identification information for each eHealth system in the cluster.

Starting Cluster Status

You access the Cluster Status utility from the eHealth Web interface Admin page.

To run Cluster Status:

1. Log in to any eHealth system in the cluster as the web administrator.
2. If necessary, select the Admin tab.

The browser opens a new window named eHealth Cluster Status, as shown in Figure 15. The window refreshes every five minutes.

Figure 15. eHealth Cluster Status
The eHealth Cluster Status window shows the following information:

- **Status.** Displays an icon that shows the status of the eHealth system. The status can be one of the following:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>The eHealth system is reachable and the eHealth server is running.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>The eHealth system is down. This image (a flashing red light) indicates that the eHealth system is unreachable or that the eHealth server has stopped.</td>
</tr>
</tbody>
</table>

Hover the mouse pointer over the status icon to display the date and time of the last good status. For example:

the last good status: 09-24-01 15:10:49

- **Members.** Displays the member name and IP address of the eHealth system. eHealth obtains the list of cluster members from the member table of the eHealth system from which you ran Cluster Status.

- **Type.** Specifies the type of eHealth system, either **System** (Distributed eHealth System) or **Console** (Distributed eHealth Console).

The Cluster Status window displays the elements in order by IP address. If one or more systems show a down status, those systems appear first in the list, followed by the eHealth systems that have a good status.

**NOTE**

If all eHealth systems show an alarm icon, check the eHealth system from which you launched Cluster Status first to determine if any servers or services have stopped, or if its network connection is down.

---

**Monitoring the Processes and Log Files**

eHealth Release 6.0 requires that you monitor the eHealth logs and processes using the features of eHealth SystemEDGE installed on each eHealth system. After you install the agent on your eHealth system, you use the nhAddSysEdgeMonEntries command to configure SystemEDGE monitoring of the critical eHealth processes and logs. This command also causes SystemEDGE to monitor the eHealth system log (system.log) and notify you when any scheduled jobs have been running for more than six hours. For instructions on installing the agent, refer to the *eHealth SystemEDGE User Guide*.

**Monitoring Logs**

The SystemEDGE agent includes a log monitoring capability that can identify user-specified strings or expressions in log files. When the agent finds a matching string or expression, it can alert you with a trap or other notification process. For a Distributed eHealth system, you use the nhAddSysEdgeMonEntries command to configure the agent to monitor the log files described in Table 9.
Table 9. Log Files

<table>
<thead>
<tr>
<th>Log File Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate_baseline</td>
<td>Contains information about the Live Exceptions Baseline job which calculates a baseline of data for Live Exceptions alarm processing.</td>
</tr>
<tr>
<td>Conversation_rollup</td>
<td>Contains information about the daily database rollups for conversations elements (Traffic Accountant) that helps reduce the eHealth database size.</td>
</tr>
<tr>
<td>Data_analysis</td>
<td>Contains information about the daily analysis of data used in scheduled reports such as Health, Service Level, and MyHealth reports.</td>
</tr>
<tr>
<td>Database_save</td>
<td>Contains information about the Database Save scheduled job that creates a backup of your database on a regular basis.</td>
</tr>
<tr>
<td>Delete_old_reports</td>
<td>Contains information about the Delete Old Reports job that removes old report directories and old files in the /ehealth/tmp directory.</td>
</tr>
<tr>
<td>Find_Cluster_Differences</td>
<td>Contains information about differences in object information among cluster members.</td>
</tr>
<tr>
<td>License</td>
<td>Contains information about the eHealth license manager processes.</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Contains information about the eHealth Maintenance default job, which resets the eHealth system once each week.</td>
</tr>
<tr>
<td>Statistics_index</td>
<td>Contains information about the Index Stats job that indexes database tables of polled data that are currently not indexed.</td>
</tr>
<tr>
<td>Statistics_rollup</td>
<td>Contains information about the daily database rollups for statistics elements (all others except Traffic Accountant) that help reduce the eHealth database size.</td>
</tr>
<tr>
<td>Synchronize</td>
<td>Contains information about the scheduled Synchronization job that runs on Distributed eHealth Consoles.</td>
</tr>
<tr>
<td>System</td>
<td>Contains information written to the eHealth console system messages area.</td>
</tr>
</tbody>
</table>

You specify the action to take when the agent finds the following words in any of the log files:

<table>
<thead>
<tr>
<th>Term</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>A failed process or problem.</td>
</tr>
<tr>
<td>warning</td>
<td>A problem that might have affected the process or its outcome, but probably did not cause a failure</td>
</tr>
<tr>
<td>info</td>
<td>Text that should be reviewed or brought to an administrator’s attention</td>
</tr>
<tr>
<td>unknown</td>
<td>An unknown file, directory, or subject that could not be resolved with the information provided</td>
</tr>
</tbody>
</table>

**Note**

The nhAddSysEdgeMonEntries command configures eHealth SystemEDGE to monitor the default log files for these specific string cases; however, you can use SystemEDGE commands and utilities to increase the monitoring capabilities. For more information, refer to the *eHealth SystemEDGE User Guide*. 

*Distributed eHealth Administration Guide*
Monitoring Processes

SystemEDGE also includes a process and service monitoring capability that you can use to configure the agent to monitor processes, services, and applications on a system. In a Distributed eHealth environment, you configure the agent on each eHealth system to monitor critical eHealth processes and alert you when any of the processes stop. For a Distributed eHealth system, you use the nhAddSysEdgeMonEntries command to configure the agent to monitor the processes listed in Table 10:

Table 10. Processes

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Definition</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>nhiCfgServer</td>
<td>eHealth configuration manager</td>
<td>Handles changes in the poller configuration and for objects such as elements, groups, and group lists.</td>
</tr>
<tr>
<td>nhiDbServer</td>
<td>eHealth database server</td>
<td>Handles requests to and from the database.</td>
</tr>
<tr>
<td>nhiHttpd</td>
<td>eHealth web server process</td>
<td>Manages requests for the eHealth Web interface.</td>
</tr>
<tr>
<td>nhiHttpSvc</td>
<td>eHealth web server service on Windows systems</td>
<td>Manages requests for the eHealth Web interface.</td>
</tr>
<tr>
<td>nhiLiveExSvr</td>
<td>Live Exceptions server</td>
<td>Manages the processes and messages sent to the Live Exceptions application.</td>
</tr>
<tr>
<td>nhiMsgServer</td>
<td>eHealth message server process</td>
<td>Handles the passing of messages to various processes throughout the eHealth system.</td>
</tr>
<tr>
<td>nhiPoller</td>
<td>Statistics poller</td>
<td>Collects data at regular intervals from the SNMP statistics elements in your poller configuration. The Statistics Polling Status dialog monitors its activity.</td>
</tr>
<tr>
<td>nhiPoller -dlg</td>
<td>Conversations poller</td>
<td>Collects data at regular intervals from the SNMP conversations elements (probes) in your poller configuration. The Conversations Polling Status dialog monitors its activity.</td>
</tr>
<tr>
<td>nhiPoller -import</td>
<td>Import poller</td>
<td>Collects data at regular intervals from the imported elements in your poller configuration. The Import Polling Status dialog monitors its activity.</td>
</tr>
<tr>
<td>nhiPoller -live</td>
<td>Live Trend fast sampling poller</td>
<td>Collects the data at 30-second intervals for elements that are monitored by Live Trend as fast-sampled elements.</td>
</tr>
<tr>
<td>nhiReplServer</td>
<td>eHealth replication server process</td>
<td>Controls the replication processes for cluster members.</td>
</tr>
<tr>
<td>nhiRftIn and nhiRftOut</td>
<td>Remote File Transfer (RFT) processes</td>
<td>Send files to and receive files from the other cluster members. They are part of the RCS interface.</td>
</tr>
<tr>
<td>nhiRmtIn and nhiRmtOut</td>
<td>Remote Message Transfer (RMT) processes</td>
<td>Send messages to and receive messages from the other cluster members. They are part of the RCS interface.</td>
</tr>
<tr>
<td>nhiServer</td>
<td>main eHealth server process</td>
<td>When you stop the eHealth server using the nhServer command or the Console → Stop Server menu, or you stop the eHealth Service, you stop this process. When it stops, it also stops a variety of related subprocesses.</td>
</tr>
</tbody>
</table>
In addition, you can configure the agent to monitor scheduled jobs and alert you when jobs have been running for more than six hours. The nhAddSysEdgeMonEntries command configures SystemEDGE to monitor the default processes and react when any of them stops; however, you can use SystemEDGE commands and utilities to increase the monitoring capabilities. For more information, refer to the eHealth SystemEDGE User Guide.

**Configuring SystemEDGE on a Cluster Member**

After you install SystemEDGE on each eHealth system in the cluster, you must configure each agent to monitor the eHealth logs and processes and take action when specific events occur. You use the nhAddSysEdgeMonEntries command to configure the agent. If you have purchased the eHealth AdvantEDGE View product for managing your SystemEDGE agents, you can use it to configure the SystemEDGE agents that monitor your cluster members. For more information, refer to the AdvantEDGE View User Guide.

The command adds entries to the sysedge.cf file which is located in /etc on UNIX systems and %SystemRoot%\system32 on Windows systems. The command also stops and restarts the SystemEDGE agent to effect the changes.

**To run the nhAddSysEdgeMonEntries command:**

1. If the eHealth system is a UNIX system, do the following:
   a. Log in to the eHealth system as root. (Only the root user can configure the SystemEDGE agent.)
   b. Open a terminal window and change to the eHealth installation directory as follows:

   ```
cd /ehealth
   ```
   c. Source the nethealthrc resource file for your shell type. For an overview of the commands for each shell environment, refer to Table 7 on page 52.

2. If the eHealth system is a Windows system, log in as the eHealth administrator and open a DOS command window.

3. In the command/terminal window, use the following command:

   ```
   nhAddSysEdgeMonEntries arguments
   ```

**NOTE**

You can also use the nhRunCommand to run the nhAddSysEdgeMonEntries on more than one system in a cluster at the same time. For example:

```
nhRunCommand -all -cmd "nhAddSysEdgeMonEntries args"
```

Several examples of the command follow.
**Example 1.** Send traps to destination 1.2.3.4 when a critical process stops or when a log file contains the word *error*:

```
nhAddSysEdgeMonEntries -trapList 1.2.3.4 -trapExpr "error"
```

The command outputs the following information:

```
Stopping SystemEDGE Management Agent
SystemEDGE Management Agent starting.
```

**Example 2.** Send e-mail to account username@mycompany.com when a critical process stops or when a log file contains the word *error* or *unknown*:

```
nhAddSysEdgeMonEntries -mail username@mycompany.com -mailExpr "error, unknown"
```

**NOTE**

If your eHealth system is a Windows system, you must define the NH_NT_SMTP_SERVER environment variable to allow eHealth processes to send e-mail. If you did not set the variable during eHealth installation, refer to the eHealth Installation Guide for the Windows platform.

**Example 3.** Send traps to destination 1.2.3.4 when a critical process stops or when a log file contains the word *error*; send e-mail to account username@mycompany.com when a critical process stops or when a log file contains the word *unknown*:

```
nhAddSysEdgeMonEntries -trapList 1.2.3.4 -trapExpr "error" -mail username@mycompany.com -mailExpr "unknown"
```

**Monitoring the Health of the Distributed eHealth Cluster**

If you have an eHealth — System and Application license, you can use System Health reports to monitor the health and performance of the eHealth systems in your cluster. System Health reports provide you with detailed performance information on the system resource utilizations (for CPUs, disks, partitions, network I/O, and so on), availability and latency information, and other critical metrics that can help you to manage and monitor your Distributed eHealth systems.

**Discovering the eHealth Systems**

On one of the Distributed eHealth Systems in your cluster, you should create a file of the IP addresses of the eHealth systems in the cluster. You can then discover your eHealth systems using that file of IP addresses. You can also use the file for scheduled discover processes to keep your eHealth system element information up-to-date.

Although you can discover the eHealth systems from all Distributed eHealth Systems in your cluster, you should monitor them from just one Distributed eHealth System. (This ensures that you do not use multiple poller licenses to monitor the same system elements, and it helps reduce the duplicate elements in your cluster.)
Creating a Group of the Cluster Members for Reporting

After you discover your eHealth systems as system elements, create a group to organize them for reporting. You can also use that group to monitor your eHealth systems using Live Exceptions and Live Status (which you must run from the eHealth system that polls those elements). For more information about Live Health, refer to the Web Help.

Managing the Distributed eHealth Console

The following sections describe the tasks that you should follow to manage the Distributed eHealth Console systems in your eHealth cluster.

