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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audience</td>
<td>5</td>
</tr>
<tr>
<td>About This Guide</td>
<td>5</td>
</tr>
<tr>
<td>Reading Path</td>
<td>5</td>
</tr>
<tr>
<td>Revision Information</td>
<td>5</td>
</tr>
<tr>
<td>Documentation Conventions</td>
<td>6</td>
</tr>
<tr>
<td>Technical Support</td>
<td>6</td>
</tr>
</tbody>
</table>

## Chapter 1 Preparing for the Discover Process

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Purpose of Discovery</td>
<td>7</td>
</tr>
<tr>
<td>Discovering Using the DataSync Programming Interface</td>
<td>7</td>
</tr>
<tr>
<td>Identifying Your Infrastructure Resources</td>
<td>7</td>
</tr>
<tr>
<td>Understanding the Impact of Resource Changes</td>
<td>8</td>
</tr>
<tr>
<td>Discovering Logical versus Physical Elements</td>
<td>8</td>
</tr>
<tr>
<td>Your Resource Discovery Roadmap</td>
<td>8</td>
</tr>
</tbody>
</table>

## Chapter 2 Finding Your Resources: Running Discover for the First Time

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovering Interactively for the First Time</td>
<td>9</td>
</tr>
<tr>
<td>Completing a Discover</td>
<td>10</td>
</tr>
<tr>
<td>When You Need to Rediscover</td>
<td>11</td>
</tr>
<tr>
<td>Understanding the Purpose of the Log File</td>
<td>11</td>
</tr>
<tr>
<td>Types of Changes That Discover Can Resolve</td>
<td>12</td>
</tr>
<tr>
<td>Reviewing Your Discover Results in the Log File</td>
<td>13</td>
</tr>
<tr>
<td>Effectively Maintaining Your Poller Configuration</td>
<td>16</td>
</tr>
</tbody>
</table>

## Chapter 3 Best Practices for Optimal Discovery

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tailoring Your Discovery</td>
<td>17</td>
</tr>
<tr>
<td>Configuring eHealth to Discover SNMPv3 Elements</td>
<td>17</td>
</tr>
<tr>
<td>Limiting the Discovery of LANs.</td>
<td>18</td>
</tr>
<tr>
<td>Using a DCI Rule to Include or Exclude Elements</td>
<td>18</td>
</tr>
<tr>
<td>Specifying an IP List from a File.</td>
<td>19</td>
</tr>
<tr>
<td>Discovering Elements in the Current Configuration Only</td>
<td>19</td>
</tr>
<tr>
<td>Excluding Specific Elements from the Search</td>
<td>19</td>
</tr>
</tbody>
</table>
## Table of Contents

- Improving Your Discoveries ............................................. 20
  - Use the Same IP Address Discover Strategy Each Time .......... 20
  - Specify Correct Community Strings ................................. 20
  - Configure Discover to Search for Different Ports .............. 21
- Save Your Elements to Groups Automatically ..................... 21
- Enable eHealth to Find System Processes ........................... 22
- Ensure Data Continuity for Rebooted Systems ..................... 22
  - Use Unique sysName and uniqueDeviceId Values .................. 22
  - Use Unique ifDescr Values within a Device ...................... 22
  - Run Discover Again after Upgrading ............................... 23
  - Run a Discover after Each Device Change ....................... 23
  - Increase the Number of Devices That eHealth Discovers ........ 23
- Using a Scheduled Discover Job ...................................... 23
  - Controlling Updates to the Poller Configuration ............... 23
  - Saving Scheduled Discover Results ................................ 24
  - Best Practices for Running Scheduled Discoveries ............. 24
- Preventing and Reconciling Unresolved Elements ................... 25
- Troubleshooting Discover Problems .................................. 27
  - Discover Errors and Failures ...................................... 27
  - System Mode Issues .................................................. 30
  - Router Mode Issues .................................................. 30

### Appendix A

- The Discovery Process: Element Matching ......................... 33
  - The Matching Algorithm ............................................. 33
  - Matching at the Device Level ...................................... 34
    - Verifying a Device Match ......................................... 34
  - Matching at the Agent Level ...................................... 34
  - Matching at the Element Level .................................... 35
    - Using Discover Keys ............................................... 35
    - Using Physical Address ............................................ 38
    - Using MIB Attributes .............................................. 38

### Appendix B

- eHealth Element Naming Conventions ................................ 39
  - Element Creation ................................................... 39
    - Discovering LAN/WAN Elements .................................. 39
    - Discovering Router/Switch Elements ......................... 39
    - Discovering Traffic Accountant Elements ....................... 41
    - Discovering System Elements .................................. 41
    - Discovering Application Elements .............................. 42
    - Discovering Modem Pool and RAS Elements ................... 42
    - Discovering Response Elements ................................ 43
    - Discovering QoS Elements ........................................ 43
  - Naming Elements ..................................................... 44
  - Customizing Element Names ........................................ 47

### Index

  eHealth Resource Discovery Guide
This guide describes the eHealth resource discovery process—how eHealth locates and identifies the resources that you want to monitor within your infrastructure. Resource discovery is one of several primary tasks that an eHealth administrator performs. This guide supports eHealth Release 6.0 and later.

**Audience**

This guide is intended for anyone who must perform resource discovery or is responsible for managing one or more aspects of that administrative function. To use this guide, you should have a basic understanding of networking and data communications concepts; be comfortable using scripts; and be familiar with SNMP, ping, and device configuration settings.

**About This Guide**

This section describes the reading path that you should follow, as well as the revision history of this guide. It also includes the documentation conventions used in this guide.

**Reading Path**

Prior to reading this guide, you should review the *Introduction to eHealth* guide and *eHealth Administration Overview Guide*. After you have finished reading this guide, you can refer to the *eHealth Element and Poller Management Guide* for detailed information about managing your resources and the eHealth poller. These guides are available in PDF format in the eHealth Web Help and on the Technical Support Web site.

**Revision Information**

This is the first release of this guide.
**Documentation Conventions**

Table 1 lists the conventions used in this document.

<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 lines.</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>
<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](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](http://support.concord.com).
Preparing for the Discover Process

The eHealth discover process is a powerful tool that locates resources within your infrastructure so that eHealth can collect data from them and report on their performance and availability. Before you run the discover process for the first time, it is important to understand the purpose of discovery, identify the resources within your infrastructure that you want to manage using eHealth, and understand the impact of changes within your network.

The Purpose of Discovery

eHealth uses the discover process to find the critical resources within your infrastructure. After eHealth discovers a resource, it adds it as an element—its representation of your business resource—to the poller configuration.

To find all of your resources, eHealth uses Simple Network Management Protocol (SNMP) to search for the IP addresses that you specify. It then obtains information from the management information base (MIB) of each device and creates elements based on that information. When you save the discover process results, eHealth stores the element information in its database and its poller configuration. eHealth can then collect performance and availability information on those elements, analyze the data, and generate valuable reports on the health of the resources.

Discovering Using the DataSync Programming Interface

If you have a network management system (NMS) or other source at your site that collects configuration information and data for your resources, you could use the eHealth DataSync application programming interface as an alternative to the discover process to import your element information. To use DataSync successfully, however, you must have a working knowledge of eHealth and be comfortable creating programs, scripts, and files that use complex syntax. For detailed instructions, refer to the eHealth Integration Guide.

Identifying Your Infrastructure Resources

Before you perform a discovery to locate your resources, obtain a map or list of the devices in your infrastructure and their IP addresses from a network or system administrator. As a best practice, create a list of IP addresses in an electronic file for future reference. Your infrastructure typically contains resources such as routers, systems, switches, probes, modems, and other devices. These resources contain interfaces that connect them to each other and form the topology of the infrastructure. Users connect to various systems over the topology to access critical applications. You need to identify which resources you want to monitor using eHealth and then use the discover process to find them. The specific eHealth licenses that you should purchase for your organization depend on the number and types of resources that you want to monitor.
Understanding the Impact of Resource Changes

Complex devices such as large-scale routers or systems can constantly change with the addition or removal of components. These types of complex devices could experience multiple changes at one time, so you may need to spend more time to maintain the poller configuration for these elements. For simpler devices, such as stackable hubs, which are not as likely to change frequently or significantly, the discover process is more likely to resolve the changes and maintain your poller configuration automatically.

To determine how often you will need to rediscover your elements to maintain element information, you should become familiar with the various ways that your resources can change and the frequency.

Resource information can change for any of the following reasons:

- Adding or removing devices from your infrastructure.
- Changing MIB attributes or the agent at a device.
- Adding or removing interface cards, disks, CPUs, or partitions for devices.
- Upgrading or restarting applications, operating systems, or SNMP agents at devices.
- Changing interface bandwidth/speed.
- Assigning new IP addresses to interfaces.

Discovering Logical versus Physical Elements

If discover finds a device or component such as a specific port on a specific card of a specific router, discover collects the data for the port, regardless of who uses it or why. In that case, you are reporting on a physical element. If you report on a logical purpose for a device or component, such as a link from New York to Boston, you are reporting on a logical element. If the link moves from one router to another router, you would want the element and its data to move with the element; that is, you would want eHealth to retain the data for the element from the previous device and collect the new data from the new device.

Your Resource Discovery Roadmap

This chapter explains the purpose of discovery, how to identify the resources that you want to monitor, and how to understand the impact of changes within your network. Managing the discovery process involves several primary tasks (outlined in Table 2) that are discussed in the remaining chapters.

Table 2. Resource Discovery Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finding your resources</td>
<td>Chapter 2</td>
</tr>
<tr>
<td>Tailoring your discovery</td>
<td>Chapter 3</td>
</tr>
<tr>
<td>Improving your discoveries</td>
<td>Chapter 3</td>
</tr>
<tr>
<td>Scheduling a discover job</td>
<td>Chapter 3</td>
</tr>
</tbody>
</table>
Finding Your Resources: Running Discover for the First Time

As a best practice, when you run the discover process for the first time, use an interactive discover. An interactive discover allows you to find new elements and update existing elements at any time that you feel it is necessary. Depending on the licenses that you have purchased, eHealth is able to discover different types of resources within your infrastructure and create elements for them. Before discover saves the elements that it finds, eHealth gives you the opportunity to edit the results. This allows you to control the resources that it will poll within your infrastructure.

This chapter describes the basic process that you should follow to discover your elements for the first time, interpret the results, and save the data to update your poller configuration. For future discoveries, you can use the methods described in Chapter 3, “Best Practices for Optimal Discovery,” to tailor your search and schedule discover jobs to keep your poller configuration up-to-date.

Discovering Interactively for the First Time

To perform a discovery for the first time, use the Discover dialog by selecting Discover from the Setup menu in the console. After you initially start eHealth and enter your licenses, the Discover dialog automatically appears. For an initial discovery, you need to specify only the types of devices that you would like to find and their IP addresses. After you start the search, the Discovering dialog displays details about the progress of your discovery. Once it has completed its search, you have the opportunity to save the results and update your poller configuration with the new information, discard the results and begin again, or modify the results before saving them.
To run the discover process interactively for the first time:

1. Select the technology types that you want to discover. (If an element type does not appear, you do not have a license to discover that type of element.)

