VMware Smart Assurance IP Manager Concepts Guide

VMware Smart Assurance 10.0.0

vmware®

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2019 VMware, Inc. All rights reserved. Copyright and trademark information.

Contents

1 Preface 8

Purpose 8

Audience 8

VMware Smart Assurance IP Manager installation directory 9

VMware Smart Assurance IP Manager products 9

IP Manager documentation 9

Related documentation 10

Conventions used in this document 10

Where to get help 11

Your comments 11

2 Overview 12

3 Global Manager overview 14

IC Domain Configuration objects 16

4 Configuration 17

User configuration parameters 18

Configuration Scanner tool 19

IP Configuration Manager 19

Conventions for specifying an IPv6 address with port number 32

Conventions for specifying an IPv6 address in a class instance name 32

IPv4-only features 33

Understanding interface-matching filters 33

Understanding groups and settings 36

Understanding ICMP and SNMP polling 38

Understanding polling groups and settings 39

Default polling groups 40

Understanding ICMP and SNMP polling and thresholds 42

Understanding threshold groups and settings 42

Understanding device access groups and settings 43

Polling settings for IP performance monitoring 43

Thresholds tab 44

Threshold settings for IP availability monitoring 54

Threshold settings for IP performance monitoring 54

Understanding the IP tagging feature 55

Understanding tagging filter groups and settings 64

Audit logging 64

ICMP and SNMP polling overview 68
Federal Information Processing Standard (FIPS) 140 70

5 Discovery 72
Discovery process overview 74
Network system support 75
Light discovery 76
ICMP, SNMP, ASNMP, and IPv6 support 79
Support for HSRP and VRRP groups 79
Support for NAS gateways 79
Support for NIC teaming 80
Support for IPSec and IKE tunnels 80
Support for overlapping IP addresses 80
Support for SNMP v3 auto discovery 80
Support for Wireless LAN 81
Support for Virtual Port Channel 83
Link Aggregation and EtherChannel discovery 85
Support for Virtual Data Context 86
DASL integration 87
Virtual bridging 89
Incremental reconfigure 90
Default IP route 90
Asynchronous SNMP (ASNMP) 91
Link Layer Discovery Protocol (LLDP) 91
Unique VLAN naming 95
OID Info Tool 96
ESX Server Certification 97
Cisco Nexus 1000V implementation 99
Cisco Unified Computing System Certification 103
AggregateLink representation in the Map console 114
Discovery and the VPN-Tagging Server 116
About seed files 117
Seed file format 118
Seed file examples 120
Which systems to put in a seed file 122
Discovery methods 123
Discovery phases 125
Autodiscovery data sources 143
Discovery of NIC teaming 144
Discovery and short discovery of IPSec tunnels 146
Discovery and the local name resolution service 150
Hostname resolution and IP address checking 151

How discovery names a system 151
Configuring automatic name resolution 153
Preparing for manual discovery 154
Initiating manual or autodiscovery 156
Initiating postprocessing 157
About discovery filters 157
Discovery filter fields 158
Discovery filter considerations 160
Discovery filter matching 161
Examples of discovery filters 162
Discovery results 164
Discovery error scenarios and results 165
Discovery errors 166
DiscoveryError notifications 169
Pending Devices list 172
About certifying managed devices 175
Discovery hook scripts 180
Automatic and manual discovery 181
Discovery methods 181
Discovery considerations 182
Scheduling automatic discovery 182
Read community strings 183
About managing and unmanaging topology objects 183
Handling duplicate IP address errors 187
Redundancy groups 188
ICMP 188
SNMP 189
About ICMP and SNMP polling parameters 192

6 Availability Manager 193
Network notifications 193
Network topology 195
Network containment 200
Root-cause problems, exceptions, and events 201
Trap analysis 201
Redundancy groups 204

7 Performance Manager and Server Performance Manager 211
About IP Performance Manager 211
Performance monitoring 212
Performance analysis 212
Performance notifications 212

Before you start 213
SNMP traps processed 213
DisplayName and Name attributes 214

8 NAS 216
Introduction to network-attached storage 216
The VMware Smart Assurance IP NAS solution 219
NAS IP discovery 221
Celerra devices for NAS Extension functionality 222
Scenarios 223
NAS notifications 226
NAS topology map 226
NAS containment 230

9 Terminology 231
System and device 232
Modeled topology 233
Object 233
Domain Manager 233
Satellite Domain Manager 234
Device connection 234
Discovery 235
Map Console 235
NASChassis Element 235
Edge (Map Console display component) 235
IP Availability Manager 236
IP Performance Manager 236
Event 236
Icon (Map Console display component) 236
IP Manager 236
Global Manager 237
Notification 237
Notification Log 237
Repository 237
Global Console 237
Impact Analysis 237
Logical link 237
Partition 238
Managed Element 238
Root-Cause Analysis 238
Topology Browser 239
Adapter 239

FileServer Element 239
Certification 239
Certification Matrix 239
Codebook 239
Correlation Model 239
Correlator 240
Broker 240
Containment 240
ICMP 240
ICMP operation 241
SNMP 241
SNMP key components 241
SNMP basic operations 241
SNMP versions 241
SNMPv1, SNMPv2c, and SNMPv3 security 242
NASChassis 242
FileServer 243
Comparable terms 243
Wildcard patterns 244

Preface

1

As part of an effort to improve its product lines, VMware periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your VMware technical support professional if a product does not function properly or does not function as described in this document.

Note This document was accurate at publication time. Go to VMware Support (docs.vmware.com)

This chapter includes the following topics:

- Purpose
- Audience
- VMware Smart Assurance IP Manager installation directory
- VMware Smart Assurance IP Manager products
- IP Manager documentation
- Related documentation
- Conventions used in this document
- Where to get help
- Your comments

Purpose

This document is part of the VMware Smart Assurance IP Manager documentation set. It provides conceptual information on IP Manager products.

Audience

This document is intended for IT managers and system administrators.

VMware Smart Assurance IP Manager installation directory

In this document, the term BASEDIR represents the location where VMware Smart Assurance software is installed.

- For UNIX, this location is: /opt/InCharge/<productsuite>.

  On UNIX operating systems, IP Manager is, by default, installed to: /opt/InCharge/IP/smarts. This location is referred to as BASEDIR/smarts.

  Optionally, you can specify the root of BASEDIR to be something different, but you cannot change the <productsuite> location under the root directory.

  The VMware Smart Assurance System Administration Guide provides more information about the directory structure of VMware Smart Assurance software.

VMware Smart Assurance IP Manager products

The VMware Smart Assurance IP Manager includes the following products:

- VMware Smart Assurance IP Availability Manager
- VMware Smart Assurance IP Performance Manager
- VMware Smart Assurance IP Availability Manager Extension for NAS

IP Manager documentation

The following documents are relevant to users of the IP Manager product. Electronic versions of the updated manuals are available at VMware Online Support: docs.vmware.com

- VMware Smart Assurance IP Manager Release Notes
- *VMware Smart Assurance Open Source License and Copyright Information*
- *VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers*
- *VMware Smart Assurance IP Manager Deployment Guide*
- *VMware Smart Assurance IP Manager Concepts Guide*
- *VMware Smart Assurance IP Manager User Guide*
- VMware Smart Assurance IP Manager Reference Guide
- *VMware Smart Assurance Topology Split Manager User Guide*
- VMware Smart Assurance IP Manager Troubleshooting Guide
- *VMware Important Notice*
- *VMware Smart Assurance IP Manager Certification Matrix*

Related documentation

The following VMware Smart Assurance publications provide additional information:

- *VMware Smart Assurance System Administration Guide*
- *VMware Smart Assurance Common Information Model (ICIM) 1.11 Reference for Non-Service Assurance Manager Products*
- *VMware Smart Assurance Common Information Model (ICIM) Reference for Service Assurance Manager*
- Smarts Foundation EMC Data Access API (EDAA) Programmer Guide
- *VMware Smart Assurance ASL Reference Guide*
- *VMware Smart Assurance Perl Reference Guide*
- *VMware Smart Assurance MODEL Reference Guide*
- *VMware Smart Assurance Dynamic Modeling Tutorial*

Conventions used in this document

VMware uses the following conventions for special notices:

Note NOTICE is used to address practices not related to personal injury.

A note presents information that is important, but not hazard-related.

An important notice contains information essential to software or hardware operation.

Typographical conventions

Vmware uses the following type style conventions in this document:

Bold	Use for names of interface elements
Italic	Use for full titles of publications referenced in text
`Monospace`	Use for: ■ System output, such as an error message or script ■ System code ■ Pathnames, filenames, prompts, and syntax ■ Commands and options
`Monospace italic`	Use for variables.

`Monospace bold`	Use for user input.
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections — the bar means “or”
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information omitted from the example

Where to get help

VMware Smart Assurance support, product, and licensing information can be obtained as follows:

Product information

For documentation, release notes, software updates, or information about Vmware Smart Assurance products, go to VMware Online Support at:

docs.vmware.com

Technical support

Go to VMware Online Support. You will see several options for contacting Technical Support. Note that to open a service request, you must have a valid support agreement. Contact your VMware sales representative for details about obtaining a valid support agreement or with questions about your account.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Send your feedback on the below linkdocs.vmware.com

Overview

2

IP networks are comprised of thousands of interconnected devices. Any outage can cause a flood of alarms on the console, leaving operators with the task of trying to distinguish root-cause problems from their symptoms. To provide the actionable information needed to sustain service, management solutions must automate the following functions:

- Discovery - Determine what elements exist in the environment.
- Modeling - Combine the results of discovery with data from other sources to map where the elements are located, how they are related, and how each element's behavior correlates with related elements.
- Root-cause and impact analysis - Provide the actionable intelligence necessary to sustain service by revealing what precisely are the problems that need to be fixed and what is the impact of these problems on the business.

  Most management solutions today are event-based and rely on rules to provide intelligence for a set of events that occurs. Rules writing means that for every possible event or combination of events that might occur in a complex distributed network, you need to create a rule that accounts for that condition. The problems with this approach are obvious—there's no built-in analysis, the solution can't scale with the constant change so characteristic of today's networks, and it's very expensive, time consuming and labor intensive, because the rules-writing never ends.

  VMware Smart Assurance takes a different approach since it relies on two key technologies:

- The VMware Smart Assurance Common Information Model
- The patented Codebook Correlation Technology

  The VMware Smart Assurance Common Information Model is based on the DMTF standard and is extended to include many classes and relationships. It is a conceptual and structural view of systems that applies object-oriented concepts to systems management. It provides a unified representation and classification of logical and physical entities that describes their state, behavior and relationships.

  The model uses generic objects, such as Switch, Host, Application or Service Offering, to represent IT infrastructure devices and their authentic problems. The model is basically a library of generic physical and logical objects that includes attributes, relationships it can participate in with other objects, authentic problems, characteristic symptoms of problems it can cause, and how the problems

propagate to related objects. These generic objects represent a wealth of knowledge about that type of object and its behavior. By associating business objects, such as services and customers, with applications at the edge of the network, VMware Smart Assurance can automatically correlate IT problems to business impacts.

Object and relationship instances discovered in the environment are instantiated with the library of generic objects to automatically become part of a real-time inventory of, or repository for, the managed environment.

The Codebook is layered over the model to monitor the environment, look for combinations of symptoms that indicate service-affecting problems, and deliver the actionable intelligence needed to prioritize corrective action on the problems that matter most. Problem signatures, which are a set of symptoms that create a unique identifier for a problem, are derived from the library and repository. These signatures are stored in the Codebook.

VMware Smart Assurance IP Manager provides powerful discovery, modeling, root-cause, and impact analysis capabilities across the IP network, including network-attached storage. The product suite is distributed on a CD/DVD-ROM and available for download from vmware online support. It is installed, by default, to the /IP subdirectory under the InCharge root directory. The VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers provides the procedure to install services manually for the underlying servers used in the IP Management Suite.

VMware Smart Assurance IP Manager represents the following VMware Smart Assurance Domain Managers:

- VMware Smart Assurance IP Availability Manager
- VMware Smart Assurance IP Performance Manager
- VMware Smart Assurance IP Server Performance Manager
- VMware Smart Assurance IP Availability Manager Extension for NAS
- VMware Smart Assurance IP Configuration Manager

A Domain Manager is a service assurance application that is associated with a particular type of information technology domain, such as networks, systems, applications, or application services. For IP Manager, the domain is IP network transport. Each Domain Manager is autonomous in the sense that it:

- Maintains its own data models, repository, and problem signatures.
- Monitors and analyzes the discovered objects in its own domain.

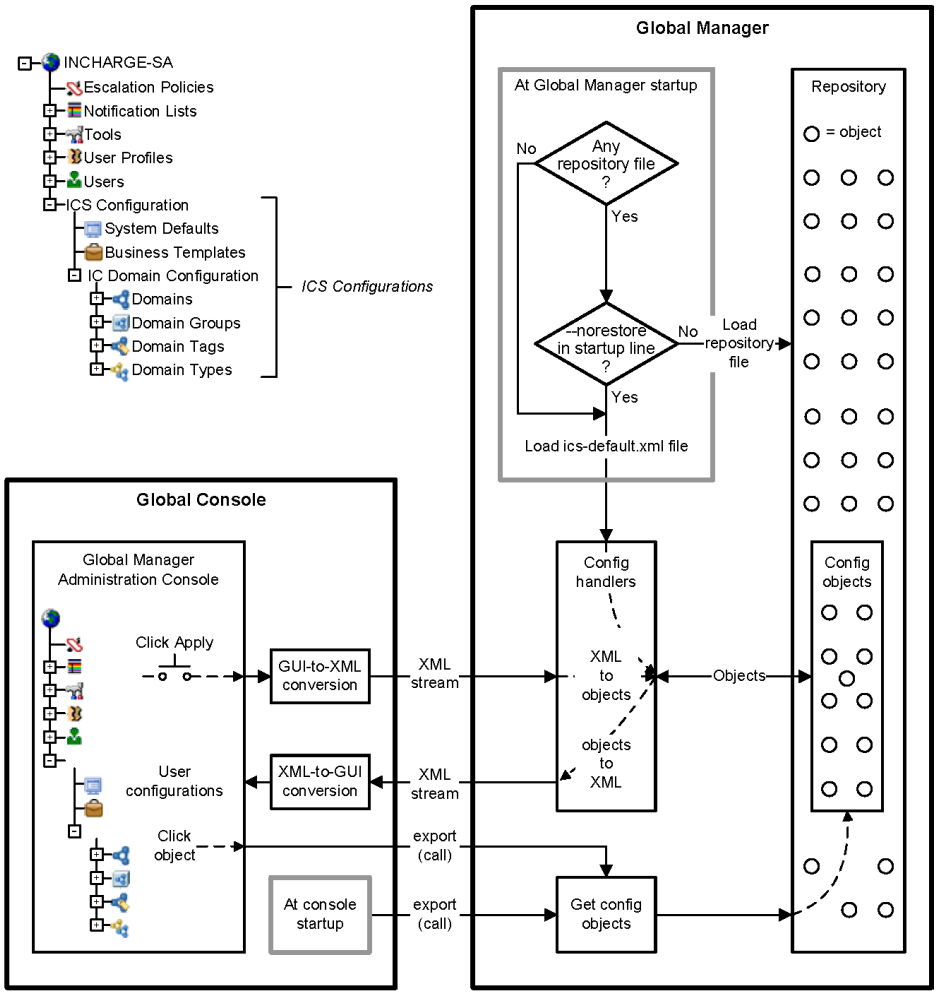
The IP Availability Manager and IP Performance Manager are examples of VMware Smart Assurance Domain Managers. You can choose to start IP Availability Manager (AM) only, IP Performance Manager (PM) only or a combination of both IP Availability Manager and IP Performance Manager (AM-PM). The configurations and licenses that are applied to an IP Manager at startup determine which discovery and analysis will run.

Global Manager overview

3

The Global Manager is part of the VMware Smart Assurance Service Assurance Manager. Most of the configuration tasks for setting up the Global Manager in an IP Manager deployment are accomplished by using the Global Manager Administration Console view of the Global Console. Global Manager Administration Console interaction with the Global Manager summarizes the Global Manager Administration Console interaction with the Global Manager.

Figure 3-1. Global Manager Administration Console interaction with the Global Manager



IC Domain Configuration objects

Global Manager Administration Console presents an example of a Global Manager Administration Console that is attached to a Global Manager named INCHARGE-SA.

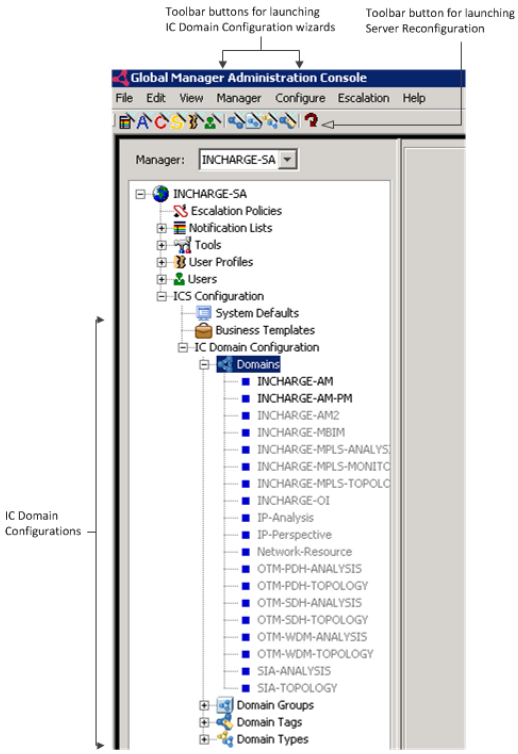


Figure 3-2. Global Manager Administration Console

Listed at the bottom of the configuration tree of the Global Manager Administration Console are the four categories of IC Domain Configuration objects: Domains, Domain Groups, Domain Tags, and Domain Types. Listed under each category are instances of these objects; for example, the default domain “INCHARGE-AM” is listed under Domains. Selecting an object in the left panel displays its attributes and their values in the right panel.

The names of enabled domains appear as black text in the Global Manager Administration Console, and the names of disabled domains appear as gray text. For example, INCHARGE-AM-PM appears in black text because it is one of the enabled domains.

Configuration

4

This chapter includes the following topics:

- User configuration parameters
- Configuration Scanner tool
- IP Configuration Manager
- Conventions for specifying an IPv6 address with port number
- Conventions for specifying an IPv6 address in a class instance name
- IPv4-only features
- Understanding interface-matching filters
- Understanding groups and settings
- Understanding ICMP and SNMP polling
- Understanding polling groups and settings
- Default polling groups
- Understanding ICMP and SNMP polling and thresholds
- Understanding threshold groups and settings
- Understanding device access groups and settings
- Polling settings for IP performance monitoring
- Thresholds tab
- Threshold settings for IP availability monitoring
- Threshold settings for IP performance monitoring
- Understanding the IP tagging feature
- Understanding tagging filter groups and settings
- Audit logging
- ICMP and SNMP polling overview
- Federal Information Processing Standard (FIPS) 140

User configuration parameters

As with most VMware Smart Assurance configuration files, the user configuration files for the IP Manager are used to set attributes for certain objects within a server's environment when the server starts up. The parameters and their values in a configuration file become the attributes and their values for the created objects.

For the discovery.conf file, as an example, the associated object is an instance of the ICF_TopologyManager class and is named ICF-TopologyManager.

The VMware Smart Assurance IP Manager User Guide provides information on how to modify user configuration parameters.

Major part of configuring the IP Manager involves the setting of user configuration parameters in the files that are identified in User configuration files for the IP Manager.

Table 4-1. User configuration files for the IP Manager

Directory under BASEDIR	Filename	Description
smarts/conf/discovery	discovery.conf	File in which you fine tune configurations to improve the performance of IP Manager discovery. VMware Smart Assurance IP Manager Reference Guide provides details.
	tpmgr-param.conf	File in which you customize IP Manager discovery. VMware Smart Assurance IP Manager Reference Guide provides details.
	name-resolver.conf	File in which you control how the IP Manager names a discovered system. VMware Smart Assurance IP Manager Reference Guide provides details.
	partition.conf	File in which you specify display names for partitions. VMware Smart Assurance IP Manager Reference Guide provides details.
	user-defined-connections.conf	File in which you specify connections between discovered network objects. VMware Smart Assurance IP Manager Reference Guide provides details.
smarts/conf/trapd	trapd.conf	File in which you configure the port number of the IP built-in trap receiver. VMware Smart Assurance IP Manager Reference Guide provides details.
smarts/conf/notifier	trap-notify.conf	File in which you configure the SNMP Trap Notifier Adapter. VMware Smart Assurance IP Manager Reference Guide provides details.
smarts/local/conf	runcmd_env.sh	File in which you set environment variables to control behavior globally. VMware Smart Assurance IP Manager Reference Guide provides details.

About commands to modify configuration parameters

You can issue any of the following commands from the BASEDIR/smarts/bin directory to change configuration parameters for the IP Manager:

- dmctl put
- dmctl invoke
- dmctl get
- sm_tpmgr

  The **dmctl** utility is described in the HTML pages that are located in the BASEDIR/smarts/doc/html/usage directory of any Domain Manager installation area. The VMware Smart Assurance IP Manager Reference Guide provides additional syntax information and also the parameters to enable and control the ConfigChange event.

Configuration Scanner tool

The Configuration Scanner tool scans for configuration changes in your current installation. It scans for customizations with regard to the following:

- **Polling and threshold settings:** The tool presents the non-default values and settings, that is, the tool generates a list of polling settings and threshold parameters that have changed from their default values, along with details of all the groups they are associated with. The output also includes tagging settings. In case of CLI settings, the tool collects and displays the username and the associated matching criteria.
- **Configuration files:** The tool generates a list of files that have changed from the default installation. The tool scans the files in the SM_SITEMOD and base installation and does a two-way difference analysis to figure out what files have changed, and displays the list of files, flagged appropriately as modified or added. For discovery.conf, name-resolver.conf, and tpmgr-param.conf files, the tool parses through the content and presents the difference at an attribute->value pair level. The clientConnect.conf, serverConnect.conf, brokerConnect.conf, .imk.dat files, and the l10n classes and Perl directories are excluded from the scan.

  Note The tool ignores service pack and hotfix files that are not modified by you.

  The VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers and the VMware Smart Assurance IP Manager User Guide provide more information on this tool.

IP Configuration Manager

The IP-Configuration Manager is an optional component and provides a method for applying a consistent set of polling and threshold settings, IP tagging settings, and CLI device-access credentials to all IP domain managers.

When the IP-Configuration Manager is not used, configure each IP domain manager with the polling and threshold settings applicable to each.

The default deployment of the IP Manager has only one IP-Configuration Manager to which any number of IP domain managers can be assigned. This works best when you deploy multiple IP domain managers with similar groups and settings.

Use the IP-Configuration Manager feature to maintain a centralized copy of the settings by attaching the Polling and Thresholds Console to the IP-Configuration Manager and pushing settings directly to those IP Managers which are "configuration-enabled". The topic, Assigning an IP domain manager to the IP-Configuration Manager explains how to designate an IP Manager as configuration-enabled.

The “IP Configuration Manager” section in the VMware Smart Assurance IP Manager User Guide provide more information on the IP Configuration Manager.

Assigning an IP domain manager to the IP-Configuration Manager

An IP domain manager is “configuration-enabled” or under the control of the IP-Configuration Manager when these conditions are met:

- The IP Manager is a target of the IP-Configuration Manager (that is, an IC_Domain instance is created for the IP domain manager in IP-Configuration Manager)
- The attribute “IsConfEnabled” of the related IC_Domain instance in IP-Configuration Manager is set to True. For example,

  ```
  IC_Domain::<IP_DM_Name>::IsConfEnabled=True
  ```

- The IP domain manager attribute ICIM_Manager::ICIM-Manager::ServiceName is not empty or is set equal to IP-Configuration Manager's name attribute.

Load settings into the IP-Configuration Manager

You can import settings from a running IP domain manager into the IP-Configuration Manager. The sm_settings.pl script can be used to export the settings from the running IP domain manager and imported into the IP-Configuration Manager. The VMware Smart Assurance IP Manager User Guide provides information on how to export the settings from a running IP domain manager and also on how to import the settings into the IP-Configuration Manager.

Considerations when attaching to the IP-Configuration Manager

The Polling and Thresholds Console may be attached either to individual IP domain managers or to the IP-Configuration Manager. The procedures for updating groups and settings are the same for either manager; however, there are some special considerations when creating a master set of policies to be managed by the IP-Configuration Manager.

By default, both new and changed settings are pushed automatically to all configuration-enabled IP domain managers that are connected to the IP-Configuration Manager. In addition, device access settings (CLI credentials) are also pushed to the IP domain managers.

If some settings apply to some IP domain managers but not others, you need to plan your configuration carefully before connecting IP domain managers to the IP-Configuration Manager; otherwise, you may overwrite unique policy settings with the master set of policies in the IP-Configuration Manager.

Settings are pushed from the IP-Configuration Manager whenever the IP Managers are connected or re-connected, or when the Push Settings action is invoked. This means that no settings should be defined locally to any IP Manager that is configured by using an IP-Configuration Manager.

Note If you want to customize settings before applying them, you should start IP-Configuration Manager first, customize settings, and then start target IP Managers. Control transfer of settings to IP domain managers section provides more information.

The default groups and settings in the Polling and Thresholds Console attached to the IP-Configuration Manager are applied to all configuration-enabled IP Managers. When needed, however, custom groups and settings may be created by copying an existing group, renaming it, and editing the settings.

The Copy Configuration function (available when you right-click one of the default groups in the left panel of the Polling and Thresholds Console) automatically moves the copied setting group immediately above its source setting group. The settings in the copied setting group will have a higher priority than those in the source setting group and any attempt to move a copied setting group below its source setting group will be rejected. This ensures that custom settings will not be overwritten with settings from the lower-level source group.

The group's settings are assigned to one or more IP Managers in the **Helper** box appearing at the top of the right panel in each Polling and Thresholds screen. For default settings groups, an asterisk appears in this field indicating that the groups and settings will be applied to all configuration-enabled component servers. For newly created groups, you must enter a value in the **Helper** box. If no value is entered, the settings will not be pushed to any target domain managers.

You may specify matching criteria by using a glob pattern, and wildcards to list a unique set of IP domain managers that will use the custom groups and settings. Custom settings are pushed only to those domains that are specified in the matching criteria.

A master set of device access credentials, also referred to as command line interface (CLI) credentials, and IP tagging settings can be configured from the Polling and Thresholds Console as well. After modification, click the **Apply** button in the right panel to update the settings in the IP-Configuration Manager and then click **Push Settings** to implement the new settings in all configuration-enabled servers.

Note New or changed configuration settings will be pushed automatically to all configuration-enabled IP domain managers only upon connect or reconnect. Once connected, you need to manually push settings by clicking the **Push Settings** button in the Polling and Thresholds Console. Disable the automatic update process until you are ready to push the master settings to the IP Managers.

Push settings from the IP-Configuration Manager to IP domain managers

When the Polling and Thresholds Console is attached to the IP-Configuration Manager, all changes are pushed to all IP Managers under control of that IP-Configuration Manager.

The following conditions affect when settings are pushed to the IP domain managers:

- Settings are automatically pushed to the IP Managers by connecting or reconnecting either the IP-Configuration Manager or IP Manager server, or by clicking the **Push Settings** button in the Polling and Thresholds Console. The Push Settings option is also available from the **Group** menu in this console.
- The **Push Settings** button is also available in the Domain Administration Console and from the **Topology** menu in this console. Only changed and new settings will be pushed and applied.
- When the attribute **IsConfEnabled** is set to **False**, the corresponding IP Manager is excluded and no settings will be pushed to it. Alternatively, remove the IC_Domain instance of the IP Manager to prevent settings from being pushed.
- If the attribute **RemoveExtraFromTarget** is set to **True**, then settings that exist only in the target IP Manager (that is, settings do not exist in the IP-Configuration Manager) will be removed.

Control transfer of settings to IP domain managers

There are several ways to control the transfer of settings to the configuration-enabled IP domain managers. Choose one of the following methods to disable the automatic transfer of settings to the IP Managers:

- Do not start IP Manager servers before settings are ready to be pushed. Complete the configuration of settings in the Polling and Threshold Console attached to the IP-Configuration Manager before starting the IP Manager servers.
- If the IP-Configuration Manager is running, but the IP domain managers are not, then access the Polling and Thresholds Console attached to the IP-Configuration Manager. Add the target IP Manager names to the IP-Configuration Manager, then edit the master groups and settings in the Polling and Thresholds Console prior to starting the IP Manager servers. If any custom groups or settings are required, set those up before starting the IP Managers. The “Add target IP domain managers through the Global Console” section in the VMware Smart Assurance IP Manager User Guide provides more information.
- Attach the Polling and Thresholds Console and edit the master groups and settings. When ready to have the IP-Configuration Manager push the settings, change the IsConfEnabled attribute back to TRUE for each target manager you want configuration-enabled. Edit the attribute value from the domain manager Administration Console. The “Add target IP domain managers through the Global Console” section in the VMware Smart Assurance IP Manager User Guide provides more information.

- If both IP-Configuration Manager and the target IP Managers are running, then use the Polling and Thresholds Console attached to the IP-Configuration Manager to set the attribute, IsConfEnabled to FALSE for each target IP Manager. This temporarily disables automatic download of the settings. When ready to have the IP-Configuration Manager push the settings, change the IsConfEnabled attribute back to TRUE for each target IP Manager. To initiate transfer of settings, click the **Push Settings** icon.

Edit the IP Manager's attribute, ICIM_Manager::ICIM-Manager::ServiceName, and set it to a value other than the IP-Configuration Manager's name. This temporarily disables automatic download of the settings. When ready to have the IP-Configuration Manager push the settings, change the Service Name attribute back to an empty value or to the IP-Configuration Manager's name. To initiate transfer of settings, click **Push Settings** .Discovery settings from the IP-Configuration Manager to IP domain managers

Topology parameters and discovery filters can be configured in IP-Configuration Manager and applied to IP domain managers using the **ApplyDiscoverySettings** attribute. This feature is not enabled by default in IP-Configuration Manager.

- The **ApplyDiscoverySettings** attribute of each IP domain manager under **IC-Domain**,controls the transfer of discovery filters and topology parameters. This attribute must be set to TRUE for discovery settings to be applied to a target IP domain manager.

Policy Management

In an IP deployment, one or more IP domain managers discover topology objects and monitor their status by polling them at regular intervals. The frequency of polling is determined by the polling and threshold settings that are configured for each IP domain manager. The IP domain manager monitors managed objects but not unmanaged objects. An unmanaged object is not polled or rediscovered by the IP domain manager.

The IsManaged attribute value of an object determines whether an object is managed or unmanaged. A managed object is an object whose IsManaged attribute is TRUE. An unmanaged object is an object whose IsManaged attribute is FALSE. In most cases, an object's management state is more accurately determined after postprocessing.

By default,

- All interfaces are managed.
- All ports that have connections are managed.

Types of objects that can be managed or unmanaged

Management state can be set for the following types of objects:

- Systems, such as routers, switches, and hosts
- Physical packages, such as cards
- Network adapters, such as ports and interfaces
- Protocol endpoints, such as IP interfaces and STP nodes
- Management agents, such as SNMP agents
- Logical devices, such as processors, temperature and voltage sensors, and filesystems
- Redundancy groups, such as redundancy groups that represent two or more cards or two or more network connections

  Except for systems (devices) and redundancy groups, the objects in the preceding bullet list are containment objects, which are internal components of systems.

  The ability to unmanage IP interfaces enables you to develop flexible management policies. When you unmanage an IP interface, the underlying physical interface continues to be managed because the status of the IP interface is determined by using ICMP, and the status of the physical interface is determined by using SNMP.

  You can unmanage the IP interfaces on systems where ICMP pings are not allowed, for example, of a firewall. When you unmanage these IP interfaces, the IP domain manager no longer polls them through ICMP. More important, however, is that the IP domain manager still manages the underlying physical interfaces through SNMP.

Enabling policy management

The Policy Management feature enables an administrator to centrally manage/unmanage objects from the IP Configuration Manager.

The policy management feature is controlled by the flag, IsUserCustomizationEnabled of the CM_Configuration class. This is defined in the icf-c/bootend.conf file. This flag has to be changed to TRUE before starting the IP-Configuration Manager server to enable the Custom Policy Groups.

```
CM_Configuration::CM_Configuration
{
    IsUserCustomizationEnabled = TRUE
}
```

If you start an IP Configuration Manager server without an RPS, modify icf-c/bootend.conf and change IsUserCustomizationEnabled =True

./sm_edit ../conf/icf-c/bootend.conf

If you start an IP Configuration Manager server with an RPS, perform the following two steps:

Step 1:

./sm_edit ../conf/icf-c/bootend.conf

and change IsUserCustomizationEnabled =True

Step 2:

./dmctl -s IP-CM invoke ICF_PolicyManager::ICF-PolicyManager loadSettings <BASE-DIR>/conf/icf-c/settings_custom.conf, where “IP-CM” is the name of the IP Configuration Manager server.

If you want to enable Policy Management in the running IP Configuration Manager server, execute the following three steps:

Step 1:

./dmctl -s IP-CM put CM_Configuration::CM-Configuration::IsUserCustomizationEnabled TRUE, where “IP-CM” is the name of the IP Configuration Manager server.

Step 2:

./sm_edit ../conf/icf-c/bootend.conf

and change IsUserCustomizationEnabled =True

Step 3:

./dmctl -s IP-CM invoke ICF_PolicyManager::ICF-PolicyManager loadSettings <BASE-DIR>/conf/icf-c/settings_custom.conf, where “IP-CM” is the name of the IP Configuration Manager server.

The following additional groups will be displayed in the Polling and Threshold Console once the Custom Policy groups have been loaded.

- Managed FileSystems
- Managed Interfaces
- Managed Logical Devices
- Managed Port
- Managed RedundancyGroups
- Managed STPNodes
- Unmanaged FileSystems
- Unmanaged Interfaces
- Unmanaged Logical Devices
- Unmanaged Port
- Unmanaged RedundancyGroups
- Unmanaged STPNodes

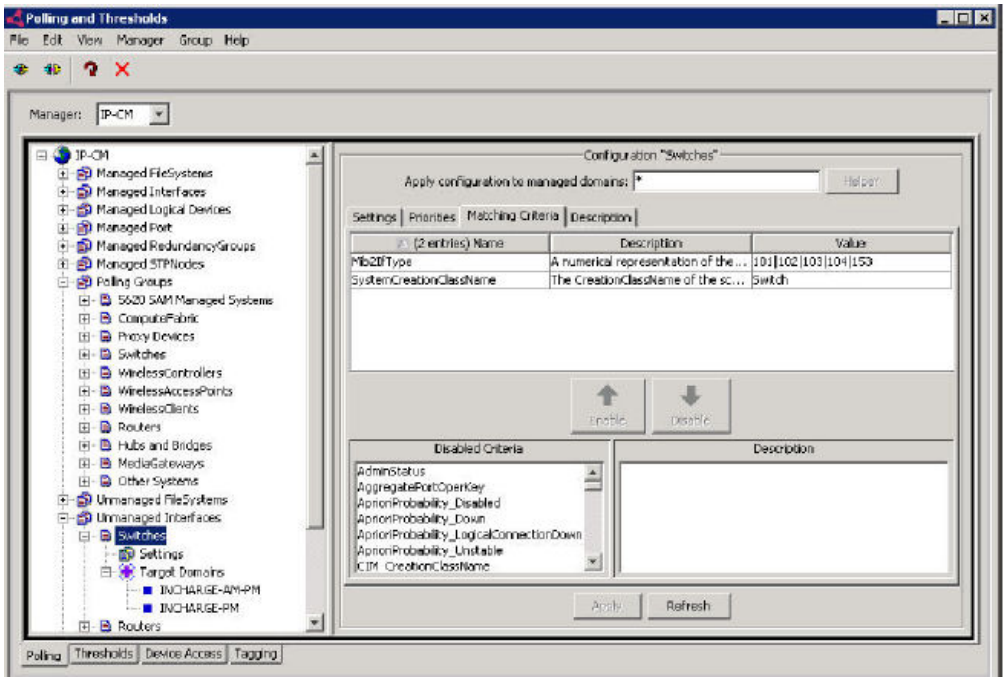
Viewing policy groups in the Polling and Threshold Console

Once the policy management feature is enabled, the following managed and unmanaged policy groups are added to the Polling tab of the Polling and Thresholds Console that is attached to the IP-Configuration Manager:

- Managed FileSystems
- Managed Interfaces
- Managed Logical Devices
- Managed Port
- Managed RedundancyGroups
- Managed STPNodes
- Unmanaged FileSystems
- Unmanaged Interfaces
- Unmanaged Logical Devices
- Unmanaged Port
- Unmanaged RedundancyGroups
- Unmanaged STPNodes

Policy groupsshows the policy groups in the Polling tab of the Polling and Threshold Console. Polling Groups, which contain all of the polling groups, always appear in the Polling tab.

Figure 4-1. Policy groups



In Policy groups, the **Matching Criteria**tab for the Unmanaged Interfaces Switches policy group is selected. Each policy group uses the attributes of the object type in its Matching Criteria tab to determine which objects become members of the group.

Objects that become members of a managed policy group will be managed, and therefore will be monitored. Objects that become members of an unmanaged policy group will be unmanaged, and therefore will not be monitored.

Default policy groups for the IP Managershows the default policy groups, their target classes, and their matching criteria.