Managing Web User Accounts

You (or an eHealth web administrator) manage the web accounts for the users who will access your Distributed eHealth Consoles to run or view reports. If you use unsecured (or public) web access, you do not need to create individual web accounts. However, if you use secure access, you must create accounts for each user (or group of users) who will access the site.

If you have several Distributed eHealth Consoles for your cluster, you can configure them to have the same web account information so that users may access any of them, or you can configure unique accounts at each site to ensure that users go to a specific Distributed eHealth Console to access eHealth. For instructions, refer to the Web Help.

Best Practice: Using DCI Tools to Manage Web Accounts

If you have a large number of web accounts to create and manage, consider using the eHealth database configuration information (DCI) tools to import and export web account information. For example, you can create a file of web account information and then import that file into each eHealth system that requires it.

To create the initial web account DCI file, you can do one of the following:

- Use OneClick for eHealth to create the web accounts on one system. You can then export the web account information to a file that you can retain, update, and import to other systems as necessary. You can also export web account information and all web reports for web users from one eHealth system directly to a Distributed eHealth Console.

- Create the web account file by following the format and instructions in the eHealth Integration Guide, and then import that file as necessary to the eHealth systems.

A file of web account information is also helpful as a centralized administration point. Rather than edit the accounts on each eHealth system that has the account, you can modify the file and import the account information to each eHealth system as necessary. You can also use rules and data files to tailor the import process. For example, if you assign different web accounts to each Distributed eHealth Console, you can use rules and data files to ensure that you import the correct accounts to each eHealth system.

Monitoring Disk Space and System Performance

For a Distributed eHealth Console, the web report directories are typically the most dynamic disk space consumers. If you schedule reports for your web accounts and you allow some or all of your web users to run reports on demand, regularly monitor the disk space to ensure that the Distributed eHealth Console does not run out of report directory space.
You can use the Delete Old Reports scheduled job to remove report directories and temporary files older than a specified number of days. By default, the job removes reports that are older than 31 days, and temporary files in the `ehealth/tmp` directory that are older than four days. You can change these defaults to ranges that better match your policies or disk space.

For Distributed eHealth Consoles, the CPU resources are also a performance consideration. When a web user runs a Trend or At-a-Glance report, for example, the eHealth web server creates a list of the elements that the user may select as a subject. If the user has permission to see a large number of elements, such as 5000 or more, it can take a few seconds to display that list. If many users are logged in and running reports on demand, the Distributed eHealth Console could encounter performance issues. It may need more CPU resources or more performance to support a large number of web users who can each view a large number of elements.

**NOTE**

For a description of reports and report options, refer to the *eHealth Report Management Guide*.

eHealth offers a variety of tools that can help you to monitor disk space and CPU resources and notify you *before* performance becomes a problem. You can configure the SystemEDGE agent to monitor these performance indicators and notify you when the disk space or CPU resources reach a certain utilization percentage, such as 80%. If you have eHealth — System and Application and you discover your cluster members (as described in “Monitoring the Health of the Distributed eHealth Cluster” on page 64), you can use At-a-Glance, Trend, and Top N reports to monitor these usage metrics over time as well. You can run System Health reports to monitor historical trends and the health of your eHealth cluster members.

**Adding Another Distributed eHealth Console**

You must have at least one Distributed eHealth Console for every set of 250,000 elements that you are monitoring in your cluster. You can add more to provide access to more web users, faster local access for a geographically dispersed cluster, or to customize the Distributed eHealth Console Web interfaces for your users.

To add another console, you install eHealth on that system and add licenses. You can then join the system to the cluster. The Distributed eHealth processes will update the Distributed eHealth Console with the object information. You can then add web accounts to the system and schedule reports.

**Managing the Distributed eHealth Systems**

The following sections describe management tasks that you should follow for the Distributed eHealth Systems in your eHealth cluster.

**Monitoring Disk Space**

Since the Distributed eHealth Systems are polling systems (as well as local reporting systems), you should monitor the disk space usage in a manner similar to that for a standalone eHealth system. For information on how to manage the size of your Oracle eHealth database, refer to the *eHealth Database Management Guide*.

**Monitoring Reports and Data Analysis**

Typically, distributed report generation is a transparent process to the administrator of a Distributed eHealth System. For any scheduled reports that you create, you should monitor the log files to ensure that the reports are finishing and that elements are not missing due to timeouts from cluster members.
Recovering a Cluster System

If the system hardware for a cluster system fails, you can recover the system if you have a recent backup of the database. For a complete description of database backup and loading processes, refer to the *eHealth Database Management Guide*. Always schedule regular database backups. If possible, save the database on a separate system that is also backed up to ensure that your backups are protected from system failures.

**CAUTION**

When you load a database on a cluster system, make sure that you use a database backup created on the same type of cluster system. That is, always load a Distributed eHealth Console with a database saved on a Distributed eHealth Console. Never load a Distributed eHealth Console with a database created on a Distributed eHealth System.

Also, avoid creating or modifying any global objects in the cluster while the database load and update is in progress. Changes to objects may be lost (overwritten) based on the timing of the change and the progress of the synchronization.

After you load the database on the cluster member, follow these steps to ensure that the system synchronizes correctly with the other cluster systems.

**To synchronize a cluster system after loading a database:**

1. Log in to the cluster system as the eHealth administrator.
2. Start the eHealth server. Select **Start → Server** on the eHealth console.

   **NOTE**
   
   Do not create or update any objects on this system yet.

3. Enter the following to update the member table on the local cluster system with the latest member tables from the other cluster members:

   ```bash
   nhUpdateClusterTable -cluster
   ```

4. Enter the following to verify that the local system has information about the other cluster members:

   ```bash
   nhListClusterMembers -all
   ```

5. Enter the following to update the database IDs table on the local system with the latest information from the other cluster members:

   ```bash
   nhUpdateDatabaseIds -cluster
   ```

6. By default, the Standard service profile currently used by the cluster systems replaces the Standard service profile on the local system. If you want all cluster systems to use the Standard service profile on the local system, enter the following:

   ```bash
   nhPutFile -cluster -fromFile
   '$(NH_HOME)/reports/dataAnalysis/standard.1.10000.dac' -toFile
   '$(NH_HOME)/reports/dataAnalysis/standard.1.10000.dac' -overwrite
   ```

   **NOTE**
   
   Type the command on one line. This example shown here cannot fit on one line on the printed page.
7. Enter the following to update all other cluster members with the information on the local system:

```
nhReplicate -cluster -objTypeAll -allElements
```

**CAUTION**

After loading a database, always run nhReplicate before nhSynchronize to ensure that you update the cluster members first, before you update the local system with the cluster information.

8. Enter the following to update the local system with the object information from the other cluster members:

```
nhSynchronize -cluster -objTypeAll -allElements
```

**NOTE**

This step recovers any objects created on the local system between the time when the backup was created and the time that the system returned to service after the database load.

You should also run the nhFindClusterDiffs command to find any objects that are not synchronized within the cluster.

### Moving a Database to a New Cluster System

You can move a database saved on one cluster system to another cluster system. For example, you might move a database to a new system when you want to move eHealth to a new, higher-performance system. In a Distributed eHealth environment, the process of moving the eHealth database to a new system requires that you prepare the new system, load the saved database from the old system, and update the cluster and object information.

**To move an eHealth database to a new system:**

1. Save the database of the existing cluster system by doing the following:
   a. Enter the following to perform an ascii save:

```
nhSaveDb -ascii -p D:\ehealth\db\savedb\asciiSave.tdb
```
   b. Save the *.tdb file to a location on the new system such as D:\ehealth\db\savedb.

For a complete description of database backup tasks, refer to the eHealth Database Management Guide.

2. Install eHealth on the new system.

**CAUTION**

Make sure that you configure the new system type correctly; that is, make sure that you configure the system as a Distributed eHealth Console if that is the type of system that you are replacing. Also, contact Technical Support to change your eHealth licenses so that they will be valid on the new system.

3. Start eHealth on the new system and enter your licenses. Use an ASCII load to load the saved database that you created in Step 1. For example:

```
nhLoadDb -ascii -p D:\ehealthdb\savedb\asciiSave.tdb
```

The load process updates the database to reflect the new system information (that is, the new member name).
4. Export web users, passwords, and web reports from the old system to the new system:
   a. On the old system, enter the following:

```
    nhExportWebAcct -dciOut users.dci -reportsFile reports.zip -passwd
```

   b. Copy users.dci and reports.zip to the new system. On the new system, enter the following:

```
    nhImportWebAcct -dciIn users.dci -reportsFile reports.zip
```

5. Start the eHealth server on the new system by running nhServer start or selecting Start → Server on the eHealth console.

6. On the new system, enter the following, where hostname is the hostname of the new eHealth system:

```
    nhListClusterMembers -host localhost
```

   In the command output, note the IP address of the new eHealth system.

7. As the eHealth administrator, log in to another eHealth system in the cluster and enter the following, where oldName is the old name of the eHealth system, hostname is the name of the new system, and ip is the IP address of the new system:

```
    nhModifyClusterMember -all -name oldName -newName hostname -ipAddress ip
```

   This command updates the member tables on all cluster members with the new member name (hostname) and IP address of the new cluster system.

8. On the new cluster system, enter the following to update the member table on the local system with the information from the other cluster members (this command should be run on any other system in the cluster except the new system):

```
    nhUpdateClusterTable -cluster
```

9. On the new system, enter the following to verify that the local system has information about all other cluster members:

```
    nListClusterMembers -all
```

10. By default, the Standard service profile that is currently used by the cluster systems will replace the Standard service profile on the local system. If you want all cluster systems to use the Standard service profile on the local system, enter the following:

```
    nhPutFile -cluster -fromFile
    $(NH_HOME)/reports/dataAnalysis/standard.1.10000.dac' -toFile
    $(NH_HOME)/reports/dataAnalysis/standard.1.10000.dac' -overwrite
```

   **NOTE**

   Type the command on one line. The command example shown here cannot fit on one line on the printed page.

11. On the new system, enter the following to replicate the object information on the local system to the other cluster members:

```
    nhReplicate -cluster -objTypeAll -allElements
```
CAUTION

After loading a database, always run nhReplicate before nhSynchronize to ensure that you update the cluster members first, before you update the local system with the cluster information.

12. On the new system, enter the following to update the local system with the object information from the other cluster members:

   `nhSynchronize -cluster -objTypeAll -allElements`

Managing the Objects in the Cluster

This section describes the following administrative practices for managing the objects in your cluster:

- Finding duplicate objects
- Identifying the Distributed eHealth System that “owns” an element (the element’s home)

Finding Duplicate Objects in the Cluster

Since you can discover elements from any Distributed eHealth System in the cluster, you could discover the same elements from different Distributed eHealth Systems. Although it is not prohibited, this is not a recommended practice. You should monitor a resource from only one Distributed eHealth System to ensure that you do not use multiple poller licenses for the same resource and that you avoid duplicate data within the cluster. For more information about duplicate objects, refer to “Naming Rules, Duplicates, and Uniqueness” on page 36.

Also, you may have duplicate objects (that is, objects that have the same name but that really are different objects) within your cluster. For example, when you add a new cluster member, it could have groups, group lists, or service profiles that have the same names as groups, group lists, or service profiles that are already in the cluster.

You can use the nhFindDupObjects command to identify duplicate objects in your cluster. You should run this command periodically, as well as after you make changes such as running a discover process or adding a cluster member.

For example, to find duplicate elements in a cluster:

   `nhFindDupObjects -all
Cluster Find Duplicates initiated:
Querying cluster .. OK
  finding duplicate elements on aladdin DONE
  finding duplicate elements on DRAcUT .. DONE
  finding duplicate group names on local machine DONE
  finding duplicate group list names on local machine DONE
  finding duplicate service profile names on local machine DONE
database-sh,dracut,09-27-01 14:39:54
database-sh,aladdin,09-28-01 13:56:17
database-sh-enet-port-2,dracut,09-27-01 14:39:54
database-sh-enet-port-2,aladdin,09-28-01 13:56:17
database-sh-cpu-1,dracut,09-27-01 14:39:55
database-sh-cpu-1,aladdin,09-28-01 13:56:17
database-sh-c:,dracut,09-27-01 14:39:56
database-sh-c:,aladdin,09-28-01 13:56:17
database-sh-d:,dracut,09-27-01 14:39:56
database-sh-d:,aladdin,09-28-01 13:56:17`
### Managing the Objects in the Cluster

In this example, the command found six duplicate elements for a system named database-sh and several of its subcomponents (a CPU, an Ethernet interface, and several disks and partitions). It identified that the duplicate elements occurred on the Distributed eHealth Systems named aladdin and DRACUT. The command also shows when the elements were last updated on each Distributed eHealth System.