2. Use one of the following formats to specify the IP addresses for which eHealth should search:
   - An inclusive range with a dash (-) separating the first and last number of the range. The IP address range must consist of a single IP subnet. Specify a range from 0 to 255, and do not use spaces between dashes or commas. For example, the range of addresses 128.12.10-12.87 searches the following IP addresses: 128.12.10.87, 128.12.11.87, and 128.12.12.87. The range 128.12.10.83-85 searches the following IP addresses: 128.12.10.83, 128.12.10.84, and 128.12.10.85.
   - Use a list of specific numbers separated by a comma (,). For example, the range of addresses 128.12.5,19,22.87 searches the following IP addresses: 128.12.5.87, 128.12.19.87, and 128.12.22.87.
   - Use a combination of dashes and commas. For example, the range of addresses 192.12.15-17.5,8 searches the following IP addresses:
     - 192.12.15.5 192.12.15.8 192.12.16.5 192.12.16.8
     - 192.12.17.5 192.12.17.8

3. If you have configured the eHealth SPECTRUM Integration and set up collections of models, select SPECTRUM Import and import data for a specific collection of elements. For instructions on using this module, refer to the eHealth SPECTRUM Integration focus topic.

4. Click Discover to start the discover process. The Discovering dialog displays details concerning the progress of the discovery.

Completing a Discover

When you complete your first discover, eHealth displays the number of new elements that it discovered and saves these messages in the discoverResults.log file. Once you have reviewed the results contained in your discover log, you can discard the changes, save them, or edit the poller configuration before saving them.

Discarding All Changes

If you do not want to retain any of the discovered data, you can click Don’t Save in the Discover Results dialog to discard the new elements without adding them to the poller configuration. eHealth then saves the log file as discoverInteractive.date.time.unsaved.log.

Saving Changes without Editing the Poller Configuration

If you want to monitor all new elements that discover found, click Save in the Discover Results dialog. eHealth creates an entry in the poller configuration for each new element and saves the results in the pollerAudit.date.time.log file in the /ehealth/log directory. After you save the results, eHealth immediately starts to poll the elements.
Editing the Poller Configuration before Saving

Before saving your discover results, you may want to remove elements that you do not want to monitor, change the names of specific devices to make them easily recognizable, or change specific information that discover was not able to update. Click Edit Before Saving in the Discovering dialog. In the Poller Configuration dialog, change the elements, as necessary, and then click OK to ensure that eHealth saves the edited results of the discover process. If you click Cancel, it discards any newly discovered elements and updates.

When You Need to Rediscover

As you add and reconfigure devices, you should run discover after the changes occur to update your poller configuration. By rediscovering regularly, you can limit the number of updates that are needed, and the number of changes that eHealth is unable to resolve based on the current information in the database. However, rediscovering a large number of elements can impact your system performance. To avoid performance problems, allow several hours to elapse between discoveries.

You should also perform a rediscovery after you do any of the following:

- Upgrade to a new release of eHealth, to ensure that you use the latest eHealth discover information.
- Upgrade devices, to resolve element changes that occurred as a result of the upgrade.
- Change the configuration of a device, to ensure that eHealth updates the element.
- Discover router/switch elements. You must rediscover to include router/switch interface elements in your LAN/WAN reports.
- Reboot a device, to update the SNMP index values.

Understanding the Purpose of the Log File

When you run a discover process, eHealth compares attributes of the discovered elements to attributes of previously discovered elements. For each discover that you run, eHealth records the results in comparison to the existing poller configuration in a file named discoverInteractive.<date.time>.log. As a best practice, you should review the discover logs to verify the changes and update the element information that discover was unable to change.

The log records the following:

- New and updated elements that it found
- Elements that it could not find
- Elements that appear to be identical
- Those elements that it was unable to update because it found conflicting information
- Missing elements (previously discovered elements that exist in the poller configuration but that discover did not find during this discovery).

The log includes the infrastructure changes that the discover process could resolve using existing information in the poller configuration. Discover is able to resolve an infrastructure change when it can update the poller configuration with the information without duplicating an element or incorrectly changing existing information. It uses a complex matching process based on discover keys to resolve changes. For an in-depth discussion, refer to Appendix A, “The Discovery Process: Element Matching.”
In a case in which discover cannot resolve the change (that is, it cannot clearly discern the exact change to make to the poller configuration), it lists the change as unresolved to give you the opportunity to edit the poller configuration and avoid losing historical data or inadvertently creating multiple elements for the same resources.

**Types of Changes That Discover Can Resolve**

Following a discovery, eHealth itemizes the types of addresses that it was unable to update by displaying the messages shown in Table 3. In the cases in which it was able to resolve all changes, it simply itemizes the number of new elements found and the number of updates it made.

**Table 3. Discovery Status Messages**

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignored Addresses</td>
<td>Number of broadcast addresses that discover ignored (it cannot discover elements at those addresses)</td>
</tr>
<tr>
<td>Excluded Addresses</td>
<td>Number of IP addresses that discover ignored because you included them in an IP exclusion file</td>
</tr>
<tr>
<td>No Response to Ping</td>
<td>Number of devices that did not respond to a ping</td>
</tr>
<tr>
<td>No Response to SNMP Requests</td>
<td>Number of IP addresses at which the SNMP request timed out before the agent responded to the request</td>
</tr>
<tr>
<td>Invalid SNMP Responses</td>
<td>Number of IP addresses at which the SNMP agent responded with an error or with an indication that it expects a different community string or port number</td>
</tr>
<tr>
<td>Unsupported Agents</td>
<td>Number of SNMP agents found that eHealth does not support</td>
</tr>
<tr>
<td>Discovered Agents</td>
<td>Number of SNMP agents found</td>
</tr>
</tbody>
</table>

In general, eHealth can resolve the following infrastructure changes:

- New devices and components added to the network
- SNMP index shifts for elements with unique discover keys or unique MAC addresses
- One of the following changes to the attributes of a device:
  - Replacement of a chassis
  - Change to the IP address
  - Change to the sysName MIB variable of a device
  - Removal or addition of device components such as cards, memory, or disks
- Change to component attributes, except for changes to attributes used to create a discover key or to compare unkeyed components.
- Total device replacement
Reviewing Your Discover Results in the Log File

Depending on the modes that you select, eHealth discovers different types of resources within your infrastructure and uses various methods to create elements for them. For each resource that eHealth discovers, the method by which it creates a corresponding element varies. To interpret your discover results correctly, it is important to understand the different ways that eHealth discovers each resource type. For a detailed discussion, refer to Appendix B, “eHealth Element Naming Conventions.” You need to carefully review the log file to identify the changes that eHealth has found within the infrastructure and determine the steps that you need to follow to update your poller configuration, if necessary. The following sections provide guidelines for making these changes. For detailed instructions on changing your poller configuration, refer to the eHealth Element and Poller Management Guide.

To view a discover log:

1. Click View Discover Logs in the Poller Configuration dialog or the Discover dialog and select the type of discover log that you want to view.

2. Click View.

3. Navigate to each section of the discover log by clicking the appropriate button at the top of the screen. Each discover file displays a header composed of the following sections:

   - Discover Results
   - Network Change Summary
   - Duplicate Analysis

4. After you review the header, review each of the following subsections by clicking the appropriate button at the top of the screen:

   - New Elements
   - Updated Elements
   - Duplicates
   - Missing Elements

Discover Results Section

The Discover Results section at the top of the log lists the time at which the discover occurred, indicates whether it was scheduled or interactive, and itemizes the options that you specified for the discovery.
Network Change Summary Section

The Network Change Summary section identifies several types of elements: new, updated, unchanged, and missing. It also lists discover key changes—the unique identifiers that eHealth uses to recognize elements.

**New Elements.** Discover identifies an element in the discover log as *new* when it finds a device in the infrastructure or components on existing devices that are not in your current poller configuration. When you save the discover results, it adds the new elements and their components to the configuration.

**Updated Elements.** Discover identifies an element in the discover log as *updated* if it is in your poller configuration and discover found updated information for it. For example, a device (router, system, or RAS device) would be updated by replacing one piece of hardware with another, causing a change to the hardware ID. The discover process can resolve this change if the old device has been removed from the infrastructure and the new device uses the same system name (sysName) and IP address as the old device. However, if you replace a switch or router, discover matches on agent address and sysName, not on the unique device ID. Based on the list of IP addresses, it identifies the device as a new element.

Discover differentiates between resolved and unresolved updates as follows:

- If discover is able to positively match the element to another element in your infrastructure based on the discover key or physical address, it lists it as a *resolved update*. When you save your discover results, it updates your poller configuration with the changed information.
- If discover is unable to match the element based on a discover key or physical address, but matches it on the MIB index and MTF (which is a less reliable attribute), it lists it as an *unresolved update*.

After you review each unresolved update in the log to determine what has changed, do one of the following:

- Modify the current poller configuration and run the discover process again.
- Run an interactive discover process and edit the discovered results before saving them.

If the unresolved changes are the result of a modification to a device, you can reconcile the changes in various ways by following the instructions provided in “Preventing and Reconciling Unresolved Elements” on page 25. To improve your discover results in the future, familiarize yourself with the tips and recommendations offered in Chapter 3, “Best Practices for Optimal Discovery.”

**Unchanged Elements.** Discover identifies an element as unchanged if its attributes have not changed in any way. These elements still exist in your infrastructure, and discover did not detect any new configuration information.

**Missing Elements.** Discover identifies an element in the discover log as *missing* if its IP address and UDP port fall within the discover process range, but eHealth was not able to find it for any of the following reasons:

- Discover request timed out before it could get a response.
- SNMP agent on the device is down (and, therefore, is unable to collect data from the device).
- Element no longer exists in the infrastructure.
- UDP port on which the agent is running has changed.

**NOTE**

If some, but not all, element properties change, eHealth identifies it as a new element *and* a missing element.
If missing elements are included in the Network Change Summary section of the log, scroll to the Missing Elements section. Determine how long the elements have been missing by examining the Missing Elements section of discover logs for previous days. If the elements have not responded for several days and you can verify that the elements have been removed, delete or retire them. For detailed instructions, refer to the eHealth Element and Poller Management Guide.

**To resolve a missing element that you know no longer exists:**

1. Log in to the OneClickEH console.
2. Right-click the element and disable polling for it.
3. Do one of the following:
   - If you no longer want to report on the element, right-click and delete it from the poller configuration.
   - If you want to continue to run reports on the element, right-click and retire it.
4. Delete it from any discovery IP address list file that you specify in your discoveries.

**Discover Key Changes.** The log lists the number of elements in the poller configuration for which the discover process changed the discover key. Provided that other changes did not occur to the device (that is, the IP address, hardware ID, and other attributes used to form the discover key have not changed), the discover process resolves device name changes and displays the new discover key in the discover log.

**Duplicate Analysis Section**

The Duplicate Analysis section lists the number of suspected duplicate elements, duplicate (identical) names, and duplicate keys identified in the database. If duplicate elements were found, click Duplicates to scroll to the Duplicate Analysis Details section. To avoid taxing your eHealth server, it is good practice to resolve all duplicates and avoid rediscovering them in the future.

**Suspected Duplicate Elements.** TheSuspected Duplicate Elements section lists elements in the poller configuration that appear to be duplicate elements. The attributes of each set of suspected duplicate elements closely match, but they are not identical. If these elements are duplicates, eHealth may be polling the same element twice.

Suspected duplicate elements occur when you save the results of a discover process that include a change in an index value of an element that eHealth cannot track with a discover key. Systems and system agents are very dynamic. On some systems, element indexes can change as a result of inserting a CD-ROM into a drive. Router interface elements can change indices when the router reboots, a card is inserted or removed, or when you install firmware upgrades. To avoid this problem, regularly update the original element with its current index value.