Table 4-2. Default policy groups for the IP Manager

Policy group	Target class and matching criteria
1 Unmatchable value: To use DisplayName, change value to a matchable value (for example, DisplayName = NY*). 2 Matchable (real) value: Accept or change value to a different matchable value.	
Managed FileSystems and Unmanaged FileSystems	
FileSystems	CIM_FileSystem with attributes: ■ CreationClassName = FileSystem ■ DisplayName = Manage or Unmanage 1
Managed Interfaces	
Switches	Interface with attributes: ■ SystemCreationClassName = Switch ■ Mib2IfType = 10000 2
Routers	Interface with attributes: ■ SystemCreationClassName = Router ■ Mib2IfType = 10000 2
Hubs and Bridges	Interface with attributes: ■ SystemCreationClassName = Hub|Bridge ■ Mib2IfType = 10000 2
Other Systems	Interface with attributes: ■ SystemCreationClassName = ~Switch|Router|Hub|Bridge ■ Mib2IfType = 10000 2
Unmanaged Interfaces	
Switches	Interface with attributes: ■ SystemCreationClassName = Switch ■ Mib2IfType = 101|102|103|104|153 2
Routers	Interface with attributes: ■ SystemCreationClassName = Router ■ Description = Virtual-Access*|Description Async* 2 ■ Mib2IfType = 100 2
Hubs and Bridges	Interface with attributes: ■ SystemCreationClassName = Hub|Bridge ■ Description = Virtual-Access*|Description Async* 2 ■ Mib2IfType = 100 2

Table 4-2. Default policy groups for the IP Manager (Continued)

Policy group	Target class and matching criteria
Other Systems	Interface with attributes: ■ SystemCreationClassName = ~Switch|Router|Hub|Bridge ■ Description = Virtual-Access*|Description Async* 2 ■ Mib2IfType = 100 2
Managed Logical Devices and Unmanaged Logical Devices	
Switches	CIM_LogicalDevice with attributes: ■ SystemCreationClassName = Switch ■ CreationClassName = Memory|Processor|TemperatureSensor|VoltageSensor|Fan|Disk ■ DisplayName = Manage or Unmanage 1
Routers	CIM_LogicalDevice with attributes: ■ SystemCreationClassName = Router ■ CreationClassName = Memory|Processor|TemperatureSensor|VoltageSensor|Fan|Disk ■ DisplayName = Manage or Unmanage 1
Hubs and Bridges	CIM_LogicalDevice with attributes: ■ SystemCreationClassName = Hub|Bridge ■ CreationClassName = Memory|Processor|TemperatureSensor|VoltageSensor|Fan|Disk ■ DisplayName = Manage or Unmanage 1
Other Systems	CIM_LogicalDevice with attributes: ■ SystemCreationClassName = ~Switch|Router|Hub|Bridge ■ CreationClassName = Memory|Processor|TemperatureSensor|VoltageSensor|Fan|Disk ■ DisplayName = Manage or Unmanage 1
Managed Ports	
Switches	Port with attributes: ■ SystemCreationClassName = Switch ■ Mib2IfType = 101|102|103|104|153 2
Hubs and Bridges	Port with attributes: ■ SystemCreationClassName = Hub|Bridge ■ Description = Virtual-Access*|Description Async* 2 ■ Mib2IfType = 100 2
Other Systems	Port with attributes: ■ SystemCreationClassName = ~Switch|Hub|Bridge ■ Description = Virtual-Access*|Description Async* 2 ■ Mib2IfType = 100 2
Unmanaged Ports	
Switches	Port with attributes: ■ SystemCreationClassName = Switch ■ Mib2IfType = 10000 2
Hubs and Bridges	Port with attributes: ■ SystemCreationClassName = Hub|Bridge ■ Mib2IfType = 10000 2

Table 4-2. Default policy groups for the IP Manager (Continued)

Policy group	Target class and matching criteria
Other Systems	Port with attributes: ■ SystemCreationClassName = ~Switch|Hub|Bridge ■ Mib2IfType = 10000 2
Managed RedundancyGroups and Unmanaged RedundancyGroups	
RedundancyGroups	RedundancyGroup with attributes: ■ CreationClassName = CardRedundancyGroup|NetworkAdapterRedundancyGroup|HSRPGroup|VRRPGroup ■ DisplayName = Manage or Unmanage 1
Managed STPNodes and Unmanaged STPNodes	
Switches	ICIM_ServiceAccessPoint with attributes: ■ SystemCreationClassName = Switch ■ CreationClassName = STPNode ■ DisplayName = Manage or Unmanage 1

Managed and Unmanaged IP domain managers

An IP domain manager can either be managed by an IP-Configuration Manager or by itself. An IP domain manager is said to be managed if its ICIM_Manager::ICIM-Manager::ServiceName is not empty. The non-empty value is interpreted as the name of the IP-Configuration Manager that manages it, but the referred IP-Configuration Manager may or may not actually exist. An IP domain manager is said to be unmanaged if its ICIM_Manager::ICIM-Manager::ServiceName is empty. A managed IP domain manager puts itself under the control of an IP-Configuration Manager. The IP Configuration Manager is responsible for configuring and applying settings to the managed IP domain managers. To prevent the managed IP domain manager from overwriting the settings configured by the IP-Configuration Manager, the setting editing capability of the managed IP domain manager is disabled when it is attached to the console. The console does not dynamically track whether an IP domain manger is managed by an IP Configuration Manager. If an IP domain manager changes from managed to unmanaged, or vice-versa, it should be detached from the console and then reattached to the console to get its setting editing capability updated.

An unmanaged IP domain manager is not controlled by any IP-Configuration Manager and is responsible for managing its own settings.

An IP-Configuration Manager manages an IP domain manager if the following three conditions are met:

- The IP domain manager is a target of the IP Configuration manager, that is, there is an IC_Domain instance created for the IP domain manager in the IP Configuration Manager.
- The attribute IC_Doamin::<target_server>::IsConfEnabled is set to true.
- The IP domain manager's attribute ICIM_Manager::ICIM-Manager::ServiceName is set to the name of the IP Configuration Manager or, when the IP Configuration Manager's attribute CM_Configuration::CM-Configuration:: ConfUnmanagedTarget is set to True, is empty.

Configuration Options

Table 4-3. List of configuration options

Class	Option	Function
IC_Domain	IsConfEnabled ■ boolean ■ defaults to TRUE	Specify whether the target domain manager is configured by this Configuration Manager or not.
	RemoveExtraFromTarget ■ boolean ■ defaults to TRUE	Specify whether to remove settings from the target domain manager if they do not exist in this Configuration Manager.
CM_Configuration	EnforceConfigPriority ■ boolean ■ defaults to TRUE	Specify whether to perform sanity check and enforce that any copied setting group must have higher priority than the corresponding source setting group.
	ConfUnmanagedTarget ■ boolean ■ defaults to TRUE	Specify whether to configure target domain managers that have not yet been configured by other Configuration Managers. The attribute ICIM_Manager::ICIM-Manager::ServiceName of the target domain manager should either have a value equal to the name of the Configuration Manager currently configuring it, or be empty when it is not configured by any Configuration Manager.

List of configuration options provides a list of Configuration options.

Secure connections

The security configuration files, clientConnect.conf and serverConnect.conf, which are located in the BASEDIR/smarts/conf directory of each VMware Smart Assurance software installation, enable you to set up secure connections between the component applications in your deployment. By default, the configuration option settings in the clientConnect.conf file and the serverConnect.conf files set minimally secure connections between the components.

An IP Manager deployment that employs secure communications has the following environment variables set to the same value in each component application's BASEDIR/smarts/local/conf/runcmd_env.sh file:

```
SM_INCOMING_PROTOCOL=[1,0 or 2,0 or 3,0]
SM_OUTGOING_PROTOCOL=[1,0 or 2,0 or 3,0]
```

For example:

```
SM_INCOMING_PROTOCOL=2,0
SM_OUTGOING_PROTOCOL=2,0
```

The VMware Smart Assurance System Administration Guide presents detailed information about VMware Smart Assurance secure communications.

IP versioning

The Broker can support IPv4 and IPv6 protocols. If the component applications in your deployment are identified by IPv4 and IPv6 addresses, the Broker must reside on a dual-stack host so that it has both an IPv4 address and an IPv6 address.

If all component applications in your deployment use the same IP protocol version, the Broker must reside on a host with an IP address that accepts that protocol version. For example, if all component applications in your deployment support the IPv6 protocol, the Broker must reside on a host with an IPv6 address. That host may be dual-stack or IPv6-only.

Broker and server-to-server communications are also controlled by the SM_IP_VERSIONS environment variable. If all component applications in your deployment are communicating with the same IP protocol version, the SM_IP_VERSIONS variable should be set to the same value in each component application's BASEDIR/smarts/local/conf/runcmd_env.sh file. In an IPv6-only environment, for example, the SM_IP_VERSIONS environment variable should be set to v6 or v6v4, as shown in IPv6-only environment example and variable settings.

Table 4-4. IPv6-only environment example and variable settings

SM_IP_VERSIONS setting	Product suite	Components
SM_IP_VERSIONS=v6-or-SM_IP_VERSIONS=v6v4	IP Manager version 9.1	■ IP Availability Manager ■ IP Performance Manager ■ IP Server Performance Manager ■ IP Availability Manager Extension for NAS VMware Smart Assurance IP Manager User Guide provides information for setting the SM_IP_VERSIONS environment variable for IP Manager components.
SM_IP_VERSIONS=v6-or-SM_IP_VERSIONS=v6v4	Service Assurance Manager version 9.1	■ Global Manager ■ Global Console ■ Adapter Platform ■ SNMP Trap Adapter The VMware Smart Assurance Service Assurance Manager Configuration Guide provides information for setting the SM_IP_VERSIONS environment variable for Service Assurance components.
SM_IP_VERSIONS=v6-or-SM_IP_VERSIONS=v6v4	Either suite listed above	Broker

Conventions for specifying an IPv6 address with port number

Sometimes when you use an VMware Smart Assurance command line utility, such as dmctl, sm_tpmgr, or sm_snmpwalk, you want to specify a Broker other than the default Broker, which is identified in the BASEDIR/smarts/local/conf/runcmd_env.sh file of the installation directory from which the command line utility is invoked; for example, localhost:426. To do so, you include the -b <*Broker*> option in the command line invocation.

The <*Broker*> syntax is hostname or IP address, followed by a colon (:) delimiter, followed by a port number. The combination of an IP (v4 or v6) address and port number is also called a socket.

“Conventions for specifying an IPv6 address with port number” section in Chapter 8, IPv6 Address Conventions, in the VMware Smart Assurance IP Manager Reference Guide provides information on the syntax.

Conventions for specifying an IPv6 address in a class instance name

For the **dmctl** utility, if your class instance name contains either square brackets or double colons, you need to specify one of the following conventions:

- In dmctl command line mode, if the instance uses square brackets or double colons in its name, enclose the instance with single and double quotation marks. Here are two examples:

  ```
  MyClass::”’[2001:db8:0:0:0:0:0102:0304]’”
  MyClass::”’::instance’”
  ```

- In dmctl interactive mode, if the instance uses square brackets or double colons in its name, enclose the instance with either single or double quotation marks. Here are four examples:

  ```
  MyClass::’[2001:db8:0:0:0:0:0102:0304]’
  MyClass::”[2001:db8:0:0:0:0:0102:0304]”
  MyClass::’::instance’
  MyClass::”::instance”
  ```

  “Conventions for specifying an IPv6 address in a class instance name” section in Chapter 8, IPv6 Address Conventions, in the VMware Smart Assurance IP Manager Reference Guide provides information on the valid values for the SM_IP_VERSIONS environment variable.

IPv4-only features

The IP Manager uses Internet Control Message Protocol (ICMP) and Simple Network Management Protocol (SNMP) to discover and poll IPv4-only, IPv6-only, and dual-stack systems in the managed network environment. The IP Manager supports IPv4 ICMP (also known as ICMPv4), IPv6 ICMP (also known as ICMPv6), IPv4 SNMP, and IPv6 SNMP.

Note Unless specified otherwise, the term ICMP is used to refer to IPv4 and IPv6 ICMP polling, and the term SNMP is used to refer to IPv4 and IPv6 SNMP polling.

Because many Management Information Bases (MIBs) have not yet been modified to accommodate IPv6 addresses, the following IP Manager features are supported for IPv4 only:

- Autodiscovery
- Light discovery for satellite Domain Managers
- Creation of CLI device access objects for satellite Domain Managers
- IP tagging and IP tag filters
- IPSec tunnel discovery
- Virtual router discovery
- Hot Standby Router Protocol (HSRP) and Virtual Router Redundancy Protocol (VRRP) discovery, monitoring, and analysis.

Understanding interface-matching filters

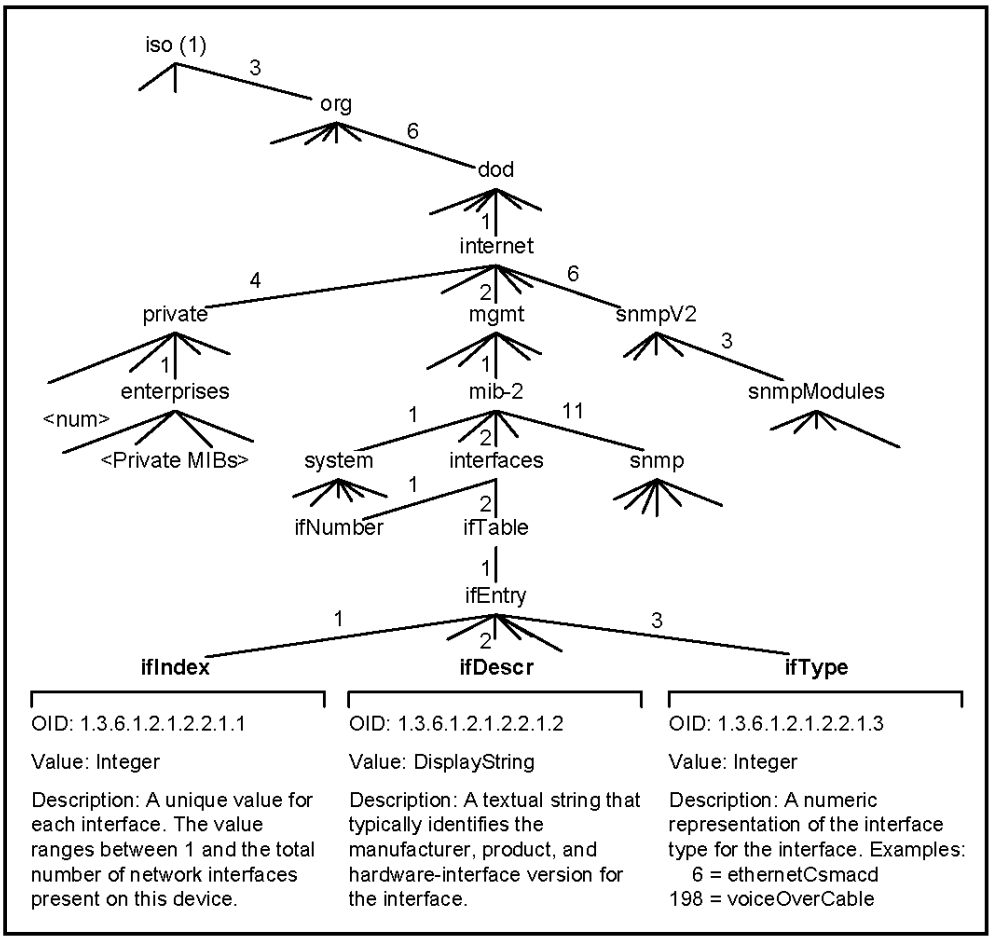
There are several reasons for wanting to exclude device-access interfaces from the discovered topology, foremost of which is to limit the interfaces that are discovered for access devices. An access device is a network device that contains interfaces (also known as access ports) that provide connections for end users or node devices, such as routers or servers. Devices that are certified as access devices are identified in the *VMware Smart Assurance IP Certification Matrix*.

To exclude interfaces for a discovered device, create an interface-matching filters for the device and adds the filters to the tpmgr-param.conf file. The filters are based on the following interface table (ifTable) objects of the device's MIB:

- ifIndex
- ifDescr
- ifType

  The object identifier (OID) paths to these objects in the Internet-standard MIB-II are shown in OID-tree view of the MIB-II ifTable's ifIndex, ifDescr, and IfType objects.

Figure 4-2. OID-tree view of the MIB-II ifTable's ifIndex, ifDescr, and IfType objects



For a device that is using a private MIB, the OID paths to the ifIndex, ifDescr, and ifType objects all begin with the prefix 1.3.6.1.4.1. As shown in OID-tree view of the MIB-II ifTable's ifIndex, ifDescr, and IfType objects, a prefix of 1.3.6.1.4.1.<*number*> points to the objects in an enterprise's private MIBs.

The following display is an example of a set of interface-matching filters that are created for a Redback SMS 1800 Router device.

Key
(interface pattern types for this device)

Value
(Interface-matching patterns)

```
IFIndexPattern-.1.3.6.1.4.1.2352.1.3 ~*

IFDescrPattern-.1.3.6.1.4.1.2352.1.3 ~LOGICAL INTERFACE~PPP LAYER~0

IFTypePattern-.1.3.6.1.4.1.2352.1.3  ~23~53~0
```

The Key part of an interface-matching filter specifies the interface pattern type (IFIndex, IFDescr, or IFType) for the device, where the device is identified by its system object identifier (sysObjectID). (The sysObjectID for a Redback SMS 1800 Router is 1.3.6.1.4.1.2352.1.3.) The Value part of an interface-matching filter specifies the interface-matching pattern for the filter's interface pattern type.

Any interface retrieved from the target device that matches any interface-matching pattern for any interface pattern type for the device is included in the discovered topology. Conversely, any interface retrieved from the target device that does not match any of the interface-matching patterns is excluded from the discovered topology.

As indicated in the example, a set of interface-matching filters for a device consists of three different interface pattern types:

- IFIndexPattern pertains to the ifIndex MIB object type.
- IFDescrPattern pertains to the ifDescr MIB object type.
- IFTypePattern pertains to the ifType MIB object type.

The default value (interface-matching pattern) for IFIndexPattern is “~*”, the default value for IFDescrPattern is “~*”, and the default value for IFTypePattern is “*”. As examples:

- IFIndexPattern-.1.3.6.1.4.1.164.6.1.3.183 ~*
- IFDescrPattern-.1.3.6.1.4.1.164.6.1.3.183 ~*
- IFTypePattern-.1.3.6.1.4.1.164.6.1.3.183 *

The “*” wildcard pattern matches anything, and the “~*” wildcard pattern matches nothing because of the leading tilde. Refer Chapter 9 Terminology for more information on wildcard syntax.

Each pattern type and its value is considered an interface-matching filter.

The interface-matching filters for a device are inclusive, meaning that if an interface that is retrieved from the target device matches just one of the matching patterns that are specified for IFIndexPattern, IFDescrPattern, or IFTypePattern, the interface will be included in the discovered topology. So, by default, because of the “*” in the IFTypePattern, all of the interfaces that are retrieved from the target device are included in the topology.

Interface-matching filters in the tpmgr-param.conf file

Use the interface-matching filters in the tpmgr-param.conf configuration file as a starting point to configure interface-matching filters for access devices. The tpmgr-param.conf file contains the IFTypePattern and IFDescrPattern interface-matching filters. These filters are listed in a table in the VMware Smart Assurance IP Manager Reference Guide. Modify any of these filters or add your own filters to the file.

Other interface-limiting parameter types in the tpmgr-param.conf file

The tpmgr-param.conf file also contains other interface-limiting parameter types, most of which are listed in a table in the VMware Smart Assurance IP Manager Reference Guide. The table includes an example, the default value, and the definition of each parameter type. Several parameter types in the table are interface-matching filters that are also based on the ifIndex, ifDescr, and ifType objects. However, they have slightly different interface-matching pattern names and employ slightly different syntax than the standard interface-matching filters previously discussed.

The VMware Smart Assurance IP Manager User Guide provides information on configuring interface matching filters.

Understanding groups and settings

A *group* contains one or more settings, and a *setting* contains one or more parameters. Managed objects (systems discovered in the managed environment) become members of a group based on the matching criteria defined for the group.

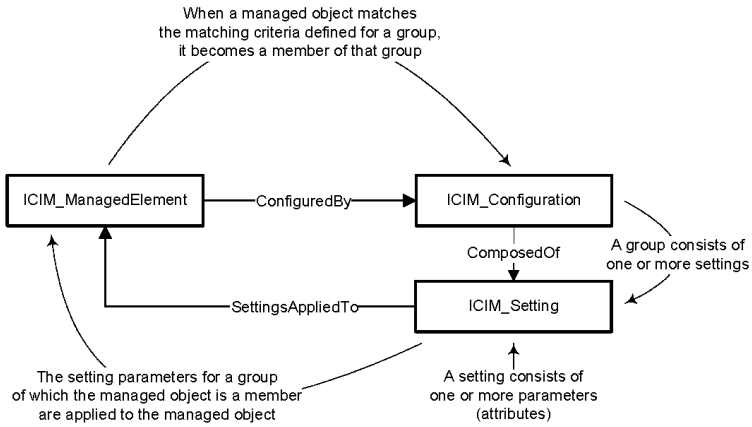
How managed objects are assigned to groups

A managed object may be a member of one and only one group within a particular group category. When the IP Manager performs discovery, it automatically assigns each managed object to one or more groups based on:

- The matching criteria defined for a group.
- The priority of a group, which determines membership when an object meets the matching criteria for more than one group within a particular group category.

  Relating managed objects to groups and settings illustrates the relationships between managed objects, groups, and settings.

Figure 4-3. Relating managed objects to groups and settings



- The VMware Smart Assurance ECIM-R core class for group (in the sense of “groups and settings”) is ICIM_Configuration. This class is the parent of all group classes that are used by IP Manager or any other VMware Smart Assurance Domain Manager.

- The ECIM-R core class for setting is ICIM_Setting. This class is the parent of all setting classes that are used by IP Manager or any other VMware Smart Assurance Domain Manager.
- The ECIM-R core class for managed object is ICIM_ManagedElement. This class is the parent of all managed object classes used by IP Manager or any other VMware Smart Assurance Domain Manager.

Default groups and settings

The Polling, Thresholds, Device Access, and Tagging tabs of the Polling and Thresholds Console show the default polling groups, the default threshold groups, the default device access groups, and the default IP tagging filter groups shipped with the IP Manager software.

The tabs also identify the matching criteria defined for each group, the default settings that are associated with each group, and the default parameter values that are associated with each setting. The VMware Smart Assurance IP Manager Reference Guide provides information on all these settings.

Matching criteria

The matching criteria for a group are the attribute values defined for the target class that is associated with the group. For polling and threshold groups, the target class is Port, Interface, or UnitaryComputerSystem (such as a Switch, Router, Hub, or Bridge). For device access and IP tagging filter groups, the target class is UnitaryComputerSystem.

Any managed object that meets the matching criteria for a group becomes a member of the group. When a managed object becomes a member of a group, all parameter values that are defined for the group are applied to the managed object.

Prioritized lists of groups

Within each group category (polling groups, threshold groups, device access groups, and IP tagging filter groups), groups can be arranged in a high-to-low priority. A managed object is compared against the highest priority group first and becomes a member of the first group for which there is a criteria match.

A managed object may become a member of different groups in different group categories, but a managed object may only be a member of one group within a particular group category.

The VMware Smart Assurance IP Manager User Guide provides information on working with groups and settings and the VMware Smart Assurance IP Manager Reference Guide provides information on Polling settings, Threshold settings, default polling groups, their target classes, matching criteria and their default settings.

Understanding ICMP and SNMP polling

The IP Availability Manager and IP Performance Manager uses the Internet Control Message Protocol (ICMP) and Simple Network Management Protocol (SNMP) polling to monitor the IP network. The results of this polling are then compared to threshold values that define acceptable and unacceptable levels of connectivity or performance. The thresholds, in conjunction with traps, are used to diagnose the failed or degraded objects that interrupt network connectivity or adversely affect network performance.

The monitoring of the network is controlled by ICMP and SNMP polling-related and threshold-related settings that are created for the IP Availability Manager through the Polling and Thresholds Console.

This discussion focuses on the polling-related settings. Threshold related settings are described in the VMware Smart Assurance IP Manager User Guide.

ICMP and SNMP polling and analysis

The IP Manager and monitors the health of the IP network by periodically sending ICMP polls and SNMP polls to the discovered network devices. The results of the polling are compared to threshold values that define acceptable and unacceptable levels of connectivity or performance.

The polled data and the thresholds, in conjunction with received traps, are used by the IP Availability Manager to detect the failed or degraded objects that adversely affect network performance. In addition, the polled data, thresholds, and received traps are used by the IP Availability Manager to collect or determine status updates, which are imported by the IP Manager to diagnose the failed or degraded objects that interrupt network connectivity.

The VMware Smart Assurance IP Manager User Guide provides a description of ICMP and SNMP polling for correlation analysis.

The VMware Smart Assurance IP Manager Reference Guide describe the events that are diagnosed by the IP Managers.

Understanding polling groups and settings

Using the Polling and Thresholds Console, you can customize ICMP and SNMP polling for the IP Availability Manager and IP Performance Manager by modifying default polling groups, or by creating new polling groups. The polling groups periodically poll the managed IP devices to collect availability and performance data.

The IP Availability Manager and IP Performance Manager uses the *settings* in a polling group to assign polling parameters to a particular group of managed devices. From a polling and threshold perspective, a setting is a collection of parameters that are common to a particular type of analysis; for example, connectivity polling or environment polling.

Polling settings for IP availability monitoring

Through the polling settings for IP availability monitoring, you control the ICMP and SNMP connectivity polling of the managed network objects in your network. The polled information serves as input to IP Availability Manager‘s availability analysis of switches, routers, hubs, bridges, and other systems.

The IP Availability Manager provides the following polling settings for IP availability monitoring:

- Connectivity Polling
- Connectivity Polling - External Poller
- Connectivity Polling for IPv6 Addresses
- Connectivity Polling for IPv6 Addresses - External Poller

- Default IP_Route Connectivity Polling

  The “Connectivity Polling” and “Connectivity Polling - External Poller” settings are mutually exclusive, which means that only one or the other (but not both) can be applied to a polling group at any given time. The “Connectivity Polling for IPv6 Addresses” and “Connectivity Polling for IPv6 Addresses - External Poller” settings are also mutually exclusive.

  The VMware Smart Assurance IP Manager Reference Guide provides information on connectivity polling setting parameters and external poller settings.

Default IP_Route Connectivity Polling setting

The Default IP_Route Connectivity Polling setting determines the polling intervals that are used by the IP Availability Manager to monitor default routes. The IP Availability Manager monitors the default routes by probing the SNMP ipRouteTable (.1.3.6.1.2.1.4.21.0.0.0.0) on certain router devices.

Routers in an organization generally point the default route to the router that has a connection to a network service provider. By doing so, packets with destinations that are outside of the organization’s IP address space will be forwarded to that service provider. The default route address in IPv4 is 0.0.0.0/0.

The lack of a default route on a router is a good indication that an organization’s service provider is having connectivity issues.

When default-route polling fails, the IP Availability Manager generates a MissingDefaultRoute event. Specifically, the IPRoute object that is created as part of the Default IP_Route Connectivity Polling setting, itself, generates the MissingDefaultRoute event.

The VMware Smart Assurance IP Manager Reference Guide provides information on the Default IP_Route Connectivity Polling setting parameters.

Default polling groups

The IP Manager provides a polling category named “Polling Groups,” which contains the following default polling groups:

- ComputeFabric
- MediaGateways
- 5620 SAM Managed Systems
- Proxy Devices
- Switches
- Routers
- Hubs and Bridges
- WirelessControllers
- WirelessAccessPoints
- WirelessClients

- Other Systems

Polling setting for optimization

Through the Cleanup Unused Monitoring Instances setting, you control the removal of instrumentation objects that no longer have an associated managed object to monitor. This situation occurs when you remove a setting from a polling group.

For example, when you apply a setting such as Connectivity Polling to a polling group, the IP Manager creates the corresponding instrumentation objects. If you subsequently remove the Connectivity Polling setting from the polling group, the IP Manager responds in one of two ways:

- With theCleanup Unused Monitoring Instances setting applied to the polling group, the IP Manager removes the corresponding instrumentation objects from memory.
- With the Cleanup Unused Monitoring Instances setting *not* applied to the polling group, the IP Manager does *not* remove the corresponding instrumentation objects from memory.

  By default, the Cleanup Unused Monitoring Instances setting is applied to the following polling groups:

- Switches
- Routers
- Hubs and Bridges
- Other Systems

  “Default polling groups for the IP Manager” table in the VMware Smart Assurance IP Manager Reference Guide, provides the default polling groups, their target classes and matching criteria, and their default settings.

  Settings affected by the Cleanup Unused Monitoring Instances setting lists the settings that are affected by the Cleanup Unused Monitoring Instances setting.

Table 4-5. Settings affected by the Cleanup Unused Monitoring Instances setting

Polling setting	Instrumentation objects created for these classes
Connectivity Polling	■ Card ■ Interface ■ Port ■ EVCEndPoint
Connectivity Polling for IPv6 Addresses	■ IPv6
Environment Polling	■ Fan ■ PowerSupply ■ TemperatureSensor ■ VoltageSensor
File System Polling	■ Disk ■ File System

Table 4-5. Settings affected by the Cleanup Unused Monitoring Instances setting (Continued)

Polling setting	Instrumentation objects created for these classes
Performance Polling - Ports and Interfaces	■ Interface ■ Port
Performance Polling - Processor and Memory	■ Memory ■ Processor

You can specify matching criteria for a polling group to limit its membership, as explained in VMware Smart Assurance IP Manager User Guide.

Understanding ICMP and SNMP polling and thresholds

The IP Availability Manager and the IP Performance Manager uses Internet Control Message Protocol (ICMP) and Simple Network Management Protocol (SNMP) polling to monitor the IP network. The monitoring of the network is controlled by ICMP and SNMP polling-related and threshold-related settings that are created for the IP Availability Manager and the IP Performance Manager through the Polling and Thresholds Console.

This discussion focuses on the threshold-related settings. Polling-related settings are described in the VMware Smart Assurance IP Manager User Guide.

ICMP and SNMP polling, thresholds, and analysis

The IP Availability Manager and the IP Performance Manager monitor the health of the IP network by periodically sending ICMP polls and SNMP polls to the discovered network devices. The results of the polling are compared to threshold values that define acceptable and unacceptable levels of connectivity or performance.

The polled data and the thresholds, in conjunction with received traps, are used by the IP Availability Manager and the IP Performance Manager to detect the failed or degraded objects that adversely affect network performance. In addition, the polled data, thresholds, and received traps are used by the IP Availability Manager and the IP Performance Manager to collect or determine status updates, which are imported by the IP Manager to diagnose the failed or degraded objects that interrupt network connectivity.

VMware Smart Assurance IP Manager User Guide provides a description of ICMP and SNMP polling for correlation analysis.

The VMware Smart Assurance IP Manager Reference Guide describe the events that are diagnosed by the IP Availability Manager and the IP Performance Manager.

Understanding threshold groups and settings

Using the Polling and Thresholds Console, you can customize ICMP and SNMP polling thresholds for the IP Availability Manager and the IP Performance Manager by modifying default threshold groups, or by creating new threshold groups.

The IP Availability Manager and the IP Performance Manager use the *settings* in a threshold group to assign threshold parameters to a particular group of managed devices, interfaces, or ports. From a polling and threshold perspective, a setting is a collection of parameters that are common to a particular type of analysis; for example, connectivity polling or environment polling.

Understanding device access groups and settings

Because the IP Manager is a component of many VMware Smart Assurance deployments, the IP Availability Manager has been selected as the central repository for storing the device access credentials for other Domain Managers — such as command line interface (CLI) login credentials required by Network Protocol Manager for EIGRP and Network Protocol Manager for IS-IS to perform CLI discovery and, optionally, CLI polling. The satellite Domain Managers import the device access credentials as well as initial topology from IP Availability Manager when they add IP Availability Manager as a source.

Polling settings for IP performance monitoring

Through the polling settings for IP performance monitoring, you control the ICMP and SNMP performance polling of the managed network objects in your network. The polled information serves as input to IP Performance Manager‘s fault and performance analysis of interfaces and ports, processors and memory, disks and file systems, fans and power supplies, and voltage and temperature sensors.

The IP Performance Manager provides the following polling settings for IP performance monitoring:

- Environment Polling
- Environment Polling - External Poller
- File System Polling
- Performance Polling - Ports and Interfaces
- Performance Polling - Processor and Memory

  The “Environment Polling” and “Environment Polling - External Poller” settings are mutually exclusive, which means that only one or the other (but not both) can be applied to a polling group at any given time.

Environment Polling setting

The Environment Polling setting configures the polling intervals that are used to monitor system components such as the fan, memory, power supply, temperature sensor, and voltage sensor objects.

The VMware Smart Assurance IP Manager Reference Guide provides information on the Environment Polling setting parameters.

Environment Polling - External Poller

The Environment Polling - External Poller setting serves two purposes:

- Provides a means of testing the system-component instrumentation and analysis functionality of the IP Performance Manager.
- Provides a means for the IP Performance Manager to import system-component status updates from an VMware Smart Assurance adapter, such as the Adapter for Alcatel-Lucent 5620 SAM EMS.

  The VMware Smart Assurance IP Manager Reference Guide provides information on the Environment Polling - External Poller setting parameters.

File System Polling setting

The File System Polling setting configures the polling intervals that are used to monitor a host system's file system. This setting is a valid choice only for the Other Systems polling group.

The VMware Smart Assurance IP Manager Reference Guide provides information on the File System Polling setting parameters.

Performance Polling - Ports and Interfaces setting

The Performance Polling - Ports and Interfaces setting configures the polling intervals that are used to monitor the performance of managed ports and interfaces.

The VMware Smart Assurance IP Manager Reference Guide provides information on the Performance Polling - Ports and Interfaces setting parameters.

Performance Polling - Processor and Memory setting

The Performance Polling - Processor and Memory setting configures the polling intervals that are used to monitor a system's processor and associated memory objects.

The VMware Smart Assurance IP Manager Reference Guide provides information on the Performance Polling - Processor and Memory setting parameters.

Thresholds tab

The IP Manager provides the following categories of threshold groups, each of which contains default threshold groups:

- File Systems
- Interface Groups
- Port Groups - Access Ports
- Port Groups - Trunk Ports
- System Resource Groups

- System VLAN Tag Groups
- TemperatureSensor
- VoltageSensor
- VLAN Impact Propagation Groups

  The “VLAN Impact Propagation Groups” category does not automatically appear. You must issue a certain command for the VLAN Impact Propagation Groups category to appear.

File Systems

File system is treated as a logical network object. The File System Management functionality allows you to define and configure thresholds for file system space utilization.

In the earlier version, a single threshold value is set for all file systems using the System Resource group. The disadvantage is that each file system cannot be monitored individually.

The following enhancements are added:

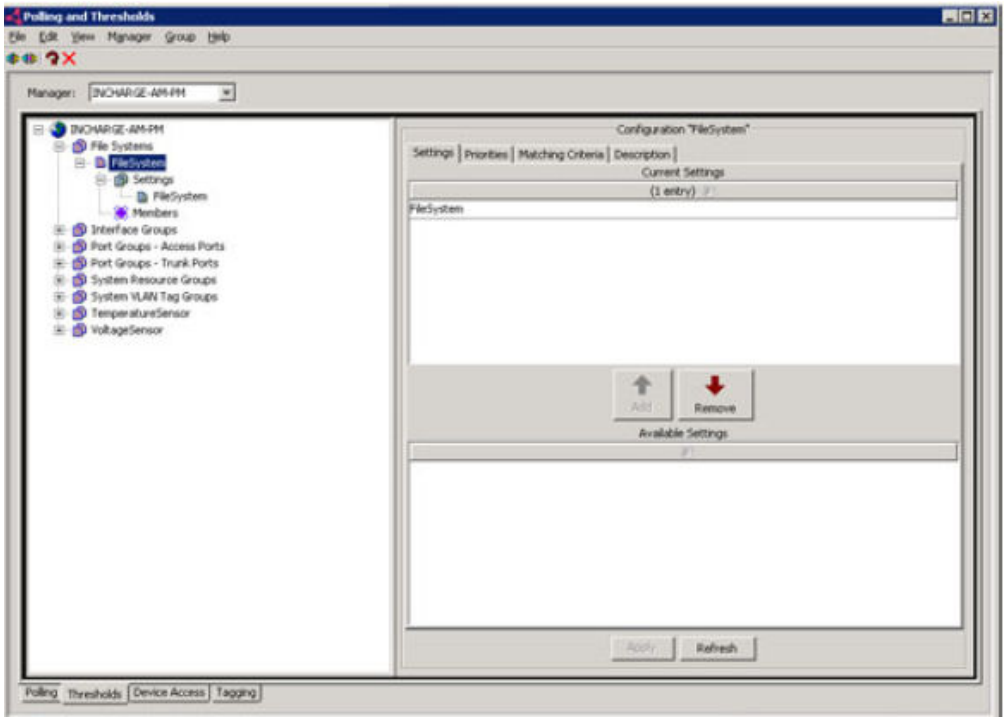
- Ability to define thresholds for each file system instance
- Ability to define thresholds using the current wildcard mechanism. Wildcards are enabled to be used to define inclusion and/or exclusion lists of either servers or file systems.
- Ability to change default thresholds per file system
- Ability to manage and unmanage file systems

Define thresholds for individual file system instances

You can now configure thresholds for individual file system instances. A default group called File System is available.

The utilization thresholds can be configured for each file system instance. As a result, a host with multiple file systems can have different thresholds defined individually for each file system instance. Each file system instance is monitored individually.

Figure 4-4. Groups for file system instances



- You can configure two types of utilization thresholds (in percentage) for each file system instance. They are *MaxUtilizationPct* and *ModerateUtilizationPct.*
- The events generated are *HighUtilization* and *ModerateUtilization*.

Ability to define thresholds using wildcard

The file system instances can be grouped based on one or more matching criteria. You can use wildcards to configure utilization thresholds for each file system group. The configured thresholds are applied to all the file system instances of that group.

You can also create more file system groups, add new matching criteria, and configure the thresholds for each file system group.

This can be done in the **Matching Criteria** tab for the selected File System group in the **Polling and Thresholds** screen.