Although aladdin appears to have the most recent update, you should run a Trend report for the last month for each system element to determine which Distributed eHealth System has the longest baseline of data. If the Distributed eHealth Systems have the same amount of data, you should remove the duplicate elements from the Distributed eHealth System with the oldest element information (in this case, DRACUT). If a Distributed eHealth System clearly has a longer history of data for the duplicate elements, you should rediscover the elements on that Distributed eHealth System and then delete the elements on the Distributed eHealth System that has less history for the elements.

The example command did not find any duplicate group, group list, or service profile names. However, if you run the command and encounter duplicate names, you should rename one of the objects to ensure that all groups, group lists, and profiles have unique names within the cluster.

### Identifying Where an Element is “Homed”

From any eHealth system in the cluster (Distributed eHealth Console or Distributed eHealth System), you can identify which Distributed eHealth System is polling and retaining data for an element. This can be helpful if you want to log in to that Distributed eHealth System to perform management tasks such as deleting the element, modifying the element information, adding the element to a group, or running local (non-distributed) reports or Live Trend applications for the element. You can use locate an element in a cluster by using the **Find Element Location** feature on the Web user interface or by running the `nhFindElement` command.

#### Using the Find Element Location Feature

The Find Element Location feature of the eHealth Web interface allows users who are logged in as the eHealth web administrator (the `admin` user) to look for elements within the cluster. It identifies which Distributed eHealth System is polling and collecting data for the element. You can access the Find Element Location feature from the Admin and Organization pages.

**To use Find Element Location on the Admin page:**

1. Log in to the Web interface of any eHealth system in the cluster as the `admin` user; then select the **Admin** tab.
2. Click **Site Management** → **Find Element Location**.
3. In the Find Element Location window (Figure 16), perform a **manual** search or go to Step 4 to perform an express search. Select an element associated with a specific technology by doing one of the following:
• Optionally, filter the list by specifying an element name or a wildcard character in the Element filter field; then select an element from the list.

• Select a technology from the Technology list. The elements associated with that technology appear in the Available elements list. Select an element.

eHealth automatically begins searching for elements, and a page similar to the following appears:

<table>
<thead>
<tr>
<th>ELEMENT NAME</th>
<th>MACH NAME (TYPE)</th>
<th>ELEM TYPE</th>
<th>IP ADDRESS</th>
<th>DATE LAST MODIFIED</th>
</tr>
</thead>
<tbody>
<tr>
<td>aladdin-EHsystem1</td>
<td>aladdin (server)</td>
<td>genericLand</td>
<td>172.1.7.33</td>
<td>01-08-03 17:42:02</td>
</tr>
</tbody>
</table>

The page displays the name of the element, the Distributed eHealth System that owns the element, the element type, the IP address associated with the element (or its parent), and the date the element was last modified.

4. Perform an express search. Do one of the following in the Single-/Multi-Element Express Search field:

• Specify a specific element name; then click Find. The system searches all Distributed eHealth Systems within the cluster and finds all elements that match that name exactly.

• Specify an element name (or portion of a name) with a wildcard; then click Find. For example, if you specify AUG*, the system searches all Distributed eHealth Systems within the cluster and finds all elements with a name that starts with the letters AUG.

**NOTE**
This feature supports the use of the asterisk (*) symbol as a wildcard.

eHealth displays a page that lists all elements that matched the search criteria. The page also displays a message to identify the number of elements that were found at each eHealth system, as follows:

- star (server) had 117 matches out of 2206 total elements.
- sea (server) had 0 matches out of 29,876 total elements.
- sky (server) had 12 matches out of 23,444 total elements.

To use the Find Element Location feature on the Organization page:

1. Log in to the Web interface of any eHealth system in the cluster as the admin user.

2. Click the Organization tab.

3. Click Elements, and navigate through the element list until you select the element that you want to find.

4. In the right frame, click the Find Element Location icon at the top.
Using nhFindElement

You use the nhFindElement command to identify the Distributed eHealth System that owns an element. For example, to identify the Distributed eHealth System where the element Scout7300-seg-1 is homed:

```
nhFindElement -name Scout7300-seg-1
```

```
searching...

ELEMENT NAME       MACH NAME(TYPE)   ELEM TYPE   IP ADDRESS   DATE LAST MODIFIED
Scout7300-seg-1    aladdin(server)  ethernetIf   12.14.15.1    09-20-01 18:03:21

ERRORS:
NONE
```

The command output shows that the element is homed on the Distributed eHealth System named aladdin.

You can specify an asterisk (*) or a percent sign (%) as a wildcard character. For example, if you specify the string Boston* or Boston%, the command locates all elements that have element names that begin with the characters Boston (but not boston or BOSTON).

**NOTE**

On UNIX systems, if you use the asterisk as a wildcard, you must enclose the value of -name in double quotation marks (for example, "Boston*").

If you run the command and specify an element name that has duplicate entries on several Distributed eHealth Systems, the command lists all Distributed eHealth Systems that own the element, as shown in the following example:

```
nhFindElement -name DATABASE-SH
```

```
searching...

ELEMENT NAME       MACH NAME(TYPE)   ELEM TYPE   IP ADDRESS   DATE LAST MODIFIED
DATABASE-SH        aladdin(server)  sysEdgeNtSystem 1.16.5.24 09-28-01 13:56:17
DATABASE-SH        DRACUT(server)   sysEdgeNtSystem 1.16.5.24 09-27-01 14:39:54
```

Managing Security for the Cluster

Within a Distributed eHealth cluster, consider the following security precautions.

- The cluster password prevents an unauthorized eHealth system from joining the cluster. An eHealth administrator must enter the password for the nhJoinCluster command to succeed.
- If your policies are to change passwords at a regular interval (such as every month or after a certain number of days), you can also change the cluster password using the nhSetClusterPassword command at a similar interval.

**NOTE**

After you change the cluster password, you should restart the eHealth server on each cluster member to ensure that the new password takes effect.

- Secure links such as VPN connections between the eHealth systems in the cluster add another level of security to help prevent unauthorized access to the eHealth systems.
- For any cluster members that are outside the firewall or within a DMZ, consider making those systems non-trusted systems to help prevent anyone from running commands on those systems and that affect the other cluster members.
Upgrading the Cluster Systems

If you upgrade to a newer eHealth release, you must upgrade all eHealth systems in the cluster to the new release. For a description of the upgrade process and requirements, refer to the eHealth Installation Guide for your operating system platform.

Installing Service Packs and Certification Releases

If you install an eHealth service pack or certification release, you must install the service pack or release on all eHealth systems in the cluster. For a description of the service pack contents and how to install the service pack, refer to the readme file provided in the service pack kits.
eHealth uses the processes of replication and synchronization to maintain the latest information on each eHealth system in a cluster. This chapter describes how to manage the replication processes and the best practices for avoiding problems.

**About Replication and Synchronization**

eHealth uses replication to “push” information about the objects defined within an eHealth system in the cluster to one, more, or all other eHealth systems in the cluster. The synchronization process “pulls” information from one or more of the other cluster systems to update the local system. The replication and synchronization processes are essentially the same process—but they occur as a result of different commands. Replication occurs when a local system updates one or more other systems. Synchronization occurs when the local eHealth system updates itself with information obtained from one or more other eHealth systems.

---

**NOTE**

Throughout this chapter and guide, the word replication generally refers to the process of updating object information on one or more eHealth systems.

---

The replication and synchronization processes can occur in the following ways:

- **Immediate replication** – Changes to certain object types on the local eHealth system cause it to immediately update the other cluster members.
- **Scheduled synchronization** – Two jobs that run each day on a Distributed eHealth Console to update itself with information from the other cluster members for any objects that are not immediately replicated or that have somehow changed since the last synchronization.
- **Manual replication/synchronization** – Updates on-demand that you initiate for files that are not automatically replicated or if you do not want to wait for the scheduled Synchronization job.

The following sections describe the immediate, scheduled, and manual replication processes and how to manage them.
About Immediate Replication

eHealth immediately replicates information for the following object types when you add, modify, rename, or delete them:

- Groups
- Group lists
- Group list members
- Service profiles
- Data analysis records (which result when you add or modify a scheduled report on a Distributed eHealth Console)

These objects are **global objects** because their information is shared with all eHealth systems in the cluster. When you save the changes for any one of these global objects, eHealth replicates the information immediately to all other eHealth systems in the cluster. However, eHealth does not show empty groups for reporting on a Distributed eHealth Console.

- If you create and populate a group on a Distributed eHealth System, you can immediately see the group on another Distributed eHealth System and modify it.
- If you immediately try to select the same group on a Distributed eHealth Console when setting up a report, you will not see the group.

The group does, in fact, replicate immediately, but it is not immediately available for reporting until the nhSynchronize system job runs and populates the nh_group_members table on the Distributed eHealth Console. If you need to make a group available immediately for setting up a report, run nhSynchronize -cluster from the Distributed eHealth Console. This command populates the nh_group_members table on the Distributed eHealth Console with information about the new group’s members from the Distributed eHealth Systems.

**NOTE**

After you delete a global object, you cannot immediately add an object with the same name. For guidelines, refer to "Replication and Soft-Deletion" on page 83.

Monitoring Immediate Replication

When you create or modify a global object, eHealth displays a progress indicator to show you the replication status. For example, if you use the Edit Service Profile dialog to create a new service profile, when you click **OK**, the Update Status dialog appears.

![Update Status dialog](image)
The Update Status dialog displays a progress bar to show you the status of the replication. eHealth contacts each eHealth system in the cluster and attempts to update its object information. If the local eHealth system cannot contact another eHealth system in the cluster, the replication request times out for that remote system. The message “Unable to connect to port number” appears at the bottom of the dialog. When all systems have replied, or when all timeouts have expired, the progress bar shows 100% and the dialog disappears.

**Understanding Timeouts**

Timeouts during replication and synchronization processes can occur for any of the following reasons:

- The eHealth server is stopped on the remote cluster member.
- The eHealth cluster member is unreachable due to network issues.
- The remote eHealth system is down.

When a replication or synchronization process encounters a timeout on a remote system, it displays a message in the Update Status dialog for each system that did not respond. (If all remote systems do not respond, you might first check the network connection for your local system to determine if it is down.) Use the Cluster Status tool to determine whether the local system can communicate with the other systems in the cluster. For instructions on increasing the replication timeout, refer to “Replication Timeout Errors” on page 101.

If any of the connections or remote systems are down, correct the problems with the eHealth systems and/or the network connections between them, and use the nhReplicate command to update the remote systems again.

**Avoiding Simultaneous Changes to Objects**

Because the replication occurs immediately, avoid making simultaneous changes to any global object from different Distributed eHealth Systems. For example, if two administrators change a group’s name at the same time from different Distributed eHealth Systems, the simultaneous changes could cause the cluster information for that group to become unsynchronized, or “out of sync.” eHealth replicates the changes as soon as you save them; thus, the replications could overwrite one another. The last saved change for an object is always considered the most current information; eHealth overwrites any previous changes.

**NOTE**

To avoid editing conflicts, establish policies among the administrators to ensure that changes to global objects occur on only one Distributed eHealth System, preferably the system on which the object was originally created. You can also establish naming conventions that indicate the Distributed eHealth System on which an object was created, and establish policies that modifications or deletions should occur only on the Distributed eHealth System on which the object was created.

**About Scheduled Synchronization**

eHealth does not immediately replicate information for elements and group members (elements assigned to groups) when you add, modify, or delete them. Since elements and group members are known only to the eHealth system on which they are created and the Distributed eHealth Consoles—these objects are not considered global objects.
After you discover new elements, modify or delete elements, or change group memberships, and you save the changes locally (on the Distributed eHealth System that you are using), eHealth does not replicate the changes to the Distributed eHealth Consoles in the cluster. Instead, Distributed eHealth Consoles use the scheduled Synchronize job and Element Synchronize job to obtain these changes during the day.

By default, the Synchronize job runs at 11 P.M. each day to gather these changes from the Distributed eHealth Systems in the cluster. When the job runs, each Distributed eHealth Console sends a request to the other eHealth systems in the cluster to obtain the latest object information that they have. Each eHealth system responds with its information. You can change the time when the Synchronize job runs to select a low-activity time on the eHealth system.