Discover identifies an element as a suspected duplicate element if either of the following occurs:

- The element duplicates another element’s attributes that are normally used to identify it uniquely.
- The physical address and/or discover keys match, but the MIB indexes and MTF are different.

For each suspected duplicate that discover identifies, discover provides the name of the suspected duplicate and the name of the oldest element that duplicates it.

**CAUTION**

In some cases, discover identifies elements of similar agent types as duplicates because they share similar properties, but they are not truly duplicates. Use caution when deleting them. If you know that two elements are actually duplicates, delete the true duplicate. Otherwise, contact Technical Support for assistance.
**Duplicate Element Names.** Discover identifies a duplicate element name if one of the following occurs:

- It was unable to update it because it is identical to other element names in the poller configuration, but has one or more attributes that do not match.
- The element’s system name, hardware ID, discover key, or agent type is not unique in the infrastructure.
- The interface was replaced with another type.
- You specified the element name incorrectly in the Modify Element dialog.

To resolve a duplicate element name, you can use the OneClick for eHealth console (OneClickEH) to change the name of one of the elements. If you leave them as is, discover will display them with a -A and -B suffix. For example: east-SH, east-SH-A, east-SH-B.

**Duplicate Keys.** The Duplicate Keys section lists all elements in the poller configuration that have identical discover keys. If a discover process detects a duplicate key, it must identify the elements through other means, by comparing device and element information and then by attempting to match on the physical address.

Discover identifies an element as a duplicate key if one of the following occurs:

- It has the same ifDescr variable across many interfaces in the device.
- The sysName and unique device ID are not unique in the infrastructure.

To resolve duplicate keys:

1. Update the device’s configuration to ensure that attributes such as sysName and unique device ID are globally unique.
2. Update the device’s configuration to ensure that the ifDescr for the interface is unique on the device.
3. Rediscover.

**Effectively Maintaining Your Poller Configuration**

Your infrastructure changes as you replace and remove devices, add new applications, and move systems from one location to another. Without regular rediscovery, your poller configuration can become out-of-date quickly. If you select Poller Configuration in the Discover dialog, eHealth rediscovers and updates your elements at the same IP addresses and with the same community strings that are specified in the file.

The next chapter describes best practices that you should follow to ensure optimal discoveries, explains how to prevent and reconcile unresolved elements found in your discover logs, and offers some tips on effectively troubleshooting common problems. It also describes how to schedule discover processes to update your element information automatically.
Best Practices for Optimal Discovery

To keep the poller configuration up-to-date with your infrastructure, you should run discover on a regular basis. This chapter describes best practices that you should follow to ensure optimal discoveries, and explains how to prevent and reconcile unresolved elements found in your discover logs, and troubleshoot common problems.

Tailoring Your Discovery

To run an interactive discovery, you must specify the IP addresses for the resources that you want to find. Over an extended period, you may determine that eHealth repeatedly discovers elements that you do not want to add to your poller configuration. To remove unwanted elements after you discover, you must modify your poller configuration by following the instructions and guidelines provided in the eHealth Element and Poller Management Guide. As a best practice, ensure that eHealth finds the elements that you want to monitor by using these methods to tailor your discovery:

- Configure eHealth to discover SNMPv3 elements.
- Limit the discovery of LAN resources.
- Exclude or include elements based on their attributes.
- Search for specific IP addresses rather than ranges.
- Discover only addresses that already exist in your poller configuration.
- Exclude specific elements from the search.

Configuring eHealth to Discover SNMPv3 Elements

To monitor SNMPv3 devices in your infrastructure, you need to configure eHealth to discover and poll these types of elements using the SNMPv3 protocol. The discover process may take longer than a typical discovery due to the extra security validation involved with SNMPv3.

To configure eHealth to discover SNMPv3 devices:

1. Install and configure the SNMP Security Pack™ software (from SNMP Research International) on your eHealth system. For more information, refer to the SNMP Security Pack Quick Start Guide provided by SNMP Research.

**NOTE**

The SNMP Security Pack software installation program prompts you to identify the network management software that you want to use with the service pack. Select Some other manager application.
2. Set the NH_SNMP_PROXY_ADDRESS environment variable to specify the IP address and port number of the BRASS™ server installed as part of the SNMP Security Pack software (Default: 127.0.0.1:4747). For instructions on using this variable, refer to the Web Help.

3. Import SNMPv3 key configuration information for each of the devices that you want to discover using SNMPv3 protocols. For instructions, refer to the eHealth Integration Guide.

4. Perform a discovery. The first time that you discover an element using the SNMPv3 protocol, the discover process may time out due to the extra security validation involved with SNMPv3. If the process times out, and you receive a NoResponse error message, increase the setting of the NH_DISCOVER_TIMEOUT environment variable. For instructions, refer to the Web Help.

**Limiting the Discovery of LANs**

When performing a discovery of LAN/WAN elements, select the Find MIB2 LANs option in the Discover dialog to limit the discover to LAN interfaces that support basic MIB2 statistics such as In/Out/Total packets. This method is useful for devices that have an uncertified SNMP agent installed. When you use this option, eHealth generates a basic element which allows for reporting of availability and basic packet count information. To ensure that you collect vendor-specific interface information when discovering certified devices, do not use this option.

**Using a DCI Rule to Include or Exclude Elements**

By creating a DCI rule, you can include or exclude elements based on certain element attributes such as element name substring, element type (such as Ethernet), index range, and enterprise ID. If you specify that rule during discovery, discover filters its search based on that rule. For specific instructions on creating DCI rules, refer to the eHealth Integration Guide.

To filter discovered elements by specifying a DCI rule:

1. In the Discover dialog, select Use DCI rules and click Specify.
2. Select Use a DCI rules file and browse to select a file. Otherwise, select Create new rule or use rule named and select an existing rule.
3. Specify or change the conditions for the rule to include or exclude elements based on criteria:
   - For the Name field, use regular expression syntax to filter based on element names. For example, to filter all element names that match -PVC-, specify the following:
     
     .*-PVC-.*
   - For the Index field, use commas and hyphens to specify a range. For example: 1,2,5-8
   - For the Enterprise ID field, specify one or more enterprise IDs using a device-specific value. For more information, consult the documentation associated with the device.
4. Click **Save and Select** to save the rule with a unique name, or click **OK** to apply the rule to the current discover task. If you do not save the rule, eHealth discards it when the discover process ends.

**Specifying an IP List from a File**

If you want eHealth to discover a specific set of elements, you can create an ASCII file of specific IP addresses and community strings. The discover process interprets any entry in the format `aa.bb.cc.dd` as an IP address. eHealth does not recognize an IP address range or pattern in a file. Use the following format for each entry:

```
IP_Address  community_string  port
```

For example, to discover an element with an IP address of 10.20.30.40 on SNMP port 6200, a read-write community string of acmeRw, and a read-only community string of acmeRd, enter the following:

```
10.20.30.40  acmeRw/acmeRd  6200
```

When you rediscover elements using an IP address file, if the environment variable `NH_DISCOVER_UPDATE_IPADDR` is set to `yes`, eHealth updates the IP address for each element when it redisCOVERS them. Similarly, if you also have set the environment variable `NH_DISCOVER_UPDATE_COMM_STR` to `yes`, eHealth updates the community string for each element when it redisCOVERS them.

To use a file to discover elements:

1. In the Discover dialog, select **IP List from File**.
2. Specify the full pathname of an ASCII file that you created, or browse to select the file.
3. If you included community strings in the file, select **Use Community Strings from File**; then click **Discover**. Discover will base its search on the community strings specified in any files that you use, including IP exclusion files. If you do not specify community strings in the file, eHealth uses the ones that you specify in the Discover dialog.

**Discovering Elements in the Current Configuration Only**

To discover all IP addresses for elements that exist in the current configuration, select **Poller Configuration** under **Discover Using**. Discover searches for those elements only and updates your poller configuration with any changes.

**Excluding Specific Elements from the Search**

If you know the IP addresses of resources that you do not want to monitor, add the addresses to an IP exclusion file. If you specify that file in the **IP Exclusion File** field, discover ignores the addresses.

The IP exclusion file can be any text file that contains a list of individual IP addresses. eHealth does not recognize an IP address range or pattern in a file. It does not require the file to have a specific format and it ignores any text that is not a valid IP address. You can create a file or use an existing file. For example, you can use the `discoverResults.log` file in the `log` directory of your eHealth installation to ignore any IP addresses specified in the previous discover process.

**NOTE**

When you enable the **Use Community Strings from File** option, **always** specify matching community strings in IP exclusion files to ensure that Discover excludes the IP addresses properly.
Improving Your Discoveries

This section outlines some infrastructure guidelines that you should follow to help the discovery process identify changes to eHealth elements that represent the devices that you are monitoring.

Use the Same IP Address Discover Strategy Each Time

If you typically discover from a list of IP addresses, use a list each time. Often, discovering from a list or a file of IP addresses is faster than discovering a large range of IP addresses. Also, for devices that have multiple IP addresses, specify the same IP address for each discover process to avoid discovering duplicate elements for the device.

Specify Correct Community Strings

By default, eHealth performs a discovery based on the public community string. The community string is an SNMP password that administrators assign to devices to control read and write access to data that is stored in the MIB. For most devices, eHealth needs to know only the community string for read access (a read-only community string).

Sometimes, device agents use several strings to protect information in the MIB. Some elements require read-write community strings to allow eHealth to monitor them. For example, to collect alternate latency information or data from service response elements, specify a community string that permits read and write access (a read-write community string) to the element when you discover or modify its information in the poller configuration.

You can specify one or more read-write and read-only community strings in the Discover dialog by following these guidelines:

- Specify a maximum of 64 single-byte or 32 double-byte characters using the letters A through Z and a through z, backslashes, and the numbers 0 through 9. Do not use the word All, spaces, and commas.
- When using the eHealth console on UNIX or Windows, do not supply the extra backslash escape character.

NOTE

If Technical Support advises you to run the discover process through the CLI, follow these guidelines:

- On a UNIX system, always include an additional backslash as an escape character.
- On a Windows system, do not supply the extra backslash escape character.

- Specify the correct letter casing. The community string is case-sensitive.
- If you specify more than one string, separate each string with a comma, but do not insert a space between the comma and the next string.
- Specify both read-write and read-only community strings by entering the read-write string, a slash (/), and then the read-only string, as follows:

  acmeRw/acmeRd

When you specify multiple community strings, eHealth runs a separate discover process for each one. The first discover process finds elements at SNMP agents that use the first community string. The second discover process finds elements at SNMP agents that use the second community string, and so on. To reduce this overhead, you can filter your discovery by using a file that lists IP addresses and their associated community strings (for instructions, refer to “Specifying an IP List from a File” on page 19).
Configure Discover to Search for Different Ports

By default, eHealth looks for SNMP agents at port 161, though it also looks for system agents at ports 1691 and 6665. Before you run the discover process, you can configure it to search for SNMP agents on other ports by setting one of the environment variables provided in Table 4.