Sample file system groups and matching criteria lists the sample file system groups along with their matching criteria.

Table 4-6. Sample file system groups and matching criteria

Sample file system groups	Matching Criteria	Applicable in
OS Only File Systems	/	Linux
Writable File Systems	/var	Linux
Temporary File Systems	/tmp	Linux

Table 4-6. Sample file system groups and matching criteria (Continued)

Sample file system groups	Matching Criteria	Applicable in
Non Critical File Systems	User defined	
Unclassified File Systems	User defined	

Events and thresholds for file systems

The VMware Smart Assurance IP Manager Reference Guide provides the events and thresholds for file systems along with their description.

The thresholds defined in the System Resource Group serve as the default and may be overwritten by values in one of the new File System groups. If a file system instance belongs to a particular File System group, it inherits the group's threshold settings. In case it does not belong to any group, the thresholds in the System Resource group that the device belongs to will apply.

When new file system groups are created, they do not have any matching criteria and are empty. You can add new matching criteria and configure the thresholds for each file system group. This can be done in the **Matching Criteria** tab for the selected file system group in the **Polling and Thresholds** window.

Each file system group is placed in the order of priority in the **Polling and Thresholds** window. You can change the priority from the Priority tab. The matching of the file system instances begin from the highest priority group and continues in the order of priority. A file system instance that matches a group is not considered for matching with the remaining groups.

Sample file system groups and matching criteria

The VMware Smart Assurance IP Manager Reference Guide lists sample file system groups, their description, and their matching criteria. You can use those as a guide to create file system groups manually.

Managing or unmanaging file systems

You can manage or unmanage an individual file system instance. When a file system is rendered managed, that file system instance is monitored. On the other hand, when a file system instance is rendered unmanaged, that file system instance is not monitored.

The VMware Smart Assurance IP Manager User Guide provides information on how to manage and unmanage file systems.

Attribute of file system describes the attribute for a file system instance.

Table 4-7. Attribute of file system

Attribute	Description	Allowed values
IsManaged	Determines if a file system instance should be monitored by the management system. An unmanaged object does not have any associated instrumentation. This attribute is read-only.	Boolean: True or False Default: True

Migration impacts

"Migrate data from older IP deployments" section in Chapter 4, Performing Migration, in the VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers provides more information about migrating settings.

Interface Groups

Threshold groups for interfaces are used to configure parameters for interface analysis. Thresholds are determined by the interface's media type (for example, Ethernet or ATM) and its role (primary, backup, or dial-on-demand).

The VMware Smart Assurance IP Manager Reference Guide provides information on the default interface threshold groups, their target classes and matching criteria, and their default settings. The target class for all interface threshold groups is Interface.

- The default group-to-setting mappings in the table are the mappings that users would typically use. The settings that are available to an interface threshold group are as follows:
  - Backup Interface Support
  - Dial-on-Demand Interface Support
  - Disable Fault Analysis for Interface/Port
  - Disable Performance Analysis for Interface/Port
  - Ethernet Interface/Port Performance
  - Generic Interface/Port Performance
  - Interface/Port Flapping
- The "Other Interfaces" threshold group has the lowest priority. Managed interfaces that do not match the criteria for the other interface threshold groups become members of the Other Interfaces threshold group.
- Only discovered objects that match the type and attribute values in the "Target class and matching criteria" column for a particular group can become members of that group.
- By default, all discovered objects that match the type and attribute values in the "Target class and matching criteria" column for a particular group become members of that group.

  You can specify matching criteria for a threshold group to limit its membership.

Port Groups - Access Ports

Threshold groups for access ports are used to configure parameters for access port analysis. Thresholds are determined by the access port's media type; for example, Ethernet or ATM.

An access port is a port that provides a connection for end users or node devices, such as routers or servers. An access port becomes managed if it has a connection or is explicitly managed.

The VMware Smart Assurance IP Manager Reference Guide provides information on the default access port threshold groups, their target classes and matching criteria, and their default settings. The target class for all access port threshold groups is Port.

- The default group-to-setting mappings in the table are the mappings that users would typically use. The settings that are available to an access port threshold group are as follows:
  - Disable Fault Analysis for Interface/Port
  - Disable Performance Analysis for Interface/Port
  - Ethernet Interface/Port Performance
  - Generic Interface/Port Performance
  - Interface/Port Flapping
- The “Other Ports” threshold group has the lowest priority. Managed access ports that do not match the criteria for the other access port threshold groups become members of the Other Ports threshold group.

Port Groups - Trunk Ports

Threshold groups for trunk ports are used to configure parameters for trunk port analysis. Thresholds are determined by the trunk port’s media type; for example, Ethernet or ATM.

A trunk port is any port that connects to another port or interface on a device that participates in the Layer 2 bridging protocol. By default, all trunk ports are managed, which means that by default all trunk ports can be associated with trunk port threshold groups.

Because trunk ports are inherently more critical than access ports, trunk port threshold groups typically have lower thresholds than access port threshold groups for utilization, errors, and so on. That is, the thresholds for trunk port threshold groups are more stringent than the thresholds for access port threshold groups.

The VMware Smart Assurance IP Manager Reference Guide provides information on the default trunk port threshold groups, their target classes and matching criteria, and their default settings. The target class for all trunk port threshold groups is Port.

- The default group-to-setting mappings in the table are the mappings that users would typically use. The settings that are available to a trunk port threshold group are as follows:
  - Disable Fault Analysis for Interface/Port
  - Disable Performance Analysis for Interface/Port
  - Ethernet Interface/Port Performance
  - Generic Interface/Port Performance
  - Interface/Port Flapping
- The “Other Ports” threshold group has the lowest priority. Managed trunk ports that do not match the criteria for the other trunk port threshold groups become members of the Other Ports threshold group.

System Resource Groups

Threshold groups for system resources are used to configure parameters for system analysis. The sensitivity of the system thresholds is determined by the role of the system.

The VMware Smart Assurance IP Manager Reference Guide provides information on the default system resource threshold groups, their target classes and matching criteria, and their default settings. The target class for all system resource threshold groups is UnitaryComputerSystem.

- The default group-to-setting mappings in the table are the mappings that users would typically use. The settings that are available to a system resource threshold group are as follows:
  - Cisco Fast and NvRam Memory
  - Connectivity
  - Environment
  - File System
  - Interface Management Policy
  - Juniper Environment
  - Processor and Memory
  - VLAN Tagging Policy
- The “Other Systems” group has the lowest priority. Managed systems that do not match the criteria for the other system resource threshold groups become members of the Other Systems group. For example, managed host systems become members of the Other Systems group.

System VLAN Tag Groups

Threshold groups for VLAN tagging are used to ensure that systems are assigned to their appropriate VLAN. These groups are described in the VMware Smart Assurance IP Manager User Guide, under “Performing tasks in the Thresholds tab”.

The VMware Smart Assurance IP Manager Reference Guide provides information on the default system VLAN tag groups, their target classes and matching criteria, and their default settings. The target class for all system VLAN tag groups is UnitaryComputerSystem.

- Even though the following settings are available to a system VLAN tag group, only the “VLAN Tagging Policy” setting is valid:
  - Cisco Fast and NvRam Memory
  - Connectivity
  - Environment
  - File System
  - Interface Management Policy

  - Juniper Environment
  - Processor and Memory
  - VLAN Tagging Policy
- The “Other Systems” group has the lowest priority. Managed systems that do not match the criteria for the Switches system VLAN tag group become members of the Other Systems group. For example, managed host systems become members of the Other Systems group.

Thresholds for temperature and voltage sensors

Voltage and Temperature thresholds are set to generate alarms when the actual values in the devices cross the defined threshold values.

Two types of values can be set for voltage threshold, that is, high and low voltage threshold. Only one threshold can be set for the temperature sensor, that is, the high temperature threshold. Accordingly, two types of alarms can be generated for voltage sensors, and only one alarm can be generated for the temperature sensor.

You can configure separate thresholds for different voltage and temperature sensors in the same device.

Note The threshold values set at the Instance level take precedence over those set at the Device Type (System Resource Group) level. When no values are specified at Instance level, Device level threshold values prevail. Additionally, if the value for the parameter, UseMIBThresholds is set to TRUE, the MIB values will take precedence over the threshold values set at IP for alarm generation.

By default, the following are available:

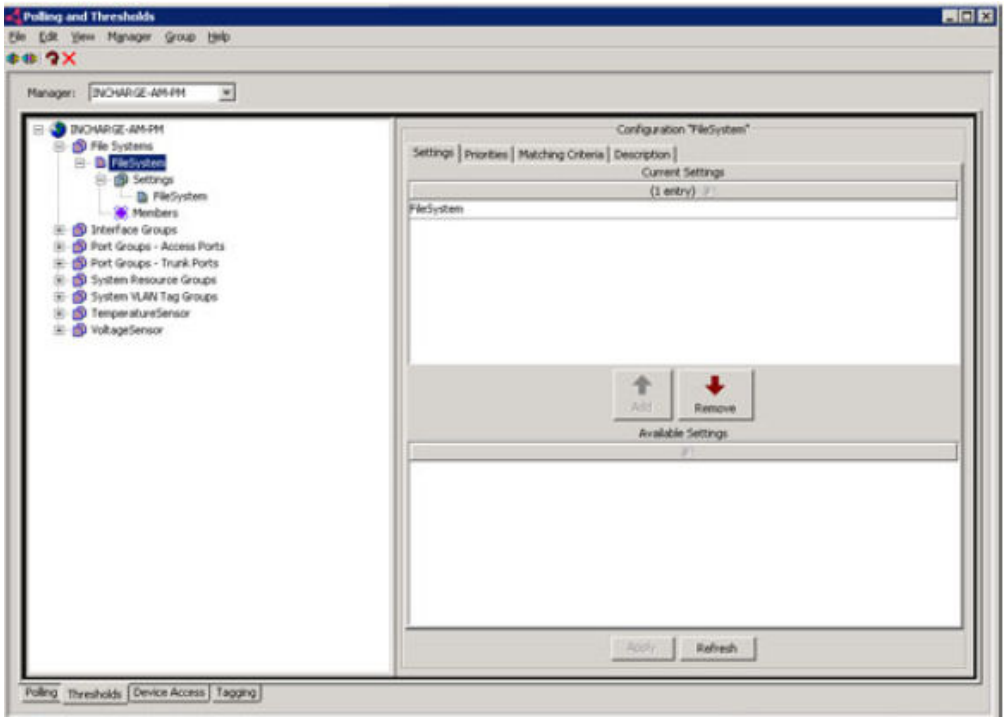
- Groups called TemperatureSensor and VoltageSensor. These groups are empty and do not match any temperature or voltage sensors.
- The thresholds defined in the System Resource Group serve as the default and may be overwritten by values in one of the new groups.

  If a temperature sensor or voltage sensor belongs to a particular group, it inherits the group’s threshold setting. In case it does not belong to any group, the thresholds in the System Resource group will apply.

Creating separate thresholds for voltage and temperature sensors

To create separate thresholds for different voltage and temperature sensors (or a group of sensors), you must create groups by using appropriate matching criteria.

Figure 4-5. Groups for temperature and voltage sensors



Device-level threshold values can be set from the **Polling and Threshold** window, under the **Threshold** tab. In the **Polling and Threshold** window, navigate to **SystemResourceGroups > Device > Settings > Environment**, and set the required threshold value.

Instance-level threshold values can be set from the **Polling and Threshold** window, under the **Threshold** tab. In the **Polling and Threshold** window, navigate to **Temperature Sensor > Voltage Sensor**, and set the required threshold value.

Also priority has to be set, with custom group taking higher priority than the default groups. The threshold values for a group which has been assigned higher priority take precedence over the others.

Note When a group's Voltage or Temperature threshold setting is removed, the group's instrumentation too should be removed.

Threshold behavior for temperature sensors

The threshold behavior for temperature sensors is different. Discovery of thresholds for temperature sensors using EntitySensor MIB provides more details.

The following sections cover details regarding thresholds and events for the sensors:

- Default thresholds and events
- Discovery of thresholds for temperature sensors using EntitySensor MIB
- Temperature sensor threshold attributes explained

Threshold settings

Default thresholds and events

The table “Events and thresholds for temperature and voltage sensors” in the VMware Smart Assurance IP Manager Reference Guide provides information on the default threshold setting and events generated for temperature and voltage sensors.

Discovery of thresholds for temperature sensors using EntitySensor MIB

Temperature sensor thresholds can be set during discovery for devices that are certified in the IP Manager using **EntitySensor MIB** for **Environment**. To verify if threshold values are discovered from the MIB, you must enable **MetaClass** and verify the **Instrumentation Class** (TemperatureSensor_Fault_CiscoEntityFRU) of the specific temperature sensor object.

There are two possibilities for setting the threshold value of the temperature sensors:

- If the threshold values returned by the agent are greater than 0, discovery reads in the threshold values from the EntitySensor MIB and automatically sets the thresholds in the instrumentation object to the values from the agent.
- If the device agent does not return a positive value for the threshold reading, then the threshold value falls back to the default setting defined in either the TemperatureSensor group or in the System Resource group.

Temperature sensor threshold attributes explained

The following attributes are explained in the VMware Smart Assurance IP Manager Reference Guide:

- MajorAlarmThreshold
- MibEntSensorThreshold
- CurrentValue
- RelativeTemperatureThreshold
- TemperatureHighThreshold
- HighThreshold

VLAN Impact Propagation Groups

Threshold groups for VLAN impact propagation are used to prevent device impact propagation in certain (user-specified) VLANs, such as management VLANs and default VLANs. These groups are described in the VMware Smart Assurance IP Manager User Guide, under “Performing tasks in the Thresholds tab”. The default VLAN impact propagation groups do not automatically appear. You must issue a certain command for the default VLAN impact propagation groups to appear.

The VMware Smart Assurance IP Manager Reference Guide provides information on the default VLAN impact propagation groups, their target classes and matching criteria, and their default settings. The target class for all VLAN impact propagation groups is VLAN.

- The only setting that is available to a VLAN impact propagation group is “Problem Impact propagation policy over VLANs.” The VMware Smart Assurance IP Manager Reference Guide provides information on the “Problem Impact propagation policy over VLANs” setting parameters
- The Other VLANs group has the lowest priority. VLANs that do not match the criteria for the VLANs group become members of the Other VLANs group.

Threshold settings for IP availability monitoring

Through the threshold settings for IP availability monitoring, you control the ICMP and SNMP connectivity polling thresholds of the managed network objects in your network. The polled information serves as input to IP Availability Manager‘s availability analysis of switches, routers, hubs, bridges, and other systems.

The IP Manager provides the following threshold settings for IP availability monitoring:

- Backup Interface Support
- Connectivity
- Dial-on-Demand Interface Support
- Interface/Port Flapping

  The VMware Smart Assurance IP Manager Reference Guide provides more information on the threshold settings for IP availability monitoring.

Threshold settings for IP performance monitoring

Through the threshold settings for IP performance monitoring, you control the ICMP and SNMP performance polling thresholds of the managed network objects in your network. The polled information serves as input to IP Performance Manager‘s fault and performance analysis of interfaces and ports, processors and memory, disks and file systems, fans and power supplies, and voltage and temperature sensors.

The IP Manager provides the following threshold settings for IP performance monitoring:

- Ethernet Interface/Port Performance
- Generic Interface/Port Performance
- Cisco Fast and NvRam Memory
- Environment
- File System
- Juniper Environment

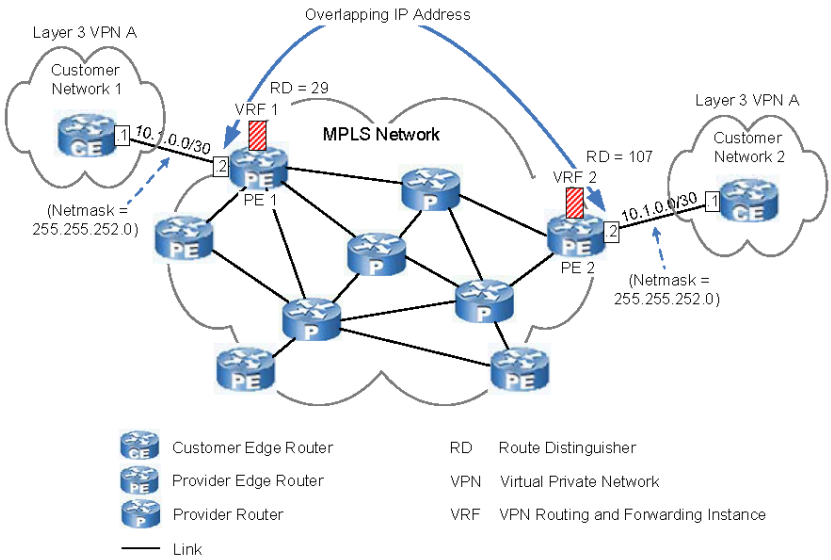
- Processor and Memory

  The VMware Smart Assurance IP Manager Reference Guide provides more information on the threshold settings for IP performance monitoring.

Understanding the IP tagging feature

In a networked environment, the same IP addresses can be reused across multiple systems (devices) running in separate networks. These IP addresses, known as overlapping IP addresses, are prominent in MPLS VPN environments and in networks centrally managed by Managed Service Providers, where private IP networks of customers often deploy identically numbered IP address spaces. Example of an overlapping IP in an MPLS VPN network is an example of an overlapping IP address in an MPLS VPN network.

Figure 4-6. Example of an overlapping IP in an MPLS VPN network



Overlapping IP addresses do not create problems across different networks; however, when a service-assurance application for a centrally managed network attempts to discover the overlapping IP addresses, IP address identity problems arise. For example, without the IP tagging feature, the IP Manager will respond as follows when it discovers an overlapping IP address:

- Create a DuplicateIP notification and a DuplicateIP object.
- Discard the most recently received overlapping address.
- Remove the initially received overlapping address from its repository.

  With the IP tagging feature, the IP Manager is able to distinguish between overlapping addresses belonging to different customers by adding a distinguishing tag to their names.

  That is, instead of using the standard naming scheme of “IP-<*IP address*>” to name an overlapping IP address instance, the IP Manager uses the “IP-<*IP address*>/<*tag*>” naming scheme to name the instance.

IP tagging scenario

Consider the following scenario:

- In the managed network environment, two switches (SW1, SW2) in one customer network and two switches (SW3, SW4) in another customer network are using the same IP address space.
- Without creating an IP tag group for at least one of the customer networks, the IP Manager will discard the overlapping IP addresses collected from the four switches.
- By creating an IP tag group for at least one of the customer networks, the IP Manager will assign one name to the IP addresses collected from SW1 and SW2, and assign a different name to the IP addresses collected from SW3 and SW4.
- For example, if you create an IP tag group having a “York” tag and matching criteria that allow SW1 and SW2 to become members of the group, the IP Manager will assign the name IP-<*IP address*>/York to the IP addresses collected from SW1 and SW2, and assign the name IP-<*IP address*> to the IP addresses collected from SW3 and SW4.

  With the IP tagging feature, create an IP tag filter for each group of systems that is using an overlapping address space. Included in each filter is a user-specified tag value and a range of overlapping IP addresses.

Why modeling overlapping IP addresses is important

Through the IP tag filters, the IP Manager is able to create IP objects in the modeled topology that represent the overlapping IP addresses that are used by locally distinct groups of systems. This capability leads to better modeling of the managed network environment, which in turn leads to more accurate correlation analysis results.

An overlapping IP address is often on a client-side physical interface of a router or switch managed by a Managed Service Provider (MSP). The MSP wants to discover the router and all of its interfaces and IP addresses, but does not want to (and often cannot) discover the client equipment accessible through the overlapping IP address. The inclusion of the overlapping IP addresses in the modeled topology allows the MSP to observe the status of the client equipment that is associated with the overlapping IP addresses, which enables the MSP to respond immediately to customer outages.

The IP Manager monitors the status of the overlapping IP addresses by periodically sending ICMP and SNMP requests to the systems on which the overlapping IP addresses were discovered.

Overlapping IP addresses versus management IP addresses

An overlapping IP address is typically an address on an edge system's physical interface, not the address — known as a management IP address — bound to the edge system's SNMP agent. (An edge system is a system at the edge of the managed network environment.) An overlapping IP address is typically unreachable in the sense that no routes exist between the host running the IP Manager and the overlapping IP address.

Note A management IP address is the address that the IP Manager uses to discover information about a system and to monitor the system after it is discovered.

Managed IP objects versus unmanaged IP objects

Even for a pair of overlapping IP addresses that are reachable by the IP Manager, the IP Manager will not be able to monitor them because it will not be able to determine which of the systems bound to those IP addresses is responding to the ICMP or SNMP polls. If one of the overlapping IP address is reachable and the other is not, managing the reachable one and unmanaging the unreachable one would make sense.

The "Unmanage admin down" feature addresses this scenario where a duplicate IP in the network comes in picture due to network configuration change. In such a situation, one interface related to the duplicate IP may be set as "administratively down," as it is not participating in the network. Once the interface is down, the IP related to this particular interface is not discovered, while the other IP remains as it is. The "EnableUnmanageAdminDown" in the tpmgr-param.conf file enables or disables this functionality.

Note The "Unmanage admin down" and the "IP tagging" features are mutually exclusive. To utilize the "Unmanage admin down" functionality, it is required that the IP tagging for that specific set of IPs is removed and then the "unmanage admin down" feature is enabled.

By default, an IP tag filter sets all IP objects created for overlapping IP addresses to "unmanaged," and sets all IP objects created for management IP addresses to "managed." A managed object is monitored by using ICMP and SNMP — unless different access mode is specified — to determine the object's status and connectivity. An unmanaged object is not polled or rediscovered but remains present in the modeled topology.

The VMware Smart Assurance IP Manager User Guide provides more information about managed and unmanaged objects.

Automatic and user-defined IP tagging

The IP tagging feature provides two levels of IP tagging:

- Automatic
- User-defined

  By default, no IP tagging occurs. After you edit the BASEDIR/smarts/conf/discovery/tpmgr-param.conf file and change the DisableOverlappingIP value to FALSE, automatic IP tagging is available.

Automatic tagging

Automatic IP tagging works as enumerated below:

1. For every MPLS-related VRF IP address collected for a discovered system, the IP Manager creates an IP object and tags its name with the route distinguisher (RD) tag value that is associated with the VRF to which the VRF IP address belongs. The IP Manager obtains the RD tag by reading the vendor VPN MIB on the discovered system.
2. The IP Manager tags the name of the IP object with a route distinguisher value regardless of whether the VRF IP is — or is not — an overlapping IP address.

   The IP Manager also creates a corresponding IPNetwork object and tags it with the same value that is used to tag the IP objects. The IP objects have a ConnectedVia relationship with the IPNetwork object.

   Note **Automatic IP tagging is also known as VRF IP tagging.**

User-defined tagging

User-defined IP tagging occurs when you create an IP tag filter group in IP Availability Manager or IP Performance Manager based on specific deployment and specifies a tag value and an overlapping IP address range for the group. The user-specified tag may be applied to both non-VRF overlapping IP addresses and VRF IP addresses, where a non-VRF overlapping IP address is defined as an overlapping IP address that is not part of an MPLS Layer 3 VPN. In addition, users can configure an IP tag filter group to double-tag the name of the objects created for discovered VRF IP addresses, or to replace the route distinguisher tag in the name of these objects with the user-specified tag.

For every IP address that falls within the overlapping IP address range of an IP tag filter group, the IP Manager creates an IP object and tags its name with the user-specified value. The IP Manager also creates a corresponding IPNetwork object and tags it with the same value that is used to tag the IP objects. (The IP objects have a ConnectedVia relationship with the IPNetwork object.)

IP tag naming scheme

The tagged name of a IP or IPNetwork object is defined in the object's Name and DisplayName attributes. The tagged name that is specified in the DisplayName attribute is the name of the object displayed in the Global Console, while the tagged name that is specified in the Name attribute is the name of the object as known in the IP Manager's repository.

In the tagged naming examples that follow, IP address 10.1.0.2 in Figure 18, "Example of an overlapping IP in an MPLS VPN network," is the overlapping IP address.

Naming scheme for automatic tagging

The naming scheme for automatic tagging is based on the route distinguisher (RD) tags obtained from the vendor VPN MIB on the discovered system.

Note **The following discussion assumes DisableOverlappingIP = FALSE.**

IP objects representing VRF IP addresses appear in the topology as:

Name: *IP-<IP address>/<RD tag>*

DisplayName: *<RD tag>:<IP address> [<system containing this IP>]*

Examples:

Name: IP-10.1.0.2/29

DisplayName: 29:10.1.0.2 [PE 1]

Name: IP-10.1.0.2/107

DisplayName: 107:10.1.0.2 [PE 2]

Corresponding tagged IPNetwork objects appear in the topology as:

Name: *IPNET-<IP network>/<RD tag>*

DisplayName: *<RD tag>:<IP network>*

Examples:

Name: IPNET-10.1.0.0/29

DisplayName: 29:10.1.0.0

Name: IPNET-10.1.0.0/107

DisplayName: 107:10.1.0.0

Naming scheme for user-defined tagging

The naming scheme for user-defined tagging is based on user-specified tags in user-created IP tag filters. The tagged naming examples included in this section are based on the following two IP tag filter groups:

- Router-PE1 containing:
  - Matching criteria that allows only router PE 1 to become a member of the Router-PE1 group
  - User-specified tag = cust1
  - Overlapping IP address range = 10.1.0.2
  - Route distinguisher = enabled (default)
- Router-PE2 containing:
  - Matching criteria that allows only router PE 2 to become a member of the Router-PE2 group
  - User-specified tag = cust2
  - Overlapping IP address range = 10.1.0.2
  - Route distinguisher = enabled (default)

Note **The following discussion assumes DisableOverlappingIP = FALSE**

Tagged IP objects representing non-VRF IP overlapping addresses appear in the topology as:

Name: *IP-<IP address>/<user-specified tag>*

DisplayName: *<user-specified tag>:<IP address> [<system containing this IP>]*

Corresponding tagged IPNetwork object:

Name: *IPNET-<IP network>/<user-specified tag>*

DisplayName: *<user-specified tag>:<IP network>*

Tagged IP objects representing VRF IP addresses appear in the topology as:

Name: *IP-<IP address>/<user-specified tag>-<RD tag>*

DisplayName: *<user-specified tag>-<RD tag>:<IP address> [<system containing this IP>]*

Examples:

Name: IP-10.1.0.2/cust1-29

DisplayName: cust1-29:10.1.0.2 [PE 1]

Name: IP-10.1.0.2/cust2-107

DisplayName: cust2-107:10.1.0.2 [PE 2]

Corresponding tagged IPNetwork object:

Name: IPNET-*<IP network>*/*<user-specified tag>-<RD tag>*

DisplayName: *<user-specified tag>-<RD tag>*:*<IP network>*

Examples:

Name: IPNET-10.1.0.0/cust1-29

DisplayName: cust1-29:10.1.0.0

Name: IPNET-10.1.0.0/cust2-107

DisplayName: cust2-107:10.1.0.0

The preceding examples demonstrate the double-tagging capability. Because IP address 10.1.0.2 is a VRF IP and because the route distinguisher parameter is enabled in the Router-PE1 and Router-PE2 IP tag filter groups, the IP objects created for IP address 10.1.0.2 are double-tagged.

By disabling the route distinguisher parameter in the filter groups and invoking a rediscovery of the topology, the IP objects for IP address 10.1.0.2 will receive a single tag:

Name: IP-10.1.0.2/cust1

DisplayName: cust1:10.1.0.2 [PE 1]

Name: IP-10.1.0.2/cust2

DisplayName: cust2:10.1.0.2 [PE 2]

By disabling the route distinguisher parameter in just one of the groups, say Router-PE1, and performing a rediscovery of the topology, the IP objects for IP address 10.1.0.2 will be named as follows:

Name: IP-10.1.0.2/cust1

DisplayName: cust1:10.1.0.2 [PE 1]

Name: IP-10.1.0.2/cust2-107

DisplayName: cust2-107:10.1.0.2 [PE 2]

Shadow IP and IPNetwork objects

When the EnableShadowIP parameter is set to TRUE (the default) in the BASEDIR/smarts/conf/discovery/tpmgr-param.conf file, the IP Manager creates a shadow IP object for each tagged, nonoverlapping, non-VRF IP object that it creates, and creates a shadow IPNetwork object for the IPNetwork object to which the IP object is ConnectedVia. Earlier versions of satellite Domain Managers will import the shadow objects, rather than their tagged counterparts, when importing topology from the IP Manager (specifically, from IP Availability Manager).

Shadow object rationale

Even though you will not intentionally tag a nonoverlapping, non-VRF IP address, it can still happen. (For example, the IP address might just happen to fall within the overlapping IP address range that is specified for an IP tag filter group.) The creation of a shadow IP object for this IP address, which earlier versions of satellite Domain Managers will import, ensures that the Domain Managers become aware of this IP address.

The IP Manager does not create shadow IP objects for tagged, overlapping, non-VRF IP addresses because earlier versions of satellite Domain Managers expect to import overlapping non-VRF IP addresses as DuplicateIP objects. Because DuplicateIP objects are not useful to the Domain Managers, the absence of shadow IP objects for tagged, overlapping, non-VRF IP addresses ensures that the Domain Managers do not become aware of these IP addresses.

So, from the perspective of earlier versions of satellite Domain Managers, the only shadow IP objects that make sense are the ones for tagged, nonoverlapping, non-VRF IP addresses.

Shadow object naming schemes

Shadow IP object names comply with the following naming scheme:

Name: *IP-<IP address>*

Display name: *[Shadow]:<IP address> [<system containing this IP>]*

where *<IP address>* is the IP address of the tagged, non-VRF IP object that the shadow IP object is shadowing.

Shadow IPNetwork object names comply with the following naming scheme:

Name: *IPNET-<IP network>*

Display name: *[Shadow]:<IP network>*

where *<IP network>* is the IP network of the tagged, non-VRF IPNetwork object that the shadow IPNetwork object is shadowing.

Shadow object unique characteristics

A shadow IP object is associated with its tagged IP object counterpart by using the ShadowOf relationship. Similarly, a shadow IPNetwork object is associated with its tagged IPNetwork object counterpart by using the ShadowOf relationship. All shadow objects have an attribute named IsShadow that is set to TRUE.

A shadow IP object has a ConnectedVia relationship to a shadow IPNetwork object, and a shadow IPNetwork object has ConnectedTo relationships to its shadow IP objects.

Note As the shadow IP objects are unmanaged, the IP Manager does not include them in its correlation analysis.

Shadow object use by earlier versions of satellite Domain Manager s

Since the shadow IP object has the same interface and system layering information as its tagged IP object counterpart, earlier versions of satellite Domain Managers can use Shadow IP objects in the same way that they used untagged IP objects: To create the correct connections and to determine the hosts that are hosting the IPs.

Understanding tagging filter groups and settings

The tagging filter groups and settings provide a means to create an IP tag filter and specify a unique tag for each group of systems that are using the same but locally distinct IP addresses. Because the tags enable the IP Manager to distinguish between overlapping addresses belonging to different customers, the IP Manager is able to represent the customer-related overlapping IP addresses in the modeled topology.

Each discovered system that meets the defined matching criteria for an IP tag filter group is subjected to a second line of filtering defined by the parameters defined in the group's setting, including a user-specified tag and a range of overlapping IP addresses. In this way, different filtering values can be applied to different groups of systems.

Audit logging

Audit logging relates to the VMware Smart Assurancearchitecture of an VMware Smart AssuranceGlobal Manager Administration Console (Global Console) interacting with VMware Smart Assuranceanalysis servers such as IP Availability Managerand Performance Manager (IP AM/PM).

Audit logging is enabled on the analysis servers when a modification is made to the server using the Global Console. Chapter 5, Audit Logging, in the
VMware Smart Assurance IP Manager User Guideprovides information on the audit logging format, how to configure audit logging, how to start the audit log adapter, and the list of user actions captured in audit logs.

In raw format, the audit log output on each server contains specific remote API calls made from the Global Console to the server. The raw output is then processed by the VMware Smart AssuranceAudit Log Adapter which parses the output to provide more easily readable and understandable audit log information.

Raw audit log output shows the creation of raw audit log information. The Global Consoles connect to VMware Smart Assuranceanalysis servers, such as IP AM or PM. These servers print information to a file named *<server_name>_audit.log*for all remote API calls sent from the Global Console to the server. Each server maintains one log file regardless of the number of connected Global Consoles. You must run one Audit Log Adapter per server and the Audit Log Adapter must be run on the same host as the server whose audit log it will parse.

Raw audit log output

The raw audit log information includes the details of the VMware Smart Assurance remote API calls made from the Global Console to the server, such as the name of the VMware Smart Assurance user who invoked the call and the time of the invocation. These API calls include calls that interact with the remote object repository in each VMware Smart Assurance server (for example, the API names may be *put*, *insert*, and *invokeOperation*).

The raw audit log output is written to a file named *<server_name>_audit.log* in the *BASEDIR*/*<productsuite>*/smarts/local/logs directory. For example, the raw audit log for an VMware Smart Assurance IP AM server would be *INCHARGE-AM_audit.log which would be located in the BASEDIR /IP/smarts/local/logs* directory. The raw audit log file automatically rolls to a new file when the size of the log file exceeds 1 MG. The size at which the log file will roll is configurable via the SM_AUDIT_CLIENT_LENGTH environment variable. You may set this environmental variable by adding it to the *runcmd_env.sh* file.

In some cases, information in the raw audit log may not correspond to expected higher level functions. In these cases, the raw audit log information is the same as that printed in the final audit log file.

Some of the configuration functions available from the VMware Smart Assurance Global Console are aggregated into XML source data passed from the Global Console to the server. The XML data may contain one or more changes related to different types of configurable entities. In these cases, the VMware Smart Assurance Audit Log Adapter does not interpret the XML data to determine the specific changes made, but instead indicates that one of the set of configurable entities has changed and provides the specific XML data. The final audit log output will indicate that one of these has changed with the pertinent XML source data providing the changed information.

Final audit log output

The raw audit log information often is not easily understandable. Therefore, the VMware Smart Assurance Audit Log Adapter is invoked to process and modify the raw audit log data to make it more user-friendly in the form of a final audit log. The adapter interprets specific remote API calls, and creates audit log messages that describe the higher level functions being invoked by the user. For example, an operation which may be invoked by the user may be the creation of a new discovery filter. The remote API calls to modify discovery filters would be invokeOperation ICF_TopologyManager ICF-TopologyManager applyXMLFilters. The VMware Smart Assurance Audit Log Adapter interprets this invokeOperation call and creates a new audit log message in the final audit log file which more clearly states "Modify Discovery Filters”.

The final audit log output includes the following fields:

- Date, time and time zone (that is, the local time zone of the server)
- Time in seconds from epoch
- VMware Smart Assurance user login name, VMware Smart Assurance system user name and client ID. Final Audit Output Sample provides an example of a complete final audit log entry.
- Action type—indicates the type of action performed, such as "Add Discovery Filter" or "Invoke Operation *xxx*"
- Object class related to action
- Object instance related to action

- Text—includes additional information relevant to the actions, such as a property name, an event name, or arguments to the action

  The format of both the raw and final output files are the same. The fields are separated by tabs, which allows for easy export to Excel spreadsheets.

  For comparison, a sample of raw audit output versus final audit output is provided below in the following figures. Note that in the raw audit log output in Raw Audit Output Sample, all of the actions relating to creating a topology subgroup in the entries with a time in seconds from epoch of 1143150029 and 1143150044 have been modified into clearer and more concise entries in the final log audit output shown in Final Audit Output Sample. Also note that actions such as checking licenses that show up in the raw audit log (the sixth and seventh entries in Raw Audit Output Sample) do not appear in the final audit log.

23-Mar-2006 16:40:29 EST	1143150029	IC Console(admin - millec - 88)	invokeOperation	SelectiveGroup	HGRP-group1	makeChildGroup { subgroup1 }
23-Mar-2006 16:40:29 EST	1143150029	IC Console(admin - millec - 88)	put	SelectiveGroup	HGRP-group1/subgroup1	TargetClass UnitaryComputerSystem
23-Mar-2006 16:40:29 EST	1143150029	IC Console(admin - millec - 88)	invokeOperation	SelectiveGroup	HGRP-group1	changed { }
23-Mar-2006 16:40:29 EST	1143150029	IC Console(admin - millec - 88)	put	SelectiveGroup	HGRP-group1/subgroup1	Description subgroup1 desc
23-Mar-2006 16:40:33 EST	1143150033	IC Console(admin - millec - 89)	invokeOperation	Map_Service	Map-Service	checkBusinessLicense { }
23-Mar-2006 16:40:33 EST	1143150033	IC Console(admin - millec - 89)	invokeOperation	Map_Application	Map-Application	checkLicense { }
23-Mar-2006 16:40:45 EST	1143150044	IC Console(admin - millec - 89)	invokeOperation	SelectiveGroup	HGRP-group2	makeChildGroup { subgroup2 }
23-Mar-2006 16:40:45 EST	1143150044	IC Console(admin - millec - 89)	put	SelectiveGroup	HGRP-group2/subgroup2	TargetClass UnitaryComputerSystem
23-Mar-2006 16:40:45 EST	1143150044	IC Console(admin - millec - 89)	invokeOperation	SelectiveGroup	HGRP-group2	changed { }
23-Mar-2006 16:40:45 EST	1143150044	IC Console(admin - millec - 89)	put	SelectiveGroup	HGRP-group2/subgroup2	Description subgroup2 desc

Figure 4-7. Raw Audit Output Sample

Figure 4-8. Final Audit Output Sample

23-Mar-2006 16:40:29 EST	1143150029	IC Console(admin - millec - 88)	Create Topology Sub Group	SelectiveGroup	HGRP-group1/subgroup1	makeChildGroup UnitaryComputerSystem
23-Mar-2006 16:40:29 EST	1143150029	IC Console(admin - millec - 88)	put	SelectiveGroup	HGRP-group1/subgroup1	Description subgroup1 desc
23-Mar-2006 16:40:45 EST	1143150044	IC Console(admin - millec - 89)	Create Topology Sub Group	SelectiveGroup	HGRP-group2/subgroup2	makeChildGroup UnitaryComputerSystem
23-Mar-2006 16:40:45 EST	1143150044	IC Console(admin - millec - 89)	put	SelectiveGroup	HGRP-group2/subgroup2	Description subgroup2 desc

Adapter behavior after shutdown

The Audit Log Adapter is designed to re-read the raw audit log file if the adapter is shut down and restarted. The adapter maintains the last timestamp processed from its previous execution. When re-reading the raw audit log file, the adapter will process all entries with the timestamp equal to or greater than the last timestamp from the previous execution. In some cases, this may mean that the same entry is processed twice in the final audit log file.