By default, the Element Synchronize job runs first at 3:30 A.M. each day, and then every four hours thereafter, to gather new, modified, and deleted element information and changes in group membership information from the other eHealth systems in the cluster. When the job runs, each Distributed eHealth Console sends a request to the Distributed eHealth Systems in the cluster to obtain the element and group membership changes that have occurred since the last Element Synchronization. Each eHealth system responds with its information. You can change the time and the frequency of the Element Synchronize job to ensure that the Distributed eHealth Consoles are kept current with the changes in element information.

**To modify either scheduled job:**

1. Select Setup → Schedule Jobs on the console.
2. In the Schedule Jobs dialog, select the job name from the list of jobs in the Application column; then click Modify.
3. Change the time of day when the job runs and click OK.
4. Click OK in the Schedule Jobs dialog.

**Understanding the Update Delay on Distributed eHealth Consoles**

Since changes to elements and group memberships do not replicate immediately, a delay could occur between the time that you make these changes on a Distributed eHealth System, and the time that the changes appear at the Distributed eHealth Consoles in your cluster. Any interfaces that display lists of elements such as the Organization page and the subject fields of Run Reports pages for example, will not reflect the latest changes until after the scheduled Element Synchronize job runs at the Distributed eHealth Consoles.

If you make extensive changes to elements or group memberships at a Distributed eHealth System, and you want the Distributed eHealth Consoles to reflect those changes immediately, you can manually replicate the changes to the Distributed eHealth Consoles.

**About Manual Replication**

Rather than wait for the scheduled Synchronize or Element Synchronize job to run each day, you can run a manual replication to send changes to one or more eHealth systems in the cluster. For example, if you discover new elements, modify the information, and assign them to groups, you might want to send these changes to the Distributed eHealth Consoles immediately. Or, if a previous replication reported timeout messages, you might want to replicate the changes on demand to update one or more eHealth systems that did not receive the new information.
You use the nhReplicate command to send the changes from a local eHealth system to one or all of the other eHealth systems in the cluster. (You could also use the nhSynchronize command to update the local system with information from one or more remote eHealth systems in the cluster.) The following section describes how to manually replicate information.

**NOTE**

By default for best performance, nhReplicate sends only the element information that has changed (new, modified, or deleted elements) since the last time that the element information was replicated. You can force the nhReplicate command to replicate all element information, regardless of the last time it changed, by including the \-allElements argument.

### Manually Replicating Information

You use the nhReplicate command to manually replicate information from the local eHealth system to one or more other eHealth systems in the cluster.

**To use the nhReplicate command:**

1. Log in as the eHealth administrator to the eHealth system that has the latest information for an object.
2. If the eHealth system is a UNIX system, source the nethealthrc resource file for your shell environment. For an overview of the commands, refer to Table 7 on page 52.
3. Enter the following command:
   
   ```
   nhReplicate arguments
   ```

4. For example, if you recently created a group on a Distributed eHealth System and received messages that a remote system named DRACUT was not updated, you could run the following command:

   ```
   nhReplicate -host DRACUT -objType group
   Cluster Replicate initiated:
   Replicating objects of type group .. OK
   ```

   This command updates the group names on DRACUT. If DRACUT is a Distributed eHealth Console, you can run the command with the groupSection value to update the group names and the information for the elements assigned to the groups, as follows:

   ```
   nhReplicate -host DRACUT -objType groupSection
   Cluster Replicate initiated:
   Replicating objects of type groupSection .. OK
   ```

**NOTE**

Run the command from the eHealth system that has the latest object information.
Manually Synchronizing Information

You use the nhSynchronize command to manually synchronize the information on the local eHealth system with information from one or more other eHealth systems in the cluster.

**NOTE**

By default for best performance, nhSynchronize requests only the element information that has changed (new, modified, or deleted elements) since the last time that the element information was synchronized on a Distributed eHealth Console. You can force the nhSynchronize command to synchronize all element information, regardless of the last time it changed, by including the -allElements argument.

To use the nhSynchronize command:

1. Log in to the eHealth system that you want to update as the eHealth administrator.
2. If the eHealth system is a UNIX system, source the nethealthrc resource file for your shell environment. For an overview of the commands, refer to Table 7 on page 52.
3. Enter the following command:

   `nhSynchronize arguments`

4. For example, if you want to obtain the latest group information from a Distributed eHealth System named DRACUT, you could run the following command:

   `nhSynchronize -host DRACUT -objType group`

   Cluster Sync initiated:
   Synchronizing objects of type group .. OK

**NOTE**

Run the command from the eHealth system that you want to update.

Using Replication Log Information

When replication processes update an eHealth system, eHealth writes information to the system log, `/ehealth/log/system.log`.

For example, when the local eHealth system receives new or changed object information from another cluster member, the following message appears:

Local configuration change...

Finding Differences in Object Information

You should regularly monitor the object information in your cluster and check for differences in object information among the cluster members. Differences can result from incomplete replications or synchronizations, as well as from conflicting edits on global objects.

For example, if two eHealth administrators modify a service profile at the same time on different cluster members, and their changes are different, the profile becomes inconsistent within the cluster. As another example, if an eHealth administrator deletes a group but connections to some of the cluster members are down, or very slow, the replication might not be able to update all cluster members with the information. In both cases, the cluster members have different object information.
eHealth provides a scheduled Find Cluster Differences job that, by default, runs each day at 12:00 P.M. (midnight) and runs again every four hours after that. This scheduled job runs the nhFindClusterDiffs command and identifies object information that varies on one or more eHealth systems in the cluster. You can also run the nhFindClusterDiffs command at any time to find cluster differences.

You can change the time that the job first runs, and how often during the day that the job repeats. If your eHealth cluster usually does not change object information frequently during the day, you might configure the job to run only once each day. If your eHealth cluster has frequent changes to objects during the day, you should run the job more frequently to identify any differences more quickly.

### Reviewing the Scheduled Job Log Files

The Find Cluster Differences job creates a log file each time it runs. The log file, named `Find_Cluster_Differences.jobId.log`, resides in the `/ehealth/log` directory. Each time the job runs, it copies the current log file to `Find_Cluster_Differences.jobId.log.bak` and creates a new `Find_Cluster_Differences.jobId.log` file.

A sample `Find_Cluster_Differences.jobId.log` file follows.

```
-----
Job started by Scheduler at '10/02/2001 08:00:23 AM'.
-----

-----
$NH_HOME/bin/sys/nhiFindClusterDiffs -cluster
-----

Cluster Find Differences initiated:
Searching for cluster differences between objects of type group.... OK
Searching for cluster differences between objects of type groupList.... OK
Searching for cluster differences between objects of type groupListContents.... OK
Searching for cluster differences between objects of type dataAnalysisRec.... OK
Searching for cluster differences between objects of type serviceProfile.... OK
*******************************************************************************

nhFindClusterDifferences Report

Machine Run On:               aladdin             10/02/2001 08:00:39 AM

Report Summary table:          |
-------------------------------|-----------------------------------------------------|
Object Type                  | Machines | Objects                     |
-------------------------------|----------|----------------------------|
|                               |          | Polled | Responded | Processed | Need Update |
-------------------------------|----------|--------|-----------|-----------|-------------|
dataAnalysisRec               | 3        | 3      | 0         | 0         |
Element                        | 3        | 3      | 0         | 0         |
Group                          | 3        | 3      | 10        | 0         |
GroupSection                   | 3        | 3      | 0         | 0         |
GroupList                      | 3        | 3      | 1         | 0         |
GroupListContents              | 3        | 3      | 1         | 0         |
serviceProfile                 | 3        | 3      | 1         | 0         |

*******************************************************************************

-----
Scheduled Job ended at '10/02/2001 08:00:39 AM'.
-----
```
In the sample log file, the Find Cluster Differences job did not detect any differences in the object information. (Note that the Need Update column indicates that none of the eHealth systems in the cluster need to be updated.) Thus, a log of this type indicates that the cluster members have the same object information.

The following sample log file shows a difference in object information.

-----
Job started by Scheduler at '10/02/2001 12:00:23 AM'.
-----

-----
$NH_HOME/bin/sys/nhiFindClusterDiffs -cluster
-----

Cluster Find Differences initiated:
Searching for cluster differences between objects of type group.... OK
Searching for cluster differences between objects of type groupList.... OK
Searching for cluster differences between objects of type groupListContents.... OK
Searching for cluster differences between objects of type dataAnalysisRec.... OK
Searching for cluster differences between objects of type serviceProfile.... OK
*******************************************************************************
Machine Run On:               aladdin             10/02/2001 12:00:37 PM
Report Summary table:
-----------------------------+-------------------+-----------------------------
|     Machines      |          Objects |
| Object Type                  | Polled   Responded|  Processed    Need Update |
|-----------------------------+-------------------+-----------------------------
dataAnalysisRec                     3         3             0             0
| element                             3         3             0             0
| group                               3         3             6             1
| groupSection                        3         3             0             0
| groupList                           3         3             1             0
| groupListContents                   3         3             1             0
| serviceProfile                      3         3             1             0
|-----------------------------+-------------------+-----------------------------
*******************************************************************************
-------------------------------------------------------------------------------
Object Type:                  group
Machines Polled:              3
Responses Received:           3
Object(s) Processed:          6
Object Update(s) Required:    1
-------------------------------------------------------------------------------
Object Name:                  TestRtrAladdin [ Router/Switch ]
Object Id:                    739247905@1000006
(+ 1 other)
Machine(s) to Update:         DRACUT              <No Objec>
*******************************************************************************
-----
Scheduled Job ended at '10/02/2001 12:00:39 AM'.
-----

In this sample log file, the group named TestRtrAladdin is out-of-date on the eHealth system named DRACUT. Two other systems in this cluster share the same information about the group, but the system named DRACUT does not.
Correcting Cluster Differences

To correct the cluster differences identified in the Find Cluster Differences log file, you need to update the systems that are out-of-sync with information from systems that have current information. For the sample log in the previous section, you must update the system named DRACUT with the group information. From the system named aladdin (which has the latest version), run the following nhReplicate command:

```
nhReplicate -host DRACUT -objType group
```

The following messages appear:

```
Cluster Replicate initiated:
Replicating objects of type group .. OK
```

Replication and Soft-Deletion

With Distributed eHealth, you should familiarize yourself with a concept for objects called soft-deletion. When you delete an object such as a group, service profile, or group list, the deletion process immediately removes the object from your menus, subject selections, and other visible interface areas. The replication process also updates the other eHealth systems in the cluster to remove the deleted object. However, a record of the object remains in the eHealth database until the scheduled FSA Scrubber job runs. It removes temporary files in the FSA as well as the soft-deleted objects that still have entries in the database. (You can also run the nhScrubFsa command on demand to remove the soft-deleted objects or the FSA files.)

If you try to create a new object that has the same name as a soft-deleted object in the database, eHealth displays the following error:

```
Database error: object has been marked for deletion. Choose another name.
```

To avoid this problem, specify a different object name or run the nhScrubFsa command to remove the soft-deleted objects. After the command finishes, save the new object again; the save should finish without error.

By default, the command removes all objects that are a week or more older. However, you can set the value of the NH_DB_DELETED_OBJECTS_TIME2KEEP environment variable to the number of hours that you want to retain soft-deleted objects. You can also use the -deletedObjectHours argument to specify the age of the files that should be removed. For example, to delete all soft-deleted objects older than four hours, use the following command:

```
nhScrubFsa -deletedObjectHours 4
```

To delete all soft-deleted objects, use the following command:

```
nhScrubFsa -deletedObjectHours 0
```

Configuring the Replication Timeout

By default, the eHealth replication server waits up to 5 minutes for a response from another cluster member’s replication server. You can use the NH_REPL_MSG_TIMEOUT environment variable to configure how long the replication server should wait. The variable specifies the time in seconds that the server should wait. The default is 300 seconds (5 minutes).
Managing Distributed Reports

When you run a report from a Distributed eHealth Console in an eHealth cluster, the report is a distributed report. A **distributed report** is a standard eHealth report that contains elements managed by one or more Distributed eHealth Systems in the cluster. Since the Distributed eHealth Console does not have any elements of its own, any report that you run from it is a distributed report, even if the report is for only one element.

This chapter describes the process and concepts associated with distributed reporting, and how to administer and manage the reporting tasks. For a detailed description of a report type, refer to the Web Help for reports.

**Generating Distributed Reports**

Distributed reports are *identical* to the standard eHealth reports. You generate them the same way, either by scheduling them from the Schedule Jobs dialog in the console, or running them on-demand from the console dialogs and Web interface Run Report pages. They have the same appearance and contents, and the same drill-down capabilities when viewed from the eHealth Web interface.