Table 4. Setting Environment Variables to Search for Ports

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH_DISCOVER_APPLICATION_PORTS</td>
<td>Specifies the agent ports for application service elements.</td>
</tr>
<tr>
<td>NH_DISCOVER_SERVER_PORTS</td>
<td>Specifies the agent ports for system elements.</td>
</tr>
<tr>
<td>NH_DISCOVER_PORTS</td>
<td>Specifies the agent ports for LAN/WAN, router, remote access, or probe elements.</td>
</tr>
<tr>
<td>NH_DISCOVER_RESPONSE_PORTS</td>
<td>Specifies the agent ports for response elements.</td>
</tr>
</tbody>
</table>

**Changing Ports on a UNIX System.** When you add or modify ports on an eHealth UNIX system in the nethealthrc.sh.usr file, separate them with a space and enclose them in double quotation marks as follows:

```
NH_DISCOVER_PORTS "161 5080 5081 5082"
```

**Changing Ports on a Windows System.** When you add or modify the port list on an eHealth Windows system using the System Properties dialog, separate each value with a space as follows:

```
161 1691 5080 5081
```

For instructions on setting environment variables, refer to the Web Help.

Save Your Elements to Groups Automatically

eHealth provides a grouping capability to allow you to report on groups of related elements. During discovery, you can use the Save to Groups option to automate this process. Discover can add elements to groups as it discovers the elements, or it can add discovered elements to existing groups.

When using this option, follow these guidelines:

- If you specify more than one group in this field, separate each group name with a comma. eHealth adds all elements as members to all of the groups.
- The Browse dialog shows all groups, both local and remote, in an eHealth cluster. Choose one group, and eHealth adds all of the elements that you discover to that group.
- If you specify a group name that does not exist, eHealth creates that group.
- If you used the Options dialog to specify an element filter, be aware that eHealth adds the discovered elements to the specified groups. If you specify a different group name in which to save the discovered elements, you will not be able to view the elements in the Poller Configuration dialog until you change the element filter.
Enable eHealth to Find System Processes

If you have installed a system agent, you can define process sets for the applications that are running on your systems. This enables eHealth to find those processes during the discover process. After discover finds systems, eHealth polls the agent to obtain statistics on the discovered processes and creates elements for them in your poller configuration. After eHealth completes the discover process and successfully polls the process set data, you can generate reports to determine the impact of a process set on certain system variables, such as CPU utilization.

If you are discovering system elements, and you plan to monitor processes and process set elements on that system, follow this process:

1. Select System from the Mode list.
2. Select Find Processes (Specify write community).
3. Click Define.
4. Create your process set by following the instructions provided in the eHealth — System and Application Administration Guide.
5. Specify the IP addresses for which eHealth should search.
6. Specify a community string that has read-write permissions.
7. Click Discover.

Ensure Data Continuity for Rebooted Systems

If a device experiences a failure, you may not be able to poll it successfully after the device reboots because the SNMP index values might have changed. If a system reboots, you must rediscover it to ensure that eHealth continues to collect data for it.

To identify all elements that have rebooted, run the following command:

```
nhListElements -rebooted -outfile file.txt
```

eHealth searches the messages.stats.log file for rebooted devices and generates a seed file containing a list of their IP addresses and community strings. To enable eHealth to collect data for the devices, perform an interactive discover and specify the seedfile in the IP List from File field.

If you want to automate this process, contact CA Technology Services for assistance in formulating Live Health Notifier rules that can work with the command line interface to enable the rediscovery of devices after they reboot.

Use Unique sysName and uniqueDeviceId Values

If two or more devices have the same sysName, uniqueDeviceId, or chassisId values, the discover process might not be able to identify these devices as unique devices based on the matching algorithm described in Appendix A. Use unique values for each device.

Use Unique ifDescr Values within a Device

The ifDescr is a MIB variable that describes an interface. For devices that have multiple interfaces such as switches and routers, each interface should have a unique ifDescr value. If the devices use the same ifDescr value for two or more interfaces, the discover process might not be able to differentiate among the interfaces.
Run Discover Again after Upgrading

Device information could change after you upgrade your operating system or SNMP agent software. Discover the device before the upgrade and resolve all unresolved changes; then discover the device again after the upgrade to resolve the element changes that occurred as a result of the upgrade.

Run a Discover after Each Device Change

If you plan to make several changes to an element (such as changing sysName, IP addresses, or ifDescr values), make one change at a time and discover the device after each change. If you change two or more attributes at the same time, discover might interpret the changes to be an entirely new device rather than updates to an existing device. For instructions on interpreting discover results, refer to “Reviewing Your Discover Results in the Log File” on page 13.

Increase the Number of Devices That eHealth Discovers

During discover, eHealth attempts to discover a maximum of four devices at a time. If you have a high-performance workstation, you can improve discover performance by changing the NH_FINDER_PROC_LIMIT environment variable to increase the number of devices that eHealth discovers at a time. Increasing the discover limit can, however, affect the system performance by consuming too much CPU during discovery. For more information about this variable, refer to the Web Help.

Using a Scheduled Discover Job

After you run an interactive discovery once, create a scheduled discover job to run periodically. A scheduled discover job automatically maintains the element definitions in your poller configuration to ensure that they continue to respond to eHealth polls. When elements do not respond to polls, eHealth cannot collect data from those elements. In contrast to the interactive discover, the scheduled discover job merges and saves the data to the database without allowing you to review it first. As a best practice, always review the discover log of a scheduled discover job to identify unresolved changes; then run an interactive discover process or modify the element information to resolve the changes.

Controlling Updates to the Poller Configuration

When a scheduled discover job runs, eHealth discovers elements at the specified IP addresses and creates the Discover.jobId.log file, which contains a list of the results for each IP address discovered. This file contains the same information that appears in the Discovery Results window of the Discovering dialog. eHealth saves this file and the discoverMerge.time.date.log file in the /ehealth/log directory. It saves a minimum of seven files for each type of file in this directory, and deletes files in excess of the seven files that are older than seven days.

Optionally, eHealth updates the poller configuration with information about new and changed elements unless you specify that the scheduled discover job should not automatically save the changes. You can review the changes in the discoverScheduled.time.date.log file and run an interactive discover process to apply the changes that you want to implement.
Saving Scheduled Discover Results
When you save the results of a scheduled discover job, eHealth saves the elements and updates the information relating to updated elements. It ignores duplicate elements. During the discover process, it also reports any previously discovered elements in the specific IP range that were not found during the current discover process. It might not find devices for the following reasons:

- The device is powered off or has been removed from the IT infrastructure.
- The device is not the same type that you were discovering. For example, RMON2 probes are not discovered during a LAN/WAN discover process.

Because scheduled discover jobs save changes automatically, they are more conservative in making changes to the poller configuration than the interactive discover process. By default, scheduled discover jobs add new elements, update existing elements, and add new keyed component elements to existing devices in the poller configuration. However, scheduled discover jobs never make unresolved updates to the poller configuration (that is, it does not match the element based on less reliable attributes such as MIB index and MTF if it cannot match on discover key or physical address).

When you schedule a discover job, you can specify the save options for the job. eHealth uses the current discover configuration for ports and element naming conventions.

Best Practices for Running Scheduled Discoveries
When scheduling a discover job, follow these important guidelines:

- Do not modify, add, or delete elements while a scheduled discover job is running. If you make changes using the Poller Configuration dialog, you must cancel the changes after the discover has finished.
- If you access the Poller Configuration dialog while a scheduled discover is running, eHealth may not provide current information about individual elements. Scheduled discover jobs do not update in an open Poller Configuration dialog. To view the changes that were made by the scheduled discover job, you must close and reopen the dialog.
- To prevent eHealth from rediscovering elements that you do not want to poll, disable polling or add them to an IP exclusion file, as described in “Excluding Specific Elements from the Search” on page 19. If you delete elements and do not use an IP exclusion file, eHealth could discover those elements again the next time that it runs a scheduled discover process.

Setting Up a Scheduled Discover Job
To set up a scheduled discover job, use the Add Scheduled Discover dialog. The information that you specify is similar to the information that you specify for an interactive discover process.
To schedule a discover process:

1. Select Setup → Schedule Jobs on the console.
2. Select Add Discover from the Add list in the Schedule Jobs dialog. Select one or more element types.
3. Specify the IP addresses for which discover should search.
4. Specify the method to use to save changes. Select Report Only if you do not want the scheduled discover process to update the poller configuration; otherwise, accept the default options.

**NOTE**

If the ifDescr attribute for a device changes, discover creates new elements for the device. (This is an unresolved element creation.) If you often change the ifDescr value for your devices, deselect Create new elements and use an interactive discover process to identify updates to devices that have new or changed ifDescr values.

5. Optionally, specify an e-mail address of a user in the Notify User field. eHealth sends the discover log file to this user.
6. Specify the days and time on which to run the scheduled discover job; then click Schedule.

### Preventing and Reconciling Unresolved Elements

Table 5 lists some common changes to components that the discover process might not be able to resolve. It provides suggestions and guidelines for preventing and reconciling unresolved elements. For detailed instructions, refer to the *eHealth Element and Poller Management Guide*. 
Table 5. Resolving Changes to Components

<table>
<thead>
<tr>
<th>Change</th>
<th>Suggested Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicating ifDescr values</td>
<td>Ensure that the ifDescr MIB variable is unique for each element on a device.</td>
</tr>
<tr>
<td>Using the same ifDescr MIB variable for more than one element on a device.</td>
<td></td>
</tr>
<tr>
<td>Adding new components</td>
<td>Ensure that <strong>Create new element</strong> is selected when you schedule a discover job. Although the discover log lists new components as unresolved, discover adds the element to the poller configuration if you select this option. (eHealth selects this option by default.)</td>
</tr>
<tr>
<td>For example, adding a new interface to a router that has been previously discovered.</td>
<td></td>
</tr>
<tr>
<td>Moving, removing, or upgrading (replacing) a component</td>
<td>Delete or retire the old element and then allow eHealth to discover the new element and create a new element in the poller configuration. Because eHealth assigns a new agent type to the element, it cannot track the new element as a replacement for the old element.</td>
</tr>
<tr>
<td>For example, replacing a token ring interface card with an FDDI interface.</td>
<td></td>
</tr>
<tr>
<td>Reusing a component</td>
<td>Delete or retire the old interface and then allow eHealth to discover the new interface and create a new element in the poller configuration.</td>
</tr>
<tr>
<td>Changing how an existing component is used, signified by changing the value of the ifDescr MIB variable; for example, redefining an Ethernet port as a token ring port.</td>
<td></td>
</tr>
<tr>
<td>Multiple changes at once</td>
<td>Modify information in the poller configuration so that the discover process does not find more than one change.</td>
</tr>
<tr>
<td>Changing the ifDescr, partition name, or device name MIB variable at the same time as changing a device-level attribute.</td>
<td></td>
</tr>
<tr>
<td>Restarting the device</td>
<td>If you know about these events before you run a discover process, you can change the SNMP index in the poller configuration. Typically, you need to reconcile an index shift after the discover process by examining the discover log and your poller configuration; then modify the poller configuration accordingly.</td>
</tr>
<tr>
<td>Causes SNMP indexes to shift; this applies to elements that do not have a discover key or unique MAC address.</td>
<td></td>
</tr>
</tbody>
</table>

Table 6 lists device-level changes that the eHealth discover process might not be able to resolve and provides suggestions for managing these changes.

Table 6. Resolving Changes to Devices  (Page 1 of 2)

<table>
<thead>
<tr>
<th>Infrastructure Change</th>
<th>Suggested Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicating the system name</td>
<td>Ensure that each device on the infrastructure has a unique sysName value.</td>
</tr>
<tr>
<td>Assigning the same value to the sysName MIB variable of more than one device on the infrastructure</td>
<td></td>
</tr>
</tbody>
</table>
Troubleshooting Discover Problems

This section describes typical problems that you might encounter when running a discover process, and offers some possible solutions.