If the raw audit log file rolls over while the Audit Log Adapter is shut down, the adapter will only process entries in the current raw audit log file. Any entries in the previous audit log file (named *<server_name>_audit.log.bak*) are not processed. In this case, you must ensure that the previous raw audit log file is saved to retain information on the changes made during the time when the adapter was shut down.

Configuring the Audit Log Adapter

The Audit Log Adapter uses the *audit_table.conf* configuration file. This configuration file includes the following four tables:

- IgnoreActionTable—This table contains actions which may be included in the raw output but correspond to functions which are not modifications. Actions in this table are not written to the final output file. For example, invokeOperation ICS_ActionManager ICS-ActionManager getClientActions is written to the raw output file. All calls to invokeOperation are written to the raw output file, but some of these calls correspond to operations which do not modify anything. The getClientActions does not modify anything. Therefore, this is included in the IgnoreActionsTable. Anything in this table is not written to the final output file.
- SingleActionTable—This table maps actions from the raw output to the improved text in the final audit log output. This information directs the audit log adapter on how to format the final audit log output.
- ActionSequenceTable—This table includes sequences of actions that correspond to one logical output action.
- MultiInProgressTable—This table is used internally to track multi-action sequences that are currently in progress. It should not be modified.

  An example of the table entries that are displayed in the *audit_table.conf* file is shown in Sample Entries in the audit_table.conf File.

Figure 4-9. Sample Entries in the audit_table.conf File

```
AuditLogging-IgnoreActionTable {
# { "func::class::instance::prop" "" "" "" "" }
  { "createInstance::Map_PhysicalConnectivity::PhysicalConnectivityMap::" "" "" "" "" }
  { "invokeOperation::Map_Service::Map-Service::checkBusinessLicense" "" "" "" "" }
  { "invokeOperation::ICS_ActionManager::ICS-ActionManager::getClientActions" "" "" "" "" }
  { "invokeOperation::ICS_ActionManager::ICS-ActionManager::getObjectActions" "" "" "" "" }
  { "invokeOperation::ICF_TopologyManager::ICF-TopologyManager::getAddAgentHeader" "" "" "" "" }
  { "invokeOperation::AD_Pending::AD-Pending::getPending" "" "" "" "" }
  { "invokeOperation::ICF_TopologyManager::ICF-TopologyManager::getXMLFilters" "" "" "" "" }
  { "invokeOperation::ICF_TopologyManager::ICF-TopologyManager::syncCompleted" "" "" "" "" }
}

AuditLogging-SingleActionTable {
# { "func::class::instance::prop" "outputAction" "outputClass" "outputInstance" "outputProperty" }
  { "invokeOperation::ICF_TopologyManager::ICF-TopologyManager::xmlCreate" "Import Topology" "ICF_TopologyManager" "ICF-TopologyManager" "xmlCreate" }
  { "invokeOperation::ICF_TopologyManager::ICF-TopologyManager::probe" "Rediscover" "ICF_TopologyManager" "ICF-TopologyManager" "probe" }
  { "invokeOperation::ICF_TopologyManager::ICF-TopologyManager::refresh" "Discover All/Pending" "ICF_TopologyManager" "ICF-TopologyManager" "refresh" }
  { "invokeOperation::Configuration::*::makeSetting" "Add New Setting" "Configuration" "*" "makeSetting" }
  { "put::ICF_TopologyManager::ICF-TopologyManager::autoReprobe_pending" "Modify Topology/Discovery Parameters" "ICF_TopologyManager" "ICF-TopologyManager" "put" }
  { "put::ICF_TopologyManager::ICF-TopologyManager::reprobePeriod_pending" "Modify Topology/Discovery Parameters" "ICF_TopologyManager" "ICF-TopologyManager" "put" }
  { "put::ICF_TopologyManager::ICF-TopologyManager::reprobePeriod" "Modify Topology/Discovery Parameters" "ICF_TopologyManager" "ICF-TopologyManager" "put" }
  { "put::ICF_TopologyManager::ICF-TopologyManager::autoReprobe" "Modify Topology/Discovery Parameters" "ICF_TopologyManager" "ICF-TopologyManager" "put" }
}

AuditLogging-ActionSequenceTable {
# { "action-sequence" "total_num_seq" "function" "class::instance::property" "output class::instance::property" }
  { "Create Sub Group-1" "2" "invokeOperation" "SelectiveGroup::*::makeChildGroup" "SelectiveGroup::*::" }
  { "Create Sub Group-2" "2" "put" "SelectiveGroup::*::TargetClass" "SelectiveGroup::*::" }
  { "Create Topology Top Group-1" "4" "invokeOperation" "HierarchicalGroupManager::ICS-HierarchicalGroupManager::makeGroup" "HierarchicalGroupManager::ICS-HierarchicalGroupManager::makeGroup"
  { "Create Topology Top Group-2" "4" "put" "SelectiveGroup::*::DisplayName" "HierarchicalGroupManager::ICS-HierarchicalGroupManager::makeGroup" }
  { "Create Topology Top Group-3" "4" "invokeOperation" "SelectiveGroup::*::changed" "HierarchicalGroupManager::ICS-HierarchicalGroupManager::makeGroup" }
  { "Create Topology Top Group-4" "4" "put" "SelectiveGroup::*::TargetClass" "HierarchicalGroupManager::ICS-HierarchicalGroupManager::makeGroup" }
  { "Create Topology Sub Group-1" "2" "invokeOperation" "SelectiveGroup::*::makeChildGroup" "SelectiveGroup::*::makeChildGroup" }
  { "Create Topology Sub Group-2" "2" "put" "SelectiveGroup::*::TargetClass" "SelectiveGroup::*::makeChildGroup" }
}

AuditLogging-MultiInProgressTable {
# { "user" "action" "last_seq" "total_num" "" }
}
```

ICMP and SNMP polling overview

After the initial discovery and modeling of the network, the IP Manager uses Internet Control Message Protocol (ICMP) and Simple Network Management Protocol (SNMP) polling to monitor the status of the discovered systems. The type of ICMP and SNMP polling that IP Manager uses depends upon the protocol of the discovered systems.

The IP Manager uses ICMP and SNMP to discover and poll IPv4-only, IPv6-only, and dual-stack systems in the managed network environment. The IP Manager supports IPv4 ICMP, IPv6 ICMP, IPv4 SNMP, and IPv6 SNMP. IPv4 ICMP and IPv6 ICMP are also known as ICMPv4 and ICMPv6.

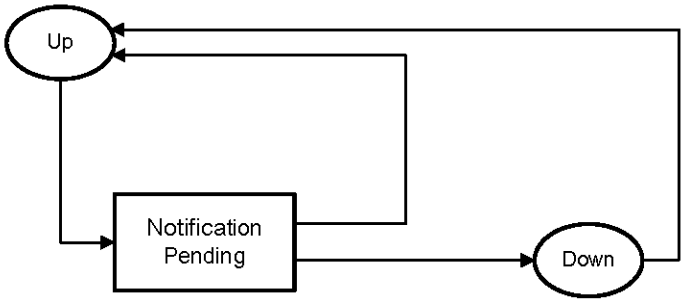
The parameters for controlling ICMP and SNMP polling and for setting thresholds for polled data are accessed through the Polling and Thresholds Console.

ICMP poller

The IP Manager uses a high-performance, asynchronous ICMP poller to send messages for use with IPv4 and IPv6. The ICMP poller uses two asynchronous threads; one thread sends polls, and the other receives polls. Because the send and receive threads operate separately, slow response times or excessive timeouts do not affect the polling rate. Thus, the ICMP poller performs at a consistent polling rate.

Possible states of an object during an analysis polling cycle shows the three possible states of a monitored object as determined by its response to an ICMP poll. The states are Up, Notification Pending, and Down.

Figure 4-10. Possible states of an object during an analysis polling cycle



An object is in the up state until it fails to respond to an ICMP poll. When it fails to respond, the status of the object changes to the Notification Pending state and stays in that state until the IP Manager can determine whether the object is Up or Down.

If the object responds to an ICMP poll before the maximum failure retry count is exceeded for the object, the status of the object changes back to the Up state.

If the maximum failure retry count is exceeded for the object, the status of the object changes to the Down state, and the IP Manager does not poll the object again until the next scheduled polling cycle. An object stays in the Down state until it responds to an ICMP poll. When the object responds, the IP Manager changes the status of the object to the Up state.

This polling sequence is performed for each IP endpoint that is managed.

SNMP poller

The IP Manager uses a synchronous, multithreaded SNMP poller to send messages for use over IPv4 and IPv6. By default, the SNMP poller uses 10 synchronous polling threads.

The SNMP poller fully supports the SNMPv1 and v2c protocols, and supports authentication and data encryption for the SNMPv3 protocol. With SNMPv1, the correlation model uses 32-bit counters in its correlation analysis. With SNMPv2c or v3, the correlation model uses high-capacity 64-bit counters in its correlation analysis. Using 64-bit counters is critical for performance analysis of high-speed data links because using 32-bit counters might result in wrapping (overflow) of the counters between polls.

Polling for devices with multiple IP addresses is supported because the SNMP poller supports multiple IP addresses for each SNMP agent. The SNMP poller automatically switches to an alternate IP address during failures, thereby ensuring the integrity of the IP Manager correlation analysis during an outage.

Just-in-time polling

The SNMP poller's MIB variable poll list is driven by a Just-In-Time polling algorithm, which ensures that only those MIB variables needed for correlation are polled. For example, if a port monitored for performance data is disabled, or goes down, the SNMP poller automatically removes the relevant MIB variables from the poll list. If the port is reenabled, or comes back up, the variables are automatically put back onto the MIB poll list.

Request-consolidation polling

Issuing a single SNMP GET request that requests 10 variables is more efficient than issuing 10 GET requests each requesting a single variable. The SNMP poller consolidates as many variables as possible into a single SNMP GET request. The consolidation is not restricted to variables from the same SNMP table. Polling consolidation continually adapts to changes in the MIB variable poll list.

Upon encountering a nonfatal error during polling consolidation, the SNMP poller responds differently to an SNMPv1 agent than to an SNMPv2c or v3 agent for the following reason: Where an SNMPv1 agent stops processing a request upon encountering an error, an SNMPv2c or v3 agent continues processing a request upon encountering an error. An SNMPv2c or v3 agent handles errors on a per-OID basis.

If a nonfatal error is encountered by an SNMPv1 agent during a GET request seeking multiple variables, the SNMP poller suspends the polling of the affected variable. It suspends polling to prevent reduced performance of the SNMPv1 agent, since continuous polling of the affected variable would require the resending of the remainder of the request after receiving the error. The SNMP poller continues to poll the unaffected variables. (An example of an affected variable is one that has become unavailable due to a configuration change.) This behavior enables the SNMP poller to operate efficiently with an SNMPv1 agent during unexpected changes to a device's configuration.

In contrast, if a nonfatal error is encountered by an SNMPv2c or v3 agent during a GET request seeking multiple variables, the SNMP poller continues the polling of the affected variable as well as the unaffected variables.

ICMP and SNMP polling coordination

The IP Manager links its ICMP and SNMP pollers so that the SNMP poller does not send requests to any SNMP agent address that the ICMP poller determines to be unreachable. IP addresses that are unresponsive to ICMP polls are added to a "do not poll" list, which the SNMP poller checks before sending SNMP requests.

The checking process is as follows:

- If an SNMP agent address is on the "do not poll" list, the SNMP poller does not send a request to that address.
- If an SNMP agent has multiple IP addresses, the SNMP poller checks for each address in the "do not poll" list.
  - For an address that does not appear in the list, the SNMP poller sends a request to that address.
  - If all addresses for an agent are on the list, the SNMP poller deems the agent as unreachable and temporarily suspends sending SNMP requests to that agent.

    As soon as an agent's IP address becomes responsive, as determined by the ICMP poller, the address is removed from the list, and SNMP polling resumes.

Federal Information Processing Standard (FIPS) 140

The Federal Information Processing Standard (FIPS) Publication 140-2 is a U.S. government computer security standard governing cryptographic modules and is required for any software purchased by the US government and US military.

Open-source derived cryptographic algorithms implemented in DMT's C/C++ code base were removed and replaced with RSA's BSAFE Micro Edition Suite (MES).

RSA's BSAFE MES is a library that implements algorithms and provides both FIPS-validated and non-FIPS implementations of the cryptographic algorithms. The non-FIPS implementations are used by default. When placed in FIPS 140 mode, DMT causes MES to use the FIPS 140 validated implementations.

FIPS 140 mode allows you to use only SNMPv1, SNMPv2c, and SNMPv3 except MD5 and DES authentication protocols. If you run SNMPv1 or SNMPv2c, then FIPS 140 mode has no impact. If you are using encryption, then you must use a certified encryption library and only certain encryption routines.

When you discover an SNMPv3 device, you need to select the option “V3” in the “Add Agent” window. The “Authentication Protocol” option lists only SHA and not MD5 and the “Privacy Protocol” option lists only AES and not DES. This is because MD5 and DES are not supported in FIPS 140 mode. If you discover SNMPv3 devices with MD5 and DES protocol as seed, discovery fails and the devices go to the Pending List and display as “Invalid” or “Unsupported SNMP V3 protocol”. ASL error exception messages are also observed in the IP server logs.

“Support for FIPS 140-2 in IP 9.1” section in Chapter 3, Installing IP Manager, in the VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers, provides more information about the FIPS 140 mode implementation.

Discovery

5

This chapter includes the following topics:

- Discovery process overview
- Network system support
- Light discovery
- ICMP, SNMP, ASNMP, and IPv6 support
- Support for HSRP and VRRP groups
- Support for NAS gateways
- Support for NIC teaming
- Support for IPSec and IKE tunnels
- Support for overlapping IP addresses
- Support for SNMP v3 auto discovery
- Support for Wireless LAN
- Support for Virtual Port Channel
- Link Aggregation and EtherChannel discovery
- Support for Virtual Data Context
- DASL integration
- Virtual bridging
- Incremental reconfigure
- Default IP route
- Asynchronous SNMP (ASNMP)
- Link Layer Discovery Protocol (LLDP)
- Unique VLAN naming
- OID Info Tool
- ESX Server Certification

- Cisco Nexus 1000V implementation
- Cisco Unified Computing System Certification
- AggregateLink representation in the Map console
- Discovery and the VPN-Tagging Server
- About seed files
- Seed file format
- Seed file examples
- Which systems to put in a seed file
- Discovery methods
- Discovery phases
- Autodiscovery data sources
- Discovery of NIC teaming
- Discovery and short discovery of IPSec tunnels
- Discovery and the local name resolution service
- Hostname resolution and IP address checking
- How discovery names a system
- Configuring automatic name resolution
- Preparing for manual discovery
- Initiating manual or autodiscovery
- Initiating postprocessing
- About discovery filters
- Discovery filter fields
- Discovery filter considerations
- Discovery filter matching
- Examples of discovery filters
- Discovery results
- Discovery error scenarios and results
- Discovery errors
- DiscoveryError notifications
- Pending Devices list
- About certifying managed devices
- Discovery hook scripts

- Automatic and manual discovery
- Discovery methods
- Discovery considerations
- Scheduling automatic discovery
- Read community strings
- About managing and unmanaging topology objects
- Handling duplicate IP address errors
- Redundancy groups
- ICMP
- SNMP
- About ICMP and SNMP polling parameters

Discovery process overview

During per-device discovery, the IP Manager sends ICMP and SNMP polls to the network devices to discover the network topology. The IP Manager uses VMware Smart Assurance object classes to create a data-model representation of the discovered topology. The data model includes representations of the managed network devices and their internal components.

During postprocessing, the IP Manager uses VMware Smart Assurance object classes to create representations of the physical and logical connections between the managed network devices. The IP Manager also establishes device relationships.

The data model that is supported by the IP Manager is described in the VMware Smart Assurance IP Manager Reference Guide

Note Throughout this chapter and the chapters that follow, the terms “system” and “device” are used interchangeably. Usually, “system” is used.

The configurations and licenses that are applied to an IP Manager at startup determine which discovery and analysis will run. By default, discovery and analysis for both IP Availability Manager and IP Performance Manager will run.

“Components of a typical IP Manager deployment” on page 36 identifies the elements of a typical IP Manager deployment and shows the flow of information among the elements.

The IP Manager discovers Layer 2 (data link) and Layer 3 (network) connectivity in multivendor, switched, and routed networks. Layer 2 network systems use machine addresses to transmit packets from network node to network node, and Layer 3 network systems use network addresses to route data to different LANs and WANs. The IP Manager discovers the network systems by sending them ICMP and SNMP polls.

In addition, depending on the configuration, IP Availability Manager also sends network topology to the VMware Smart Assurance Service Assurance Manager Adapter Platform (Adapter Platform) and to satellite Domain Managers.

Running separate instances of IP Availability Manager and IP Performance Manager is necessary if the two applications are running on different hosts. When running on the same host, a more common configuration would be IP Availability Manager and IP Performance Manager running as a single process.

After building a data model of the discovered network topology, IP Availability Manager, IP Performance Manager, and IP Server Performance Manager use SNMP polling to monitor the topology. The results of this polling, in conjunction with the received SNMP traps, are used by:

- IP Availability Manager to diagnose the failed network objects that interrupt network connectivity.
- IP Performance Manager and IP Server Performance Manager to diagnose emerging problems in the service delivery infrastructure.

  IP Availability Manager, IP Performance Manager, and IP Server Performance Manager export their topology data and analysis results (events) to the VMware Smart Assurance Service Assurance Manager (Global Manager).

  Whenever discovery completes, the IP Manager prints a report to its log file identifying the number of network objects that are created during the discovery. Naming of the log depends on the locale and encoding set by the user and is controlled by the SM_LOCALE and SM_ENCODING environment variables. The log file is named <IP Manager instance name>_<locale>_<character encoding>.log (for example, INCHARGE-AM_en_US_UTF-8.log) and is located in the BASEDIR/smarts/local/logs directory in the IP Manager installation area.

Network system support

The IP Manager discovers and models the following types of network systems:

- Bridges
- Hosts
- Hubs
- Nodes
- Probes
- Routers
- Switches
- Terminal servers

  The VMware Smart Assurance IP Manager Reference Guide provides information about the classes that are used by the IP Manager to model these systems.

Fully tested devices

The IP Manager supports the discovery, monitoring, and analysis of IP networks in many vendor-specific environments. The *VMware Smart Assurance IP Certification Matrix* identifies the devices that have been fully tested for this release of IP Manager. This document is found on VMware Online Support at: docs.vmware.com.

Routing-device types

The following routing-device types are supported by the IP Manager:

- Physical routers
- Virtual routers
- Route switch modules (RSMs)
- Route switch feature cards (RSFCs)
- Multilayer switch feature cards (MSFCs)

  Virtual routers are implemented within physical routers or switches.

  RSMs, RSFCs, and MSFCs are routing-device type cards in Layer 3 (network) switches. The cards perform routing between virtual LANs (VLANs).

  Layer 3 switch devices that house routing-device type cards are called hybrids.

Light discovery

In addition to performing its traditional discovery of network objects, the IP Manager, as shown in Exporting topology collection sets to satellite Domain Managers, can be configured to create topology collection sets from the discovered network objects, one set for each of the satellite Domain Managers that are supported by the IP Manager. The creation of topology collection sets is known as “IP Manager light discovery” or simply “light discovery.”

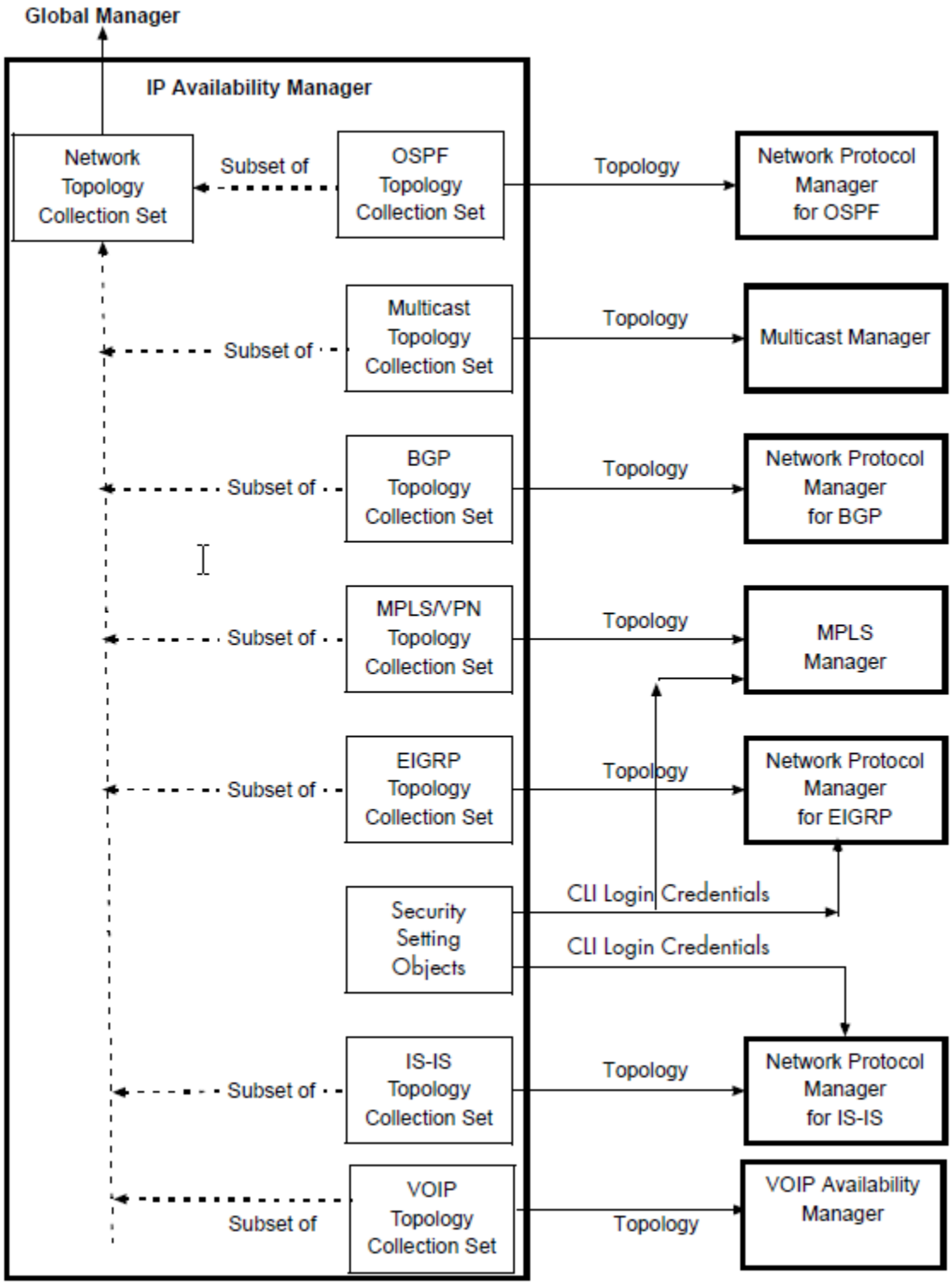


Figure 5-1. Exporting topology collection sets to satellite Domain Manager s

The term “satellite,” as used here, refers to any of the VMware Smart Assurance Domain Managers that import topology and topology updates from the IP Manager.

Topology import

From the IP Manager, a satellite Domain Manager not only receives its initial topology and subsequent topology updates, but it also receives Layer 1 (physical) and Layer 2 (data link) status updates that are relevant to the imported topology. As an example of the types of objects that are imported from the IP Manager, Network Protocol Manager for EIGRP imports router and switch objects along with the following objects that are associated with the routers or switches:

- Card
- Interface
- NetworkConnection
  - Cable
  - TrunkCable
- IP
  - DuplicateIP
- IPNetwork
- SNMPAgent
  - VRAgent

Note The indentations in the bullet list indicate class hierarchy. The VMware Smart Assurance IP Manager Reference Guide provides descriptions of these object types.

Although all satellite Domain Managers import topology and topology updates from the IP Manager, not all import topology collection sets. That is, some satellite Domain Managers use a means other than light discovery to import topology and topology updates from the IP Manager. The individual configuration guides for the satellite Domain Managers state which Domain Managers import topology collection sets.

By default, light discovery is disabled.

The VMware Smart Assurance IP Manager User Guide provides instructions on configuring the IP Manager to create topology collection sets.

CLI device-access object import

Network Protocol Manager for EIGRP, Network Protocol Manager for IS-IS, and Multiprotocol Label Switching (MPLS Manager) also import one or more of the following types of CLI device-access objects from the IP Manager:

- Configuration
- ICF_Criteria

- SelectiveConfiguration
- CLI_AccessSetting
- ICF_ConfigurationSelector

  The CLI device-access objects carry the login credentials and other criteria that are needed by a satellite Domain Manager to access devices in the managed network for the purpose of invoking CLI commands from the devices.

  The VMware Smart Assurance IP Manager User Guide provides information about creating CLI device-access objects.

ICMP, SNMP, ASNMP, and IPv6 support

The IP Manager uses Internet Control Message Protocol (ICMP) and Simple Network Management Protocol (SNMP) to discover IPv4-only, IPv6-only, and dual-stack systems in the managed network environment. A dual-stack system is configured with both IPv4 (32-bit) and IPv6 (128-bit) addresses.

The IP Manager supports IPv4 ICMP (also known as ICMPv4), IPv6 ICMP (also known as ICMPv6), IPv4 SNMP, and IPv6 SNMP. Because many Management Information Bases (MIBs) have not yet been modified to accommodate IPv6 addresses, certain IP Manager features are not available to IPv6.

Note Starting from IP 8.1.1, IPv6 addresses on a device that are link-local (FE80::), site-local (FEC0::), multicast (FF00::), or loopback (::1) do not get discovered and are not added to the topology. For example, a valid link-local IPv6 address FE80:0:0:0:20F:35FF:FEC1:2400, will not be discovered.

The IP Manager also supports SNMPv1, v2c, and v3.

The VMware Smart Assurance IP Manager User Guide provides information about ICMP, SNMP, and Asynchronous SNMP (ASNMP). Unless specified otherwise, the term ICMP is used to refer to IPv4 and IPv6 ICMP polling, and the term SNMP is used to refer to IPv4 and IPv6 SNMP polling.

Support for HSRP and VRRP groups

The IP Manager is able to discover and monitor redundancy groups of routers that are involved in Hot Standby Router Protocol (HSRP) or Virtual Router Redundancy Protocol (VRRP) groups. It discovers HSRP groups, HSRP router endpoints, VRRP groups, and VRRP router endpoints, and represents them as instances of the HSRPGroup, HSRPEndpoint, VRRPGroup, and VRRPEndpoint classes, respectively.

The VMware Smart Assurance IP Manager Reference Guide provides descriptions of these object types.

Support for NAS gateways

The IP Manager discovers and monitors VMware Celerra network-attached storage (NAS) gateways and NAS connectivity problems. It discovers NAS chassis, Control Stations, and Data Movers, and represents them as instances of the NASChassis, Host, and FileServer classes, respectively.

In IP 9.2, once the Control Station is discovered, by default, it will automatically move the File servers attached to it to the Pending List.

Note Monitoring of File System of File Servers and DataMovers are not supported.

The VMware Smart Assurance IP Manager Reference Guide provides descriptions of these object types.

Support for NIC teaming

The IP Manager discovers and monitors network interface card (NIC) teaming between hosts and switches. NIC teaming is the grouping of two or more physical NICs into a single logical NIC, for the purpose of achieving network fault tolerance or network traffic load balancing or both. NIC teaming is described in Discovery of NIC teaming.

Note Setting up NIC teaming is device and vendor specific. Vendor specific documentation on how to set it up in a customer environment with different vendors is not provided in the manuals. For any device or vendor specific setup information, it is recommended to refer to the relevant vendor specific documentation.

Support for IPSec and IKE tunnels

The IP Manager is able to discover and monitor IPSec tunnels and Internet Key Exchange (IKE) tunnels between IPSec-enabled routers on IPv4 networks. IPSec is a standardized framework for securing IP communications at the network layer, and IKE is a standardized key exchange protocol that is used to set up IPSec tunnels. Tunnel discovery is described in Discovery and short discovery of IPSec tunnels.

Support for overlapping IP addresses

The IP Manager is able to discover and create tagged IP objects for overlapping IPv4 addresses, which are IP addresses that are reused and configured across multiple network devices in the same managed network environment.

The VMware Smart Assurance IP Manager User Guide describes the IP tagging feature and provides instructions on configuring IP tagging and creating IP tag filters. The VMware Smart Assurance IP Manager Reference Guide identifies the configuration parameters that control IP tagging.

Support for SNMP v3 auto discovery

IP Manager has been enhanced to support auto discovery of devices in an IP network using SNMP v3 polling mechanism. With the support for SNMP v3 polling mechanism, you can discover and monitor those devices in the network that supports SNMP v3. This has been facilitated by the following changes in the IP Domain Manager Administration Console user interface:

- Enhancement to Discovery Filters tab
- Enhancement to Topology tab

- Supported Encryption Standards

  The VMware Smart Assurance IP Manager User Guide provides more information on the above enhancements including supported encryption standards.

Support for Wireless LAN

IP supports discovery and monitoring a wireless LAN. As part of wireless LAN discovery and monitoring, IP discovers Wireless Controllers, WirelessAccessPoints, Wireless Clients and Authentication server. An authentication server provides authentication service for different network elements such as Wireless Clients.

The VMware Smart Assurance IP Manager Reference Guide provides more information on topology classes and root-case analysis scenarios for Wireless LAN.

The Wireless certification is currently supported for two vendors, namely Cisco and Aruba. The certification of the two vendors can be found in the following conf files:

- oid2Type_Cisco.conf in <*BASEDIR*>\IP\smarts\conf\discovery
- oid2Type_Aruba.conf in <*BASEDIR*>\IP\smarts\conf\discovery

Note IP server does not generate any alert if the status of the authentication server is UNKNOWN. So, if the WirelessController is Down, the related authentication server status is UNKNOWN. Therefore, to get the proper status of the authentication server, rediscover any other running WirelessContoller that is serviced by the same authentication server.

Wireless LAN short discovery

Short discovery handles wireless topology changes, which typically occur during WirelessClient movement. In addition, it dynamically detects Rogue Wireless Access Points.

Wireless LAN settings lists the flags that determine the settings for Wireless LAN discovery.

Table 5-1. Wireless LAN settings

Flag	Allowed value	Description	File
WIFI_autoReprobe_Short **Note** Server restart is required.	Default value: FALSE ■ TRUE — Enables WIFI related short discovery. ■ FALSE— Disables WIFI related short discovery.	Enables or disables WIFI related short discovery.	discovery.conf
WIFI_reprobePeriod_short	Default value:900 seconds **Note** Value should be positive and greater than 300.	Sets the interval between successive WIFI short discoveries. **Note** Decreasing the discovery interval will increase the CPU usage. If you decrease the short discovery interval, use small decrements and monitor your current CPU usage.	tpmgr-param.conf

Table 5-1. Wireless LAN settings (Continued)

Flag	Allowed value	Description	File
EnableRogueAPShortDiscovery	Default value: TRUE	Determines the discovery of Rogue AP. Disabling this flag will stop the short discovery of Rogue AP.	tpmgr-param.conf
EnableWirelessClientDiscovery	Default value: TRUE	Determines the discovery of the WirelessClient.	tpmgr-param.conf
EnableWirelessClientShortDiscovery	Default value: TRUE	Determines the short discovery of the WirelessClient	tpmgr-param.conf

Wireless license

Note The wireless LAN management is a licensed feature in IP 9.1.

The wireless license determines the discovery of wireless network components, that is, Wireless Controller and Wireless Access Points. The basic wireless feature is enabled by the license key AP_WIFI.

Even if a customer has the basic AP_WIFI license, the number of Wireless Access Points or WirelessController discovery is limited by the appropriate license count.

- WirelessAccessPoint — The number of WirelessAccessPoint to be discovered is controlled by the count of the license key AP_AP_CNT.
- WirelessController — The number of WirelessController to be discovered is controlled by the count of the license key AP_CTL_CNT.

  Note The discovery of WirelessClient is not controlled by license counts.

  If you do not have a wireless license, you can only discover Wireless Controllers and Thick Wireless Access Points with older IP certification. Discovery will not include Thin Wireless Access Points and Wireless Clients.

  If the license block for System volume or WirelessAccessPoint or WirelessController exceeds the license count, then a new license file with more license blocks should be obtained.

  Restart the license server with new license file.

  Restart the the dependent IP servers.

Wireless LAN map

A wireless LAN map displays a wireless LAN map. The details are explained in Descripiton of a WIFI LAN map.

Figure 5-2. A wireless LAN map

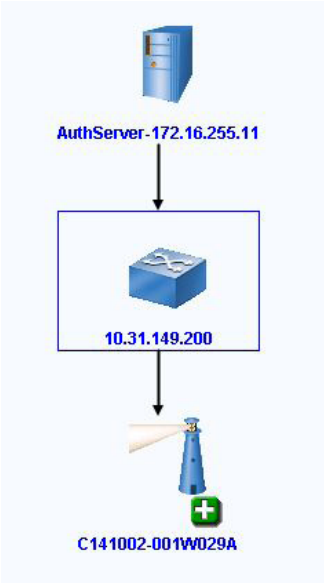


Table 5-2. Descripiton of a WIFI LAN map

Icons	Representation	Relationships
	WirelessAccessPoint	■ *ControlledBy* relationship with WirelessController ■ *Hosts* relationship with WirelessClients **Note** Click on the + icon to view the WirelessClients that are hosted by this WirelessAccessPoint.
	WirelessController	■ *Controls* relationship with WirelessAccessPoint ■ *HostsStations* relationship with WirelessClient
	AuthenticationServer	*AuthenticationServiceTo* relationship with the WirelessController.

Support for Virtual Port Channel

IP 9.1 supports Virtual Port Channel (vPC) management on Cisco Nexus 5000 switches. A Virtual Port Channel is formed by the logical aggregation of ethernet switch ports that belong to two different switches.

The Virtual Port Channel is represented as vAggregatePort in IP topology.

Note IP 9.1 supports vPC on Cisco Nexus 5000 switches through CLI discovery. Provide read-only credentials of the Cisco Nexus 5000 switch in CLI Access Group with matching criteria in Polling and Threshold console > Device Access.

The *vAggregatePort* class is defined by the following relationships:

- *LayeredOver/Underlyin*g relationship with Switches.
- *ComposedOf/PartOf* relationship with underlying AggregatePorts (vPC member ports).
- *Uses/UsedBy* relationship with AggregatePorts participating in peer link.

Enabling vPC discovery

The flag *EnableVPCDiscovery* controls vPC discovery. By default, vPC discovery is enabled in IP 9.1.

To disable vPC discovery, set the value of the flag *EnableVPCDiscovery* in <*BASEDIR*>/conf/tpmgr-param.conf file.

```
EnableVPCDiscovery= FALSE
```

For information on the events and root-cause analysis of vAggregatePort, refer to VMware Smart Assurance IP Manager Reference Guide.

Viewing vPC information

To view the information about the virtual port channels (vPCs), run the command on the Nexus switch on which vPC is configured:

```
show vpc brief
```

The command displays the following information:

- vPC domain ID
- Peer-link status
- Keepalive message status
- Configuration consistency status
- Number of vPC confgured

Note This command is available only if the vPC feature is enabled.

Sample output of the command

```
switch(config)# show vpc brief
Legend:
                (*) - local vpc is down, forwarding via vPC peer-link
vPC domain id                   : 10
Peer status                     : peer adjacency formed ok
vPC keep-alive status           : peer is alive
Configuration consistency status: success
vPC role                        : primary
```

```
Number of vPC configured        : 1
vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po10   up     1-100
vPC status
----------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- -------------------------- ------------
20   Po20   up     success     success                    1-100
```

Link Aggregation and EtherChannel discovery

IP 9.1 has been enhanced with functionality to manage link aggregation, which is recognized by industry as Etherchannel or Portchannel. This means that IP server recognizes multiple connections between two devices as a single consolidated connection.

Link Aggregation allows one or more ports to be aggregated together to form a Link Aggregation Group (LAG), such that a MAC Client can treat the Link Aggregation Group as if it were a single link.

Note Link Aggregation Group (LAG) discovery is supported on Cisco Nexus 5000 switches through CLI discovery. LAG can be formed using LACP / PAGP (Cisco proprietary) signalling protocol. Both LACP and PAGP protocols are supported by IP.

This feature allows:

- Detection of failure of logical groups. When there is a signal failure or a failure in all of the discrete connections, then a single **Down** problem is reported on the LAG.
- Discovery of peer port connectivity and Logical Group connectivity in Cisco devices.

  Note This is true even when the neighbor discovery protocol is not enabled.

- Detection of traffic load imbalance among ports that participate in an aggregation. Generates the event *UnbalancedAdapterUtilization* for AggregatePort when traffic load is relatively imbalanced.
- Chapter 1 in VMware Smart Assurance IP Manager Reference Guide provides more information on how the this event is calculated.

Representation of Etherchannel

The discovered etherchannel connections are represented by the following classes:

- AggregatePort — This class represents the etherchannel adapter in IP Availability Manager. It is defined as the logical grouping of network adapters.