However, you can generate distributed reports only from a Distributed eHealth Console. If you generate a report from a Distributed eHealth System in the cluster, the report contains only the elements that are owned (homed) by that Distributed eHealth System.

**Note**

eHealth imposes several restrictions and limitations on distributed reports. For example, you cannot generate a distributed Traffic Accountant report, nor can you run distributed Live Trend or Live Status applications. To generate Traffic Accountant reports or monitor elements using Live Trend or Live Status, you must log in to the specific Distributed eHealth System that manages those elements and generate them from the Distributed eHealth System.

**Running Distributed Reports On-Demand**

This section describes the processes and steps that occur when you generate an eHealth report on demand from the Distributed eHealth Console. The example shows a Top N report, but the process is similar for all distributed report types.
To run a distributed Top N report:

1. Use your web browser to log in to the eHealth web server running on a Distributed eHealth Console in your cluster.
2. Select the Run Reports tab.
3. Under Available Reports in the left frame, click Standard under Top N.
4. On the Run a Standard Top N Report page, select the subject of the report, time frame options, and other information as needed.

    The Groups list includes all groups created at any of the Distributed eHealth Systems in the cluster. Note that you can see only those groups that match your web user account subject filtering options. (That is, if your account can view only those groups named Sales*, the list contains only the groups with names that begin with the characters Sales.)
5. Click Generate Report to create the Top N report.

Processing the On-Demand Report

When you generate a report on demand from the Distributed eHealth Console, eHealth performs the steps described in Table 11.

Table 11. Running Distributed Reports on Demand  (Page 1 of 2)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. User runs a report from the Distributed eHealth Console.</td>
<td>Distributed eHealth Console</td>
</tr>
<tr>
<td>2. The Distributed eHealth Console sends a data specification information (DSI) file to each Distributed eHealth System. (For reports run against a group or group list, all Distributed eHealth Systems receive a DSI file. For element reports, only the Distributed eHealth Systems that own any elements in the report receive the DSI file.) The DSI file provides the information that the Distributed eHealth System needs to generate the requested report for its elements.</td>
<td>Distributed eHealth Systems</td>
</tr>
<tr>
<td>3. Each Distributed eHealth System runs the report to generate the data for its elements. Each Distributed eHealth System sends WIP messages to the Distributed eHealth Console. It sends the report data when finished.</td>
<td>Sends work-in-progress (WIP) messages, and the report data when complete</td>
</tr>
</tbody>
</table>
Table 11. Running Distributed Reports on Demand  (Page 2 of 2)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td>The Distributed eHealth Console collects all report data from each Distributed eHealth System. It collates the data and creates the final report.</td>
</tr>
</tbody>
</table>

**Scheduling Distributed Reports**

This section describes the processes and steps that occur when you schedule an eHealth report from the Distributed eHealth Console. The example shows a Health report, but the process is similar for all distributed report types.

**To schedule a distributed Health report:**

1. Log in to the eHealth system as the eHealth administrator.
2. Schedule a report using one of the following methods:
   - Use the nhSchedule command to schedule the Health report from the command line.
   - Start the eHealth console (redirecting it to your workstation if you are logged in remotely), and use the Schedule Jobs dialog to schedule the Health report.

   For a description of these methods, refer to the *eHealth Scheduled Job Management Guide*.

The distributed report sends updated data analysis records to the Distributed eHealth Systems. The data analysis records specify when the Data Analysis job should run on a Distributed eHealth System; the records cause each Distributed eHealth System to adjust the job run time, if necessary, so that it runs before the earliest scheduled Health, Service Level, or MyHealth report. (This step is omitted if you scheduled a report that does not use data analysis such as Trend, At-a-Glance, What-If, or Top N.)

**NOTE**

The scheduled Data Analysis job creates the analyzed data for any scheduled Health, Service Level, or MyHealth reports before those reports run. The pre-analysis reduces the time required for the scheduled report to run.
### Processing the Scheduled Report

When you schedule a report from the Distributed eHealth Console, eHealth performs the steps described in Table 12.

**Table 12. Scheduling Distributed Reports**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>User schedules a report from the Distributed eHealth Console.</td>
</tr>
<tr>
<td>2.</td>
<td>If the scheduled report is a Health, Service Level, or MyHealth report, the Distributed eHealth Console sends a data analysis record to each Distributed eHealth System.</td>
</tr>
<tr>
<td>3.</td>
<td>Each Distributed eHealth System compares and adjusts the Data Analysis job to ensure that it starts before the earliest scheduled Health, Service Level, or MyHealth report each day.</td>
</tr>
<tr>
<td>4.</td>
<td>The scheduled report runs at its designated time. The process is identical to Steps 2 through 4 in Table 11 on page 86.</td>
</tr>
</tbody>
</table>

### Using the Large-Scale Reports

eHealth has three default reports that are designed for large-scale groups and group lists:

- StandardLite Health report
- ITManagerTop100 Service Level report
- ServiceCustomerTop100 Service Level report

A **large-scale** group or group list is one that contains thousands of elements (that is, 20,000 or more). These large-scale reports are variations of the existing default reports (Standard Health, ITManager, and ServiceCustomer, respectively). To improve report performance and usability, they reduce the number of elements shown in certain charts or tables that, in the existing default reports, display *all* elements. These improvements help the reports to run faster, and the resulting reports are easier to scan since they are much shorter in length. The following sections describe each large-scale report.

**NOTE**

You can continue to use the existing Standard, ITManager, and ServiceCustomer reports in large-scale and distributed environments. However, use caution when running these reports on a large number of elements. They could take many hours to finish and could contain several thousand pages.
StandardLite Health Report

The large-scale StandardLite report is based on the Standard Health report. It does not include some sections that show element-by-element summaries for all elements in the report. This report differs from Standard Health in the following ways:

- It displays the Element Top N section, but it does not display the Element Detail section.
- It does not include the Supplemental section.

ITManagerTop100 Service Level Report

The large-scale ITManagerTop100 report is based on the ITManager Service Level report. However, the ITManagerTop100 report differs in that the Element Summary chart shows only the top 100 elements in the group list rather than all elements in the group list.

ServiceCustomerTop100 Service Level Report

The large-scale ServiceCustomerTop100 report is based on the ServiceCustomer Service Level report. However, the ServiceCustomerTop100 report differs in that the Element Summary charts show only the top 100 elements in the group list rather than all elements in the group list.

The Element Summary charts consist of one or more charts that show an element-by-element summary of the critical performance variables. In a ServiceCustomer Service Level report, these charts span many pages to display each element in the group list. In the ServiceCustomerTop100, these charts span only a few pages to show the top 100 elements in the group list.

The Heartbeat Timer

The Distributed eHealth Console uses a heartbeat timer to continuously check the status of a distributed report. When a distributed report runs, either on demand or scheduled, the Distributed eHealth Console maintains a constant communication with each Distributed eHealth System to ensure that the system is responding and that the report generation is progressing.

If a report fails at a Distributed eHealth System, or if the Distributed eHealth System stops responding, the heartbeat timer informs the Distributed eHealth Console. Thus, the Distributed eHealth Console will not wait in a loop for report data that it will never receive. When one or more of the Distributed eHealth Systems do not return their data for a distributed report, the report includes a note that indicates how many of the elements are missing from the report. You can configure the heartbeat timer values. For more information, refer to “Configuring the Heartbeat and Message Limit Timers” on page 98.

The Impact of Missing Elements on Reports

If a distributed report does not include all elements in the subject, the report is incomplete. For example, if you run a report for a group of elements that are managed by different Distributed eHealth Systems, and one or more of the systems do not respond with their report data, the resulting report only includes the data for the elements on the Distributed eHealth Systems that responded.

The missing elements result in totals (such as total volume, or total errors) that are lower or smaller than they should be, and Top N or Leaders charts that do not show elements that normally would have been listed. The report data is potentially inaccurate because it did not include the data for the missing elements.
When distributed reports indicate that elements are missing, you should use caution when interpreting the results of the report. You should first correct the cause of the missing elements (by fixing the links to cluster members, or restarting cluster members that may have stopped) and then rerun the distributed reports to ensure that they contain the complete data for all elements in the subject.

**Important Information about Distributed Reports**

As you manage and administer distributed reports, note the following important details about the process:

- The Distributed eHealth Console displays WIP indications while the report runs. WIP always shows the progress of the slowest Distributed eHealth System. While other Distributed eHealth Systems could complete their tasks quickly, the report does not finish until the last Distributed eHealth System has responded with its data.

  **NOTE**
  
  eHealth shows report WIP in two ways: for reports run from the console, the WIP is the Generating Report dialog; for reports run from the Web, the WIP is a page that displays the message “The report is running” and a progress bar.

- For reports that take one or more elements as subjects (such as element Trend, What-If, and At-a-Glance reports), the Distributed eHealth Console sends the DSI file to only the Distributed eHealth Systems that own any of the element subjects.

- For reports that take groups or group lists as subjects (such as Health, Top N, Service Level, and Group Trend reports), the Distributed eHealth Console sends the DSI file to all Distributed eHealth Systems in the cluster. Each Distributed eHealth System processes the DSI file, but if it does not have any elements in the group or group list, it returns a completion message to the Distributed eHealth Console that does not include any report data.

- The report does not finish until the Distributed eHealth Systems that received a DSI file have responded.

  For a description of how eHealth monitors the status of the report, refer to “The Heartbeat Timer” on page 89.

**Understanding Distributed Report Performance**

In normal situations, a distributed report takes less time to complete than a standalone-system version of the report for the same number of elements. Because each Distributed eHealth System generates the report only for those elements that it owns, report generation occurs in parallel and each Distributed eHealth System returns its results to the Distributed eHealth Console.

Distributed reports could take longer to complete if a Distributed eHealth System is overutilized and the data generation process is taking longer than usual. Since the Distributed eHealth Console waits for each Distributed eHealth System to respond, the report will not complete until the last Distributed eHealth System returns its data (or a timeout occurs for Distributed eHealth Systems that have not responded).
About Time Zones

If your cluster member systems use more than one time zone, this section describes some important concepts and operations that relate to using Distributed eHealth.

Generating Reports from the Distributed eHealth Console

When you run a report from the Distributed eHealth Console, eHealth offsets the report periods for each Distributed eHealth System to ensure that the report data reflects the same hours requested at the Distributed eHealth Console. Figure 15 illustrates a sample cluster that has members in different time zones across the United States.

![Figure 15. Cluster with Members in Different Time Zones](image)

On the Distributed eHealth Console, a user runs a Top N report for the group named Corporate for yesterday. The Corporate group has members on both the San Diego and Boston systems.

When the Distributed eHealth Console sends the report request to the eHealth Distributed eHealth Systems, the Boston system runs its report for the hours of 1:00 A.M. yesterday to 12:59:59 A.M. today. The San Diego system runs its report for the hours 10:00 P.M. two days ago to 9:59:59 P.M. yesterday to match the requested report period in Chicago. Thus, when the report data returns to the Distributed eHealth Console and the report finishes, the report contains data for the same time range throughout the cluster.

**Note**

When you run a report for yesterday, the report period is 12:00 A.M. to 11:59:59 P.M. of the previous day.
Distributed Health Report Changes

The Distributed Health report determines the Prior Rank column of the Leaders charts differently. This chart is typically in the Top Ten section of a Health report. Figure 16 shows a sample Volume Leaders chart from a distributed System Health report.

### Volume Leaders in Bytes

<table>
<thead>
<tr>
<th>Rank</th>
<th>Element Name</th>
<th>Speed</th>
<th>Volume</th>
<th>Bandwidth</th>
<th>Health Index</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Speed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>sample-policy-200k</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
<tr>
<td>2</td>
<td><strong>VLAN-policy-1</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
<tr>
<td>3</td>
<td><strong>sample-policy-200k</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
<tr>
<td>4</td>
<td><strong>sample-policy-200k</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
<tr>
<td>5</td>
<td><strong>sample-policy-200k</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
<tr>
<td>6</td>
<td><strong>sample-policy-200k</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
<tr>
<td>7</td>
<td><strong>sample-policy-200k</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
<tr>
<td>8</td>
<td><strong>sample-policy-200k</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
<tr>
<td>9</td>
<td><strong>sample-policy-200k</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
<tr>
<td>10</td>
<td><strong>sample-policy-200k</strong></td>
<td>100.0</td>
<td>6.0</td>
<td>0.1%</td>
<td>1.2%</td>
</tr>
</tbody>
</table>

* Prior rank was greater than 100.