Discover Errors and Failures

eHealth provides a number of environment variables to control discovery. By modifying these variables, you can resolve many of the typical errors and failures that you encounter:

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH_DISCOVER_IF_IP_ADDR</td>
<td>Enables eHealth to use the specific IP address of an interface to poll elements.</td>
</tr>
<tr>
<td>NH_DISCOVER_SERVER_PORTS</td>
<td>Specifies the port on which a system agent is running.</td>
</tr>
<tr>
<td>NH_DISCOVER_IOS_SVC</td>
<td>Discovers ATM PVCs on a Cisco Router.</td>
</tr>
<tr>
<td>NH_SNMP_RETRIES</td>
<td>Extends the number of times that eHealth will attempt to poll a device before timing out.</td>
</tr>
<tr>
<td>NH_SNMP_TIMEOUT</td>
<td></td>
</tr>
<tr>
<td>NH_DISCOVER_DISABLE_PING</td>
<td>Disables discover’s ability to ping an address before it sends SNMP packets (if ping is not allowed or recognized by the target machines).</td>
</tr>
</tbody>
</table>

Devices Are Not Certified

If you encounter any of the following problems, you may be trying to discover a device that is not certified:

- Fail to discover PVCs for ATM probes
- Fail to discover multiple routers with channels
- Unable to identify SNMP agents that allow eHealth to discover systems
- Unable to determine which systems to poll

To determine if devices are fully certified, navigate to the Certified Devices page on the product Web site.
Encountering a Fatal Error on a UNIX System after a Discover

If you receive a fatal error in the eHealth console system messages window after attempting to perform a discover, you may need to change the ownership and permissions of some of your files.

To change permissions:

1. Change directory to `/ehealth/bin/sys`.
2. Confirm that the following files are owned by root and have the setuid bit set:
   - nhiDiscover
   - nhiPoller
   - nhiTrapServerCmu
3. If any of the above files are not owned by root, change the permissions on each file to root.

Unable to Discover Gigabit Ethernet Interfaces on a System

When eHealth discovers gigabit Ethernet interfaces on a system, they appear in the poller configuration as LAN or WAN elements. eHealth does not recognize the gigabit Ethernet interface description, so it assigns a description of MIB2 (WAN port) to it by default.

To ensure that these elements appear as system elements in a report, do one of the following:

- Submit a certification request to certify the interfaces.
- Use the Poller Configuration dialog to change the agent type of the element from MIB2 (WAN port) to MIB2 (LAN port). Also, change the speed to one gigabit and assign the system as the parent object.
- Contact CA Technology Services to request assistance in modifying your version of eHealth.

Unable to Perform a Discovery in a Particular Mode

If you are unable to perform a discovery in a particular mode, check the status of your licenses by using OneClickEH to confirm that the appropriate license is present and has not expired.

Unable to Save Data for an Element in the Database

Although eHealth can discover an element that is unavailable, it does not save data for an element in the database until the device becomes available and the setting of the ifOperStatus field changes to up. If you run reports on the element when this field has been set to down, the generated reports are blank because the database does not contain any data for the element. For more information, refer to the chapter on creating a MIB translation file (MTF) in the eHealth Customizing Variables Guide.

Receiving Incorrect Line Speeds on Devices after Discovery

If you encounter incorrect line speeds on devices after you discover them, you must modify the elements to use the correct speed. For instructions, refer to the eHealth Element and Poller Management Guide.

Reporting No Data for New Variables

When you upgrade to a new release of eHealth, you should rediscover elements to update them for any changes, such as new variables that are available. If you do not rediscover the elements, your database will not have this information. If a variable has changed, eHealth updates the elements after you perform a discovery on them.
Discovered Different Agent Types for One Element

An element can have two different agent types as a result of the way in which it was discovered. If you discover a device using multiple modes, the discover process creates interface elements that are children of a parent for aggregate statistics using one agent type, and also creates interface elements for individual port statistics with another agent type. This is normal eHealth functionality. You must decide how you would like to manage your devices, and only discover these dual-approach agents using one mode.

Unable to Poll and Report on ATM Ports, Paths, and Channels

eHealth defines WAN links, including those used for ATM or to carry Frame Relay PVCs, as WAN elements. However, to poll and report on ATM ports, paths, and channels, you must have an eHealth – ATM license; to poll Frame Relay PVCs, you must have an eHealth – Frame Relay license. After you add these licenses (if they are not already in your license.dat file), you must run discover again.

Unable to Send Scheduled Discover Results to Multiple Recipients

To send scheduled discover results to multiple recipients, do not use the same method that you would use to send reports to multiple people. You must enter all e-mail addresses in the Notify User field of the Add Scheduled Discover Jobs dialog, and separate each one with a comma. Do not use spaces.

Unable to View Elements after Specifying a Group Name

If you used the Options dialog to specify an element filter, eHealth adds the discovered elements to the specified groups. If you specify a different group name in which to save the discovered elements, you will not be able to view the elements until you change the element filter.

Unable to Discover the Speed of Elements

When discovering ATM elements, eHealth discovers the Speed Out variable if it is included in the device MIB. With some devices, you can override this. If you change the committed information rate (CIR) speed on a router, eHealth does not automatically update it; you must rediscover that device.

Discovering RAS Devices as Router Elements

Because remote access server (RAS) devices perform routing services, the discover process for routers could discover RAS devices as router elements. If you discover a RAS as a router element, eHealth includes the RAS in router reports, but not in remote access reports. Manage your RAS devices using the eHealth — Remote Access Server license—not a Router license. If you choose to report on a RAS as both a router element and a RAS element, eHealth polls the RAS device and its elements twice.

Unable to Collect Alternate Latency Information or Data from Response Sources

To collect alternate latency information or data from response sources, specify a community string that gives you read and write access to the element when you discover or modify its information in the poller configuration.
**System Mode Issues**

The following are typical issues that you may encounter when discovering in System mode.

**Unable to Exclude Data for Remotely Mounted Partitions**

To exclude data for remotely mounted partitions, drives, or volumes from your report, disable polling for the remote partitions. If you delete the remote partitions, eHealth will rediscover them the next time that it discovers the system.

**Unable to Create Process Set Elements for a System**

eHealth creates process set elements and process elements for a system if the system has a system agent and if the application running on that system is defined in the process definition file. eHealth can only poll process elements for a system for which you have specified a community string that has read-write permissions. For more information, refer to the eHealth — System and Application Administration Guide.

**Unable to Discover Sun/System Token Ring Interfaces**

When you discover token ring interfaces under System mode, eHealth discovers them as MIB2 LAN elements and reports generic statistics. To obtain true token ring specifics, you must discover the interface under LAN/WAN mode.

**Router Mode Issues**

The following are typical issues that you may encounter when discovering in Router mode.

**Recording Statistics Data for an Element**

If you discover a router interface or system interface element with the Record detail data option enabled in the Modify Element dialog, and then later deselect it and perform a rediscovery, Discover checks the setting of the option for each interface on the router.

- If only a few of your interfaces (or none) are configured to record detail data, OneClickEH discovers the router/switch elements using only Router/Switch mode, updates them, and adds any new elements. You must enable the Record detail data option for all interfaces manually using the Poller Configuration dialog.

- If all interfaces on the router/switch have the option enabled, OneClickEH redisCOVERS the interfaces using both Router/Switch and LAN/WAN modes. If it finds new interfaces, it adds them and enables the option by default. If you want to disable the option, you must do so manually.

This option appears (but does not require a poller license) for individual system process elements if the value of the NH_STORE_PROCESS_DATA environment variable is set to Yes.

**Receiving Incorrect Statistics When Rediscovering Lite Switches**

If you do not delete the Lite switch from the poller configuration before rediscovering it, eHealth includes the Lite backplane by aggregating the statistics from the interfaces and the backplane, which results in inflated total statistics for the switch. Likewise, before rediscovering an existing traditional or Plus switch element as a Lite switch, you must delete it.
Unable to Track Data for an Element during Link Change from One Router to Another

Sometimes eHealth cannot seamlessly track the data for an element during a change. If the element type changes, such as from a token ring interface to an FDDI interface, eHealth cannot use one element to track both types of data. You must retire the old element and create a new element for the new element type. For instructions, refer to the eHealth Element and Poller Management Guide.

Double Polling Routers That Are Also Systems

If you run discover for both Router and System modes, eHealth discovers elements that are both routers and systems twice: once as router elements, and once as system elements. eHealth also polls those elements twice. You can retire one of the elements to prevent double polling of the same element. For information about retiring elements, refer to the eHealth Element and Poller Management Guide.

As a best practice, limit your discovery of elements that could be discovered as both systems and routers to the mode in which you are interested. If you want to save detail data on the interfaces to the database for reporting, discover the elements in either LAN/WAN and System modes or in LAN/WAN and Router modes, but not in all three. If you are not concerned with detailed data but want to run System reports or Router reports, run discover in Router and System modes, but not in LAN/WAN mode.
The Discovery Process: Element Matching

eHealth uses a very complex matching algorithm to identify changes within your infrastructure. This appendix describes the matching algorithm in detail.

The Matching Algorithm

The discover process tracks changes and correctly updates the poller configuration by matching at the nmsSource level first. It then matches at these three abstract levels, as illustrated in Figure 1:

- Device
- Agent
- Element

Figure 1. Hierarchy for Discover Matching
Matching at the Device Level

At the device level, discover first attempts to match all of these attributes:

- **System name** – Name assigned to the sysName variable in the MIB on the device.
- **Unique device ID** – Typically, a chassis ID or similar attribute found in the MIB. Some devices allow you to override the unique device ID with a chassis ID. These values must be unique across your IT infrastructure.
- **IP address** – IP address assigned to the IpAddr variable in the MIB on the device (for devices, eHealth always uses the polled IP address of the device).

If discover does not find a device match based on the above, it follows this process:

1. It attempts to match two-out-of-three of those attributes.
   - If a match on all three attributes or the two-out-of-three match produces multiple possibilities, discover attempts to match the set of interface physical addresses and then the set of interface IP addresses.
   - If none of the devices match at least two of the three attributes, discover attempts to match sets of physical addresses and then sets of IP addresses. If discover does not find any matches through these methods, the device is new.

2. After matching attributes, discover compares the set of physical and IP addresses from the device interfaces with those that are stored in the poller configuration. The greater the number of matches, the greater the confidence in the device match.
   - If attribute matching indicated that a device is new, lack of an address match improves the confidence that the device is new.
   - If discover does not identify any attribute matches but does identify a near-perfect address set match, the attributes for an existing element have changed significantly. eHealth accepts the address set match (by default) and updates the attributes in the poller configuration to match the newly discovered attributes.

Verifying a Device Match

After matching attributes, discover compares the set of physical and IP addresses from the device interfaces with those that are stored in the poller configuration. The greater the number of matches, the greater the confidence in the device match.

If attribute matching indicated that a device is new, lack of an address match improves the confidence that the device is new. If discover does not identify any attribute matches but does identify a near-perfect address set match, the attributes for an existing element have changed significantly. eHealth accepts the address set match (by default) and updates the attributes in the poller configuration to match the newly discovered attributes.