- AggregateLink — This class represents the logical links between AggregatePorts. During post processing, all the AggregatePorts instances are retrieved from the discovered topology. If the constituents ports of the AggregatePort are connected and the peer ports of those constituent ports are associated with AggregatePort, then AggregateLink is created between those two AggregatePort.

  The VMware Smart Assurance IP Manager Reference Guide and the VMware Smart Assurance IP Manager Release Notes provide more information.

Identifying AggregationPorts through CLI

You can identify AggregatePorts and figure out ports that are part of an aggregation by running the CLI command:

```
show port-channel summary
```

Chapter 3 in VMware Smart Assurance IP Manager User Guide provides more information.

Support for Virtual Data Context

IP 9.1 supports the Virtual Data Context (VDC) feature in a Cisco Nexus switch. This feature allows a Cisco Nexus 7000 series switch to be logically segmented into four different virtual switches or device contexts.

Enabling VDC discovery

The flag EnableVirtualDataContextDiscovery in tpmgr-param.conf file controls VDC discovery.

Default Value=TRUE

Disabling this flag will stop the Virtual Data Context (VDC) discovery in Nexus 7000 switches.

Note Provide the read-only or Administrator VDC credentials under the CLI Access Group in Polling and Threshold > Device Access.

Viewing VDC information

While discovering the default VDC, information about other VDCs created by the default VDC can be viewed by running the CLI command show vdc.

A sample output of the command:

```
Router# show vdc
vdc_id  vdc_name                          state               mac
------  --------                          -----               ----------
1       switch                            active              00:18:ba:d8:3f:fd
2       Payroll                           active              00:18:ba:d8:3f:fe
3       RSUDAM2                           active              8c:b6:4f:e9:f3:42
4       RNORM6                            active              10:8c:cf:1d:fd:42
```

It gives the MAC address of individual VDC belong to the Nexus 7k Switch. This is used to update default VDC *ComposedOf* relation with the other VDCs that are created.

Note If the default VDC goes down, then the other VDCs are also impacted. However, IP 9.1 generates only one **Down** root cause instead of four individual VDC Down problems.

DASL integration

IP 9.1 has been enhanced to integrate the Device Access Scripting Language (DASL) framework to facilitate communication with devices with XML API and CLI interfaces.

Note DASL framework is not supported on Solaris. It is applicable only RHEL 5, and RHEL 6 platforms.

DASL is used for creating device-independent access methods. DASL framework integration allows IP to provide multiple-vendor support quickly and easily. It is designed to encapsulate all device-specific information in a highly modifiable manner, allowing the main products to remain device-independent. The access methods include:

- Telnet
- SSH
- HTTP
- HTTPS

CLI interface

The CLI framework provides IP server the capability to communicate with devices with CLI interfaces. For all communications with the corresponding devices, it uses Telnet or SSH. The devices could be any UnitaryComputerSystem with Telnet or SSH enabled. The commands are executed on the devices as line by line text, and the response processed as required in the IP server. The framework provides an excellent means of session caching as it improves performance by avoiding the building up of connection every time.

Prerequisites:

- The devices should be certified.
- Configure device access credential in the group CLI Access Group in IP Domain Manager Administrator > Polling and Threshold > Device Access tab as shown in CLI interface.

New features using CLI

DASL CLI is used for LAG discovery on Nexus 5000, Nexus 6000, and Nexus 7000 switches. It is also used for VPC discovery on Nexus 5000 and VDC discovery on Nexus 7000.

Note In the 9.1 release, DASL CLI is used only for discovery and not for monitoring.

XML interface

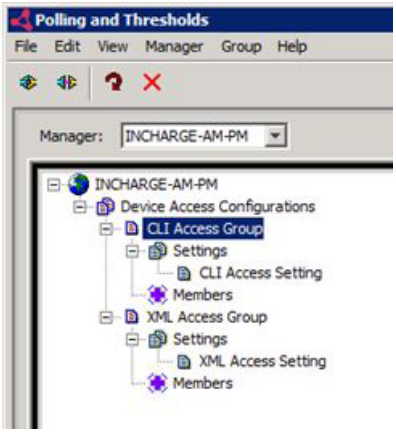
The XML framework of DASL provides IP Server the capability to communicate with and manage devices which expose XML interfaces. XML data are exchanged between the IP server and the corresponding devices using HTTP/HTTPS and by executing GET/POST commands.

Note In the 9.1 release, only the XML framework is available and is not used for implementing any feature.

Prerequisites:

- The device should be certified.
- An XML driver should be associated with it.
- Configure device access credential in the group XML Access Group in IP Domain Manager Administrator > Polling and Threshold > Device Access tab as shown in Settings group for Device Access Configuration.

Figure 5-3. Settings group for Device Access Configuration



Polling and Threshold settings

The security setting groups XML AccessGroup and CLI Access Group are added to Polling and Threshold > Device Access tab.

You can add any number of groups based on your requirement and associate them with any one of the following Settings group:

- CLI Access Group — This is a security settings group meant for CLI discovery. The settings CLI_AccessSetting is included which the user has to change when configuring device credentials for this group.

  Note For the Name attribute of this group, the Matching Criteria has been changed to ~*. This is done to avoid configuring empty settings for devices by default.

- XML Access Group — This is a security settings group meant for XML Discovery. The mechanism option is either HTTP or HTTPS. The settings XML_AccessSetting is included which the user has to change when configuring device credentials for this group.

Changes in folders and files in the IP installation directory

The following additional folders and files have been added to the IP installation directory:

- smarts/dasl — This folder contains all framework files related to the DASL framework and not expected to be edited by the end user. The files include XML, XSD, dasl, dlg, inc, and so on.
- smarts/conf/dasl — This folder includes the configuration files related for the DASL framework support in IP. The files include .import files, .conf files, and so on.
- smarts/rules/dasl — This folder includes .asl files required for initialization and discovery of objects related to CLI or XML discovery.
- smarts/local/logs/dasl — This folder contains all the logs related to the DASL framework. There will be one log for each server started, that is, one DASL log for each server started.

  Note DASL folders <BASEDIR>/smarts/dasl/logs and <BASEDIR>/smarts/dasl/local/logs/dasl are created as part of the installation even though DASL is not supported on Solaris.

Virtual bridging

In earlier releases, the bridges relationship was between MACs and the switch port. The port bridges MAC relationship is made regardless of the type of the MAC, physical or virtual. This resulted in a connection creation issue for the Hosts with ESX Server installed connected to a switch and routers with HSRP/VRRP configured connected to the switch.

In this release, Port instead of Bridges forms a new relationship called VirtualBridges with virtual MACs. During discovery, if a switch port is associated with a MAC address that matches the VirtualMACPattern configured in the tpmgr_param.conf file, a VirtualBridges relationship is created for the switch port. The VirtualBridges relation is also formed for Systemless MAC irrespective of the VirtualMACPattern.

Systemless MAC will have virtualbridges relationship with the port. The default VirtualMACPattern has the VMware specific MACs, VRRP, and HSRP MACs. By removing the VirtualMACPattern from the tpmgr-param.conf file, the Bridges relationship gets established.

The VMware Smart Assurance IP Manager User Guide provides more information on VirtualMACs.

Incremental reconfigure

After the initial discovery was completed, instead of a bulk reconfigure or if new devices were added to the topology, instead of a full reconfiguration, an incremental reconfigure happens only for newly added devices, which is faster.

The VMware Smart Assurance IP Manager User Guide provides information on how to enable incremental reconfigure.

Incremental Reconfigure Log Messages

```
        Incremental Reconfigure log messages with DebugReconfigure set to TRUE in tpmgr-param.conf
Discovered 100 devices
[December 16, 2010 2:44:37 PM GMT+05:30 +569ms] t@1319967040 ProbeCompletedRunner
ASL_MSG-*-ASLP-icf-r/iprm-reconfigure.asl: iprm-reconfigure.asl minNumberOfUCS from tpmgr-param.conf =
10
[December 16, 2010 2:44:37 PM GMT+05:30 +569ms] t@1319967040 ProbeCompletedRunner
ASL_MSG-*-ASLP-icf-r/iprm-reconfigure.asl: iprm-reconfigure.asl Triggering BULKreconfigure
[December 16, 2010 2:44:37 PM GMT+05:30 +571ms] t@1317865792
ICF_MSG-*-ICF_RECONFIGURESTART-PolicyManager: reconfigure started
[December 16, 2010 2:45:56 PM GMT+05:30 +039ms] t@1317865792
ICF_MSG-*-ICF_RECONFIGUREFINISH-PolicyManager: reconfigure finished
====
Added one device incrementally
ASL_MSG-*-ASLP-icf-r/iprm-reconfigure.asl: iprm-reconfigure.asl Triggering incremental reconfigure
[December 16, 2010 3:08:52 PM GMT+05:30 +400ms] t@1319967040 ProbeCompletedRunner
ASL_MSG-*-ASLP-icf-r/iprm-reconfigure.asl: Configuring KD014014014001.au-net.ne.jp
[December 16, 2010 3:08:52 PM GMT+05:30 +404ms] t@1319967040 ProbeCompletedRunner
ASL_MSG-*-ASLP-icf-r/iprm-device-reconfigure.asl: ===== Reconfiguring Router::KD014014014001.au-
net.ne.jp =========
```

Default IP route

Default IP routes are advertised by border routers to its peers. So this allows peers to use the advertised router as a next hop for destinations which are not listed in the routing table. This feature allows routers which are not able to maintain a huge list of routing entries in their routing table to use the default traffic to the advertised border router. You do not have to configure the static routes on the routers which are not able to maintain the routing table entries.

The IP Manager monitors the default route entry on the border router, that is, the border routers are polled to check only the default route entry and not the entire routing table, and when the default route entry is missing, the IP Manager generates an event known as *MissingDefaultRoute*. This reduces the amount of SNMP request processing that the router has to perform.

The OID which is used for this feature is *ipRouteDest-1.3.6.1.2.1.4.21.1.1*. Normally the default IP route is 0.0.0.0 which gets advertised using the update message by either the border router speaker to all its peers. The update message also contains the next hop attribute which is the IP address of the border router. That is how the peers know about the border router speaker.

Asynchronous SNMP (ASNMP)

Currently, SNMP discoveries are executed through get() and getBulk() calls for low volume exchanges, but it is inefficient when large volumes of data are fetched or when communicating with remote devices. The ASNMP feature allows you to process responses more quickly by packing more requests in one SNMP packet and by handling the code in a faster language. ASNMP is faster than Synchronous SNMP since requests are fired without waiting for a response.

In earlier releases, for any ASNMP request failure, the IP Manager displayed the following error message in the “Description” field of the pending list (AD_PendingData):

```
SNMP request timed out or had an error
```

From IP 8.x onwards, by default, ASNMP_Control (Asynchronous SNMP requests) was used. The configurable parameter DoNotUseASNMPPattern in the tpmgr-param.conf file allows you to disable asynchronous requests only for specific devices by SysOID. Add the list of sysOID entries for which you can see one of the following or both the error messages in the “Description” field of the pending list (AD_PendingData) when an ASNMP error occurs during discovery:

```
"ASNM-E-ASNMP_SEND_RECEIVE_ERROR-ASNMP request failure: Error when sending or
receiving SNMP data for Host '<IP address of the device>' (Maybe bad credentials)."
"ASNM-E-ASNMP_SNMP_ERROR-ASNMP request failure: Error <Error Code>:
<Error Message> when receiving SNMP Packet from '<IP address of the device, port<port number>"
```

By looking into the error messages, you will know that the errors are related to ASNMP. These are usually caused by the device's inability to process ASNMP requests. Turning ASNMP off for these devices may resolve the issue. Add the sysOID entries separated by “|”, for example:

```
DoNotUseASNMPPattern .1.3.6.1.4.1.9.5.50|.1.3.6.1.4.9.5.44
```

The VMware Smart Assurance IP Manager User Guide and the
VMware Smart Assurance IP Manager Reference Guide provides information on how to enable ASNMP and on the flag which configures the maximum number of OIDs per packet that can be sent using ASNMP.

Link Layer Discovery Protocol (LLDP)

The Link Layer Discovery Protocol (LLDP) is a neighbor discovery protocol used by network devices to advertise their identity, capabilities, and interconnections on an IEEE 802 LAN network. LLDP is vendor-neutral. The protocol defines a standard way for Ethernet devices to advertise information about themselves to their network neighbors, and store the information they discover from other devices.

LLDP operates above the MAC service layer and can be used in any networking device that implements a MAC service. Information gathered with LLDP is stored in the device in a standard management information database (MIB) and can be queried with the Simple Network Management Protocol (SNMP). For example, a LAN switch and a router advertise chassis/port IDs and system descriptions to each other. The devices store the information they learn about each other in local MIB databases accessible through SNMP. The network management system retrieves the data stored by each device and builds a network topology map.

Frame structure

LLDP supports a set of attributes that it uses to discover neighboring devices. These attributes contain type, length, and value descriptions and are referred to as TLVs. LLDP frames are sent by each device on each port at a fixed frequency. A frame contains a Link Layer Discovery Protocol Data Unit (LLDPDU) which is a set of type-length-value (TLV) structures. LLDP supported devices use TLVs to send to and receive information from their immediate neighbors.

The LLDP frame starts with the following mandatory TLVs:

- Chassis ID
- Port ID
- Time to live (TTL)

  The LLDP frame ends with a special TLV named “End of LLDPDU”. Refer the IEEE 802.1AB specification for a description of all the TLV types.

  Information that may be retrieved includes:

- System name and description
- Port name and description
- VLAN ID
- IP management address
- System capabilities (switching, routing, and so on)
- MAC/PHY information
- Link aggregation

Transmission and reception

The transmission and reception of LLDPDUs are enabled when LLDP is enabled on a port. The LLDP agent operates in any of the following modes:

- Transmit-only mode
- Receive-only mode

- Transmit-and-receive mode

  When LLDP transmission is enabled, the LLDP agent advertises information to neighbors at regular, user-configured intervals. When LLDP reception is enabled on a port, the LLDP agent receives advertised information from remote neighboring devices and stores this data in the remote systems' MIB.

  Note On Cisco devices, if both LLDP and Cisco Discovery Protocol (CDP) are enabled, LLDP will be given priority and connections will be created using LLDP. If LLDP is enabled on a device, and CDP is enabled on a different device, connection is created only as a bridge.

  It is important for both endpoints to support LLDP (meaning LLDP enabled) but at least one endpoint should have LLDP MIB support. The other endpoint should have basic MIB support to figure out the port and interface.

Implementation

LLDP Implementation lists various connection scenarios when LLDP is enabled or disabled between two devices A and B.

Table 5-3. LLDP Implementation

System B		LLDP enabled		LLDP not enabled
System A		Full MIB support for LLDP	LLDP MIB not supported	
LLDP enabled	Full MIB support for LLDP	Create the connection by using information from either system A or system B	Create the connection using information only from system A	No connection
	LLDP MIB not supported	Create the connection by using information only from system B	No connection	
LLDP not enabled		No connection		

LLDP supports creating instances of Device connection classes:

- NetworkConnection
- TrunkCable
- Cable

  LLDP supports multiple physical connections.

MIB OIDs

IP Manager creates connections between network devices using the following LLDP MIB OIDs:

- localSystemNameOID {".1.0.8802.1.1.2.1.3.3"}
- localSystemDescOID {".1.0.8802.1.1.2.1.3.4"}

- localPortOID {".1.0.8802.1.1.2.1.3.7.1.3" }
- localPortDescOID {".1.0.8802.1.1.2.1.3.7.1.4" }
- remotePortOID {".1.0.8802.1.1.2.1.4.1.1.7" }
- remotePortDescOID {".1.0.8802.1.1.2.1.4.1.1.8" }
- remoteSystemNameOID {".1.0.8802.1.1.2.1.4.1.1.9"}
- remoteSystemDescOID {".1.0.8802.1.1.2.1.4.1.1.10"}
- remoteManAddrIfOID {".1.0.8802.1.1.2.1.4.2.1.4"}

Installation impact

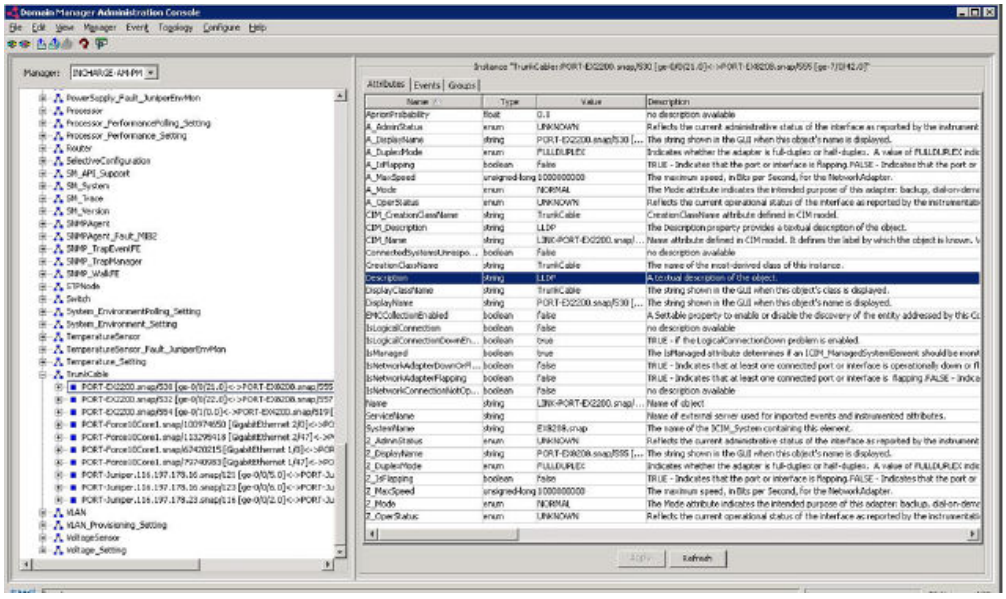
Make files are changed in order to incorporate the following new ASL files:

- ic-lldp-post-processor.asl
- ic-lldp.asl

Connectivity relationship

LLDP connection shows an LLDP connection. In the example, the LLDP connection is created between two switches under the TrunkCable class. The Description attribute is set to LLDP.

Figure 5-4. LLDP connection



Impact on discovery

A new probe named LLDP probe has been added.

Post-processing impact

The IP Availability Manager discovers LLDP enabled devices, and the global data structure is loaded with the connection details. This data structure is processed during the post-processing phase in the IP Availability Manager. The IP Availability Manager runs the ic-lldp-post-processor.asl file to do the post-processing.

NetworkConnections, TrunkCable, and Cable connections are created in the IP Availability Manager which processes the global data structure based on data gathered during discovery.

Field certification and validation of connections

If you want to do a field certification and validate the connections, invoke the LLDP framework by specifying the NEIGHBOR probe as LLDP in the configuration file oid2type_<*vendor*>.conf.

Supported devices

Cisco - C2960-24TT

Juniper - EX3200, Ex4200, EX8208, EX2200, and EX4500

Force10 - E600

Huawei Quidway Switches

Unique VLAN naming

The collection of bridges in a local area network (LAN) can be depicted as a graph whose nodes are bridges and LAN segments (or cables), and whose edges are the interfaces connecting the bridges to the segments. To break loops in the LAN while maintaining access to all LAN segments, the bridges collectively compute a spanning tree.

For all switches in a network to agree on a loop-free topology, a common frame of reference must exist to use as a guide. This reference point is called the Root Bridge. An election process among all connected switches chooses the Root Bridge. Each switch has a unique Bridge ID.

When a switch first powers up, it assumes that it is the Root Bridge itself. Every switch begins by sending out Bridge Protocol Data Units (BPDUs) with a Root Bridge ID equal to its own Bridge ID and a Sender Bridge ID that is its own Bridge ID. The Sender Bridge ID simply tells other switches who is the actual sender of the BPDU message. After a Root Bridge is decided on, configuration BPDUs are sent only by the Root Bridge. All other bridges must forward or relay the BPDUs, adding their own Sender Bridge IDs to the message.

In earlier releases, IP Manager created VLANs using VLAN IDs. This resulted in duplication of VLAN objects when multiple domains were configured with the same VLAN ID. VLANs were represented as VLAN-<VLAN ID>, for example, VLAN-1003. So the user could not discover all VLAN objects since the topology did not have all the VLAN objects because of VLAN ID duplication.

In this release, IP Manager creates unique VLANs from the Root Bridge ID and VLAN ID, that is, VLAN-<Root Bridge ID>-<VLANID>.

Each VLAN has a different Root Bridge ID because the participating members may be different. From the BRIDGE-MIB, dot1dStpDesignatedRoot (Root Bridge ID) is derived. The dot1dStpDesignatedRoot (OID: 1.3.6.1.2.1.17.2.5) is a combination of dot1dStpPriority (1.3.6.1.2.1.17.2.2 - Value of the write-able portion of the Bridge ID) and dot1dBaseBridgeAddress (The MAC address used by this bridge when it must be referred to in a unique fashion. When concatenated with dot1dStpPriority, a unique BridgeIdentifier is formed which is used in the Spanning Tree Protocol). The combined representation of Root Bridge ID and VLANID in this release is: VLAN-<Root Bridge ID>-<VLANID>.

Depending on the device types, there could be other ID added to the VLAN name, for example:

Alcatel-PR driver - Device name is added after "VLAN" class name:

VLAN-196.36.122.235-00-E0-B1-62-C8-3B-1

VLAN-196.36.122.235-00-E0-B1-62-C8-3B-2

Alcatel-Omni driver - The "Alcatel-Omni" driver is added after the VLAN class name:

VLAN-Alcatel-Omni-00-D0-95-4E-A2-81-1

A new parameter “STPVLANNamingEnabled” is added in the discovery.conf file. STP based VLAN unique naming discovery is enabled by incorporating the root bride ID as part of the VLAN name.

Default Value = TRUE

STPVLANNamingEnabled = TRUE

It is set to TRUE (by default) to enable VLAN unique naming by incorporating STP root bridge ID as part of the VLAN name during IP Manager discovery. If it is set to FALSE, the STP root bridge ID will not be included in the VLAN name.

Note Here the Root Bridge ID is formed by using only dot1dBaseBridgeAddress (MAC address). The dot1dStpPriority part from dot1dStpDesignatedRoot is not used.

If the Root Bridge ID changes due to network configuration changes, then the SVLAN name needs to be changed too. Any change to the SVLAN name will reflect after performing a "Rediscover All".

For Cisco devices supporting CISCO-VTP-MIB, the combined representation of Root Bridge ID and VLANID is: VLAN-<Root Bridge ID>-<VLANID>[VTP name], for example, VLAN-00-19-AA-CE-1B-00-10[VSAN_NEW].

OID Info Tool

The OID Info tool enables you to view the list of OIDs that are used to discover and monitor devices such as Host, Router, Switch, and so on. The OID information helps identify faults in the configuration of the SNMP agent of the device. The OID Info tool displays the list of OIDs and their SNMP values for a device. It fetches the value of the OIDs from the device and compares the SNMP values against the values in the topology.

The OID Info tool is introduced in the IP Manager and SAM Console versions from the 9.2 release and is not available in older versions. The OIDs can be viewed through the command line or through the **OID Info** option in the Domain Manager Administration Console.

The VMware Smart Assurance IP Manager User Guide provides more information on the OID Info tool.

ESX Server Certification

The VMware ESX server has been certified as a host. The interface monitoring of an ESX server is done using VMware SNMP agent.

The ESX server certification allows:

- Connection establishment between an ESX host, which could be a standard virtual switch (vSS) or a virtual distributed switch (vDS) and a physical switch. This is done using Cisco Discovery Protocol.
- Impact propagation of the physical switch failure on VMs and ESX.

Attributes for ESX Host

“Attributes of ESX Host” section in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide provides more information about the attributes for ESX Server.

Supported ESX versions

- VMware ESX 4.0
- VMware ESX 4.0i
- VMware ESX 4.1
- VMware ESX 4.1i
- VMware ESX 5.0

Pre-requisites

The following are the pre-requisites for an ESX server discovery and monitoring:

- VMware SNMP agent should be enabled on the ESX server.
- Cisco Discovery Protocol (CDP) should be enabled on both the ESX server and the physical switch.
- DNS server should have an entry mapping the ESX server name to its IP address.
- Do not enable the Net-SNMP Agent. Enable only the VMware SNMP Agent. If the proxy VMware SNMP Agent / Net SNMP Agent is enabled, the ESX server will be classified as a Linux Host and is treated like any other Linux host.

Supported functionality

The ESX server certification behavior and the supported functionalities are summarized in Supported functionality of ESX server certification.

Table 5-4. Supported functionality of ESX server certification

Supported ESX server version	Pre-requisites		IP 9.0 functionality
	VMware SNMP Agent	CDP	
ESX 4.0	Enabled	Enabled	■ Proper interface status supported ■ Connection between ESX and physical switch supported
	Enabled	Disabled	■ Proper interface status supported ■ Connection between ESX and physical switch NOT supported
	Disabled	NA	■ Improper interface status **Note** When VMware SNMP agent and CDP are not enabled, then the correct interface status is not shown. ■ Connection between ESX and physical switch NOT supported
ESX 4.0i	Enabled	Enabled	■ Proper interface status supported ■ Connection between ESX and physical switch supported
	Enabled	Disabled	■ Proper interface status supported ■ Connection between ESX and physical switch NOT supported
	Disabled	NA	■ Improper interface status **Note** When VMware SNMP agent and CDP are not enabled, then the correct interface status is not shown. ■ Connection between ESX and physical switch NOT supported
ESX 4.1	Enabled	Enabled	■ Proper interface status supported ■ Connection between ESX and physical switch supported
	Enabled	Disabled	■ Proper interface status supported ■ Connection between ESX and physical switch NOT supported
	Disabled	NA	■ Improper interface status **Note** When VMware SNMP agent and CDP are not enabled, then the correct interface status is not shown. ■ Connection between ESX and physical switch NOT supported

Table 5-4. Supported functionality of ESX server certification (Continued)

Supported ESX server version	Pre-requisites		IP 9.0 functionality
	VMware SNMP Agent	CDP	
ESX 4.1i	Enabled	Enabled	■ Proper interface status supported ■ Connection between ESX and physical switch supported
	Enabled	Disabled	■ Proper interface status supported ■ Connection between ESX and physical switch NOT supported **Note** In IP 9.0, connection through bridge is disabled if one of the peer is an ESX server.
	Disabled	NA	■ Improper interface status **Note** When VMware SNMP agent and CDP are not enabled, then the correct interface status is not shown. ■ Connection between ESX and physical switch NOT supported

Unsupported functionality

The current release does not support the following functionalities:

- Connection through bridge is not supported if one of the peer is an ESX server.
- LLDP protocol is not supported on ESX.
- NetSNMP agent features such as Host Resource Monitoring (Memory, Processor and so on) and Process Monitoring in EISM.

Cisco Nexus 1000V implementation

The Nexus 1000V switch allows the connection between a physical switch and an ESX host by using CDP (Cisco Discovery Protocol) data. The Nexus 1000V switch is installed on an ESX host.

The main features of the Nexus 1000V implementation in IP:

- It shows the impact of a physical switch failure on the VMs that are connected to the Nexus 1000V switch.
- It allows the creation of cable connection between a physical switch and an ESX server, that hosts the Nexus 1000V switch.

Requirements

The following are the requirements for the Nexus 1000V implementation:

- The physical switch should be a Cisco switch.
- CDP (Cisco Discovery Protocol) should be enabled on the physical switch and Nexus 1000V.
- SNMP should be enabled on Nexus 1000V
- The discovery of the ESX host and the Nexus 1000V switch is required, to allow the discovery of the CDP cable connection.

Note If the ESX host is not discovered, then the CDP cable connection cannot be discovered.

To create connections between a physical switch and an ESX server, Cisco Discovery Protocol (CDP) should be enabled on both the physical switch and the vSwitch (Standard vSwitch, Distributed vSwitch and Nexus 1000 vSwitch) installed on the ESX server. CDP should be running in both Listen and Advertize mode. If CDP is not enabled in both Listen and Advertize mode, then the connections between the physical switch and the ESX server will not be created.

The ESX hostname should have an entry in DNS server mapping to ESX IP address.

Cisco Nexus 1000v certification

The Cisco Nexus 1000V switch, with system ObjectID .1.3.6.1.4.1.9.12.3.1.3.840, has been recertified as vSwitch. Earlier, it was certified as a switch.

Relationship between Nexus 1000v, VMs and ESX server

Relationship and their description lists the four new relationship that has been created for the vSwitch class, to show the virtual connection between a Nexus 1000V and ESX server and Nexus 1000V and VMs.

Table 5-5. Relationship and their description

Relationship	Description
ConnectedVMs	Shows the virtual connection between Nexus 1000V and VMs
SwitchedViaVirtualSwitches	To see this two-way relationship, both the Nexus 1000V and VMs (hosting the Nexus 1000V) should be discovered.
ConnectedESXs	Shows the virtual connection between Nexus1000V and ESX servers.
RelatedVirtualSwitches	

Autodiscovery of VMs connected to the Cisco Nexus 1000V

By default, autodiscovery of VM is enabled. With the Nexus 1000V implementation, IP Availability Manager is able to discover all the VMs connected to Nexus 1000V automatically and places them in the discovery pending list.

Note In order to auto discover the VMs, you have to discover the Nexus1000V and the connected adjacent Router or Layer 3 Switch.

To disable the VMs auto discovery feature, set the value of the parameter EnableNexus1KvVMautoDiscovery to FALSE in the tpmgr-param.conf file. The tpmgr-param.conf file is located in: <BASEDIR>\smarts\conf\discovery\tpmgr-param.conf

Note The VMs which are moved to the discovery pending list during VMs autodiscovery can be discovered only by creating a discovery filter matching the VMs. This discovery filter should be created before the VMs autodiscovery. Otherwise, rediscovery of Nexus1000V is required after creating the discovery filter.

Attributes for Cisco Nexus 1000v

“Attributes for Cisco Nexus 1000v” section in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide provides more information about the attributes for Cisco Nexus 1000v switch.

Events and Exceptions

“Events for System”, “Exceptions for System by IP Availability Manager”, and “Exceptions for System by IP Performance Manager” sections in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide provide more information about the events and exceptions that are generated for a switch.

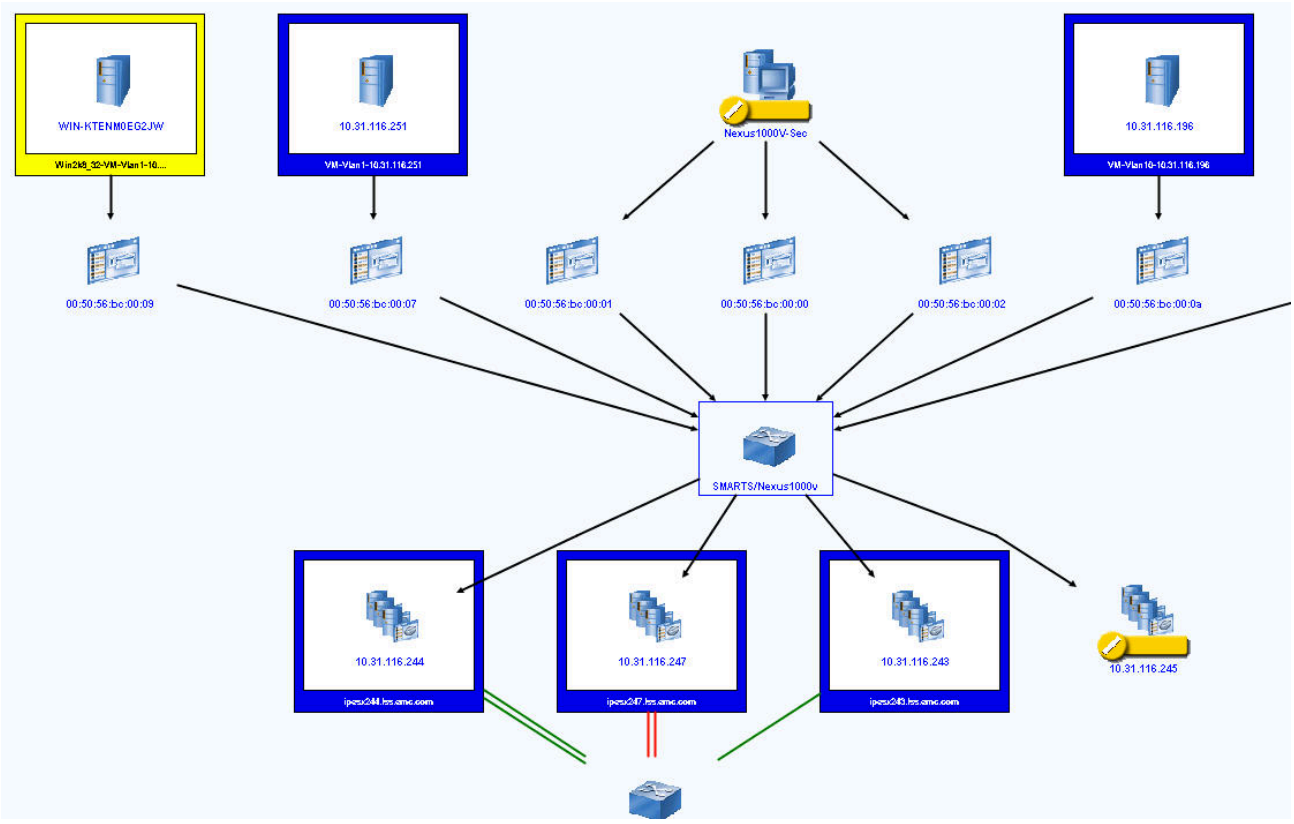
Viewing Nexus vSwitch consolidated map

A Nexus 1000 vSwitch consolidated map (both IP and EISM running) shows a Nexus 1000 vSwitch consolidated map, which is generated when both VMware Smart Assurance Server Manager and IP availability Manager are running. To display the map:

- *ConnectedESXs* and *ConnectedVMs* are relationship sets derived from IP

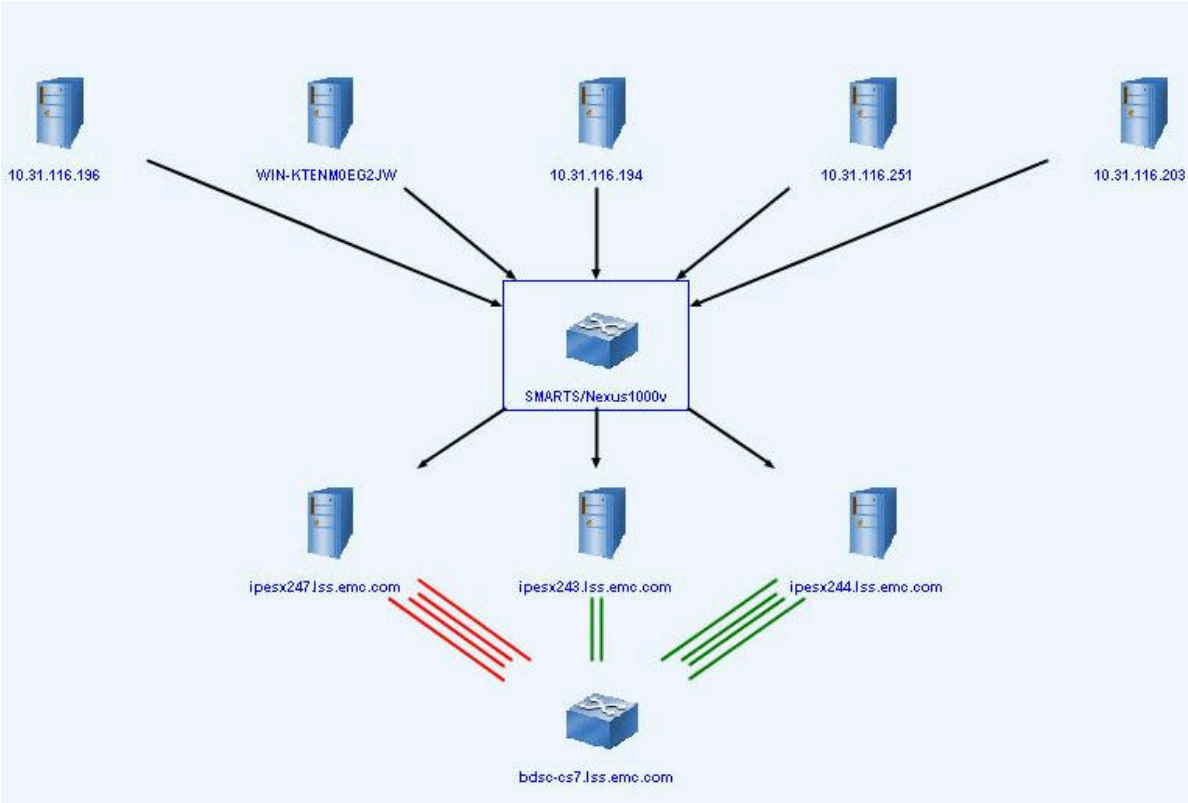
- *ConnectedVNics* and *LayeredOver* are the relationship sets derived from EISM

Figure 5-5. A Nexus 1000 vSwitch consolidated map (both IP and EISM running)



Nexus consolidated map when only IP is running shows a Nexus 1000 vSwitch when only IP is running. To display map, the *ConnectedESXs* and *ConnectedVMs* relationship sets are derived from IP.

Figure 5-6. Nexus consolidated map when only IP is running



Cisco Unified Computing System Certification

The Cisco Unified Computing System (Cisco UCS) is a next-generation data center platform that unites compute, network, storage access, and virtualization into a cohesive system designed to reduce total cost of ownership (TCO) and increase business agility.

Cisco UCS server integrates a low-latency, lossless 10 Gigabit Ethernet unified network fabric with enterprise-class, x86-architecture servers. The system is an integrated, scalable, multi-chassis platform in which all resources participate in a unified management domain.