#### About Service Profiles

Health, Service Level, and MyHealth reports use service profile settings to calculate Health Index and exception processing, as well as trends and response limits.

When you create or edit a service profile on an eHealth system in the cluster, eHealth replicates the information to the other cluster members. If you change the service profile settings for a profile that is used by scheduled reports, your changes could affect the appearance of the data in future reports. (The analyzed data for the days prior to the profile change could be very different from the analyzed data for the days following the change.) When you change a service profile, the reports include a note that indicates the date on which the profile (or the analysis settings) changed.
To avoid inconsistent data in your reports, you can use the \texttt{nhDeleteAnalyzedData} command to delete the analyzed data for one or more service profiles on one or more cluster systems. (You should use the -all option to ensure that the command runs on all \textit{eHealth} systems in the cluster.) The next time that the scheduled Data Analysis job runs on each \textit{eHealth} System, it creates new analyzed data for the scheduled reports that use those service profiles; when the scheduled reports run, the data in the reports reflects the same service profile settings.

When you run the command, you must specify the date \textit{after} which you want to delete the analyzed data. The command deletes the analyzed data starting with the day after the date you specify. To avoid inconsistent data in reports with a long baseline, you should specify a date early enough to delete the analyzed data for the entire baseline of your scheduled reports.

If your reports include any days of data prior to or including the -afterDate value, the reports continue to show the data note that indicates the date on which the profile (or the analysis control settings) changed. In this case, the note uses the date from which \textit{eHealth} deleted the analyzed data.

**Troubleshooting Report Failures**

Table 13 describes some of the problems that might cause a distributed report to fail.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report error on Distributed \textit{eHealth} Console</td>
<td>Usually a result of report problems. If you are running a default/standard \textit{eHealth} report, contact Technical Support for assistance. If you created a custom report (using the \textit{eHealth} Developer’s Kit), look for errors output by the report.</td>
</tr>
<tr>
<td>Unable to contact remote systems</td>
<td>The Distributed \textit{eHealth} Console cannot contact any Distributed \textit{eHealth} Systems in the cluster. Use the \textbf{Cluster Status} feature to check the network connections between the Distributed \textit{eHealth} Console and the other cluster systems to determine if the connections are down.</td>
</tr>
<tr>
<td>No data for report</td>
<td>The distributed report does not have any data. This problem can occur when the subject of the report resides on one or more Distributed \textit{eHealth} Systems that are down or unreachable, or that do not have licenses to run the specific type of report.</td>
</tr>
</tbody>
</table>

**Rerunning Failed Reports**

If an on-demand or scheduled report fails at the Distributed \textit{eHealth} Console, you can re-run the report. (If a problem with the network connections or the report code caused the report to fail, correct the problem before running the report again.)

If you are generating an At-a-Glance report for multiple elements (a feature available only from the \textit{eHealth} Console), note that you could receive a failed report error if the report fails for one or more elements in the subject. When you run a multi-element At-a-Glance report, \textit{eHealth} generates a report for each element. If a Distributed \textit{eHealth} System does not respond, the At-a-Glance report fails for any elements that it owns. However, the other At-a-Glance reports (for elements on other Distributed \textit{eHealth} Systems) could succeed. To alert you to the failure for some of the elements, \textit{eHealth} displays a report failure message in the Generating dialog for the report. You should correct the problems for the missing elements and re-run the At-a-Glance reports for only the elements that failed.
Troubleshooting Missing Information

If one or more Distributed eHealth Systems do not respond with data for a distributed report, the distributed report finishes, but it displays the following note to inform users that not all elements are included in the report:

Missing data for \( n \) of \( m \) elements.

The value \( n \) is the number of elements that are missing from the report, while \( m \) is the total number of elements that should be in the report. (If you run a report for parent elements, such as systems, the total number of elements also includes the subcomponents of the elements, such as CPUs, disks, interfaces, and partitions.)

If all elements are missing from a report (for example, an At-a-Glance report for an element on a Distributed eHealth System that is down), the report will fail and display a failure message to indicate the reason.

Distributed eHealth Systems might not return data for any of the following reasons:

- The connection to the Distributed eHealth System is down.
- The Distributed eHealth System is down, the eHealth server has stopped, or it was removed from the cluster.
- The Distributed eHealth System encountered an error while processing the data for the report.
- The Distributed eHealth System does not have a required report license (such as for Health or Service Level reports).

For a description of how missing elements affect reports, refer to “The Impact of Missing Elements on Reports” on page 89.

Configuring Distributed Report Timeouts and Controls

When you run a distributed report from a Distributed eHealth Console, eHealth uses a series of report timeouts to monitor the progress of the report. These timeouts control how eHealth determines when a report has failed and notifies the user accordingly. In addition, eHealth provides compression controls to help reduce the time it takes to transmit the report data from a Distributed eHealth System back to the Distributed eHealth Console. For instructions to set environment variables, refer to the Web Help.

These timeouts have default values that reflect the expected latency and response over T1 or higher speed connections. However, if your eHealth cluster uses slower links, you may need to increase these timeout values. You can configure these timeouts and report controls using environment variables on your cluster systems.

Figure 17 shows the various processes and actions controlled by the report variables. Table 14 on page 96 describes the variables in detail.
1. User runs distributed report.

2. Report is sent to Distributed eHealth Systems and acknowledged within NH_DRPT_RUN_CMD_TIMEOUT.

3. Console checks for returned report data every NH_DRPT_MAINLOOP_INTVL.

4. Console checks each System for report progress at NH_DRPT_HEARTBEAT_INTVL.

5. System responds to status within NH_DRPT_STATUS_MSG_TIMEOUT.

5a. If System does not respond to status after NH_DRPT_MISSED_MSG_LIMIT+1, the report data is considered lost.

6. System returns report data either compressed or uncompressed (NH_DRPT_COMPRESS_RDI).

7. Once the Console receives all report data, or the Console has timeouts for any non-responding systems (5a), the report generation continues and produces the final report.

Figure 17. Distributed Report Timeouts and Controls
**Distributed eHealth Console Variables**

Table 14 describes the environment variables that you can specify on the Distributed eHealth Consoles in your cluster to tailor the distributed report timeouts.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH_DRPT_RUN_CMD_TIMEOUT</td>
<td>Time, in seconds, to send a report request with a DSI file from a Distributed eHealth Console to a Distributed eHealth System. For a description, refer to Step 2 in Table 11 on page 86. To estimate this time, calculate the time that it takes to FTP a file of approximately 100 kilobytes (KB) from a Distributed eHealth Console to a Distributed eHealth System. If the timer expires without an acknowledgement from a Distributed eHealth System, the distributed report shows a “missing elements” error message (as described in “The Impact of Missing Elements on Reports” on page 89). If the timer expires for all Distributed eHealth Systems in the report, the report fails with an error. The default is 45 seconds. The valid values are 3 seconds up to a maximum of 2700 seconds (45 minutes). The value should be greater than the value of NH_DRPT_STATUS_MSG_TIMEOUT and less than the value of NH_DRPT_HEARTBEAT_INTVL.</td>
</tr>
<tr>
<td>NH_DRPT_HEARTBEAT_INTVL</td>
<td>Interval, in seconds, between the heartbeat timer checks. This variable specifies how often the Distributed eHealth Console checks each Distributed eHealth System to make sure that it is still running (processing) data for a distributed report. For more information, refer to “The Heartbeat Timer” on page 89. The default is 60 seconds. The valid values are 5 seconds up to a maximum of 3600 seconds (one hour). <strong>NOTE:</strong> This variable, in conjunction with the NH_DRPT_MISSED_MSG_LIMIT variable, can have a significant impact on report performance. For more information, refer to “Configuring the Heartbeat and Message Limit Timers” on page 98.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NH_DRPT_MISSED_MSG_LIMIT</td>
<td>Number of heartbeat messages that can be missed (not responded to) before the Distributed eHealth Console assumes that the report request has timed out or failed on a Distributed eHealth System involved in the report. The default is 2 messages. Valid values are 1 up to any valid integer. After value+1 (by default, 3) missed heartbeat messages, the Distributed eHealth Console stops checking that Distributed eHealth System for data. The final distributed report contains a &quot;missing elements&quot; error message. If all Distributed eHealth Systems fail to respond, the report fails with an error. <em>NOTE:</em> This variable, in conjunction with the NH_DRPT_HEARTBEAT_INTVL variable, can have a significant impact on report performance. For more information, refer to “Configuring the Heartbeat and Message Limit Timers” on page 98.</td>
</tr>
<tr>
<td>NH_DRPT_STATUS_MSG_TIMEOUT</td>
<td>Time that the Distributed eHealth Console waits for a response to the heartbeat timer check from a Distributed eHealth System. If the timer expires without a response, eHealth considers the heartbeat check as a missed heartbeat message. The default value is 30 seconds. The valid values are 2 seconds up to a maximum of 1200 seconds (20 minutes). The value for this variable should be less than the value of the NH_DRPT_HEARTBEAT_INTVL.</td>
</tr>
<tr>
<td>NH_DRPT_MAINLOOP_INTVL</td>
<td>Interval, in seconds, at which the Distributed eHealth Console checks to see if it has received new report data from one of the Distributed eHealth Systems involved in the report. The Distributed eHealth Systems independently process their report data and return it to the requesting Distributed eHealth Console, as described in Step 3 in Table 11 on page 86. This interval controls how often the Distributed eHealth Console checks to see if it has received new data. The default is 1 second. The valid values are 1 second up to a maximum of 600 seconds (10 minutes). This variable is most useful for on-demand reports. This variable controls how often eHealth updates the WIP indicators on the Web or the eHealth Console for the report. If set low (1 second), there is a small performance impact for this continual check and update; if set too large, there could be a delay between the time when the last of the report data returns and the time that the report continues on to the compilation and calculation process.</td>
</tr>
</tbody>
</table>
**Distributed Report Compression Variable**

You can specify the NH_DRPT_COMPRESS_RDI environment variable on all cluster systems to reduce the time required to return report data to the Distributed eHealth Consoles. This variable specifies whether the internal report data file sent by the Distributed eHealth System to a Distributed eHealth Console should be compressed. The valid values are yes and no. The default is no. The value for this variable must be the same on all cluster members.

If your cluster members are connected over slow (less than T1 speed) links, and your reports typically contain thousands of elements from a Distributed eHealth System, compressing the report data files decreases the time that it takes to send them to the Distributed eHealth Console. This can help to prevent timeouts that cause a distributed report to assume that the report data was lost and result in “missing element” messages in the report.

**Configuring the Heartbeat and Message Limit Timers**

Use caution if you change the values of the NH_DRPT_HEARTBEAT_INTVL and NH_DRPT_MISSED_MSG_LIMIT variables. These variables combine to create the greatest effect on distributed report performance. By default, the NH_DRPT_HEARTBEAT_INTVL is 60 seconds and the NH_DRPT_MISSED_MSG_LIMIT is 2, which means that if a cluster member is down or does not return data, a distributed report would wait for 180 seconds (60 times [2+1]) before ignoring that non-responding cluster system and continuing. If your distributed reports often contain the “missing elements” note, but your Distributed eHealth Systems appear to be up and running well, the problem could be that your timeouts are too low for your site.

If the links that connect your cluster members are slow, or if your cluster members typically return a large data file (for a large report), these defaults might be too low. If eHealth detects that the report process has stopped running on a Distributed eHealth System involved in the distributed report, but it has not yet received the report data from that system, the 180 seconds could expire. The distributed report continues with the assumption that the report data has been lost. Thus, you would need to increase one or both of these variables to ensure that the distributed report waits for the time it takes for the slowest Distributed eHealth System to respond with its report data. For example, you could increase the NH_DRPT_MISSED_MSG_LIMIT to 4, and the distributed report waits for 300 seconds (five minutes) before considering the report data lost and continuing.

If you increase these values, you increase the “delay” for a distributed report to complete. For example, if you set NH_DRPT_HEARTBEAT_INTVL to 3600 seconds (1 hour), and the NH_DRPT_MISSED_MSG_LIMIT to 10, and a Distributed eHealth System is not responding, your distributed report waits (or appears to hang) for 11 hours (3600 times [10+1]) before it ignores the non-responding system and continues.
This appendix describes some solutions to common problems that relate to Distributed eHealth administration and cluster management.