Matching at the Agent Level

Once eHealth finds a matching device, it proceeds to match at the agent level. eHealth sets the agent type in the poller configuration by gathering information from the MIB on the device. At the agent level, discover matches on the MIB translation file (MTF) of the parent element (if it exists), SNMP port, and enterprise ID. If an agent does not match at the parent MTF, SNMP port, and enterprise ID, it is considered a new agent.
Matching at the Element Level

Once it finds an agent, eHealth proceeds to match at the element level. Discover matches the following—in order from best to worst—to track changes and update the poller configuration:

1. Discover keys
2. Physical address and virtual interface ID (if defined)
3. MIB indexes and MTF

These methods help the discover process determine whether an element is a new addition or if it is an existing element that has been moved. You must track these changes to ensure the accuracy of eHealth reports. The discover log reports the results of these comparisons so that you can identify the changes that discover made or recommended. For detailed information on using discover logs, refer to Chapter 2, “Finding Your Resources: Running Discover for the First Time.”

Using Discover Keys

eHealth uses a unique key to identify each element during discovery. For most elements, this unique key is the value of the ifDescr field (as described in Table 7 on page 36). All interface elements of a device should have unique (that is, within the device) ifDescr values. eHealth creates discover keys for newly discovered elements based on information that it obtains from the MIB at the device.

eHealth compares the unique key for each newly discovered element with the unique key for elements that are already in the poller configuration. If it does not find an element that has the unique key, it considers the newly discovered element to be a new element. If an element in the poller configuration has the same unique key, eHealth performs additional checks:

- If the newly discovered element and the element in the poller configuration have identical information, it considers the newly discovered element to be a duplicate.
- If the two elements have different information, it considers the newly discovered element to be an updated element.

If the element does not have a discover key, the discover process uses a different method to identify it. It follows the device- and agent-matching techniques described in the previous section.

eHealth creates a discover key for certain components that meet the following requirements:

- The element is certified for use with eHealth.
- eHealth was able to obtain (from the MIB) the attributes and other information necessary to create a discover key.

NOTE

If discover keys do not match or if the keys are not unique for the agent, discover attempts to match the physical (MAC) address of the first (numerically lowest) interface card that has a MAC address on the device (through the IfPhysAddr variable) and to enhance that information with proprietary information, such as shelf, card, and slot numbers.
Discover Key Formats

eHealth uses information that it obtains from the MIB to create and maintain discover keys for component elements in the poller configuration. For interfaces, the discover key also includes a component type that eHealth assigns based on discovered information. All discover keys include different fields that are based on the type of element. Table 7 lists the types of components for which eHealth can create discover keys and describes their format.

<table>
<thead>
<tr>
<th>Component Type</th>
<th>Discover Key Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application service</td>
<td><code>applicationName</code></td>
</tr>
<tr>
<td>Application service process sets</td>
<td><code>applicationName</code></td>
</tr>
<tr>
<td>Ascend, Cisco, and Shiva ISDN interfaces</td>
<td><code>ras componentType ifDescr</code></td>
</tr>
<tr>
<td>Ascend, Cisco, and Shiva modem interfaces</td>
<td><code>pool componentType ifDescr</code></td>
</tr>
<tr>
<td>ATM path on a Cisco LightStream 1010 switch</td>
<td><code>ifDescr VPI</code></td>
</tr>
<tr>
<td>ATM channel on a Cisco LightStream 1010 switch</td>
<td><code>ifDescr VPI VCI</code></td>
</tr>
<tr>
<td>Bay Networks, WellFleet, and generic Frame Relay permanent virtual circuits (PVCs); other PVCs conforming to RFC 1604</td>
<td><code>ifDescr DLCI</code></td>
</tr>
<tr>
<td>Cisco Catalyst 5000 Switching System interfaces</td>
<td><code>port Module-module#-Port-port#</code></td>
</tr>
<tr>
<td>Generic router and system interfaces</td>
<td><code>componentType ifDescr</code></td>
</tr>
<tr>
<td>Physical disks on systems</td>
<td><code>deviceName</code></td>
</tr>
<tr>
<td>Processes</td>
<td><code>processSetName</code></td>
</tr>
<tr>
<td>Process sets</td>
<td><code>processSetName processName - arguments</code></td>
</tr>
<tr>
<td>QoS</td>
<td><code>ifDescr direction elementTypeID cbQosCMName</code></td>
</tr>
<tr>
<td>System CPU, router CPUs</td>
<td><code>Cpu cpu# or CIP-Cpu cpu#</code></td>
</tr>
<tr>
<td>System disk partitions (including Compaq and others)</td>
<td><code>partitionName</code></td>
</tr>
</tbody>
</table>

Table 8 defines each discover key field.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>applicationName</code></td>
<td>For an application service element, specifies the name of the eHealth AIM.</td>
</tr>
<tr>
<td><code>cbQosCMName</code></td>
<td>For QoS elements, specifies the user-specified name of the class of service.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| componentType         | For interfaces, specifies the component type, as one of the following:  
                        | - seg  
                        | - enet-card  
                        | - enet-port  
                        | - enet-dot3port  
                        | - fddi-port  
                        | - tr-port  
                        | - link  
                        | - ring  
                        | - modem  
                        | - isdn  
                        | eHealth assigns the component type based on information about the element that it obtains from the MIB. |
| Cpu cpu#              | For CPU elements, specifies a number that eHealth assigns to the CPU, based on its position in the list of CPUs obtained from the MIB.       |
| deviceName            | For a disk, specifies its device name, as obtained from the MIB.                                                                               |
| direction             | For QoS elements, specifies the direction of a uni-directional class of service (CoS) element (either IN or OUT).                           |
| DLCI                  | For a Frame Relay circuit, specifies its Data Link Connection Identifier (DLCI). The DLCI is a number assigned by the Frame Relay station to a specific data link connection. |
| elementTypeID         | For QoS elements, identifies the element type as one of the following:  
                        | - <> for basic CM elements  
                        | - CP for policing elements  
                        | - CS for shaping elements  
                        | For more information, refer to “Discovering QoS Elements” on page 43. |
| ifDescr               |  
                        | - For MIB2 elements, specifies the value of the ifDescr variable in the MIB.  
                        | - For application service and application service process set elements, specifies the installation location of the application.  
                        | - For QoS elements, specifies the interface on the router. |
| partitionName         | For a partition on a system, specifies the name of the partition, as obtained from the MIB.                                                   |
| portModule-module#    | For an interface in a Cisco Catalyst 5000 Switching System, specifies the interface’s module number as one of the following: 5500, 5000, 5002 |
| port-port#            | For an interface in a Cisco Catalyst 5000 Switching System, specifies the interface’s port number.                                         |
Using Physical Address

After matching on discover keys, discover compares the physical address (ipPhysAddr) of the elements with those that are stored in the poller configuration. In cases of virtual interfaces, such as Frame Relay PVCs, ATM PVCs, or workgroup hub modules (where the physical interface can be the same for more than one interface), discover uses the virtual channel identifier to identify virtual addresses within a physical one.

Using MIB Attributes

If the discover process has not been able to match an incoming element through the device-, agent-, and discover key-matching methods, it compares the MIB attributes of discovered elements to values in the poller configuration to determine whether the discovered elements are new. It compares attributes to identify components for which it cannot create a discover key and to identify devices. eHealth also uses this method to distinguish among two or more component elements that have identical discover keys.

Identifying components by comparing MIB attributes can be unreliable because the comparison includes SNMP indexes, which can shift when the device is restarted. The discover process cannot determine whether the SNMP indexes shifted. When it finds an element on an existing device that it cannot match to an element in the poller configuration, the discover log lists it as resolved-new, unless the device match is weak, in which case it lists the element as unresolved-new. For information about unresolved elements, refer to “Network Change Summary Section” on page 14.

NOTE

If MIB indexes are the only differentiating factor among elements (that is, the discover keys are not unique, and the physical address is not unique or is undefined), discover cannot accurately track any index shifting that may occur.
eHealth Element Naming Conventions

This appendix explains the modes that eHealth uses to discover your resource types, the conventions that it uses to name the elements that discover finds, and the ways in which you can customize your element names to make them more recognizable.

Element Creation

For each resource that eHealth discovers within your infrastructure, the method by which it creates a corresponding element varies. The following sections explain the process that eHealth uses to create elements and add them to your database and element configuration.

Discovering LAN/WAN Elements

When you run the discover process and select LAN/WAN mode, eHealth searches the specified IP addresses for interfaces. Upon discovering an interface, eHealth does the following:

- For each Ethernet, 802.3, token ring, or FDDI type interface, it creates a LAN element.
- For each of the following, it creates a LAN or WAN element and identifies the type of interface, as well as one or more SNMP indexes assigned to it:
  - LAN or WAN interface
  - ATM port, path, and channel
  - Frame Relay PVC

eHealth defines WAN links—including those used for ATM or to carry Frame Relay PVCs—as WAN elements. You can poll and report on ATM ports, paths, and channels, and Frame Relay PVCs.

Discovering Router/Switch Elements

When you run the discover process and select Router mode, eHealth searches the specified IP addresses for routers and switches. You can use either standard router/switch reporting or enhanced router/switch reporting. The default mode of the router/switch discover process is enhanced. If a device does not support enhanced switch reporting, eHealth discovers the switch as a traditional (standard) switch. For a list of devices that support enhanced router/switch reporting, refer to the Certified Devices Web page at http://support.concord.com.

With standard router/switch reporting, for each router or switch, eHealth creates a router/switch element and the following:

- Interface element for each interface on the router/switch
- CPU element for each CPU on the router/switch
The default element name for a CPU or interface element includes the name of the router/switch to which it belongs. Although the discover process creates an element for each interface on a router/switch, eHealth router reports do not report the statistics for the individual interfaces.

Instead, the reports combine the statistics for the interface elements. For example, router reports show the following:

- Average line utilization as a percentage of total line capacity
- Percentage of all frames with faults and frames that were discarded

eHealth summarizes the data from the interfaces to determine the amount and type of traffic that passed through the router or switch. It also analyzes the CPU utilization and buffering to measure the health of your routers or switches.

**Running Discover for Router/Switch Elements after Running It for LAN/WAN Elements**

If you run the discover process for router/switch elements after running it for LAN/WAN elements, the poller configuration uses the existing LAN element entry for the same router/switch LAN interface element and the existing WAN element entry for the same router/switch WAN interface element. If you rediscover the router/switch interface elements first, you can rediscover them in LAN/WAN mode to report on the individual elements.

**Using Enhanced Router/Switch Reporting.** With enhanced router/switch reporting, eHealth does the following:

- If the router/switch is a supported switch Plus device, eHealth creates the router/switch element and the following:
  - Interface element for each interface on the router/switch
  - CPU element for each CPU on the router/switch
  - Backplane element
- If the switch is a supported switch Lite device, eHealth creates the router/switch element and the following:
  - CPU element for each CPU on the router/switch
  - Backplane element

eHealth uses the router/switch CPU element to store CPU utilization, as well as memory and environmental statistics. Therefore, the discover process always creates a CPU element for a Plus or Lite router/switch, regardless of whether the CPU utilization data is available from the router/switch.

**Discovering Routers/Switches Using Switch Plus and Switch Lite Modes.** The default mode of the router/switch discover process is switch Plus. If eHealth discovers a router/switch that is a supported Plus device, it creates the router/switch element, its interfaces, the backplane, and the CPU. If you rediscover a traditional router/switch, eHealth simply adds the backplane and CPU elements.