The Cisco UCS components are:

- Cisco UCS Chassis — The Cisco UCS Chassis is a crucial building block of the Cisco Unified Computing System, delivering a scalable and flexible architecture for current and future data center needs, while helping reduce total cost of ownership.
- Fabric Extenders — Cisco UCS Fabric Extenders connect the Cisco UCS Blade Server Chassis to the Cisco UCS Fabric Interconnects and help simplify cabling, management, and diagnostics.
- Blade Server — A blade server is a server chassis housing multiple thin, modular electronic circuit boards. Blade servers allow more processing power in less rack space, simplifying cabling and reducing power consumption.

Note Blade servers are not discovered by IP 9.0.

- Fabric Interconnect — Fabric Interconnect is a core part of the Cisco UCS that provides both network connectivity and management capabilities to all attached blades and chassis.

  The Cisco UCS server portfolio consists of:

- Blade Server platform
- B-Series and the C-Series Rack Mount platform

  Note IP 9.0 supports UCS Manager version 1.4 only.

  The object ID is used to certify the device as Switch or ComputeFabric based on the Cisco Nexus Operating System version.

  If the IOS version is 1.4, all Cisco USC related components are discovered.

  If the IOS version is 1.3 or below, then only Switch discovery happens.

  The components to support Cisco UCS certification are:

- ComputeFabric — Represents the Cisco UCS equipment. It is a platform which converges Compute, Network, Storage Access and Virutalization into an enclosed system to provide a highly flexible infrastructure in order to optimize application environments. It is composed of:
  - Cisco UCS Chassis (represented as BladeEnclosureManager)
  - Fabric Interconnect (represented as Switch)
- BladeEnclosureManager — Represents all the Cisco UCS chassis components. It is a UnitaryComputerSystem that packages the following physical components:
  - Cards (Fabric Extenders)
  - PowerSupply
  - Fan
  - TemperatureSensor
  - VoltageSensor
- Fabric Extenders — Fabric Extenders are elements that can be plugged into the Cisco UCS Chassis. The Fabric Extender is represented as Card in the IP topology. The card has a *PartOf/ComposedOf* relationship with BladeEnclosureManager.
- Fabric Interconnect — The Fabric Interconnect is represented as Switch in the IP topology. When two Fabric Interconnect switches of an UCS are discovered, trunk cable connections are created between the Fabric Interconnect port and BladeEnclosureManager port.
  - For Fabric Interconnect port connected to the BladeEnclosureManager, IP Availability Manager sets the PortType as ACCESS.

    Note IP Availability Manager sets this port as an ACCESS port because the peer port does not belong to the switch, but belongs to the BladeEnclosureManager.

- For BladeEnclosureManager port connected to the Fabric Interconnect, IP Availability Manager sets the PortType as TRUNK.

Supported features

Supported features of Cisco UCS lists the supported features of Cisco UCS.

Table 5-6. Supported features of Cisco UCS

Components		Description	Supported features
Cisco UCS Chassis		The Cisco UCS 5100 Series Blade Server Chassis is a crucial building block of the Cisco UCS.	■ Up to eight half-width Cisco UCS B-Series Blade Servers (both half and full-width blade form factors). ■ Four single-phase, hot-swappable power supplies on the Front of the chassis. ■ Eight hot-swappable fans on the Rear of the chassis. ■ Four power connectors (one per power supply) on the Rear of the chassis.
Fabric Extenders		The Cisco UCS 2100 Series Fabric Extenders fit into the back of the Cisco UCS 5100 Series chassis.	■ Up to two fabric extenders, enabling redundancy. ■ Four 10 Gigabit Ethernet, FCoE-capable, Small Form-Factor Pluggable Plus (SFP+) ports that connect the blade chassis to the fabric interconnect. ■ Eight 10 Gigabit Ethernet ports connected through the midplane to each half-width slot in the chassis.
Blade Server	Cisco UCS B200 M1 Blade Server	It is a half-width, two-socket blade server.	■ Two Intel Xeon 5500 Series processors. ■ Up to 96 GB of DDR3 memory. ■ Two optional hot-swappable small form factor (SFF) serial attached SCSI (SAS) disk drives. ■ One mezzanine connector for up to 20 Gbps of I/O throughput.
	Cisco UCS B250 M1 Blade Server	It is a full-width, two-socket blade server featuring Cisco Extended Memory Technology.	■ Supports two Intel Xeon 5500 Series processors. ■ Supports up to 384 GB of DDR3 memory. ■ Two optional SFF SAS disk drives. ■ Two mezzanine connections for up to 40 Gbps of I/O throughput.
	Cisco UCS B200 M2 Blade Server	It is a half-width, two-socket blade server.	■ Two Intel Xeon 5600 Series Processors ■ Supports up to 96 GB of DDR3 memory ■ Two optional hot-swappable small form factor (SFF) serial attached SCSI (SAS) disk drives ■ One mezzanine connector for up to 20 Gbps of I/O throughput.

Table 5-6. Supported features of Cisco UCS (Continued)

Components		Description	Supported features
Blade Server(continued)	Cisco UCS B250 M2 Blade Server	It is a full-width, two-socket blade server featuring Cisco Extended Memory Technology.	■ Two Intel Xeon 5600 Series Processors. ■ Up to 384 GB of DDR3 memory. ■ Two optional SFF SAS disk drives. ■ Two mezzanine connections for up to 40 Gbps of I/O throughput.
	Cisco UCS B440 M1 Blade Server	It is a high-Performance, full-width, four-socket Blade server.	■ Two or four Intel Xeon 7500 Series Processors. ■ 32 dual in-line memory module (DIMM) slots and up to 256 GB at 1333 MHz. ■ 40 nanometer class (DDR3) technology. ■ Four optional front-accessible, hot-swappable Small Form-Factor Pluggable (SFFP) drives and an LSI. ■ SAS2108 RAID Controller. ■ Two dual-port mezzanine cards for up to 40 Gbps I/O per blade.
Fabric Interconnect			■ Ethernet module that provides six ports of 10 Gigabit Ethernet using the SFP+ interface. ■ Fiber Channel plus Ethernet module that provides four ports of 10 Gigabit Ethernet using the SFP+ interface, and four ports of 1/2/4-Gbps native Fiber Channel connectivity using the SFP interface. ■ Fiber Channel module that provides eight ports of 1/2/4-Gbps native Fiber Channel using the SFP interface for transparent connectivity with existing Fiber Channel networks. ■ Fibre Channel module that provides six ports of 1/2/4/8-Gbps native Fibre Channel using the SFP or SFP+ interface for transparent connectivity with existing Fibre Channel networks.

Managing Virtual Machines present on Blade Server of UCS

Cisco UCS Manager can only manage the Cisco Developed hardware such as B-Series of Cisco Blade server.

Discovery and monitoring

The end user should provide the management IP or the Cluster IP address in order to discover the Cisco UCS. However, this does not pose as a limitation as the user can discover the Cisco UCS equipment by using either the Management IP (Cluster IP) or any of the Fabric Interconnect IP address.

Note It is suggested to discover the Cisco UCS by using the Management IP or Cluster IP address.

The Cisco UCS can be discovered and managed by the end user. IP Availability Manager supports the discovery and monitoring of more than one Cisco UCS device.

Every Cisco UCS entity has the Cisco UCS 5100 series chassis which contains various physical components such as Power supply, Fan blade server and so on. These chassis are managed by the UCS Manager. When end user discovers the Cisco UCS entity, then IP is able to discover all the physical chassis that are present on the Cisco UCS entity.

IP Availability Manager discovers and monitors:

- Cards (Fabric Extender) present on the BladeEnclosureManager (Cisco UCS Chassis). “Attribute of cards (Fabric Extender)” table in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide lists the attributes of card (Fabric Extender).
- Fabric Ports on Cards (Fabric Extender) of the BladeEnclosureManager (Chassis). “Attributes and MIB Object IDs of Fabric port” table in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide lists the attributes and MIB Object IDs of Fabric port.
- Cards on the Fabric Interconnect switch.
- Ethernet Ports on Cards of the Fabric Interconnect switch.
- Trunk Cable Connection between the Fabric Interconnect switch and the BladeEnclosureManager.
- Host (Blade Server) on the BladeEnclosureManager.

  Note ESX servers hosted by the blade servers are not discovered automatically.

Relationships

ComputeFabric

The ComputeFabric has the following relationships:

- *ComposedOf/PartOf* relationship with Card (Fabric Extender)
- *PackagesSystems/SystemPackagedIn* relationship with
  - BladeEnclosureManager

  - Switch (Fabric Interconnect)

BladeEnclosureManager

The BladeEnclosureManager has the following relationships:

- *ComposedOf/PartOf* relationship with:
  - Card
  - Fan
  - PowerSupply
  - TemperatureSensor
  - VoltageSensor
- *SystemPackagedIn/PackagesSystems* relationship with ComputeFabric.

  The UCS Manager is embedded with the Fabric Interconnect (FI) which provides seamless management of the UCS components by providing the redundancy configuration. Each FI has individual IP address. In order to support the redundancy, both are configured with a management or cluster IP address. Therefore, the user should provide the management IP address (cluster IP address) in order to discover the device.

  “Attributes and MIB Object IDs” table in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide lists the attributes and the MIB ObjectIDs of the Cisco UCS entity (ComputeFabric) and the Cisco UCS chassis (BladeEnclosureManager).

Attributes and events

“Attributes and events” section in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide provides more information on the attributes and events generated for ComputeFabric and BladeEnclosureManager.

Discovery and monitoring

“Discovery and monitoring” section in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide provides more information on the attributes and the MIB Object IDs using which the Cisco UCS components are discovered and monitored.

Cisco UCS performance monitoring

The components of a Blade Enclosure Manager that are monitored by IP Performance Manager are:

- Power Supply — IP Performance Manager performs performance analysis on the PowerSupply object. Each power supply unit of the Blade Enclosure Manager is monitored and represented in the topology. A BladeEnclosureManager has a *ComposedOf* relationship with PowerSupply and conversely, PowerSupply has a *PartOf* relationship with BladeEnclosureManager.

- Voltage Sensor — IP Performance Manager performs performance analysis on the following VoltageSensor objects for:
  - Power Supply Unit
  - Blade Server motherboard
- Temperature Sensor — IP Performance Manager performs performance analysis on the following TemperatureSensor objects for:
  - IO module—Sensors present in the Fabric Extender.
  - Fan—Sensors for fan present in the chassis.
  - Motherboard Temperature counters for:
    - Motherboard Front Temperature
    - Motherboard Rear Temperature
    - Motherboard Rear Temperature Left
    - Motherboard Rear Temperature Right
  - Power supply in the chassis
- Fan—IP Performance Manager performs performance analysis of the Fan. Each Fan unit of the BladeEnclosureManager is monitored and represented in the topology. The relationships of a BladeEnclosureManager with fan are:
  - *ComposedOf* relationship with Fan
  - Conversely, *PartOf* relationship with BladeEnclosureManager

    All the thresholds settings are configured in the Polling and Thresholds console.

Faults and Exceptions

“Faults and Exceptions” section in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide provides more information on the Faults and Exceptions of Blade Enclosure Manager.

Fabric Interconnect performance monitoring

IP Performance Manager discovers and monitors the following components of a Fabric Interconnect switch:

- Temperature sensors
- Voltage sensors
- Power supply
- Fan

- Memory

  The user can discover the components present in the Fabric Interconnect switch using the MIB Object IDs that are listed in the respective tables in the following sections:

- Discovery and monitoring of temperature sensors
- Discovery and monitoring of Voltage sensor
- Discovery and monitoring of Power supply
- Discovery and monitoring of Fan
- Discovery and monitoring of Memory

Note The discovery and monitoring of the Fabric Interconnect CPU is not supported as there is no information about it in the Cisco UCS MIB.

Discovery and monitoring of temperature sensors

The temperature sensors present in a Fabric Interconnect are used to discover and monitor a Fabric Interconnect. The various temperature sensors are:

- Fabric Interconnect's Fan Inlet1
- Fabric Interconnect's Fan Inlet2
- Fabric Interconnect's Fan Inlet 3
- Fabric Interconnect's Fan Inlet 4
- Fabric Interconnect's Motherboard Outlet 1
- Fabric Interconnect's Motherboard Outlet 2
- Fabric Interconnect's Power Supply Inlet 1
- Fabric Interconnect's Power Supply Inlet 2

  "MIB Object ID of temperature sensor used for discovery" table in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide lists the MIB Object IDs used for discovery of temperature sensor in a Fabric Interconnect.

Discovery and monitoring of Voltage sensor

"MIB Object IDs used for discovery and monitoring of voltage sensor" table in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide lists the MIB Object IDs used for discovery and monitoring of voltage sensor in a Fabric Interconnect.

Discovery and monitoring of Power supply

"MIB Object IDs used for discovery and monitoring of Power Supply in Fabric Interconnect" table in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide lists the MIB Object IDs used for discovery and monitoring of power supply present in a Fabric Interconnect.

Discovery and monitoring of Fan

“MIB Object IDs used for discovery of fan in Fabric Interconnect” table in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide lists the MIB Object IDs used for discovery and monitoring of fan instances in a Fabric Interconnect.

Discovery and monitoring of Memory

“MIB Object IDs for discovery and monitoring of memory instances in Fabric Interconnect” table in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide lists the MIB Object IDs used for discovery and monitoring of memory instance in a Fabric Interconnect.

UCS Map

You can view the details about ComputeFabric, BladeEnclosureManager and all related connections in the following format:

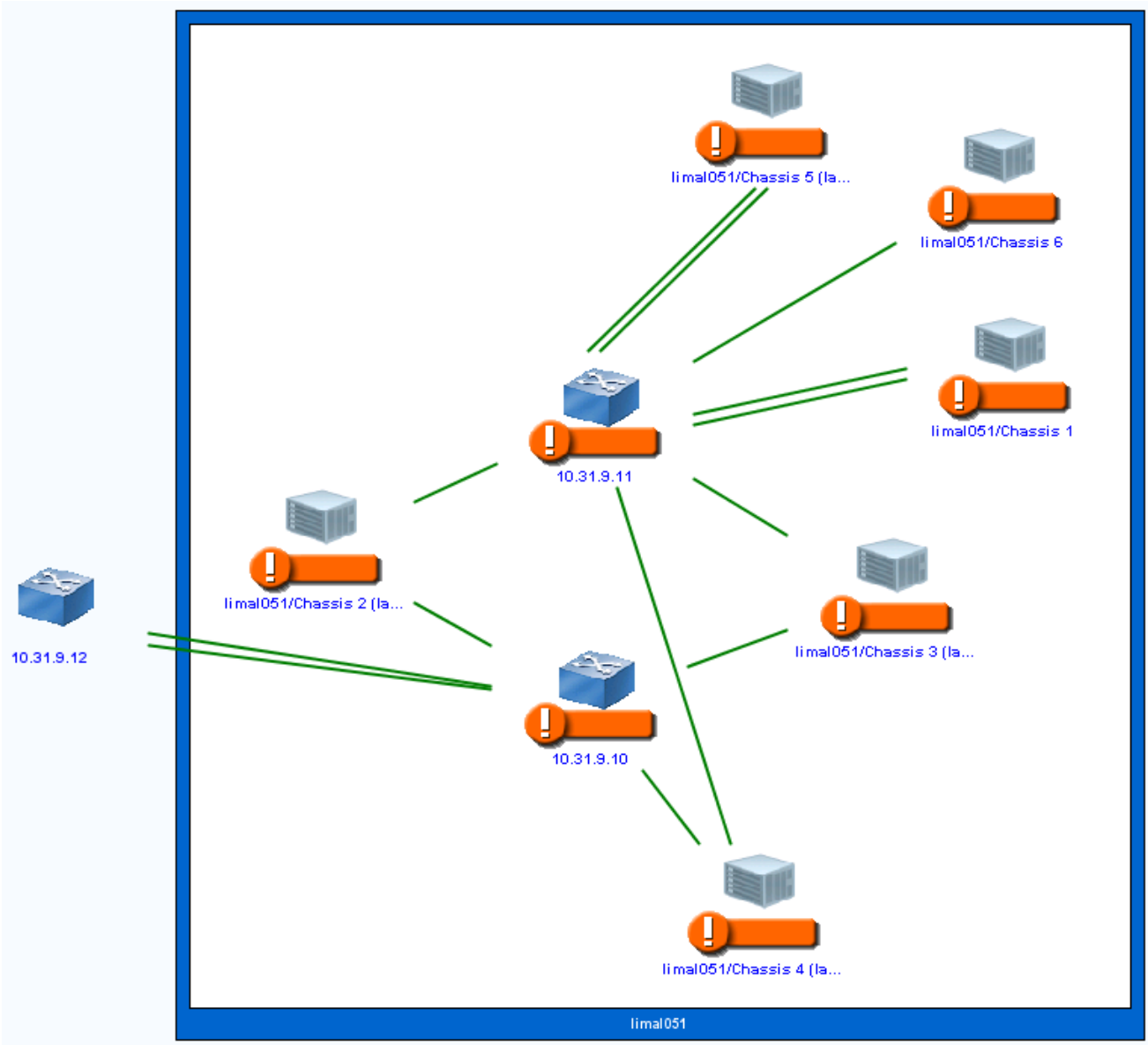
- Global Manager Administration Console
- Notification Log
- Topology Browser
- Containment view
- Maps

Viewing maps

The physical representation of a ComputeFabric is represented by the option **Show Map**. The map shows relationships among topology elements, indicating the status of each element. In a map display, a node is a graphical representation of an element, and an edge is a graphical representation of a relationship or connection between elements.

Viewing the ComputeFabric connectivity map

Figure 5-7. ComputeFabric map



ComputeFabric map shows a ComputeFabric physical connectivity map. The ComputeFabric (lima051) has *PackagesSystems* relationship with:

- BladeEnclosureManager
- Fabric Interconnect (represented as a switch)

  The relationships and the icons are described in ComputeFabric map icons .

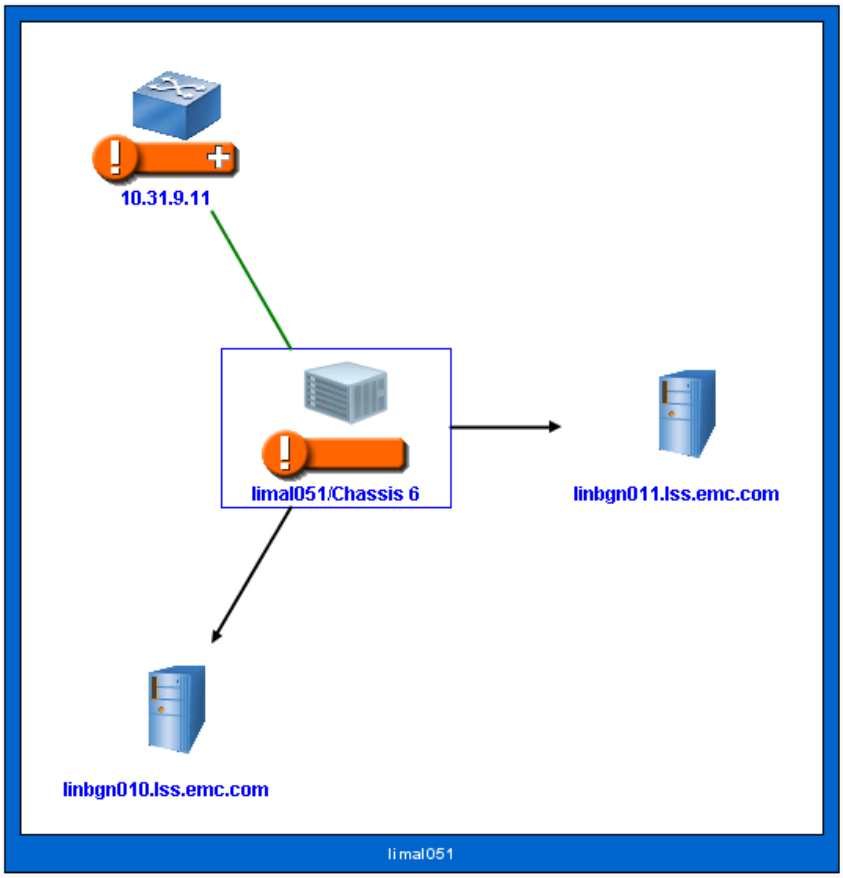
Table 5-7. ComputeFabric map icons

Icons	Representations	Relationship
	BladeEnclosureManager	■ ComposedOf Card, Fan, Port, PowerSupply, TemperatureSensor, VoltageSensor ■ *SystemPackagedIn* ComputeFabric ■ *ConnectedVia* TrunkCable ■ *MemberOf* Partition
	Fabric Interconnect (10.31.9.11 and 10.31.9.10 in this map) **Note** Fabric Interconnect is represented as a Switch in IP topology.	*SystemPackagedIn* relationship with ComputeFabric.
	External Switch (10.31.9.12 in this map)	

Viewing the BladeEnclosureManager connectivity map

BladeEnclosureManager shows the map of a BladeEnclosureManager (lima051)

Figure 5-8. BladeEnclosureManager



In this map, the BladeEnclosureManager (lima051) has:

- Two ESX hosts installed in its blade. You can view the VMware map by right clicking on the ESX host icon.
- Connected to a FabricInterconnect (10.31.9.11)

  BladeEnclosureManager map icons and their representations lists the representations and their relationship in the map.

Table 5-8. BladeEnclosureManager map icons and their representations

Icons	Representations	Relationships
	BladeEnclosureManager	■ ComposedOf Card, Fan, Host, Port, PowerSupply, TemperatureSensor, VoltageSensor ■ *SystemPackagedIn* ComputeFabric ■ *ConnectedVia* TrunkCable ■ *MemberOf* Partition
	FabricInterconnect	*Connectedsystems* BladeEnclosureManager
	ESX host installed on the blades present in the BladeEnclosureManager	*PartOf* BladeEnclosureManager **Note** Right click on the icon to view the VMware map.

Root cause analysis for external switch Down

“Root cause analysis for external switch Down” section in Chapter 1, Network Objects, in the VMware Smart Assurance IP Manager Reference Guide provides more information on root cause analysis for external switch Down.

AggregateLink representation in the Map console

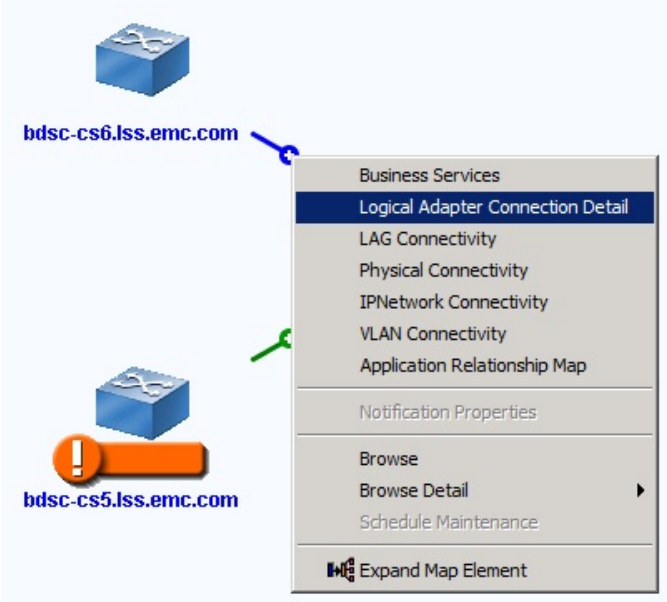
AggregateLink represents the virtual aggregated link composed of physical links between UnitaryComputerSystem instances. The AggregateLink is represented by a logical connectivity map. AggregateLink has a *ComposedOf* relationship with a set of NetworkConnections. An AggregateLink is displayed as a thicker NetworkConnection icon with a + sign on it, shown in Logical Adapter Connection Detail option, that depicts the expansion capability, but hides the NetworkConnections that it is composed of.

To view the underlying NetworkConnections in a AggregateLink:

1 Open the Map console.

2 Right click the AggregateLink edge and select *Logical Adapter Connection Detail* as shown in Logical Adapter Connection Detail option.

Figure 5-9. Logical Adapter Connection Detail option

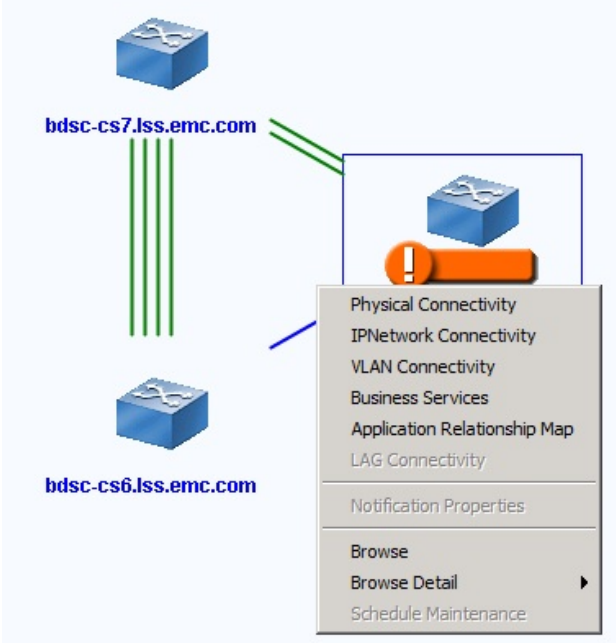


3 Other options:

- To view the underlying network connections, right click and select *Expand Map Element* as depicted in Logical Adapter Connection Detail option. This launches another map with the connecting UnitaryComputerSystems and the underlying NetworkConnections of the AggregateLink.

- To go back to the previous view, right click any entity on the map, preferably a NetworkConnection, and select *PhysicalConnectivity* as shown in Physical connectivity option.

  Figure 5-10. Physical connectivity option

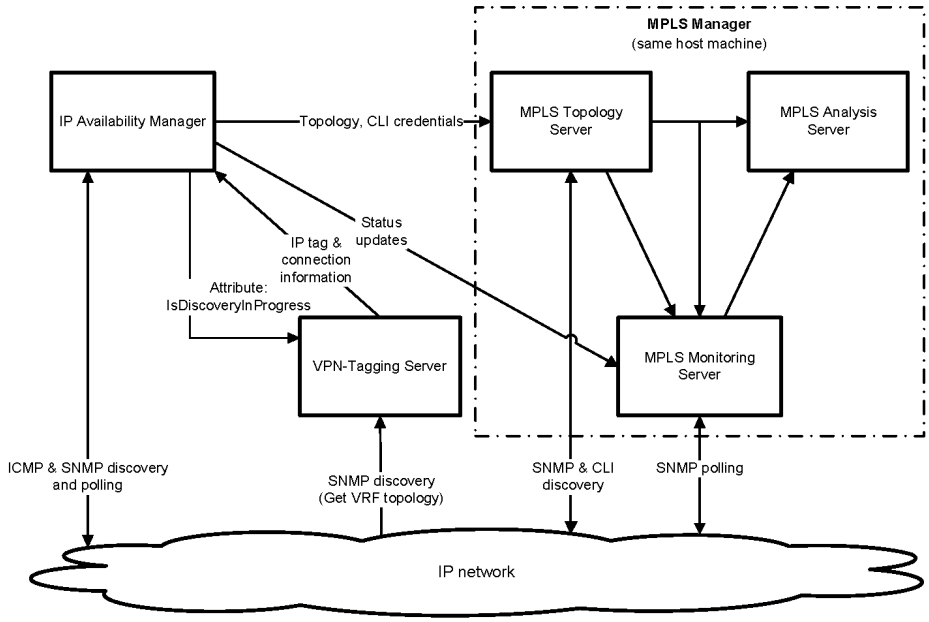


- To go back to the original map, right click and select *Reset Map*.

Discovery and the VPN-Tagging Server

The VMware Smart Assurance VPN-Tagging Server, which is an optional server in the MPLS Management Suite of products, assists the IP Manager in the creation of certain overlapping IP-address configurations. The VPN-Tagging Server discovers VPN routing and forwarding (VRF) based network connections and sends the connection information to the IP Manager so that it can distinguish overlapping IPs in the managed MPLS network. An IP Manager deployment that includes the VPN-Tagging Server shows the flow of information for the VPN-Tagging Server.

Figure 5-11. An IP Manager deployment that includes the VPN-Tagging Server



You must explicitly invoke the **Discover All** command on the IP Manager to import all the overlapping IP-address configuration information from the VPN-Tagging Server. That is, after the initial topology synchronization between an IP Manager and the VPN-Tagging Server, you must explicitly invoke the **Discover All** command on the IP Manager to import all IP tag and connection information from the VPN-Tagging Server.

The procedure for invoking the **Discover All** command is given in the VMware Smart Assurance IP Manager User Guide.

The VMware Smart Assurance MPLS Manager Discovery Guide Supplement provides information about how the VPN-Tagging Server solves the overlapping IP discovery problem.

About seed files

Preparing a seed file is one of two ways to manually add systems to the IP Manager topology. The other way is by using the **Add Agent** command, which is described in Using the Add Agent command to initiate discovery. When autodiscovery is enabled, importing a seed file or adding a candidate system by using the **Add Agent** command initiates the autodiscovery process.

In a seed file, you specify top-level network systems such as switches, routers, hubs, bridges, and hosts that are to be discovered and monitored by the IP Manager. Then, when you import the seed file, the IP Manager discovers the internal structure of these systems.

For example, in the case of a router, you provide the hostname or IP address of one interface, and the IP Manager discovers the rest of the interfaces. In the case of a switch, you provide the hostname or IP address of the switch's SNMP agent, and the IP Manager determines the switch's configuration and how the switch connects to other systems that are listed in the topology.

Seed files can be used to discover SNMPv1, v2c, or v3 systems in IPv4 or IPv6 networks.

“Seed file related tasks” section in Chapter 3, Discovery, in the
VMware Smart Assurance IP Manager User Guide, provides more information about tasks related to seed files.

Seed file template

The seed file must be stored on the host on which the IP Manager runs. By default, the IP Manager expects to find the seed file in the BASEDIR/smarts/local/conf directory. A template seed file named seedfile is provided in the BASEDIR/smarts/conf directory.

Seed file dual use

The seed entries in the seed file are read by the IP Manager’s topology manager and trap manager at startup. The topology manager uses the entries and their credentials to contact the seed systems during discovery. The trap manager uses the entries and their credentials to verify and decrypt SNMPv3 trap messages that are received from SNMPv3-capable seed systems once they are discovered.

Seed file format

A seed file consists of one or more seed entries, each of which represents a discovery candidate system. Each seed entry begins with the hostname or IP address for the candidate system, followed by options for the system.

Seed-entry system identifiers

For a seed entry that begins with a hostname, you can specify the hostname with or without an explicit IP version suffix (:v6v4, :v4v6, :v6, or :v4). Here is an example hostname that includes a v4 suffix: Core-ROUTER-1:v4.

If you do not include an explicit IP version suffix, the SM_IP_VERSIONS environment variable controls how the local name resolution service resolves the hostname to a corresponding IP address. By default, SM_IP_VERSIONS is set to v6v4.

For a seed entry that begins with an IP address, you can specify an IPv6 or IPv4 address. If you specify an IPv6 address *and* a port number, you must enclose the address in square brackets; for example: [2001:0db8::0010]:200.

Be careful not to use an overlapping IP address as a seed system.

Hostname syntax, the SM_IP_VERSIONS environment variable, and overlapping IP addresses are discussed in the VMware Smart Assurance IP Manager User Guide.

Notation and syntax

You use <keyword>=<value> to specify options for a seed entry. (Blank spaces before and after the equal sign (=) are not permitted.) The <keyword>=<value> options for a seed entry may be arranged in any order, and they may span several lines for an entry. Here is an example of a seed entry that begins with a hostname that includes a v4v6 suffix:

```
CoreROUTER-2:v4v6 SNMPPORT=200 COMMUNITY=p100 ACCESSMODE=ICMPSNMP SNMPVERSION=V2C
```

The first character of a comment line is the pound sign (#). For example:

```
# The seed file is used to import top-level network systems.
```

To make your seed file more readable, you can include blank lines and comments.

Specifying a port

For a seed entry, you can specify the port number of the SNMP agent that is running on the candidate system in one of two ways:

- By typing a colon (:) and the port number after the hostname or IP address; for example:

  ```
  [2001:0db8::0010]:200
  ```

- By using the SNMPPORT keyword; for example:

  ```
  2001:0db8::0010 SNMPPORT=200
  ```

  The default port number, 161, is specified in the defaultSNMPPort parameter in the discovery.conf file.

Specifying a community string

For a candidate system that has an SNMPv1 or v2c agent, the second field of a seed entry is the optional read community string. If you do not provide a community string for a v1 or v2c candidate system, the IP Manager uses the default read community string or strings that are specified in the Read Community Strings window (Topology tab) of the Domain Manager Administration Console.
VMware Smart Assurance IP Manager User Guide provides details about how to specify community strings in the Read Community Strings window. If the community string starts with the pound sign (#), you must use the COMMUNITY keyword to specify the read community string.

Mandatory keyword options for SNMPv1 or v2c

For a candidate system that has an SNMPv1 or v2c agent, no keyword options are mandatory because all global and SNMPv1/v2c keywords have default values that are applicable to an SNMPv1 or v2c agent. So, specifying just a hostname or an IP address for an SNMPv1/v2c seed entry will not cause the IP Manager to throw an exception when importing the seed file.

Mandatory keyword options for SNMPv3

For a candidate system that has an SNMPv3 agent, two keyword options are mandatory:

- SNMPVERSION
- USER

  If you do not specify both keyword options for an SNMPv3 seed entry, the IP Manager will throw an exception for the seed entry when reading the seed file.

  To authenticate or decrypt an SNMPv3 trap, you must include an entry in the seed file that matches the source of the trap.

  When the IP Manager receives an SNMPv3 trap, it reads the SNMPv3 agent's username and Engine ID, searches the seed file for an SNMPv3 seed entry match, and uses the credentials that are associated with that seed entry to authenticate or decrypt the trap message.

Seed file examples

The seed file entries in this section illustrate the use of the system identifiers and the options for an SNMPv1 or v2c and SNMPv3 candidate system. The system identifiers are described in Seed-entry system identifiers, and the options are described in the VMware Smart Assurance IP Manager Reference Guide.

Seed file example for v1/v2c candidate systems

The following three seed entries are examples of how to present the same seed entry as a hostname, as an IPv4 address, and as an IPv6 address. A dual-stack discovery candidate system could be presented in any of these three ways. Each seed entry spans multiple lines and specifies the SNMP version and the read community string:

```
Core-ROUTER-1 SNMPVERSION=V1
      COMMUNITY=xxxxyyyy#%@*&^)(:"=|\][}{
192.35.144.12 SNMPVERSION=V1
      COMMUNITY=xxxxyyyy#%@*&^)(:"=|\][}{
3FFE:80C0:22C:101:219:56FF:FE3F:8A50 SNMPVERSION=V1
      COMMUNITY=xxxxyyyy#%@*&^)(:"=|\][}{
```

The following three seed entries specify the SNMP port (2161) that the IP Manager should use to access the system. Notice that the seed entry with the hostname system identifier includes a v6 IP version suffix, and that the seed entry with the IPv6 system identifier encloses the IPv6 address in square brackets.

```
ROUTER2:v6:2161 SNMPVERSION=V2C COMMUNITY=public1
192.35.144.12:2161 SNMPVERSION=V2C COMMUNITY=public1
[3FFE:80C0:22C:109:203:BAFF:FEE5:7BE1]:2161 SNMPVERSION=V2C
      COMMUNITY=public1
```

The following seed entry requires the keyword COMMUNITY because the first character of the read community string is a pound sign (#):

```
router-2.example.com COMMUNITY=#abcdefg SNMPPORT=2222
```

The following seed entry can be specified without using keywords. The only two fields are the system identifier and the read community string. The COMMUNITY keyword is not required, even though the first character of the community string is the equal sign (=).

```
192.168.2.2  =abcdefg
```

The following seed entry assumes that NameFormat is set to TM_USEAUTONAME in the name-resolver.conf file:

```
Core-ROUTER-1 NAMEFORMAT=TM_USELOOPBACK SNMPVERSION=V1
      COMMUNITY=xxxxyyyy#%@*&^)(:"=|\][}{
```

The following seed entry assumes that NameFormat is set to TM_USESEEDNAME in the name-resolver.conf file:

```
Core-ROUTER-1 NAMEFORMAT=TM_USEAUTONAME SNMPVERSION=V1
      COMMUNITY=xxxxyyyy#%@*&^)(:"=|\][}{
```

Seed file example for v3 candidate systems

The following three seed entries are examples of how to present the same seed entry as a hostname, as an IPv4 address, and as an IPv6 address. A dual-stack discovery candidate system could be presented in any of these three ways. Each seed entry spans multiple lines and specifies the SNMP version, the username, and no authentication:

```
Core-SWITCH-1 SNMPVERSION=V3 USER=inch_n
      ENGINEID=800002b804616263
20.30.4.1 SNMPVERSION=V3 USER=inch_n
      ENGINEID=800002b804616263
3FFE:80C0:22C:101:219:56FF:FE3F:8A50 SNMPVERSION=V3 USER=inch_n
      ENGINEID=800002b804616263
```

The following three seed entries specify the SNMP port (2161) that the IP Manager should use to access the system. Notice that the seed entry with the hostname system identifier includes a v4v6 IP version suffix, and that the seed entry with the IPv6 system identifier encloses the IPv6 address in square brackets:

```
SWITCH2:v4v6:2161 SNMPVERSION=V3 USER=inch_n
20.30.4.1:2161 SNMPVERSION=V3 USER=inch_n
[3FFE:80C0:22C:101:219:56FF:FE3F:8A50]:2161 SNMPVERSION=V3
      USER=inch_n
```

The following seed entry specifies that the default authentication protocol MD5 should be used to access the system. The prefix <E-1.0> indicates that the authentication password is encrypted:

```
switch-2.example.com SNMPVERSION=V3 USER=smarts SNMPPORT=2222
  ENGINEID=6acff2d4098338 AUTHPROTOCOL=MD5
  AUTHPASS=<E-1.0>869D64B72C38FE49E881BC38BCCE6A0B
  1DE597A5871636C4B0F139FA7EB1100C
```

The encrypted password, which wraps around in the display but is actually one line, was created by using the procedure described in the VMware Smart Assurance IP Manager User Guide.