**Database Error: Object Has Been Marked for Deletion**

If you try to create a new object that has the same name as a soft-deleted object in the database, eHealth displays the following error:

*Database error: object has been marked for deletion. Choose another name.*

To avoid this problem, specify a different object name or run the nhScrubFsa command to remove the soft deleted objects that are a week or more old. After the command finishes, try to save the new object again and the save should complete without error. For more information, refer to “Replication and Soft-Deletion” on page 83.

**Troubleshooting Replication Problems and Failures**

This section describes some common problems and potential solutions for replication processes.

**Detecting Inconsistencies**

The scheduled Find Cluster Differences job looks for inconsistent object information within the cluster. By default, the job runs several times each day, and reports any inconsistencies within the log file for the job. The log file is `ehealth/log/Find_Cluster_Differences.jobId.log`. Each time the job runs, it copies the old log file to `Find_Cluster_Differences.jobId.log.bak` and creates a new `Find_Cluster_Differences.jobId.log` file.

**Recovering Failed Replications**

If you create or modify an object, and you receive errors during the object replication, the problem could be that one or more cluster members are unreachable. To correct the problem, use the Cluster Status utility to determine whether all cluster members are reachable, and if so, use the nhReplicate command to update all other cluster members. If you are using the eHealth Console to make the object changes, you can also edit the object, and without making any changes to the object, click **OK** to replicate the changes to the other cluster members again.
Recovering from Conflicting Edits

If two administrators make simultaneous, different changes to the same object from different eHealth systems, the information for that object becomes out-of-sync. The nhFindClusterDiffs command and scheduled Find Cluster Differences job will report these inconsistencies. To correct the problem, you must do the following:

1. Compare the information for the objects as defined on each Distributed eHealth System. One of the objects could contain all of the correct and latest information, or both definitions could be partial (that is, the object requires both sets of changes to be made).
2. Identify the latest (most current) object information. If necessary, edit one instance of the object to update it with the latest information.
3. Replicate the object information using nhReplicate to the other cluster members. Once replication completes, the object should be consistent across the cluster.

Timeout Errors

This section describes some common timeout problems and possible solutions to correct them.

Cluster Administration Command Timeout Errors

The cluster administration commands use a default timeout of 30 seconds. If these commands do not finish processing before 30 seconds, they fail with a timeout error message. The cluster administration commands include the following:

- nhJoinCluster
- nhModifyClusterMember
- nhListClusterMembers
- nhRemoveClusterMember
- nhRestoreClusterMember

You can change the default timeout for the cluster commands using the NH_CLUSTER_CMD_TIMEOUT environment variable. Specify the timeout in seconds. Although you can specify any valid positive number for the value, as a best practice, the value should be greater than 30 seconds but less than 600 seconds (10 minutes). A value of 300 seconds (five minutes) is a typical default for clusters that use a VPN which continually rebuilds connections and closes them after communication ceases. Also, if you change the RCS timeouts, you might need to change this variable as described in “How RCS Timeouts Affect Application Timeouts” on page 102.

nhRunCommand Timeout Errors

If you use the nhRunCommand command to run commands on other cluster members, eHealth uses the value of the NH_REMOTE_CMD_TIMEOUT environment variable to control how long the command runs before it fails due to non-responding cluster systems. By default, the value of the variable is 300 seconds (five minutes). The valid values are 1 up to any valid integer. If you change the default, be sure to select a value that is larger than the RCS timeouts as described in “How RCS Timeouts Affect Application Timeouts” on page 102. This timeout also applies to “clusterized” eHealth commands that use nhRunCommand such as nhFindElement and those listed in “Distributed Commands” on page 25.
Replication Timeout Errors

When you create a global object on a Distributed eHealth System, the system replicates the object information to the other cluster members. If the replication process fails with a timeout error, but all of the eHealth systems are up and reachable, the problem could be that the replication timeout interval is too short for your cluster environment. The value for this variable should not be lower than the RCS timeouts, as described in “How RCS Timeouts Affect Application Timeouts” on page 102.

You can change the replication timeout value by defining the NH_REPL_MSG_TIMEOUT environment variable. By default, the replication timeout is 300 seconds (five minutes). You should never decrease the timeout below 300 seconds, but you might need to increase it if your cluster members frequently timeout during the replication process. This variable controls the timeout for the following commands:

- nhReplicate
- nhSynchronize
- nhFindClusterDiffs
- nhUpdateDatabaseIds

To increase the timeout, define the NH_REPL_MSG_TIMEOUT environment variable for each eHealth system in your cluster. Specify the timeout in seconds.

RCS Timeout Errors

The RCS processes control the communication (remote file and message transfer) between cluster systems. There are three aspects of RCS timeouts that you can configure:

- RCS connection timeout
- RCS retry queue timeout
- RCS message timeout

RCS Connection Timeout

When eHealth runs a request (command, report, or other process) that must be sent to another cluster member, the RCS processes first connect to the RCS processes on the remote system to make sure that the system is reachable.

The NH_RCS_CONNECT_TIMEOUT variable specifies how long eHealth waits for the RCS connection to occur. By default, the variable is set to 5 seconds. The valid values are 1 up to any positive integer.

**NOTE**

TCP connections also have a timeout value. If the TCP timeout is less than the value of NH_RCS_CONNECT_TIMEOUT, the TCP timeout value supersedes the RCS connection timeout.

If the timer specified by this variable (or the TCP timeout if it is less) expires before a connection is established, eHealth displays the following error:

Unable to connect to port 5052, host hostname

If you see this error, use the ping command to determine whether your local cluster member can reach the remote cluster member. If the ping command fails (the request times out), the problem could be that the remote system or the network connection to it is down.
If the ping command succeeds, your systems probably require more time to establish the RCS connection. Increase the value of the NH_RCS_CONNECT_TIMEOUT variable and retry the original command that failed with the timeout.

**RCS Retry Queue Timeout**

RCS has a send queue that processes the outgoing messages to the other cluster members. If the local cluster member fails to send the message, and the command has a -retryTime option, eHealth places its message in a retry queue. By default, eHealth resends messages in the retry queue every five seconds.

You can specify how often eHealth retries the messages in the retry queue using the NH_RCS_RETRY_QUEUE_TIME environment variable. Specify the timeout in seconds.

**RCS Message Timeout**

After eHealth opens the RCS connection, it passes the necessary messages to the remote cluster member over the connection. The remote system returns acknowledgements for the messages that it receives. The NH_RCS_MSG_TIMEOUT variable specifies how long the local system waits for acknowledgements to the messages that it sent.

If the message timeout expires before the local system receives the acknowledgement, eHealth displays the following message:

Rcs timed out waiting for an ack for message value. Current state is state

This error indicates that the RCS processes need more time to receive acknowledgements from the remote systems. When it occurs, eHealth closes the RCS connection with the remote system. eHealth waits for the time specified by NH_RCS_RETRY_QUEUE_TIME, then attempts to reestablish the RCS connection and to resend the message. The default value of the variable is 5 seconds. The valid values are 1 up to any positive integer.

**How RCS Timeouts Affect Application Timeouts**

The values that you specify for the NH_RCS_MSG_TIMEOUT and NH_RCS_RETRY_QUEUE_TIME variables could impact the application-level timeouts used for distributed reports and commands. If you increase the value of either of these variables, check the value of the following timeouts:

- **NH_REPL_MSG_TIMEOUT** (default is 300 seconds)
- **NH_REMOTE_CMD_TIMEOUT** (default is 300 seconds)
- **NH_CLUSTER_CMD_TIMEOUT** (default is 30 seconds)
- **NH_DRPT_RUN_CMD_TIMEOUT** (default is 45 seconds)
- **NH_DRPT_HEARTBEAT_INTERVAL** (default is 60 seconds)
- **NH_DRPT_STATUS_MSG_TIMEOUT** (default is 30 seconds)

The value of these application-level timeouts should be at least twice the sum of the NH_RCS_MSG_TIMEOUT and NH_RCS_RETRY_QUEUE_TIME timeouts, as follows:

\[
\text{<Application timeout> = 2} \times [\text{NH_RCS_MSG_TIMEOUT + NH_RCS_RETRY_QUEUE_TIME}]
\]
If you increase the value of NH_RCS_MSG_TIMEOUT to 15 seconds, for example, the equation results in a value of 40 seconds \(2^*(5+15)\). The default value of NH_CLUSTER_CMD_TIMEOUT is 30 seconds; in this example, you should increase it to a value of 45 or 50 seconds to ensure that the cluster commands do not time-out before the RCS connection timers expire.

### nhJoinCluster Timeout Errors

If you use the nhJoinCluster command to add an eHealth system to a cluster, you could receive a timeout error if your eHealth systems cannot resolve the hostnames.

Distributed eHealth uses the hostnames for cluster member communication. If an eHealth system cannot resolve the hostname to an IP address, or if the hostname resolves to the wrong IP address, you will receive timeout errors. To avoid these errors, you can either define the hostnames in the hosts file on your eHealth systems (as described in the next section) or troubleshoot the Hosts configuration on your Domain Name Service (DNS) servers. You may have changed the IP addresses assigned to your systems and thus the hostname table is out-of-date.

### Unable to Connect to Port and Host

When an eHealth system in the cluster cannot connect to another system in the cluster, or one that is attempting to join the cluster, the following message appears in the eHealth console:

```
Friday, September 21, 2001 12:26:42 PM Error nhiRmtOut Pgm nhiRmtOut: Unable to connect to 'port 5050, host CHICAGO'.
```

This error typically results because the local system cannot reach the requesting system (CHICAGO). Typically, the system cannot resolve the hostname with an IP address to reach the remote system.

To correct the problem, check with the person responsible for the DNS configurations in your network to determine if the remote system is not being resolved. Or, you can define the remote system and its IP address in the host configuration of the local eHealth system.

**For example, to add a system to the host configuration on a Solaris system:**

1. Log in to the local eHealth system as the root user.
2. Change to the /etc directory.

   ```
cd /etc
   ```
3. Using any text editor such as vi or Emacs, edit the hosts file. A sample hosts file follows:

   ```
   # Internet host table
   #
   127.0.0.1 localhost
   12.16.7.33 localSys loghost
   12.16.5.24 DRACUT
   ```
4. To add an entry for a system, specify the IP address and the system name in the hosts file. For example, to define the IP address and hostname of the CHICAGO system:

```
# Internet host table
#
127.0.0.1     localhost
12.16.7.33     localSys loghost
12.16.5.24     DRACUT
12.16.5.119     CHICAGO
```

5. Save and close the file.

6. You may need to reboot your system for the changes in the hosts file to take effect.

You may need to define one or more of the eHealth systems in the cluster in the hosts file for each eHealth system in the cluster.

**nhListClusterMembers Returns Tables That are Not the Same**

If you run the nhListClusterMembers command and the output shows that the member tables are different on one or more systems in the cluster, run the nhJoinCluster command again to add any missing systems to the cluster. You can also run the nhUpdateClusterTable command to make the tables consistent.

The nhJoinCluster command might not update all systems in the cluster; network problems might have prevented the command from reaching all members. You can correct mismatched member tables by rerunning the nhJoinCluster command for each cluster system that has a discrepancy.

**Object Changes Are Lost**

If you change an object such as a group, service profile, or a group list on a Distributed eHealth System and, and after replication, it appears that your changes were lost, do the following:

- Confirm that the object was not changed by another administrator on another system after you made your changes.
- Confirm that each system in the cluster is synchronized to the same system time, as described in “Time Synchronization” on page 36. Object changes and information are time-stamped, and the latest time stamp indicates the most current object information. Thus, if one system in your cluster is configured to a time that is significantly later than the other systems, the object information on that system will always be considered the latest (or most current) information and any edits you make on other systems could be lost during replication or synchronization.

**Recovering Soft-Deleted Objects**

If an eHealth administrator accidentally deletes an object such as a group, group list, or service profile, you can recover the deleted object before the next scheduled FSA Scrubber job runs (or before an administrator runs nhScrubFsa). For the instructions to recover a soft-deleted object, contact Technical Support.

**Problems after Hostname Changes**

If you change a cluster system’s hostname after eHealth has been installed and the system has joined a cluster, you might encounter replication and synchronization problems, as well as problems running...
reports for elements monitored by that system. A system’s hostname is a critical part of the system identification for the cluster members. For the instructions to start and stop the eHealth server and database, refer to the Web Help.

To change a system’s hostname after eHealth is running on that system:

1. Use nhModifyClusterMember to change the hostname of the system in the cluster.
2. Change the hostname of the system.
3. Stop the eHealth server.
4. Stop the eHealth database.
5. Start the eHealth database.
6. Start the eHealth server.