To discover devices as Lite routers/switches, you must set the value of the NH_DISC_SW_MODE environment variable to lite. If eHealth discovers a switch that is a supported Lite device, it creates the switch element, the backplane, and the CPU. To change from switch Lite mode to switch Plus mode, you
must change the value to plus. However, before rediscovering a Lite switch as a Plus switch, you must first delete it from your poller configuration.

**NOTE**

If you do not delete the Lite switch from the poller configuration before rediscovering it, eHealth includes the Lite backplane by aggregating the statistics from the interfaces and the backplane, which results in inflated total statistics for the switch. For the same reason, you must delete an existing traditional or Plus switch element before rediscovering it as a Lite switch.

If a device does not support enhanced switch reporting, eHealth discovers the switch as a traditional switch, including the switch element and its interfaces. In some cases, eHealth discovers a switch as a router when it has been enhanced to include additional router information (for example, a Cisco Catalyst switch with the Router Switch Module (RSM)). For a list of enhanced switch devices, refer to the Certified Devices Web page at http://support.concord.com.

**Discovering LAN/WAN and Router/Switch Elements**

When you choose to concurrently discover LAN/WAN and router/switch elements, eHealth uses essentially the same element naming conventions with the exception of the following:

- When you discover LAN/WAN interface elements with Router/Switch mode, the router will be the parent for the interface elements.
- When you discover router interfaces with LAN/WAN mode, the Include detailed data option will be enabled for those elements in the Modify Element dialog.

EHealth appends -RH to the element name to designate router/switch parents, and the subelements such as CPUs and backplanes. These elements often use the convention *parentName*-cpu-x or *parentName*-Bkpln.

**Discovering Traffic Accountant Elements**

When you run a discover process and select Probes mode, eHealth searches the IP addresses specified for probes. A probe is a device that contains remote network monitoring (RMON) software to record information about the network traffic that it observes. A probe can also be embedded RMON software in a device. Probes collect data about the conversations between node pairs on the network. Although the information collected by each probe can vary, typical information includes the address of the sending node, the address of the receiving node, the number of packets and bytes transmitted, and the protocol or application type of the data. For information about discovering Traffic Accountant elements, refer to the Traffic Accountant Web Help.

**Discovering System Elements**

When you run the discover process and select the System mode, eHealth searches the specified IP addresses for systems. An additional discover option, **Find Processes (Specify write community)**, is selected by default to track the impact of individual processes in the system. Click Define to define process sets that associate processes as an application. For information about discovering system elements and defining process sets, refer to the Web Help.
Discovering Application Elements

When you run the discover process and select Application mode, eHealth searches the specified IP addresses for applications.

When you discover both system and application elements, eHealth discovers both licensed and unlicensed eHealth application insight modules (AIMs). When you discover response elements, eHealth discovers both licensed and unlicensed eHealth Service Availability (SA) modules. If you have licensed Fault Manager, you can view the unlicensed modules (with a suffix of unlicensed) in the Poller Configuration dialog. To view only the unlicensed system agents, SA modules, and eHealth AIMs, select Unlicensed SysEdge Element in the List Elements of Type field of the Poller Configuration dialog, and then click Refresh Element List. You can rediscover these elements after you license them by using the Systems & Apps tab of the eHealth Web interface. When they are licensed, eHealth can begin to poll them. For information about licensing these eHealth AIMs, refer to the Web Help.

Discovering Modem Pool and RAS Elements

You can run the discover process for modems, modem pools, and Remote Access Server (RAS) devices. You can discover RAS devices as RAS elements, modem pool elements, or both.

- To manage RAS devices as physical devices, discover the devices using Remote Access Server mode. This enables you to run reports on the RAS device, its CPU, and all of its modems and ISDN interfaces.
- To manage the modems and ISDN interfaces as groups within one or more RAS devices, discover the RAS devices using the Modem Pool mode. This enables you to report on modem pools that might span one or more RAS devices.

If you discover a RAS device using both Remote Access Server and Modem Pool modes, eHealth polls the elements on the RAS device twice, once for RAS statistics and once for modem pool statistics. For best poller performance, you should manage RAS devices as RAS elements or modem pool elements, but not both.

Remote Access Servers

When you run a discover process and select Remote Access Server mode, eHealth searches the specified IP addresses for RAS devices. When eHealth discovers a RAS device, it creates an element for each of the following:

- The RAS
- Each CPU in the RAS
- Each modem in the RAS
- Each ISDN interface in the RAS

eHealth assigns each CPU, modem, and ISDN element to the RAS element to which it belongs. When you run reports for the RAS device, eHealth combines the total data collected for each modem and ISDN element in the RAS to report on the health of the RAS. You can also run reports on the individual modem and ISDN elements in the RAS.

**NOTE**

Because RAS devices perform routing services, the discover process for routers could discover RAS devices as router elements. If you discover a RAS as a router element, eHealth includes the RAS in router reports, but not in remote access reports. Manage your RAS devices using Remote Access mode—not Router/Switch mode. If you choose to report on a RAS as both a router element and a RAS element, eHealth polls the RAS device and its elements twice.
Modem Pools

When you run a discover process and select **Modem Pool** mode, eHealth searches the specified IP addresses for RAS devices.

When eHealth discovers a RAS device, it creates an element for each of the following:

- Each modem pool configured at the device
- Each modem in the device
- Each ISDN interface in the device

If the RAS device agent has modem pool information defined in the MIB, eHealth uses the MIB information to configure the modem pool element name and assign modem and ISDN elements to the modem pool. If the RAS device agent does not have a modem pool definition, eHealth creates a single modem pool element for the RAS device and assigns all modem and ISDN elements in the device to it. If the default modem pool does not reflect your modem pool configuration, you can create modem pool elements and assign the modem and ISDN elements to the correct modem pools, as described in the *eHealth Element and Poller Management Guide*. When you run reports for the modem pool, eHealth combines the total data collected for each modem and ISDN element in the pool to report on the health of the modem pool. You can also run reports on the individual modem and ISDN elements in the modem pool.

Discovering Response Elements

When you run the discover process and select **Response Elements** mode, discover searches the specified IP addresses for response agents in your network such as Cisco IP SLA and Juniper RPM routers. eHealth can measure the response time of the applications and services that reside on those devices. **Response time** is the elapsed time between a user request and a system response. By measuring the actual response times that end users experience, you can detect degrading performance and declining availability of your critical services and applications and maintain service levels.

After you discover response elements and save them to the eHealth database, you must create **response destination** elements. You can create these elements from existing response source elements, or you can specify the IP address of a new element that does not yet exist in the poller configuration. You use a destination element to indicate one endpoint of a response path that eHealth monitors. You specify an existing response source to identify the other endpoint of the response path. eHealth monitors activity between the endpoints that you specify. For detailed information about the response technology, refer to the *Using eHealth – Response* guide.

Discovering QoS Elements

Quality of Service (QoS) is a technology that allows you to configure different levels of service for different types of traffic in your infrastructure. If you have devices that support QoS, you can configure your routers to recognize different types of traffic, and to treat each type differently based on a Class of Service (CoS) that you define for each traffic type. To implement QoS in your infrastructure, you must identify and define your various types of traffic and configure a CoS for each type. For detailed information about the QoS technology, refer to the Web Help. For instructions on discovering QoS elements, refer to the *Managing Cisco QoS Using eHealth* focus topic.
### Naming Elements

After discovering an element, eHealth assigns a name to it. Element names are represented as case-sensitive, but eHealth does not use case to determine the uniqueness of an element. By default, it uses the following format to name elements, as defined in Table 9:

\[ \text{devName-suffixMode-suffixType-suffixIndex1-suffixIndex2} \]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>devName</strong></td>
<td>Name of the device.</td>
</tr>
</tbody>
</table>
| **suffixMode** | For router and system elements only, indicates the type of parent:  
  - POOL for modem and ISDN pools  
  - RH for routers  
  - RAS for remote access servers  
  - SH for systems  
  - RS for response sources  
  - RD for response destinations  
  - AP for application paths  
  - AH for applications  
  - QOS for QoS |
| **suffixType** | Indicates the type of element, for example:  
  - Apache for Apache application elements (for system agents with the appropriate eHealth AIM)  
  - atm-path for ATM paths  
  - atm-port for WAN ports used for ATM  
  - Cpu for CPUs in routers and systems  
  - disk for system disk elements (system partition elements do not use this variable)  
  - dlc for endpoints of a Frame Relay PVC  
  - enet-port for Ethernet LAN and MIB2 elements  
  - Exchange (for system agents with the eHealth AIM for Microsoft Exchange)  
  - IIS for Microsoft IIS application elements (for system agents with the eHealth AIM for Microsoft IIS)  
  - isdn for ISDN devices  
  - link for WAN links, including those used for Frame Relay  
  - modem for modems  
  - OracleDB for Oracle application elements (for system agents with the eHealth AIM for Oracle)  
  - NetworkSvcs_ServiceName for Network Services for UNIX application elements (for system agents with the eHealth AIM for Network Services for UNIX) |

(continued on next page)
Table 9. Element Name Components (Page 2 of 2)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>suffixType</td>
<td>• NetworkSvcsNT_ServiceName for Network Services for Windows application elements (for system agents with the eHealth AIM for Network Services for Windows)</td>
</tr>
<tr>
<td></td>
<td>• probe for RMON probe elements</td>
</tr>
<tr>
<td></td>
<td>• rptGroup for repeater groups</td>
</tr>
<tr>
<td></td>
<td>• RD for response destination elements (for system agents with eHealth Service Availability)</td>
</tr>
<tr>
<td></td>
<td>• RS for response source elements (for system agents with eHealth Service Availability)</td>
</tr>
<tr>
<td></td>
<td>• seg for Ethernet LAN RMON elements</td>
</tr>
<tr>
<td></td>
<td>• SONET for fiber-optic cable elements</td>
</tr>
<tr>
<td></td>
<td>• SQLServer for Microsoft SQL application elements (for system agents with the eHealth AIM for Microsoft SQL Server)</td>
</tr>
<tr>
<td></td>
<td>• tr for token ring LAN elements</td>
</tr>
<tr>
<td></td>
<td>• unlicensed for unlicensed system agents, Service Availability modules, and eHealthAIMs</td>
</tr>
</tbody>
</table>

| suffixIndex1     | The first index from the interface table for the element. System partitions use the partition name (not the index number).                      |
| suffixIndex2     | The second index from the interface table for the element.                                                                                   |

Table 10 lists the environment variables that define the name selection process that the discover process uses to create the devName value.

Table 10. Element Name Environment Variables

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH_USE_NAME_SRVC</td>
<td>Specifies whether the discover process uses the name provided through the domain name system (DNS) or the content of the sysName field.</td>
</tr>
<tr>
<td></td>
<td>• When set to no (the default), discover uses the sysName field.</td>
</tr>
<tr>
<td></td>
<td>• When set to yes, discover uses the name provided through DNS. However, router parent elements do not use the DNS name or the name specified in the /etc/hosts file regardless of the variable setting.</td>
</tr>
<tr>
<td></td>
<td>• When set to file, discover uses the hosts file defined by an internal variable, NH_HOST_FILE.</td>
</tr>
<tr>
<td>NH_NAME_IF_ADDR</td>
<td>Specifies whether the discover process uses the interface table address or the polled address for element names. By default, eHealth sets this variable to yes, and the discover process uses the interface address for calculating the name. When the variable is set to no, the discover process uses the polled address.</td>
</tr>
</tbody>
</table>
| NH_NAME_SRVC_PAT     | Specifies a regular expression that indicates the portion of the name to truncate. For a description of how to specify regular expressions, refer to any UNIX documentation. By default, eHealth sets this variable to \( \ldots \cdot \) $. On a UNIX system, surround the setting with quotation marks (""), for example: * (\ldots\cdot) $. To use this variable, you must also set NH_USE_NAME_SRVC to yes.
Appendix B  eHealth Element Naming Conventions

The values of the NH_NAME_IF_ADDR and NH_USE_NAME_SRVC environment variables control how discover selects element names. For each of the four combinations of environment variable settings, the discover process uses the first address or name that it finds by following the steps listed in Table 11.