The following seed entry specifies that the authentication protocol SHA should be used to access the system, and that the privacy protocol AES should be used to encrypt the SNMP messages that are sent to the system. The prefix <E-1.0> indicates that both the authentication password and the privacy password are encrypted:

```
128.221.19.8 SNMPVERSION=V3 USER=shaAesUser AUTHPROTOCOL=SHA
  AUTHPASS=<E-1.0>123ABC456#$%123abc456#$%
  PRIVPROTOCOL=AES
  PRIVPASS=<E-1.0>456123abc456#$%456123ABC456#$%
  ENGINEID=0000000902000003E333C440
```

Which systems to put in a seed file

The seed file should list the top-level network systems such as routers, switches, hubs, bridges, and hosts that are to be managed by the IP Manager. Routers and switches are better choices than hosts for seed systems because their MIBs contain the IP addresses of many additional network systems.

Do not include overlapping IP addresses in the seed file. Overlapping addresses are discussed in the VMware Smart Assurance IP Manager User Guide.

Seed file sources

To create the seed entries in a seed file, you can usually find the information that you need from existing computer-accessible sources:

- If you are currently using CiscoWorks with Resource Manager Essentials, you can dump the hardware inventory to a text file, which can be converted into a seed file. Similar operations might be possible with other network management platforms.

- If you store backup router or switch configuration files at a central location, you can construct a seed file from them.
- Domain name resolution files, such as hosts.txt or zone files, are often a useful source. Note that, at sites where multiple administrative authorities select different read community strings, the hosts that are listed in a given zone file often share an administrative authority and hence a read community string. Terminal Access Controller Access Control System (TACACS) configuration files might be useful for similar reasons.
- Large sites usually maintain hardware inventories in a database, or at least in some computer-readable form. These inventories often contain the network information that you need to create the seed file.
- Some sites run network sniffer tools to keep track of active IP addresses. The lists of addresses that they generate can form the basis for a seed file.

Autodiscovery seed systems

If you plan on using autodiscovery, consider including a system that is enabled with Cisco Discovery Protocol (CDP), Extreme Discovery Protocol (EDP), or Foundry Discovery Protocol (FDP) in the seed file when possible to improve autodiscovery coverage. The discovery process is able to probe CDP, EDP, FDP, and other tables to take advantage of the abundance of topology information that they hold.

SNMPv3 systems

Autodiscovery cannot be used to discover systems that use SNMPv3 agents. To discover SNMPv3 systems, add them to the seed file.

Multiple seed entries for multiple IP-address systems

After you have identified which systems to include in the seed file, you might want to include multiple seed entries for systems that have multiple IP addresses. Routers, for example, have multiple IP addresses.

The discovery process will realize that the multiple IP addresses actually belong to a single system. Accordingly, the IP Manager will be able to reach the system by using an alternative address if the primary address is inaccessible, due to a failure, during the initial probe of the system.

After a system is discovered and all its IP addresses are known, the IP Manager will automatically try up to 10 different IP addresses to route around failures and reach the system.

Discovery methods

Discovery methods shows the discovery methods and process for adding systems to the topology that is modeled by the IP Manager. The discovery methods are:

- Manual discovery

- Autodiscovery

  The VMware Smart Assurance IP Manager User Guide provides information on how to configure the IP Manager and the Global Manager for discovery.

Manual discovery

With manual discovery, shown in Manual discovery, the IP Manager discovers a set of systems that are specified in a seed file, or discovers an individual system that is specified in an Add Agent command. This method is useful in networks where the topology information is available from a well-maintained database.

Using this method with a comprehensive seed file is the quickest method for discovering a network because all systems are discovered by using the system hostnames or IP addresses that are already in the seed file. VMware Smart Assurance IP Manager User Guide provides instructions on creating a seed file.

Autodiscovery

With autodiscovery, shown in Discovery process, the IP Manager takes IPv4 addresses that are obtained from discovered systems to find additional discovery candidate systems. The initial IP addresses that are available to autodiscovery correspond to the neighboring systems of the systems that are discovered through manual discovery. The autodiscovery method is useful when topology information is incomplete or unavailable, or when manual updates cannot be maintained.

Autodiscovery requires creating discovery filters, which are described in VMware Smart Assurance IP Manager User Guide. For systems that match a discovery filter and are added to the modeled topology, they too are probed for IP addresses of their neighbors. The autodiscovery cycle continues until no more new IP addresses match the discovery filters. The discovery filters, which are configurable, ensure that the IP Manager sweeps only the networks that the customer systems are configured to access.

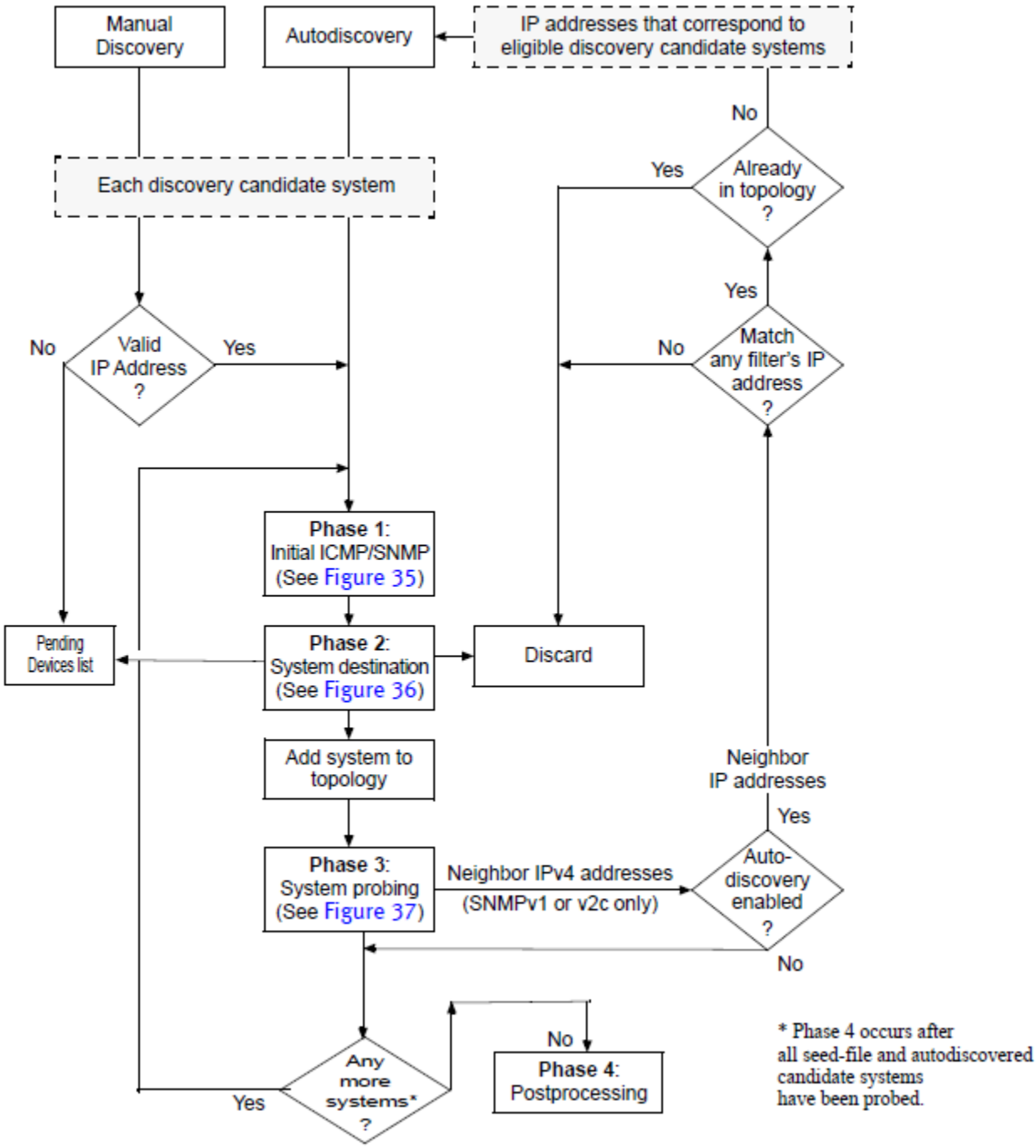


Figure 5-12. Discovery process

Discovery phases

The discovery is divided into four phases, shown in Discovery phases, that are performed sequentially for each discovery candidate system:

1 Perform initial polling of the candidate system

2 Determine the destination of the candidate system: add to topology, place on Pending Devices list, or discard

3 Probe the system that is added to the topology

4 Post-process the discovery information

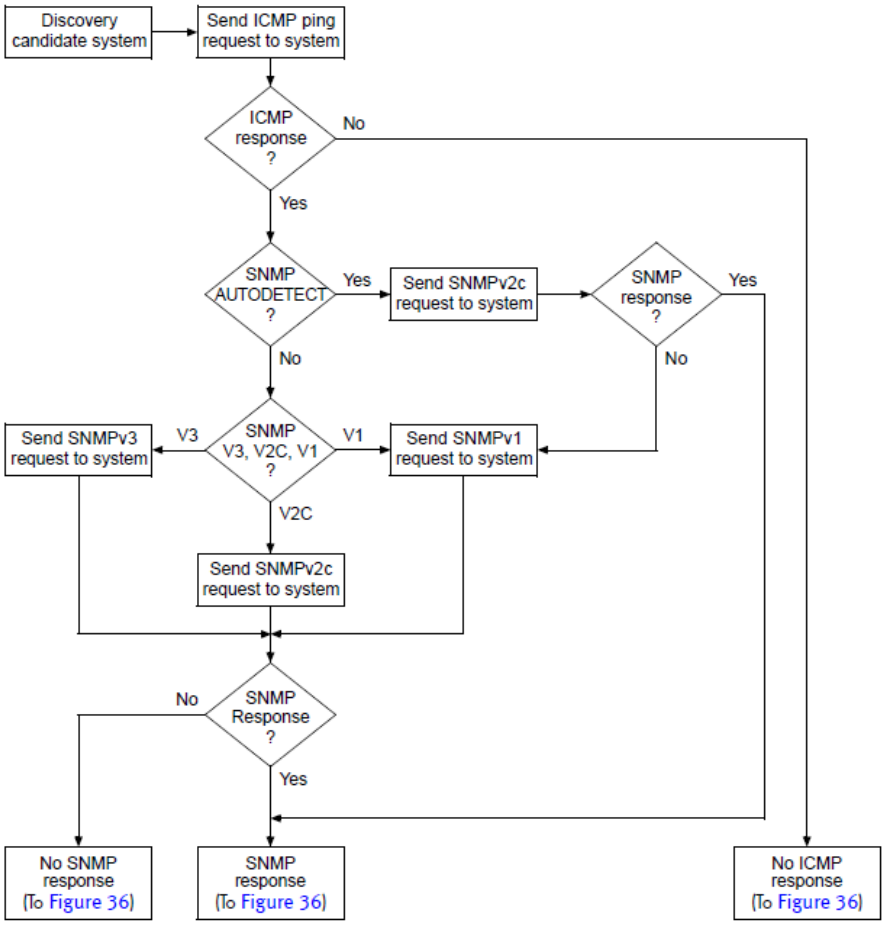
Note The VMware Smart Assurance IP Manager User Guide provides information on how to prepare for and initiate discovery.

Phase 1: Perform initial polling of the candidate system

Discovery for each discovery candidate system begins with a hostname or an IP address, which is provided by the user or automatically discovered. The discovery process resolves the hostname to one or more IP addresses through the local name resolution service. An IP address may be IPv4 or IPv6.

During the first phase of the discovery process, shown in Phase 1: Initial polling of a candidate system, the discovery process sends an ICMP request to determine if the discovery candidate system is reachable, and then sends an SNMP request to identify the system.

Figure 5-13. Phase 1: Initial polling of a candidate system



The discovery process pings the IP address of the discovery candidate system to see if the address is a reachable address. If the address is reachable, the discovery process sends an SNMP request to the system to request the following system-related information:

- sysName
- sysDescr
- sysObjectID
- sysContact
- sysLocation

  By default, the discovery process sends an SNMPv2c request to the candidate system. If that request fails after the allotted number of retries, the discovery process sends an SNMPv1 request. If the SNMP version is explicitly specified (V1, V2C, V3) for the candidate system (through a discovery filter, a seed file, or an Add Agent command), the discovery process only sends an SNMP request by using the specified version.

  Note Discovery filters support SNMP version V1 and V2C, but not V3.

  For an SNMPv1 or v2c request to a candidate system that has multiple read community strings, the discovery process sends multiple SNMP requests to the system simultaneously, where each request contains a different community string. You can specify multiple read community strings for a candidate system.

  The results of the ICMP and SNMP polling serve as input to the second phase of the discovery process.

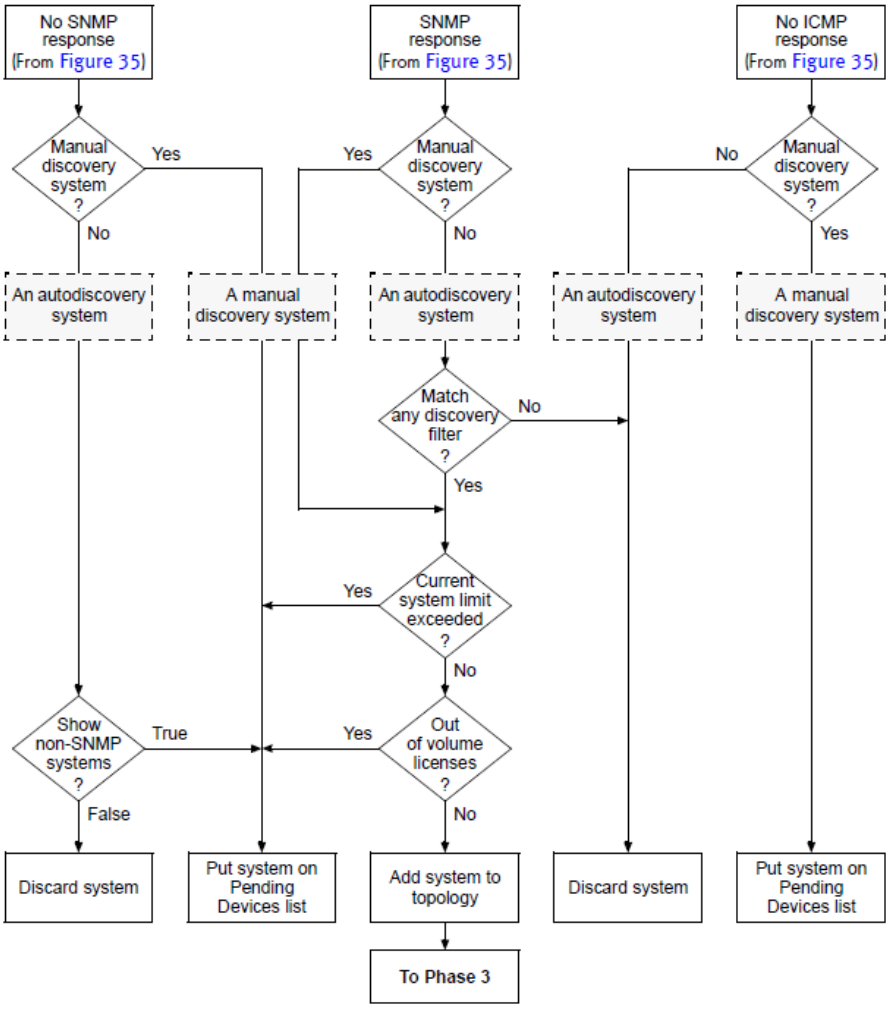
Phase 2: Determine the destination of the candidate system

The second phase of the discovery process, shown in Phase 2: Determining the destination of a candidate system, determines the destination of a discovery candidate system:

- Add to topology
- Place on the Pending Devices list

- Discard

Figure 5-14. Phase 2: Determining the destination of a candidate system



As shown in Phase 2: Determining the destination of a candidate system, the destination of the candidate system depends on many factors, including:

- Whether the candidate system is an autodiscovery or manual candidate system
- Whether the system passes ICMP or SNMP polling
- Whether the system matches a discovery filter
- Whether the current system limit is exceeded
- Whether any volume licenses are available
- Whether ShowPendingNONSNMP in the BASEDIR/smarts/conf/discovery/discovery.conf file is set to TRUE

Factors that affect the destination of a candidate system summarizes how these factors determine the destination of a candidate system when sufficient volume licenses are available.

Table 5-9. Factors that affect the destination of a candidate system

Candidate system	Initial polling		Match filter	Current system limit exceeded 1	ShowPendingNONSNMP	Destination
	ICMP	SNMP				
Autodiscovery	Passed	Passed	Yes	No	Not applicable	Topology
				Yes	Not applicable	Pending Devices list
			No	Not applicable	Not applicable	Discard
		Failed	Not applicable	Not applicable	TRUE	Pending Devices list
					FALSE (default)	Discard
	Failed	Not applicable	Not applicable	Not applicable	Not applicable	Discard
Manual Discovery	Passed	Passed	Not applicable	No	Not applicable	Topology
				Yes	Not applicable	Pending Devices list
		Failed	Not applicable	Not applicable	Not applicable	Pending Devices list
	Failed	Not applicable	Not applicable	Not applicable	Not applicable	Pending Devices list

1 50 by default. The VMware Smart Assurance IP Manager User Guide provides information on how to change the default.

The “Match filter” column in the table pertains to the matching of discovery-filter criteria other than “IP address.” An autodiscovery candidate system is subjected to two levels of discovery-filter matching: initially, a matching of IP address, and then a matching of other criteria after the candidate system passes ICMP or SNMP polling. The two levels of discovery-filter matching are described in VMware Smart Assurance IP Manager User Guide.

If a candidate system cannot be discovered and is not discarded, the IP Manager places the system on the Pending Devices list. The Pending Devices list is described in Pending Devices list.

Phase 3: Probe the system added to the topology

During the third phase of the discovery process, shown in Phase 3: Probing the system added to the topology (Sheet 1 of 2), each system that is destined for the topology is probed to gather additional information about the system, which includes the system’s components (such as cards, ports, interfaces, and so on), IP addresses, VLAN membership, environment sensors, system resources (such as memory, CPU, and so on), IP networks, netmasks, and neighboring systems. The IP Manager uses this information to create the topology for the system.

Figure 5-15. Phase 3: Probing the system added to the topology (Sheet 1 of 2)

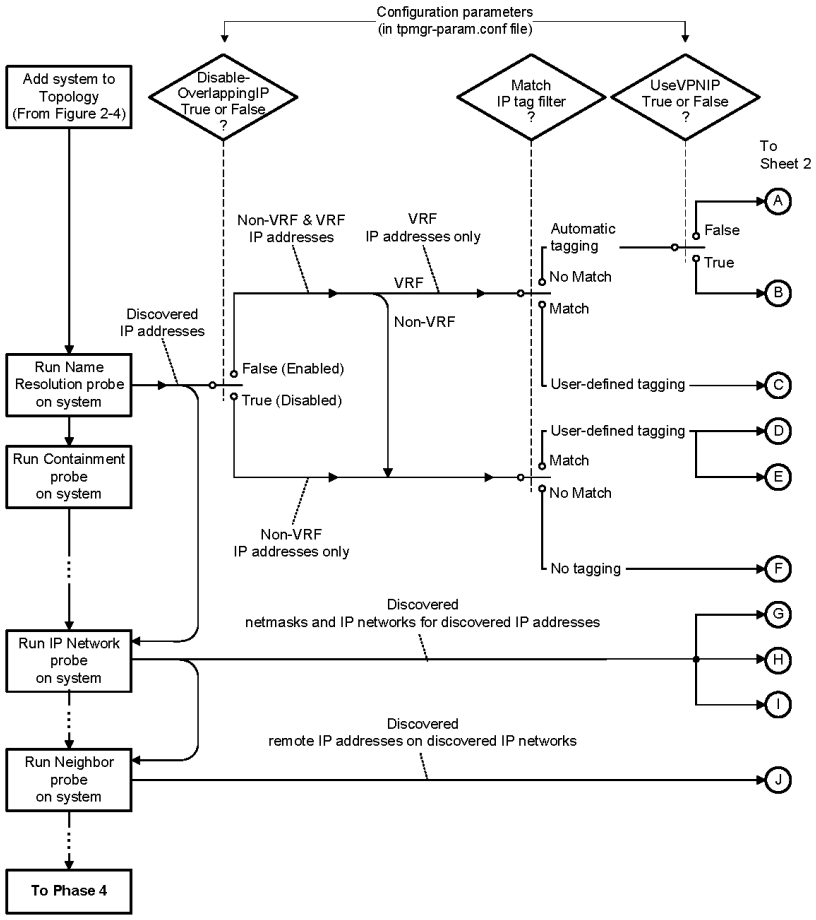
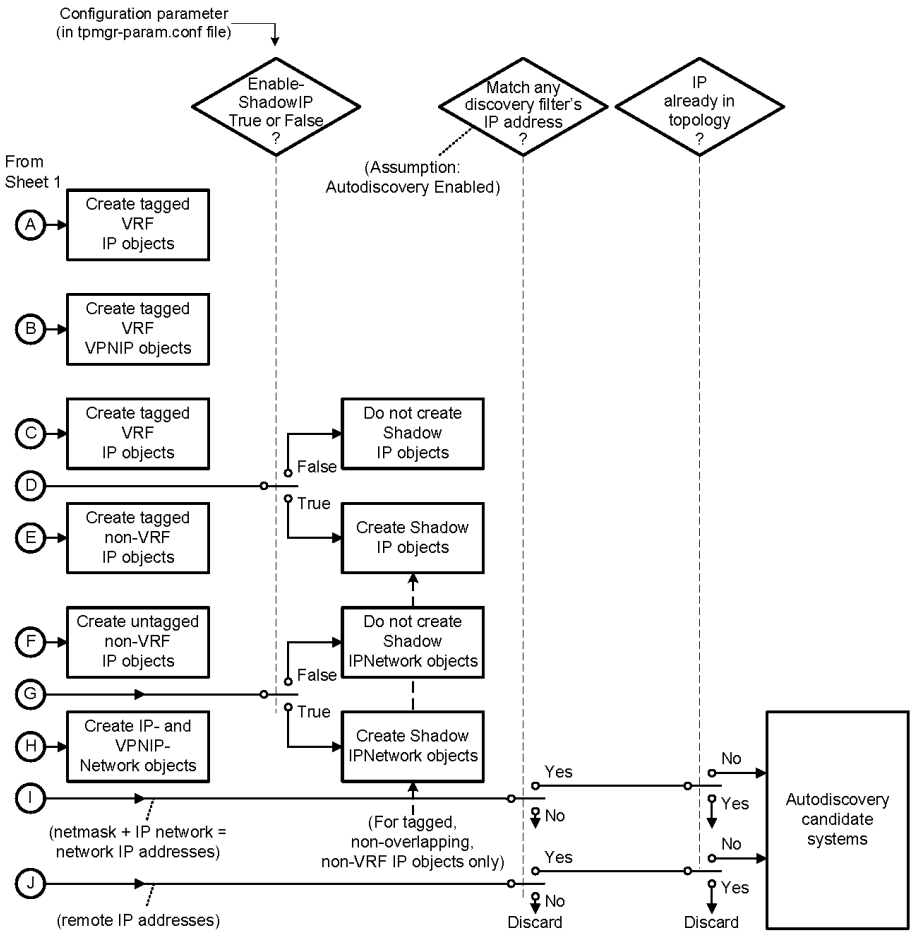


Figure 5-16. Phase 3: Probing the system added to the topology (Sheet 2 of 2)



Among other things, the network IP addresses that are formed from the discovered IP networks and netmasks serve as input to the autodiscovery process. Autodiscovery, if enabled, will compare each of these IP addresses against one or more discovery filters and add each IP address that matches a filter to the discovery queue. Discovery filters are described in the
VMware Smart Assurance IP Manager User Guide

Note Autodiscovery is available for IPv4 systems that have SNMPv1 or v2c agents.

The discovery process uniquely identifies the system within the topology and determines the system's level of certification. These tasks are accomplished by using the sysName, sysDescr, and sysObjectID information retrieved from the initial SNMP poll.

oid2type configuration files

The discovery process matches the sysObjectID to an OID in the oid2type configuration files to identify:

- System's type (Host, Router, Switch, and so on), vendor, model, and certification level.
- Which discovery drivers to use to probe the system.
- Which instrumentation classes to use to monitor the system.

  The discovery process searches for the drivers in the .import files, which are located in the BASEDIR/smarts/conf/discovery directory as the oid2type files.

Certification

Certification is a process by which VMware, Inc. ensures that the devices that are managed by the IP Manager are properly represented in the modeled topology, and that the devices are correctly monitored for connectivity and performance analysis. Devices are specific models of systems.

The certification process consists of three major parts:

- Discovery of the device, including the device's components (such as cards, ports) and endpoints (such as IP addresses, MAC addresses)
- Building of Layer 2 (data link) and Layer 3 (network) connectivity
- Verification of operational status of the device and its components

  The *VMware Smart Assurance IP Certification Matrix* identifies the vendor-specific models that are supported by the IP Manager.

Certification levels

A device may have one of three certification levels:

- CERTIFIED — VMware, Inc. has successfully tested a simulation of the device by using standard and proprietary MIBs that are supplied by the vendor.

  The IP Manager discovers the device by using standard MIB-II data and proprietary MIBs.

- TEMPLATE — VMware, Inc. assumes that the device supports some standard and proprietary MIBs, but VMware, Inc. has not tested the device.

  The IP Manager recognizes the object ID (OID), but has no information about the MIBs that this device supports.

  All field-certified devices are assigned a TEMPLATE certification level.

- GENERIC — VMware, Inc. does not have the OID for the device.

  The IP Manager does not recognize the OID and responds by classifying the certification level of the device as GENERIC, adding the device to the Node class, and analyzing the device by using MIB-II data.

  Note A certification level named VALIDATED still appears in the oid2type configuration files but is no longer used. VALIDATED has essentially the same meaning as CERTIFIED.

Standard and proprietary MIBs

The IP Manager uses proprietary MIBs in addition to standard MIBs because standard MIBs do not always contain the information that is needed by the IP Manager to discover and manage certain objects. Standard MIBs, for example, do not represent the processor or memory objects that are necessary for performance analysis. In these cases, proprietary MIBs offer more detailed information.

Note Proprietary MIBs are also known as enterprise, vendor-specific, or private MIBs. Standard MIBs are also known as generic MIBs.

Discovery probes and drivers

The probing of topology is accomplished by using the discovery drivers that are identified in the following .import files:

```
BASEDIR/smarts/conf/discovery/DISCOVERY_<vendor>.import
BASEDIR/smarts/conf/discovery/DISCOVERY_CUSTOM.import
```

The main drivers are identified in the DISCOVERY_<vendor>.import files, and the custom drivers, also known as discovery hook scripts, are identified in the DISCOVERY_CUSTOM.import files. VMware Smart Assurance IP Manager User Guide provides information about the custom drivers. By default, the custom drivers are not configured and therefore do not run.

The discovery process calls the discovery probes that are listed in Discovery probes, in the order that the probes are listed. Each probe will call one or more discovery drivers.

Table 5-10. Discovery probes

Probe name 1	Applicable to these systems	Available to these IP Manager products
1 Not all probes are listed in this table. 2 Performance probes and Host Resources probes are also known as Health probes. 3 Requires a feature license to enable the Performance probe. 4 Requires a feature license to enable the Host Resources probe.		
Name Resolution	All systems	All IP Manager products
Containment	All systems	All IP Manager products
VLAN	Bridges, switches	All IP Manager products
Bridge	Bridges, switches	All IP Manager products
HSRP	Routers	All IP Manager products
VRRP	Routers	All IP Manager products
IP Network	All systems	All IP Manager products
Virtual Router	Routers, switches	All IP Manager products
Neighbor	All systems	All IP Manager products
Performance 2	All systems	IP Availability and Performance Manager running as a single process, IP Performance Manager 3,IP Server Performance Manager 3
Host Resources 2	All systems	IP Server Performance Manager 4

The actual probes and associated discovery drivers that run for a specific device are specified in the oid2type configuration files. A specific device is identified by a unique OID. If a probe is not specified for a specific device, the IP Manager skips the probe.

The probes operate in series, one immediately following the other. In the event that one of the discovery probes is unable to complete, the discovery process does not stop. Instead, discovery proceeds with the next probe. Events that can prevent a discovery probe from completing are discussed in Discovery error scenarios and results.

Name Resolution probe

The Name Resolution probe calls multiple drivers, including:

- A driver that gathers IPv4 and IPv6 addresses and other MIB-II data for the system.
- A driver that gathers VPN routing and forwarding (VRF) IP addresses and route distinguisher tags from individual vendor MIBs on the system.
- A driver that provides a unique name for the system, as described in How discovery names a system.

  The Name Resolution probe discovers almost all of the IP and IPv6 addresses that are available for the system, including the VRF IP addresses if the DisableOverlappingIP configuration parameter is set to FALSE. Matrix of IP-related objects created by the Name Resolution probe summarizes the IP-related objects that are created by the Name Resolution probe.

Table 5-11. Matrix of IP-related objects created by the Name Resolution probe

Configuration parameters in tpmgr-param.conf file			
DisableOverlappingIP	**EnableShadowIP**	**VRF IP addresses discovered?**	**Shadow IP objects created?**
1 Shadow IP objects are created only for tagged, non-overlapping, non-VRF IP objects.			
VRF IP addresses			
TRUE (default)	Don't care	No	Not applicable
FALSE	Don't care	Yes	No
Non-VRF IP addresses			
Don't care	TRUE (default)	Not applicable	Yes 1
Don't care	FALSE	Not applicable	No

IP object tagging

All VRF and non-VRF IP objects that are created by the Name Resolution probe are subject to user-defined IP tagging *assuming* that the DisableOverlappingIP parameter is set to FALSE. The VMware Smart Assurance IP Manager User Guide provides instructions on creating IP tag filter groups and explains the naming scheme for applying tags to IP objects.

The IP objects have a Tag attribute. For a tagged IP object, the Name Resolution probe sets the Tag attribute to the object's tag value. For an untagged IP object, the Name Resolution probe leaves the Tag attribute empty.

Shadow IP object creation

When the EnableShadowIP parameter is set to TRUE (default), the Name Resolution probe creates a shadow IP object for each tagged, non-overlapping, non-VRF IP object. The probe uses the following naming scheme to name the shadow IP object:

IP-<*IP address*>

(Display name: [Shadow]:<*IP address*> [<system containing this IP>])

where <*IP address*> is the IP address of the IP object that the shadow IP object is shadowing.

The Name Resolution probe sets the IsShadow attribute of the shadow IP object to TRUE, to identify the object as a shadow IP object. The probe also creates a ShadowOf relationship between the shadow IP object and the IP object that is being shadowed.

Refer the VMware Smart Assurance IP Manager User Guide for more information on shadow IP objects.

Duplicate IP object creation

If no IP tagging is in effect and the Name Resolution probe finds an IP address that is already in the modeled topology, the probe responds as follows:

- Discards the overlapping IP address
- Removes the initially received overlapping IP address from the topology

- Creates a DuplicateIP object

  For a similar scenario in an IPv6 network, the Name Resolution probe will discard the overlapping IPv6 address, remove the initially received overlapping IPv6 address from the topology, and create a DuplicateIPv6 object.

IP address and MIB information distribution

All IP address and MIB information that is gathered by the Name Resolution probe for the discovered system is made available to all other discovery probes, including the IP Network probe.

Containment probe

The Containment probe discovers the components of a system, including ports, interfaces, MAC endpoints, and any cards or modules. It queries the following MIB variables:

- ifType
- ifDescr
- ifSpeed
- ifMtu
- ifPhysAddress
- ifAlias
- ifName
- Other variables

  If the Containment probe has access to vendor-specific MIBs, it queries additional variables to gather additional information, such as:

- Duplex mode (full-duplex, half-duplex) of ports and interfaces.
- Data link connection identifiers (DLCIs) for Frame Relay virtual connections.
- NIC-teaming ports and interfaces for teamed connections, as described in Discovery of NIC teaming.
- IPSec and Internet Key Exchange (IKE) interfaces for IPSec and IKE tunnels, as described in Support for IPSec and IKE tunnels.
- Virtual and real IP address and port information for load balancer virtual connections.

  For bridges or switches, the Containment probe might also query the dot1dBasePortIfIndex MIB variable.

VLAN probe

The VLAN probe collects VLAN information for a discovered bridge or switch, including VLAN identifiers, VLAN trunks, VLAN trunk ports, and VLAN port memberships. It uses vendor-specific MIB information.

The VLAN probe creates VLAN objects and Port or Interface object VLAN members, and uses the PartOf/ComposedOf relationship to associate the VLAN objects with their VLAN members.

All VLAN objects that are created by the VLAN probe are subject to user-defined VLAN tagging, which ensures that bridges and switches are assigned to the appropriate VLAN. The VMware Smart Assurance IP Manager User Guide provides instructions on creating VLAN tag groups, and explains the naming scheme for applying tags to VLAN objects.

Bridge probe

The Bridge probe collects bridge-forwarding tables and Spanning Tree Protocol (STP) information for a discovered bridge or switch. Discovery uses this information to create MAC objects, and to build the bridging relationships between MAC endpoints and ports. The Bridge probe uses standard and vendor-specific MIB information.

One Bridge probe is run for each VLAN that is discovered on a bridge or switch, to gather the bridge-forwarding and STP information for the VLAN.

HSRP probe

The HSRP probe collects information from a redundancy group of routers that are involved in an HSRP group. The routers use the proprietary Cisco Hot Standby Router Protocol to communicate with one another. The HSRP probe discovers the group number, the virtual IP address, and the virtual MAC address for an HSRP group; the priority number and status for each router endpoint that is participating in the group; and the currently active router in the group. The HSRP probe uses vendor-specific MIB information.

The HSRP probe creates HSRPGroup objects and HSRPEndpoint objects.

VRRP probe

The VRRP probe collects information from a redundancy group of routers that are involved in a VRRP group. The routers use the industry standard Virtual Router Redundancy Protocol to communicate with one another. The VRRP probe discovers the group number, the virtual IP address, and the virtual MAC address for a VRRP group; the priority number and status for each router endpoint that is participating in the group; and the currently active router in the group. The VRRP probe uses standard MIB information and creates VRRPGroup objects and VRRPEndpoint objects.

IP Network probe

For an IPv4 system, the IP Network probe examines the system's IP address table (ipAddrTable) to combine the IP addresses and the netmask for each interface on the system to create IPNetwork objects, and to build the connected relationships between IP endpoints and IP networks.

For an IPv6 system, the IP Network probe examines the system's IP address table to create IPv6Network objects, and to build the connected relationships between IPv6 endpoints and IPv6 networks.

For a dual-stack system, the IP Network probe examines the system's IP address table to create IPNetwork and IPv6Network objects, to build the connected relationships between IP endpoints and IP networks, and to build the connected relationships between IPv6 endpoints and IPv6 networks.

The IP Network probe uses standard MIB-II information to discover IP addresses.

The IP Manager attempts to discover and monitor IPv6 addresses on devices that implement and populate either of the following MIBs:

- IPV6-MIB (.1.3.6.1.2.1.55)

  or

- CISCO-IETF-IP-MIB (.1.3.6.1.4.1.9.10.86)

  For devices that are implementing other MIBs, the IP Manager discovers and monitors only IPv4 addresses.

IPNetwork object tagging

All VRF and non-VRF IPNetwork objects that are created by the IP Network probe are subject to user-defined IP tagging *assuming* that the DisableOverlappingIP parameter is set to FALSE. The VMware Smart Assurance IP Manager User Guide provides instructions on creating IP tag filter groups and explains the naming scheme for applying tags to IP objects and IPNetwork objects.

The IPNetwork objects have a Tag attribute. For a tagged IPNetwork object, the IP Network probe sets the Tag attribute to the object's tag value. For an untagged IPNetwork object, the IP Network probe leaves the Tag attribute empty.

Shadow IPNetwork object creation

When the EnableShadowIP parameter is set to TRUE (default), the IP Network probe creates a shadow IPNetwork object for each IPNetwork object to which a tagged, non-overlapping, non-VRF IP object is "ConnectedVia." The probe uses the following naming scheme to name the shadow IPNetwork object:

IPNET-<*IP network*>

(Display name: [Shadow]:<*IP network*>)

where <*IP network*> is the IP network of the IPNetwork object that the shadow IPNetwork object is shadowing.

The Name Resolution probe sets the IsShadow attribute of the shadow IPNetwork object to TRUE to identify the object as a shadow IPNetwork object. The probe also creates a ShadowOf relationship between the shadow IPNetwork object and the IPNetwork object that is being shadowed.

Autodiscovery IP address formation

The IP Network probe also combines each discovered IP network with its netmask to form network IP addresses, for input to the autodiscovery process. For example, consider the IP network 192.168.71.0 that has a netmask of 255.255.255.0. The IP Network will create network IP addresses in the range 192.168.71.<1-254> for this IP network and netmask. Any network IP address that matches a discovery filter and is not already in the modeled topology becomes an autodiscovery candidate system.

Virtual Router probe

The Virtual Router probe discovers the virtual routers in an IPv4 physical router or switch that contains virtual routers. The Virtual Router probe discovers the following additional information: the physical card on which the virtual router resides, if such information is available, and the virtual router's interfaces and their associated physical ports. The Virtual Router probe uses vendor-specific MIB information.