Redirecting Users When a Distributed Console Fails

If a Distributed eHealth Console fails, you can provide backup access to eHealth reports if you have another Distributed eHealth Console. You can instruct your web users to use the other Distributed eHealth Console until you can recover the Console they normally use.

Note the following if you redirect web users to a different Distributed eHealth Console:

- You may need to create accounts for the users on the backup/secondary Distributed eHealth Console if it does not already have accounts for those users. If you have a DCI file for the web accounts, you can import the web accounts to the backup console. Follow the instructions provided in the eHealth Integration Guide.
- Users will not be able to access any previously generated reports from the failed console. If users require access to one or more of the previously generated reports, you can copy the report directories from the failed console to the backup console (assuming that the failed console system is running and accessible, or that you have backups of the failed console system that you can use to obtain the files). For a description of the web report directories, refer to the eHealth Report Management Guide.

Troubleshooting Distributed Report Messages

This section describes some common error and warning messages that can occur for distributed reports.

Missing Data Notes in Reports

If one or more Distributed eHealth Systems do not respond with data for a distributed report, the report includes the following message:

Missing Data for \( n \) of \( m \) elements

A MyHealth report shows a slightly different error message:

Data for some elements is not included in the report.

These messages indicate that the report does not contain the data for all elements in the subject. If your cluster members are connected, communicating, and operating normally, you may need to increase the distributed reporting timeout and controls. For more information, refer to “Configuring the Heartbeat and Message Limit Timers” on page 98 and “Troubleshooting Missing Information” on page 94.
Report Failed

A distributed report could fail when all Distributed eHealth Systems that own elements in the subject do not respond. (If some, but not all, of the Distributed eHealth Systems respond, the report shows a Missing Data note as described in the previous section.) For more information on failures, refer to “Troubleshooting Report Failures” on page 93.

For multi-element At-a-Glance reports (that is, At-a-Glance reports for multiple subject elements), the report shows a report failed message if one or more of the At-a-Glance reports fail. This problem typically occurs when one or more elements are owned by Distributed eHealth Systems that did not respond to the report request. To resolve it, you must fix the problems that prevented the Distributed eHealth Systems from responding, and then rerun the At-a-Glance report for only those elements that failed.

Application Response Reports Fail

When you run a report for Application Response elements from a Distributed eHealth Console, you could receive the following error:

Error: The list of report data generators is empty. Report failed.

This error occurs when the Distributed eHealth Console has insufficient information to run the report. If you use Application Response in a cluster, some information on the elements is sent to the Distributed eHealth Console. Although Distributed eHealth does not support running reports for Application Response elements from a Distributed eHealth Console, users could select reports for these elements.

Users should run reports on Application Response elements from the Distributed eHealth System that manages those elements. The web administrator can also remove access to groups and group lists of response elements for web reports. You should only use this option if the groups and group lists contain Application Response elements only. If your Distributed eHealth System has other response agents, such as Cisco or Service Availability, users would then be unable to run reports for these elements.

Troubleshooting Cluster Communication Problems

This section describes errors that result when one or more cluster members are unreachable.

Connection Refused Errors

When you run cluster commands, you could receive the following error:

Error: Unable to connect to port 5050, host hostname (Connection refused).

This error occurs when the host hostname is not configured for open traffic on TCP port 5050. For more information, refer to “Opening Ports in the Firewall” on page 40.
Index

A
administration tasks, overview 24
analyzed data
 avoiding inconsistent 93
deleting 93
audience of guide 7
automatic replication 19

C
central site, remote polling 12
clean system 45
cluster
 about 13
 adding, first member 52
 administration tasks 24
 checking status 54
 Cluster Status tool 19, 59
 correcting differences 83
 creating 17, 45
 description 16
 differences
 correcting 83
 existing eHealth systems 42
 listing members 54
 managing
 members 59
 tasks 59
 member table 54
 membership, administration tasks 45
 modifying members 56
 monitoring systems 64
 out-of-sync 21
 removing members 57
 restoring members 58
 security planning 41
 status, managing 59
 TCP ports 51
 troubleshooting
 communication problems 106
 overview 99
 cluster members
 adding, first 52
 listing 54
 listing removed 56
 membership, managing 59
 modifying 56
 removing 57
 restoring 58
 cluster password
 description 17
 setting 17
 Cluster Status
 application 19
 icon descriptions 60
 starting 59
 utility 59
 commands
 cluster options 26
 distributed 25
 nhScrubFsa 83
 conflicting edits, recovering from 100
 connection refused errors 106
 connection requirements 35
D

Data Analysis job, updating 87
data analysis records 87
data specification information (DSI) file 86
database ID, about 19
disk space
  managing 65
  managing on Distributed eHealth System 66
distributed commands, list of 25
Distributed eHealth cluster, creating 45
Distributed eHealth Console
  about 13
  accessing 15
  actions after failure of 105
  adding more 66
  comparison to central site 14
  generating reports from 91
  high-availability solutions 42
  installation notes 45
  installation prompt 45
  installing on clean system 45
  license requirements 34
  managing 65
  outside firewalls 14
  overview 15
  recovering failed 67
  unsupported features 24
  update delays 78
Distributed eHealth site
  about 13
  choosing 14
  planning 29
Distributed eHealth System
  about 13
  accessing other applications 15
  actions 13
  administering 14
  information replicated 15
  installation notes 46
  license requirements 33
  managing 66
  overview 14
  recovery 67
Distributed eHealth, concepts 14
Distributed license 14
distributed polling site
  See remote polling site.
distributed reports
  about 14, 23, 85
  failures 106
  finish point 90
  generating
    about 85
    in parallel 23
  heartbeat timer 89
  impact of missing elements 89
  important notes 90
  missing information 94
  performance impacts of timers 98
  performance overview 90
  prior rank calculation 92
  process
    for scheduled 88
    overview 86
  re-running failed reports 93
  restrictions 85
  running
    about 24
    in time zones 91
    on demand 85
  scheduled, data analysis records 87
  scheduling 87
  time zones 91
  timeouts, configuring 94
  troubleshooting
    Application Response 106
    failures 93
    messages 105
    missing data 105
DMZ, about 40
DSI file. See data specification information file.
duplicate elements, avoiding 36
duplicate object names, avoiding 37
E
edits, recovering from conflicting 100
eHealth administrator, accounts, planning 34
eHealth Cluster Status window
  information 60
eHealth commands, cluster options 26
eHealth systems
  clustering existing 42
  clustering existing standalone 46
F
failed reports
re-running 93
troubleshooting 93
File Staging Area (FSA) 18
Find Cluster Differences
job 81
log file 83
Find_Cluster_Differences jobId.log file 81
firewalls 13
first cluster member, adding 52
FSA Scrubber scheduled job 83

G
global objects 19, 76
group lists, membership across cluster 23
groups, membership across cluster 23

H
handle, object 19
Health reports
Prior Rank Maximum attribute 92
understanding prior rank 92
heartbeat timer 89
high speed connections, using 42
high-availability solution, Distributed eHealth Consoles 42
home, of an element 71
host, unable to connect to 103
hostnames, troubleshooting problems after changes 104

I
immediate replication
about 76
monitoring 76
inconsistencies, detecting 99
ITManagerTop100 Service Level report 89

L
large-scale reports 88
license requirements 33
local cluster system 18
location requirements 36
log files, about 23
logs, monitoring 60

M
manual
replication 75, 78
synchronization 75, 80
member ID 18
member Name 18
member table
listing 54
overview 18
members of cluster
adding first 52
listing 54
managing 59
modifying 56
removing 57
restoring 58
missing data notes in reports 105
missing elements
  impact on distributed reports 89
  in reports 94

N
naming rules 36
NH_CLUSTER_CMD_TIMEOUT environment variable 100
NH_DB_DELETED_OBJECTS_TIME2KEEP environment variable 83
NH_DRPT_COMPRESS_RDI environment variable 98
NH_DRPT_HEARTBEAT_INTVL environment variable 96
NH_DRPT_MAINLOOP_INTVL environment variable 97
NH_DRPT_MISSED_MSG_LIMIT environment variable 97
NH_DRPT_RUN_CMD_TIMEOUT environment variable 96
NH_DRPT_STATUS_MSG_TIMEOUT environment variable 97
NH_RCS_CONNECT_TIMEOUT environment variable 101
NH_RCS_MSG_TIMEOUT environment variable 102
NH_RCS_RETRY_QUEUE_TIME environment variable 102
NH_REMOTE_CMD_TIMEOUT environment variable 100
NH_REPL_MSG_TIMEOUT environment variable 83, 101
NH_RFT_IN_PORT environment variable 51
NH_RFT_OUT_PORT environment variable 51
NH_RMT_IN_PORT variable 51
NH_RMT_OUT_PORT environment variable 51
nhAddSysEdgeMonEntries command, using 60
nhClusterDiffs command 81
nhFindElement command, using 73
nhJoinCluster command
  timeout errors 103
  using for first member 52
  using for subsequent members 53
nhListClusterMembers command
  troubleshooting differences in tables 104
  using 54
nhModifyClusterMember command, using 56
nhReplicate command, using 79
nhScrubFsa command
  object marked for deletion error 99
  using 83
nhSynchronize command, using 80
non-trusted system 16

O
object information, finding differences 80
objects
  about 19
  automatically replicated 19
  automatically synchronized 21
  avoiding simultaneous changes 77
  detecting inconsistencies 99
  error creating soft deleted 99
  finding differences 80
  global 19
  immediately replicated 19
  manually replicated 21
  manually synchronized 22
  soft deletion 83
  troubleshooting lost changes 104
  uniquely identifying 19

P
performance of distributed reports 90
planning, eHealth system locations 36
platform requirements 36
ports
  configuring TCP 51
  unable to connect to 103
Prior Rank column, changes 92
Prior Rank Maximum attribute, setting 92
processes, monitoring 60

R
RCS
  connection timeout 101
  message timeout 102
  processes, timeouts 101
  retry queue timeout 102
reading path 7
Remote Communication Service (RCS) 18
remote poller sites, remote polling 12
Index • 111

remote polling site
  central site 12
  converting 43
  overview 12
  remote poller sites 12
replication
  about 19, 75
  automatic 19
  immediate 19, 75, 76
  manual 21, 78
  recovering failed 99
  timeout value 83
  timeouts 77, 101
  troubleshooting 99
reporting front-end, about 13
reports
  data analysis records 87
  distributed 85
  generating 85
  heartbeat timer 89
  important information for distributed 90
  large-scale 88
  missing elements, impact 89
  on-demand process 86
  performance of distributed 90
  re-running failed 93
  running on demand 85
  scheduled distributed 88
  scheduling distributed 87
  time zone effect 91
  troubleshooting
    Application Response 106
    failures 93
    missing data 105
  when distributed reports finish 90
  work-in-progress indicators 90
resource inventory
  creating from 30
  planning 29
  table 32
  using to assign elements 42

S
  scheduled synchronization 75, 77
  security characteristics 41

service profiles
  changing 92
  replicating 92
ServiceCustomerTop100 Service Level report 89
simultaneous changes, avoiding 77
site configuration overview 11
site layout 38
site plan
  about 30
  preparing 30
site planning
  considerations 29
  high speed connections 42
  other options 42
  VPN connections 42
soft deleted objects
  error creating 99
  recovering 104
soft deletion 83
standalone eHealth system
  clustering 43
  elements supported 12
  overview 11
StandardLite Health report 89
status of cluster, checking 54
synchronization
  about 21, 75
  automatic 21
  manual 22, 80
  scheduled 77
Synchronize job 78
synchronizing
  time 46
system and license requirements 31
SystemEDGE agent
  configuring 63
  installing on cluster members 16
  planning 33

T
tasks, administration 24
TCP ports, configuring 51
time synchronization 36, 46
time zones
  distributed reports 91
  effect on reports 91

Distributed eHealth Administration Guide
timeouts
  about 77
  cluster administration commands 100
  nhJoinCluster command 103
  RCS
    connection 101
    message 102
    overview 101
    retry queue 102
    replication 101
    replication response 83
    troubleshooting 100
troubleshooting
  cluster communication 106
  connection refused errors 106
  differences in nhListClusterMembers tables 104
distributed reports 105
information 99
lost object changes 104
port connections 103
problems after hostname changes 104
replication response 83
report failures 106
using log files 23
trusted system 16
U
unsynchronized objects 21
Update Status dialog 77
user accounts, planning 34
V
VPN connections, planning 42
W
Web accounts
  managing 65
  planning 35
  using DCI tools 65
WIP. See work-in-progress indicators.
work-in-progress indicators
  overview 23
  reports 90
workstation requirements 36