Table 11. Name Selection Steps

<table>
<thead>
<tr>
<th>Environment Variable Settings</th>
<th>Name Selection Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH_NAME_IF_ADDR=yes</td>
<td>1. Use sysName.</td>
</tr>
<tr>
<td>NH_USE_NAME_SRVC=no</td>
<td>2. Use IP address of the interface.</td>
</tr>
<tr>
<td></td>
<td>3. Use IP address of the device polled address.</td>
</tr>
<tr>
<td>NH_NAME_IF_ADDR=no</td>
<td>1. Use DNS name of the device polled address.</td>
</tr>
<tr>
<td>NH_USE_NAME_SRVC=yes</td>
<td>2. Use sysName.</td>
</tr>
<tr>
<td></td>
<td>3. Use IP address of the device polled address.</td>
</tr>
<tr>
<td>NH_NAME_IF_ADDR=yes</td>
<td>1. Use DNS name of the interface address.</td>
</tr>
<tr>
<td>NH_USE_NAME_SRVC=yes</td>
<td>2. Use DNS name of the device polled address.</td>
</tr>
<tr>
<td></td>
<td>3. Use sysName.</td>
</tr>
<tr>
<td></td>
<td>4. Use IP address of the interface.</td>
</tr>
<tr>
<td></td>
<td>5. Use IP address of the device polled address.</td>
</tr>
<tr>
<td>NH_NAME_IF_ADDR=no</td>
<td>1. Use sysName.</td>
</tr>
<tr>
<td>NH_USE_NAME_SRVC=no</td>
<td>2. Use IP address of the device polled address.</td>
</tr>
</tbody>
</table>

Table 12. NH_NAME_SRVC_PAT Values

<table>
<thead>
<tr>
<th>Platform</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX</td>
<td>&quot;./company/.com&quot;</td>
</tr>
<tr>
<td>Windows</td>
<td>\company.com</td>
</tr>
</tbody>
</table>

This value truncates the .company.com string from each name. For instructions on setting environment variables, refer to the Web Help.
Customizing Element Names

In addition to the three variables described in Table 10 on page 45, eHealth uses the NH_ALT_NAME_SCHEME environment variable to identify a tool command language (Tcl) script that can name discovered elements. You can use this variable with those three variables.

To use the NH_ALT_NAME_SCHEME variable, edit the nethealthrc.sh file to add the following line, where `filename` is the full pathname for the Tcl script:

```
export NH_ALT_NAME_SCHEME; NH_ALT_NAME_SCHEME="source filename"
```

In a Tcl script, you can use, replace, or expand any of the variables described in Table 9 on page 44 with the variables listed in Table 13. eHealth supplies and understands these variables.

Table 13. Variables for Supplementing Element Names (Page 1 of 2)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAddr</td>
<td>IP address by which the element is discovered.</td>
</tr>
<tr>
<td>community</td>
<td>Community string of the SNMP agent.</td>
</tr>
<tr>
<td>mtfFile</td>
<td>Name of the MTF for the element; if a WAN or Frame Relay element, it is the MTF for the IN element.</td>
</tr>
<tr>
<td>mtfFile2</td>
<td>The MTF for a WAN or Frame Relay OUT element.</td>
</tr>
<tr>
<td>index</td>
<td>Contents of the Index field, displayed in the Modify or Add Element dialog.</td>
</tr>
<tr>
<td>index2</td>
<td>Contents of the Index 2 field, displayed in the Modify or Add Element dialog.</td>
</tr>
<tr>
<td>index3</td>
<td>Contents of the Index 3 field, displayed in the Modify or Add Element dialog.</td>
</tr>
<tr>
<td>modDescr</td>
<td>For Cisco elements only, contents of the ifDescr field are processed. With Cisco elements, this variable is appended to the devName variable before any other variable.</td>
</tr>
<tr>
<td>suffixedName</td>
<td>The default name assigned the element using the following convention: devName-suffixMode-suffixType-suffixIndex1-suffixIndex2</td>
</tr>
<tr>
<td>vendor</td>
<td>Contents of the vendor field. Usually, the vendor description string.</td>
</tr>
<tr>
<td>sysDescr</td>
<td>Contents of the sysDescr field.</td>
</tr>
<tr>
<td>sysLoc</td>
<td>Contents of the sysLocation field.</td>
</tr>
<tr>
<td>sysOid</td>
<td>Contents of the sysObjectID field.</td>
</tr>
<tr>
<td>sysName</td>
<td>Contents of the sysName field.</td>
</tr>
<tr>
<td>modifiedSysName</td>
<td>Contents of the sysName field with all quotation marks removed and the domain name truncated at the first period.</td>
</tr>
<tr>
<td>sysContact</td>
<td>Contents of the sysContact field.</td>
</tr>
<tr>
<td>ifDescr</td>
<td>Description of the element, usually contents of the ifDescr field.</td>
</tr>
</tbody>
</table>
In addition to the eHealth-specific variables, you can also use these two global variables, which you can access with the global varName statement:

- env – The array of available environment variables.
- Session – The SNMP session handle needed to make SNMP calls.

Table 14 describes the eHealth functions you can use to customize element names.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmpGet $Session oid var</td>
<td>Sets var to the value retrieved from an SNMP get request on oid. The value for var must have the following form: {oid type value} For example: {.1.3.6.1.2.1.1.5 DisplayString myNode} returns a value for status with 0 indicating success.</td>
</tr>
<tr>
<td>octetStringToAsciiText octetString</td>
<td>Translates an octet string to ASCII text. Use this function for converting OctetStrings to DisplayStrings.</td>
</tr>
</tbody>
</table>

The following example assumes that the ifDescr field is defined and unique for each MIB2 element in the network and is being used to set the element name:

```tcl
set IF_DESCR .1.3.6.1.2.1.2.2.1.2.
catch {snmpGet $Session $IF_DESCR.$index descr} status
if {$status != 0} {
    # could not find the name
    return ""
} else {
    set ifDescrCooked [octetStringToAsciiText [lindex $descr 2]]
    regsub -all " " "$ifDescrCooked _ ifDescr
    return $ifDescr
}
```
Index

A
application
  service 36
  service, process sets 36
autoDiscover.jobId.log file 23

C
community string, specifying 20
component
discover key format 36
  resolving changes 25
  type 37
CPU number 37

D
DCI rules, using in discover 18
devices
discovering 38
disk name 37
  resolving changes 26
  verifying match 34
discover
changing resources 8
  comparing
    MIB attributes 38
    unique keys 35
Edit Before Saving 11
filtering elements 17
identifying devices in network 7
impact of resource changes 8
IP address file, using 19
  jobs, adding 24
keys
  components 36
  creating 36
  duplicates 16
  using to identify elements 35
log
  interactive discover 11
  navigating 13
  reporting discover results 35
  Suspected Duplicate Elements 15
logical elements 8
  matching hierarchy 33
mode
  Application 42
  LAN/WAN 39
  Modem Pool 43
  Probes 41
  QoS 43
  Remote Access Server 42
  Router 39
  System 41
physical elements 8
planning for
complex devices 8
frequent network changes 8
  guidelines 17
  multiple changes to a device 23
  simple devices 8
upgrading network devices 23
  verifying device match 34
  viewing results 10
Discover dialog 9
discover process
    identifying resources 7
    resolving changes 25
    starting 10
    status messages 12
discoverResults.log file 10
disk device name 37
duplicate elements
    duplicate discover keys 16
    removing 15
    same attributes 15
    same names, different attributes 16
    suspected 15

e
    element types, discovering
    enhanced router/switch 40
    LAN/WAN 39
    Remote Access Servers 42
    router LAN/WAN 40
    router/switch 39
    switch Lite switches 40
    system 41
    Traffic Accountant 41
    elements
    duplicates 15
    filtering discoveries 17
    information, changes to 8
    type of network interface 39
    unique keys 35

F
    filtering, discovered elements 17
    Frame Relay elements, discovering 39

I
    identifying resources 7
    ifDescr variable
    as part of discover key 37
    using unique 22
    IP addresses
    discover using file 19
    discovering elements 22
    specifying during discovery 25

L
    LAN/WAN elements, discovering 39
    log files, discoverResults.log 10

M
    MAC address 35
    management information base. See MIB.
    matching, physical address 38
    MIB attributes for discover 38
    modem pool
    changing default 43
    defining in agent 43
    discovering 43
    types of elements 43
    module number, Cisco Catalyst 5000 37

N
    network
    changes, understanding 7
    identifying devices 7
    obtaining IP addresses of devices 7
    resolvable changes 12
    NH_DISCOVER_APPLICATION_PORTS 21
    NH_DISCOVER_PORTS 21
    NH_DISCOVER_RESPONSE_PORTS 21
    NH_DISCOVER_SERVER_PORTS 21
    NH_DISCOVER_TIMEOUT 18
    NH_FINDER_PROC_LIMIT 23
    NH_NAME_IF_ADDR 45
    NH_NAME_SRVC_PAT 45
    NH_SNMP_PROXY_ADDRESS 18
    NH_USE_NAME_SRVC 45

P
    partitions, name 37
    ports, specifying for discover process 21
    process sets
    creating elements 30
    defining 22
    discovering 41
    licensing 41
    naming 38
    processName arguments variable 38
    processSetName variable 38
    protocol variable, definition 38
Q
QoS elements, discovering 43

R
RAS
  discovering 29, 42
  managing
    as modem pools 42
    as physical device 42
  types of elements 42
remote access server. See RAS. 42
response elements, discovering 43
router
  discovering elements 39
  reporting
    enhanced 40
    standard 39
  troubleshooting discover issues 30
rules, using in discover 18

S
scheduled discover job 24
SNMPv3, specifying proxy for 18
specifying agent ports for discover process 21
specifying proxy for SNMPv3 discovery 18
specifying timeout for SNMPv3 discovery process 18
Suspected Duplicate Elements discover log section 15
sysName variable
  used in discover key 38
  using unique values 22
system
  definition of elements 41
  partition name 37
  troubleshooting discover issues 30

T
timeout, specifying for SNMPv3 discover process 18
topology, network 7
Traffic Accountant elements 41
troubleshooting tips
  Router mode issues 30
  System mode issues 30

U
unique keys 35

V
variables
  element naming conventions 45
  NH_DISCOVER_APPLICATION_PORTS 21
  NH_DISCOVER_PORTS 21
  NH_DISCOVER_RESPONSE_PORTS 21
  NH_DISCOVER_SERVER_PORTS 21
  NH_DISCOVER_TIMEOUT 18
  NH_FINDER_PROC_LIMIT 23
  NH_NAME_IF_ADDR 45
  NH_NAME_SRVCS_PAT 45
  NH_SNMP_PROXY_ADDRESS 18
  NH_USE_NAME_SRVCS 45
  Virtual Channel Identifier (VCI) 38
  Virtual Path Identifier (VPI) 38