The Virtual Router probe creates a Router object for the physical router and for each virtual router, and creates a Switch object for the physical switch. The probe uses the ComposedOfVRs relationship to associate the virtual-router objects with the physical-router or physical-switch object.

Neighbor probe

The Neighbor probe collects information about a system's neighbors. It reads Cisco Discovery Protocol (CDP) tables, Extreme Discovery Protocol (EDP) tables, Foundry Discovery Protocol (FDP) tables, and other tables to discover remote addresses on the already discovered networks, and to discover ATM information. The Neighbor Probe uses vendor-specific MIBs.

The Neighbor probe passes the remote IP addresses to the autodiscovery process, which adds the IP addresses to its list of potential discovery candidate systems. The probe does not create IP objects for the remote IP addresses.

The discovery process checks to see if any of the remote IP addresses correspond to IP objects that are already in the repository. The discovery process will use this information during the final phase of discovery to create cable, trunk cable, and network connections.

Performance probe

The Performance probe collects performance-related information for system objects such as ports, interfaces, processors, memory, fans, power supplies, voltage sensors, and temperature sensors. The Performance probe uses standard and vendor-specific MIB information.

The Performance probe creates Processor, Memory, Fan, PowerSupply, VoltageSensor, and TemperatureSensor objects, and monitors statistics for the objects. For Port and Interface objects, which are created by the Containment probe, the performance monitoring might be the utilization that is based on the octets that are received and transmitted. For Memory objects, the performance monitoring might be the total amount of available memory, the total number of overrun buffers, and so on.

Host Resources probe

The Host Resources probe collects additional performance-related information for system objects. It collects metrics about host disks and file systems, as well as additional information about host processors and memory. The Host Resources probe uses standard and vendor-specific MIB information.

The Host Resources probe creates Disk and File System objects, and monitors statistics for the objects. It also monitors additional statistics for the Processor and Memory objects that are created by the Performance probe.

Discovered SNMP agent

In addition to discovering the system, the discovery process also discovers the system's SNMP agent and adds it to the modeled topology. The discovery process sets the following attributes for the SNMPAgent object:

- Sets the AgentAddress attribute to the IP address that is used by the initial ICMP poll to reach the system.
- Sets the AgentAddressList attribute to a list of IP addresses that are discovered on the system. Included in the list is the IP address that is set for the AgentAddress attribute.

  The order of addresses in the list is controlled by the DiscoveryAddrPref and AccessAddressFormat parameters in the discovery.conf file. The DiscoveryAddrPref parameter places the preferred version of addresses (IPv6 or IPv4) at the beginning of the list. And within a list of IPv6 or IPv4 addresses, the AccessAddressFormat parameter places the preferred types of addresses (loopback, Ethernet, and so on) at the beginning of the list.
- Sets the PortNumber attribute (161 by default) to the port number that is used by the first SNMP poll to successfully communicate with the SNMP agent.
- Sets the SNMPVersion attribute (AUTODETECT by default) to the version that is used by the first SNMP poll to successfully communicate with the SNMP agent.
- SNMPv1 or v2c only: Sets the ReadCommunity attribute (*public* by default) to the read community string that is used by the first SNMP poll to successfully communicate with the SNMPv1 or v2c agent.
- SNMPv3 only: Sets the attributes that are shown in Values used in SNMP polling to the values that are used by the first SNMP poll to successfully communicate with the SNMPv3 agent.

Table 5-12. Values used in SNMP polling

Attribute	Description
1 Retrieved from the SNMP polling of this SNMPv3 agent.	
User	Name of user that is included in the SNMP request that is sent to this SNMPv3 agent.
AuthPass	Password that is used as the basis for the following: The creation of the localized authentication key that is used with the selected authentication protocol and this SNMPv3 agent. The password is required if the authentication protocol is MD5 or SHA. The password may be encrypted (recommended) or appear as plain text.
AuthProtocol	Protocol that is used to authenticate the SNMP request that is sent to this SNMPv3 agent: MD5, SHA, or NONE (default).
PrivPass	Password that is used as the basis for the following: The creation of the localized encryption key that is used with the selected privacy protocol and this SNMPv3 agent. The password is required if the privacy protocol is DES or AES. The password may be encrypted (recommended) or appear as plain text.
PrivProtocol	Protocol that is used to encrypt the SNMP request that is sent to this SNMPv3 agent: DES, AES, or NONE (default).
EngineID	Identifier that uniquely identifies the SNMP engine of this SNMPv3 agent; for example, 800002b804616263.

Table 5-12. Values used in SNMP polling (Continued)

Attribute	Description
EngineBoots 1	Number of times that the SNMP engine of this SNMPv3 agent has initialized or reinitialized since its last configuration.
EngineBootTime 1	Date and time that the SNMP engine of this SNMPv3 agent last initialized or reinitialized.

- Sets the sysName, sysDescr, sysObjectID, sysContact, and sysLocation attributes to the sysName, sysDescr, sysObjectID, sysContact, and sysLocation values that are retrieved from the first SNMP poll to successfully communicate with the SNMP agent.
- Sets many other SNMP agent attributes in accordance to the values that are found by the discovery probes.

Discovered VR agents

For a physical router or switch that contains virtual routers, the discovery process also discovers the VR agents for the virtual routers. Because VRAgent is a subclass of SNMPAgent, the attributes of a VR agent are the same as the attributes of an SNMP agent with the following exception: The VR agent has an additional attribute called RouterID.

Discovered topology collection sets

For satellite Domain Managers, the discovery process also discovers and creates the topology collection sets that are described in the VMware Smart Assurance IP Manager Reference Guide. By default, the creation of Domain Manager topology collection sets is disabled.

Phase 4: Post-process the discovery information

After an IP Manager probes all seed-file and autodiscovered candidate systems, it performs postprocessing. Postprocessing tasks include the reconciliation of system data objects and the clean up of chassis, SNMPAgent, IP, IPv6, and DuplicateIP objects. The message “Postprocessing topology ...” displayed in the Discovery Status section of an IP Manager’s Discovery Progress window indicates when postprocessing tasks are in progress.

The postprocessing code in the IP Manager creates Layer 2 (data link) and Layer 3 (network) connectivity between the systems in the topology. The code bases the connectivity on IP networks, standard protocols, and vendor-specific discovery protocols.

The postprocessing functions that are performed by the IP Manager include:

- Mapping IP, IPv6, and MAC endpoints in the Address Resolution Protocol (ARP) cache to the appropriate systems. For non-SNMP systems, create IP, IPv6, and MAC endpoints and layer them over a physical interface. Because non-SNMP systems are discovered, the postprocessing code creates an artificial physical interface for each IP (v4 or v6) address that is associated with the system.
- Removing MAC endpoints that do not belong to any systems in the topology. Any MAC endpoint that is bridged by a port but is not hosted by any system is removed.

- Managing ports. Any port that is BridgedVia a MAC is set to managed.
- Converting IP objects to DuplicateIP objects. Any IP object that is discovered is converted to a DuplicateIP object.
- Converting IPv6 objects to DuplicateIPv6 objects. Any IPv6 object that is discovered is converted to a DuplicateIPv6 object.
- Resolving duplicate devices, as explained in VMware Smart Assurance IP Manager User Guide.
- Creating connections by using the collected bridge-forwarding information: create cables between ports and interfaces, create trunk cables between ports, and remove MAC endpoints that are bridged by access ports from the BridgedVia relationships of other ports.
- Creating trunk cables by using the collected Spanning Tree Protocol (STP) information.
- Creating network connections to represent WAN, or logical, connections.
- Creating any user-defined connections that are specified in the user-defined-connections.conf file.
- Creating partitions and naming them in accordance to the names that are specified in the user-defined partition.conf file.
- Creating redundancy group objects by combining group-member objects.
- Setting the DisplayNames for the discovered objects.

  The user-defined-connections.conf and partition.conf files are located in the BASEDIR/smarts/conf/discovery directory. The VMware Smart Assurance IP Manager Reference Guide describes the configuration parameters in these files and provides instructions on modifying the parameters.

More about the managing of ports

All ports that are imported are unmanaged.

The IP Manager sets a port to managed if the port has a connection. The port may be a trunk port or an access port. The IP Manager imports only those ports that are managed.

Furthermore, the IP Manager imports only managed objects and performs correlation and root-cause analysis on managed objects only. Managed and unmanaged objects are described in the VMware Smart Assurance IP Manager User Guide.

More about the conversion of IP objects to DuplicateIP objects

If multiple instances of the same IP address exist within the IP Manager domain, and when the IP Manager imports and merges the topologies, it replaces the multiple IP objects with a single DuplicateIP object. For a similar scenario in an IPv6 network, the IP Manager replaces the multiple IPv6 objects with a single DuplicateIPv6 object.

Information on how to handle DuplicateIP address notifications are described in the VMware Smart Assurance IP Manager User Guide.

More about the creation of redundancy groups

A redundancy group is a logical object that contains two or more objects that participate in a redundant configuration. Examples of redundancy groups are CardRedundancyGroup, HSRPGroup, VRRPGroup, and SystemRedundancyGroup.

IP Manager automatically creates CardRedundancyGroups during discovery for Cisco devices that support the CISCO-STACK-MIB. Each such group consists of two cards. You can view the card redundancy groups and their members by using the Domain Manager Administration Console. In addition, you can create redundancy groups for cards that do not support the CISCO-STACK-MIB.

Redundancy groups for network adapters, network connections, and systems must be manually created by using the dmctl utility or an Adapter Scripting Language (ASL). Network adapter, network connection, and system redundancy groups can be viewed through the Domain Manager Administration Console or the Topology Browser view of the Global Console.

During discovery, IP Manager automatically creates HSRPGroups for Cisco devices that support the CISCO-HSRP-MIB and are configured in an HSRPGroup. Each such group is composed of two or more routers and their interfaces. Also during discovery of the VRRP enabled routers, IP Manager creates a new redundancy group called VRRPGroup.

The VMware Smart Assurance IP Manager Reference Guide describes the redundancy group classes that can be instantiated by the IP Manager.

IP 8.1.1 discovers and monitors routers in VRRP configurations. Devices supported are Nortel devices, all devices which implement RFC 3768, Juniper ERX 700, 1400, 310, 320, and Nortel Alteon ACEdirector 4. VRRP discovery is enabled by default in IP 8.1.1 in the discovery.conf file (VRRPEnabled = TRUE).

The VMware Smart Assurance IP Manager User Guide provides more information on creation of redundancy groups.

Autodiscovery data sources

The Name Resolution, IP Network, and Neighbor probes contact the autodiscovery data sources to gather the topology and IP address table (ipAddrTable) MIBs of the seed systems to determine what other systems are known to the seed systems. As these systems are discovered by autodiscovery, they too are probed for candidate systems.

The autodiscovery process uses three different sources of information to discover candidate systems:

- Topology MIB
- IP address table MIB
- ARP table MIB

  By default, autodiscovery uses the topology and IP address table MIBs of the discovered systems (initially, of the discovered seed systems) because the MIBs are very reliable sources for topology information. The procedure for adding the ARP table MIB as a data source is presented in the VMware Smart Assurance IP Manager User Guide.

Topology MIBs data source

Topology MIBs are an important data source for discovering neighboring systems. When the topology MIB data source is enabled, autodiscovery uses the IP addresses from these sources to discover related systems. Examples of topology MIB data sources include the following:

- IP addresses stored in the Cisco Discovery Protocol (CDP) table
- IP addresses stored in the MIB-II atmInterfaceConfTable table

  In some cases, however, a system is not discovered for each IP address, if, for example, there is no route to the network from the host on which the IP Manager is running. This situation can occur when security restrictions are designed to prevent access between two separate networks.

IP address table MIB data source

The MIB-II IP address table lists all the IP interfaces of the system that is being discovered, as well as the subnet masks associated with these IP networks. By using this information, autodiscovery derives all of the possible IP addresses within each subnet. The Name Resolution probe pings each of these addresses to find candidate systems for subsequent SNMP probing.

ARP table MIB data source

The MIB-II ARP table provides a mapping of IP addresses to MAC addresses for any neighboring system that has recently communicated with the system with this MIB. However, because the ARP MIB is populated only when the system resolves the addresses of neighboring systems that are connected with a broadcast network protocol such as Ethernet, the ARP MIB is not appropriate for use in WANs, which typically use a non-broadcast protocol.

Discovery of NIC teaming

The IP Manager discovers and monitors network interface card (NIC) teaming between hosts and switches. NIC teaming provides network access redundancy and allows for rapid convergence when a failure occurs.

Note Setting up NIC teaming is device and vendor specific. Vendor specific documentation on how to set it up in a customer environment with different vendors is not provided here. For any device or vendor specific setup information, refer the relevant vendor specific documentation.

NIC teaming

NIC teaming is the grouping of two or more physical network interface cards (NICs) into a single logical NIC, for the purpose of achieving either or both of the following ends:

- Network fault tolerance

  High availability is maximized by binding two or more physical NICs to a logical NIC. If a NIC fails, the network connection continues through another NIC.

- Network traffic load balancing

  Balancing the network traffic load on a server enhances the functionality of the server and the network. The traffic is distributed amongst the members of a NIC team so that the traffic is routed on all of the teamed connections.

  While NIC teaming is possible for a number of different media types, the primary focus of the NIC teaming feature in IP Manager is Ethernet-media devices. In addition, for load balancing, the following conditions are assumed:

- The teamed connections terminate on the same remote device.
- The remote device is configured with an EtherChannel or an equivalent port-trunking protocol.

  A teamed NIC can have as many as eight NICs.

  In the IP Manager, NIC teaming support is applicable specifically to device specific discovery and occurs at three different points which are IP Availability Manager initialization, oid2Type configuration, and Discovery probes.

  In a team of two or more NICs, the secondary NIC assumes the primary or active role, if the primary NIC fails. Failover will not occur unless the primary NIC is specified. The NIC with the highest supported speed is chosen as the primary NIC. If more than one NIC runs at the highest supported speed, the last of these NICs added to the team becomes the primary. The NIC teaming functionality provides network access redundancy monitoring.

  NIC teaming discovery is enabled by default in the tpmgr-param.conf file (EnableTeamedNICDiscovery =TRUE). Discovery of Interfaces on Hosts is impacted due to support for NIC teaming discovery (for devices having OID .1.3.6.1.4.1.311.1.1.3.1.*).

  Upon discovery of the device (host) that is configured for NIC teaming, a redundancy group NetworkAdapterRedundancyGroup is created in the IP Manager. If the NetworkAdapterRedundancyGroup (NARG) is not created, either the EnableTeamedNICDiscovery flag is set to FALSE or the Host may be of a different vendor type other than Broadcom, Intel, or HP.

  Note The VMware Smart Assurance IP Manager Reference Guide provides more information on NIC teaming including attributes, root cause problems, and events for the NetworkAdapterRedundancyGroup.

Discovery MIBs

By probing the standard MIB-II tables on a host that employs a teamed NIC, the IP Manager discovers the logical NIC, but does not discover the physical NIC information that it needs to create the Layer 2 connectivity between the host and the neighboring switch or switches. To collect the physical NIC information, the IP Manager probes the enterprise MIBs on the host.

The collected physical NIC information contains the burned-in MAC (BIM) for each of the teamed-NIC access ports. A BIM uniquely identifies a teamed-NIC access port. As an aside, a teamed-NIC access port is *not* uniquely identified in the MIB-II tables; instead, in the MIB-II tables, all of the teamed-NIC access ports are identified by the same logical MAC address.

By using the collected physical NIC information, the IP Manager creates an instance of the NetworkAdapterRedundancyGroup to represent the logical grouping of the physical NICs. Doing so enables the IP Manager to create the Layer 2 cable connections between the host and the neighboring switch or switches.

Vendor support

The NIC teaming feature in IP Manager supports the following vendors:

- Broadcom
- Intel
- Compaq/HP

  Although a mixture of vendors (for example, Broadcom and Intel) can be involved in a teamed NIC, only single-vendor teamed NICs have been tested for IP Manager 8.1.1.

Redundancy group creation

Upon discovering a device that is configured for NIC teaming, the IP Manager creates an instance of the NetworkAdapterRedundancyGroup. A NetworkAdapterRedundancyGroup instance consists of two or more ports or interfaces that participate in NIC teaming.

NetworkAdapterRedundancyGroup has a ComposedOf relationship with Interface, where the Interface class is used to represent both teamed-NIC ports and interfaces. Interface has a PartOf relationship with NetworkAdapterRedundancyGroup.

The VMware Smart Assurance IP Manager Reference Guide provides more information about the NetworkAdapterRedundancyGroup, including its attributes and the events that the IP Manager generates for a NetworkAdapterRedundancyGroup.

Discovery and short discovery of IPSec tunnels

The IP Manager discovers IPSec tunnels and Internet Key Exchange (IKE) tunnels between IPSec-enabled routers on IPv4 networks. It uses Router, Interface, and IP object types to model the tunnels.

During a full discovery, the IP Manager reads the Routing Table of IPSec-enabled routers to discover the terminating subinterfaces (WAN endpoints) of IPSec tunnels and IKE tunnels. The IP Manager layers the subinterfaces over the underlying physical interfaces and creates the network connections between the local and remote subinterfaces to form the IPSec and IKE tunnels.

By default, the IP Manager is not enabled to discover IPSec tunnels.

Refer the VMware Smart Assurance IP Manager User Guide on how to enable IPSec tunnel discovery and also on how to modify parameters in the tpmgr-param.conf, discovery.conf, and the runcmd_env.sh files.

Monitoring IPSec tunnels

The IP Manager monitors the tunnels by monitoring the IsDownOrFlapping attribute of the WAN Interface objects that are associated with the tunnels. The WAN Interface objects represent the terminating subinterfaces of the tunnels.

Rediscovering IPSec tunnels

When the IP Manager detects a change in the IsDownOrFlapping attribute of a WAN interface that is associated with a tunnel, it places the host router on the short discovery queue for a short discovery for the following reason:

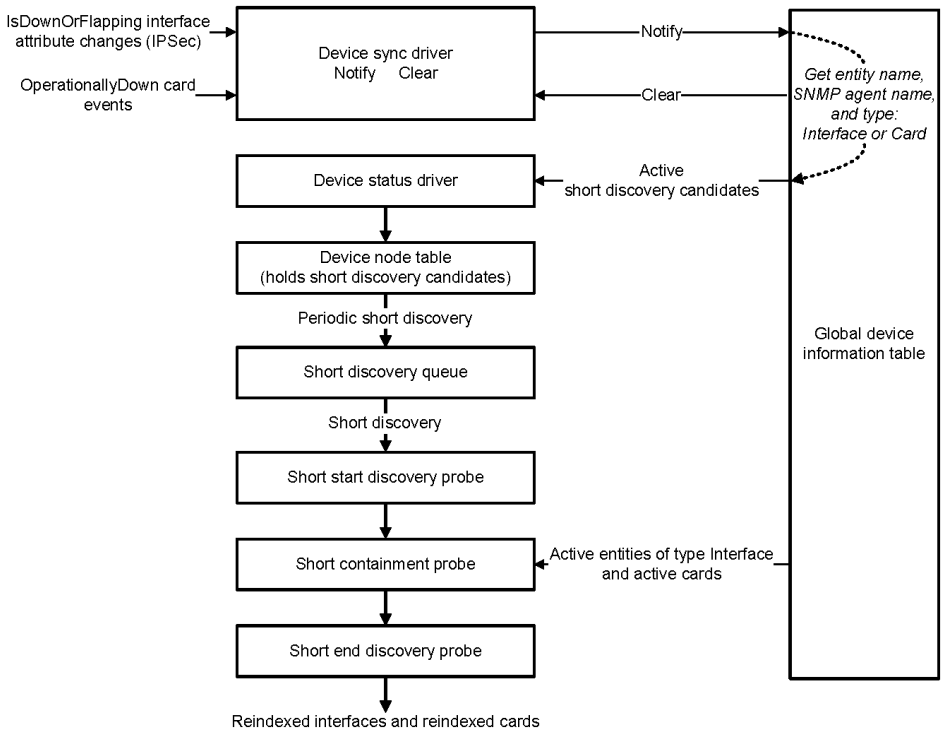
IPSec and IKE tunnels are dynamic in the sense that they can be rerouted to other WAN interfaces when an interface fails. Therefore, when the IP Manager detects a change in the IsDownOrFlapping attribute, it has no way of knowing whether the relationship between the tunnel and the associated WAN interface, as known to the IP Manager, is still valid.

By updating the dynamic indices of the interfaces on the host router, the IP Manager is able to recreate the tunnel association with the WAN interface.

Short discovery process

The short discovery process runs in parallel with the full discovery process and has its own discovery queue and its own timer interval. Short discovery process shows how the short discovery process works.

Figure 5-17. Short discovery process



IsDownOrFlapping interface attribute changes and OperationallyDown card events serve as input to the short discovery process. Periodically, the device node table is scanned for short discovery candidates, and any candidates that are found in the table are placed on the short discovery queue. The short discovery probes contact the candidates to update the dynamic indices of their interfaces and their cards for monitoring purposes.

Only the IsDownOrFlapping interface attribute changes pertain to IPSec tunnels and IPSec-enabled routers. The OperationallyDown card events pertain to cards that reside in switches.

Short discovery characteristics

Short discovery has the following characteristics:

- The device node table in Short discovery characteristics is consulted frequently (every 900 seconds, or 15 minutes, by default).
- Short discovery has its own short discovery threads, just as full discovery has its own discovery threads. By default, the number of short discovery threads is 7.
- Short discovery is performed only on routers or switches that are already in the topology. Short discovery of IPSec tunnels is performed on routers, and short discovery of cards is performed on switches.
- Short discovery changes only the attributes of already discovered router or switch objects.
- Short discovery does not perform postprocessing or codebook recomputation.

Note The VMware Smart Assurance IP Manager User Guide provides information on how to enable short discovery, change the short discovery interval, or the number of short discovery threads.

Analyzing IPSec tunnels

Here is a summary of how IPSec tunnels are discovered and monitored:

- The IP Manager discovers the WAN subinterface endpoints of the tunnels and layers them over the underlying physical interfaces.
- The IP Manager monitors the tunnels by monitoring the IsDownOrFlapping attribute of the subinterface endpoints that are associated with the tunnels and then creates the network connections between the local and remote subinterface endpoints to form the IPSec tunnels.

  “Network adapters and device connections” on page 32 provides information about physical interfaces, subinterfaces, network connections, and WANs.

  The IP Manager then uses the discovered physical interfaces, subinterfaces, and network connections, in conjunction with the subinterface statuses that are obtained from the monitoring of the IsDownOrFlapping attribute, to diagnose IPSec tunnel failures. For each physical interface, and depending on the symptoms, the IP Manager diagnoses one or more NetworkConnection::DownOrFlapping impact events, and diagnoses either an Interface::Down problem or one or more NetworkConnection::Down problems:

- Interface::Down, if the physical interface is operationally down, which explains why all of the subinterfaces that are layered over the physical interface have failed, along with any peer interfaces of those subinterfaces. “Interface” in “Interface::Down” represents the physical interface.
- NetworkConnection::Down, if the physical interface is operationally up, for each network connection failure at the subinterface level.

  Network connections at the subinterface level represent WAN circuits that are carrying IPSec tunnels.

  As a precondition for IPSec tunnel analysis, add the following line to the BASEDIR/smarts/conf/local runcmd_env.sh file:

  SM_LOGICALCONNECTIONDOWN=DISABLED

  The procedure and reason for completing this task are given in the VMware Smart Assurance IP Manager User Guide. The Interface::Down problem, NetworkConnection::Down problem, and NetworkConnection::DownOrFlapping impact events are described in the VMware Smart Assurance IP Manager Reference Guide.

Discovery and the local name resolution service

The discovery process uses the name resolution service that is provided by the local operating system to translate hostnames to IP addresses, and to translate IP addresses to hostnames. On a UNIX system, a name resolution service might be Domain Name System (DNS), NIS or NIS+, or a hosts file. Important information about name resolution services is presented in *System hostname does not resolve to an IP address*.

SM_IP_VERSIONS environment variable

If a discovery candidate system is presented as a hostname *without* an explicit IP version suffix (:v6v4, :v4v6, :v6, or :v4), the SM_IP_VERSIONS environment variable controls how the local name resolution service resolves the hostname to a corresponding IP address. By default, SM_IP_VERSIONS is set to v6v4. For v6v4, the name resolution service resolves a hostname to an IPv6 address, or if that resolution fails, to an IPv4 address.

The hostname syntax and the SM_IP_VERSIONS environment variable are described in the VMware Smart Assurance IP Manager Reference Guide.

DiscoveryAddrPref parameter

The DiscoveryAddrPref parameter controls the order of addresses in the list of IP addresses that are found on a discovered system, where the preferred addresses (IPv6 or IPv4) are first in the list. By default, DiscoveryAddrPref is set to IPV6FIRST_IPV4NEXT. For IPV6FIRST_IPV4NEXT, IPv6 addresses (if any) are first in the list, followed by IPv4 addresses.

The DiscoveryAddrPref parameter influences how the local name resolution service names a discovered system, as explained in Configuring automatic name resolution. When IPv6 is the preferred address, the global unicast address and the addresses that begin with 3FFE:* take precedence.

The DiscoveryAddrPref parameter is in the discovery.conf file and is described in the VMware Smart Assurance IP Manager Reference Guide.

Hostname resolution and IP address checking

The discovery process uses the name resolution service that is provided by the local operating system to translate hostnames to IP addresses, and to translate IP addresses to hostnames. On a UNIX system, a name resolution service might be Domain Name System (DNS), NIS or NIS+, or a hosts file.

The discovery of a candidate system starts with a hostname, which is provided by the user, or an IP address, which is provided by the user or is automatically discovered. If the system is identified by a hostname, the discovery process calls the local name resolution service to resolve the hostname to one or more IP addresses. If the hostname does not resolve, the discovery process places the system, which is identified by its hostname, on the Pending Devices list of the IP Manager.

Note The Pending Devices list is described in Pending Devices list.

If the candidate system is identified by a user-provided IP address, the discovery process checks whether the IP address conforms with the IP address format. If the IP address does not conform, the discovery process places the system, which is identified by its invalid IP address, on the Pending Devices list.

If the candidate system is identified by an automatically discovered IP address, the discovery process checks whether the IP address matches an IP address of any discovery filter. If the IP address does not match the IP address of any discovery filter, the discovery process discards the IP address.

How discovery names a system

When a system is discovered, the Name Resolution probe runs a script to determine a name for the system. The goal is to:

- Provide a unique name for the system.
- Use a consistent naming convention to ensure that the system is assigned the same name, if, for example, the system is discovered by using a different IP address.

- Update the name during a rediscovery if the name of the system has changed.

  The name that is assigned to a system is determined by the setting of the NameFormat parameter in the name-resolver.conf file, which is located in the BASEDIR/smarts/conf/discovery directory. The assigned name becomes the value of the system's Name attribute.

  Note The values of a system's Name attribute and DisplayName attribute may be different, depending on the setting of the DisplayNameFormat parameter in the discovery.conf file. The VMware Smart Assurance IP Manager Reference Guide describes the configuration parameters in the discovery.conf file and provides instructions on modifying the parameters.

  The NameFormat parameter has two options:

- TM_USEAUTONAME (default)
- TM_USESEEDNAME

  The options control how the Name Resolution probe assigns a name to a discovered system.

TM_USEAUTONAME option for automatic name resolution

When NameFormat is set to TM_USEAUTONAME, the Name Resolution probe makes calls to the local name resolution service and runs a script that resolves a name automatically by considering the following naming sources:

- Loopback IP address
- MIB-II sysName
- Public IP address
- Private IP address
- SNMP agent address

  By default, the naming sources are considered in the order shown. You can change this order to create your own prioritized list of naming sources.

  The procedure for configuring the TM_USEAUTONAME option is described in the VMware Smart Assurance IP Manager User Guide.

TM_USESEEDNAME option for seed-file name resolution

When the NameFormat is set to TM_USESEEDNAME, the Name Resolution probe uses the following keyword and value that are defined for individual systems in the IP Manager's seed file to name the systems:

SEEDNAME=<*user-specified name*>

Note Seed files are also described in the VMware Smart Assurance IP Manager User Guide.

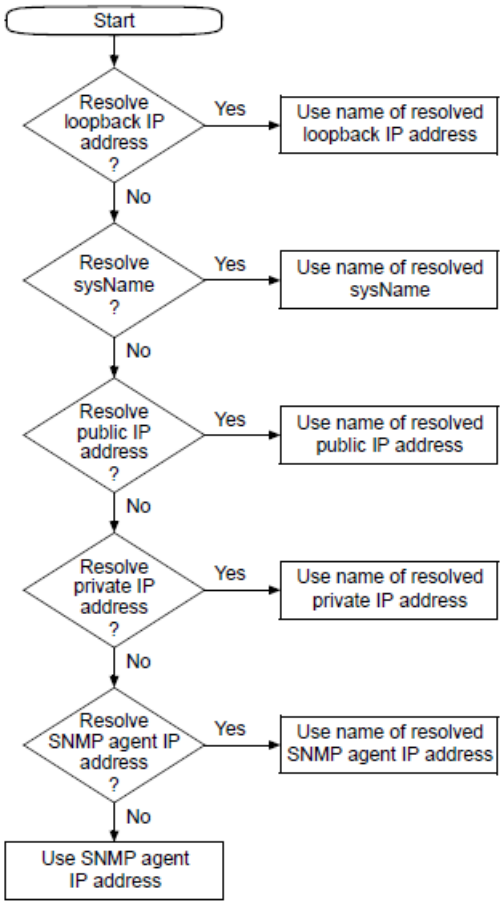
If the SEEDNAME keyword is not defined for a system, the Name Resolution probe uses the original seed name, which is the hostname or IP address by which the system was introduced to the discovery process, to name the system. For an autodiscovered system, the seed name is the first non-private IP address on the system. The TM_USESEEDNAME option is useful when the topology information is provided by an external source.

The procedure for configuring the TM_USESEEDNAME option is described in the VMware Smart Assurance IP Manager User Guide.

Configuring automatic name resolution

The TM_USEAUTONAME option in the name-resolver.conf file enables you to configure the Name Resolution probe to automatically resolve system names by using the naming sources in the order that you specify. By default, the Name Resolution probe attempts to resolve system names by using the naming-source sequence that is shown in Default order of naming sources when determining a system's name.

Figure 5-18. Default order of naming sources when determining a system's name



The DiscoveryAddrPref parameter in the discovery.conf file influences in the following way how the Name Resolution probe automatically names a system:

Determines whether the preferred loopback IP address, public IP address, private IP address, or SNMP agent IP address is IPv6 or IPv4.

By default, DiscoveryAddrPref is set to IPV6FIRST_IPV4NEXT. Also, in IPv6 terminology, a “public IP address” is referred to as a “global unicast IP address.”

Preparing for manual discovery

Seed systems serve as the starting point for building the IP Manager topology. Routers and switches are good seed system candidates because their MIBs contain many addresses and much information about additional systems in the network.

By using manual discovery without autodiscovery, the IP Manager will discover just the seed systems. The seed systems are presented to the IP Manager as entries in a seed file, rather than being presented individually by using the **Add Agent** command. A comprehensive seed file can only be created if the network topology information is complete and accessible. The VMware Smart Assurance IP Manager User Guide provides information on the **Add Agent** command.

Note The Federal Information Processing Standard (FIPS) Publication 140-2 is a U.S. government computer security standard governing cryptographic modules and is required for any software purchased by the US government and US military. FIPS 140 mode allows you to use only SNMPv1, SNMPv2c, and SNMPv3 except MD5 and DES authentication protocols. If you run SNMPv1 or SNMPv2c, then FIPS 140 mode has no impact. If you are using encryption, then you must use a certified encryption library and only certain encryption routines. When you discover an SNMPv3 device, you need to select the option “V3” in the “Add Agent” window. The “Authentication Protocol” option lists only SHA and not MD5 and the “Privacy Protocol” option lists only AES and not DES. This is because MD5 and DES are not supported in FIPS 140 mode. When you discover SNMPv3 devices with MD5 and DES protocol as seed, the devices go to the Pending List and display as “Invalid” or “Unsupported SNMP V3 protocol”.

“Support for FIPS 140-2 in IP 9.0” section in the VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers and the VMware Smart Assurance IP Manager Release Notes provide more information on the FIPS 140 mode implementation.

In situations where the network topology information is incomplete, unavailable, or constantly changing, manual discovery is combined with autodiscovery. The seed file can be much smaller because the seed systems that are specified for manual discovery also serve as a starting point for autodiscovery. Autodiscovery automatically discovers the managed network from the seed systems.

For manual discovery, or manual discovery that is combined with autodiscovery, you create a seed file by editing the BASEDIR/smarts/conf/seedfile template in the IP Manager installation area. The VMware Smart Assurance IP Manager User Guide provides the procedure for creating seed files.

Manual discovery can be used to discover SNMPv1, v2c, or v3 systems in IPv4 or IPv6 networks.

Select autodiscovery data sources

The IP Manager probes the topology and IP address table MIBs of the seed systems to find IPv4 addresses of additional systems. Those IP addresses become candidate systems for the autodiscovery process.

The autodiscovery process can use three different data sources of information to discover candidate systems:

- Topology MIB
- IP address table MIB
- ARP table MIB

  The data sources are described in Autodiscovery data sources. By default, discovery uses the topology and IP address table MIBs of the discovered systems (initially, of the discovered seed systems) because these MIBs are very reliable sources for topology information.

Considerations when selecting autodiscovery data sources

Here are some things to consider when selecting which data sources are to be used for autodiscovery:

- Topology MIBs

  If you disable the topology MIB data source, autodiscovery may not be able to find all the systems in the managed network.

- IP address table MIB

  By default, the IP Manager pings the addresses in networks with an 8-bit host address (254 hosts), to prevent the indiscriminate pinging of large numbers of addresses. If necessary, you can change the default 8-bit host address setting to a larger value by editing the MaximumHostBits parameter in the discovery.conf file. The VMware Smart Assurance IP Manager Reference Guide describes the configuration parameters in the discovery.conf file and provides instructions on modifying the parameters.

  Depending on your network configuration, it can take the IP Manager a long time to ping the entire list of addresses when the MaximumHostsBits is set to a large value, such as 16 bits (more than 65,000 addresses). With a value of 24 bits, for example, it could take the IP Manager up to several hours to ping all the available addresses.

- ARP table MIB

  By default, this data source is not enabled because autodiscovery does not typically gain many IP addresses by using the ARP table MIB. In addition, autodiscovery does not read the ARP table MIB in hosts.

Initiating manual or autodiscovery

The IP Manager provides two options for initiating a manual discovery. Both options are initiated through the Domain Manager Administration Console:

- Option 1: Import a seed file that lists one or more seed systems to be added to the topology. This option is best for adding a set of systems.
- Option 2: Add seed systems to the topology one at a time by using the **Add Agent** command. This option is best for adding a small number of systems.

  When autodiscovery is enabled, use either option to initiate the autodiscovery process. When enabled and initiated, autodiscovery will occur whenever either a full discovery occurs or a system on the Pending Devices list is successfully discovered.

Using the Add Agent command to initiate discovery

The Add Agent command is used to enter individual seed systems. You can specify a seed system as a hostname or as an IP address:

- For a hostname, you can specify a hostname with or without an explicit IP version suffix (:v6v4, :v4v6, :v6, or :v4).

  Here is an example hostname entry that includes a v4 suffix: Core-ROUTER-1:v4.

  If you do not include an explicit IP version suffix, the SM_IP_VERSIONS environment variable controls how the local name resolution service resolves the hostname to a corresponding IP address. By default, SM_IP_VERSIONS is set to v6v4.
- For an IP address, you can specify an IPv6 or IPv4 address.

  If you specify an IPv6 address, enclose the address in square brackets; for example: [2001:0db8::0010].

  Here is an example IPv6 address entry that includes a port number: [2001:0db8::0010]:200.

  Be careful not to add an overlapping IP address as a seed system.

  Hostname syntax, the SM_IP_VERSIONS environment variable, and overlapping IP addresses are discussed in the VMware Smart Assurance IP Manager User Guide.

Stopping autodiscovery

You can stop the autodiscovery process from the Discovery Progress window. (If the Discovery Progress window is not open, select Topology > Show Discovery Progress in the Domain Manager Administration Console to open the window.) In the Discovery Progress window, click Stop Auto Discovery to stop autodiscovery.

Clicking the Stop Auto Discovery button stops the finding of candidate systems for autodiscovery during Phase 3 of the discovery process. However, any systems already in the discovery queue continue to be processed, and the progress of this processing appears in the Discovery Progress window.

Stopping autodiscovery also disables future autodiscovery action until autodiscovery is re-enabled. The VMware Smart Assurance IP Manager User Guide provides information on how to re-enable autodiscovery.

Initiating postprocessing

Postprocessing uses the data collected during discovery probing to create the connectivity between systems. The information collected from various probes is processed to create the connections between systems in the topology.

Phase 4: Post-process the discovery information provides more information on postprocessing.

About discovery filters

The purpose of discovery filters is to control or limit which systems are added to the discovery process during autodiscovery. Appropriate discovery filtering ensures that autodiscovery is efficient.

You use discovery filters to ensure that certain network-accessible systems that are not intended for management by the IP Manager are not discovered. For example, an Internet service provider (ISP) might want to discover a router and its interfaces, but not the systems that are connected to the router on the ISP's client side. Configuring the appropriate discovery filters will enforce this discovery restriction.

Seed systems in a seed file, or seed systems that are specified by using the **Add Agent** command, are not subject to the discovery filters. Only the systems that are found by the autodiscovery's probing of the seed systems are subject to the discovery filters.

The VMware Smart Assurance IP Manager User Guide provides information on all the tasks related to the creation of discovery filters.

Inclusive-type filtering

Discovery filters are inclusive-type filters, which mean that an autodiscovery candidate system must match an IP address of a discovery filter before it is eligible for discovery. Thus, you must create at least one discovery filter to use autodiscovery.

To prevent specific systems from being discovered, you can use the ipExcludeList parameter in the BASEDIR/smarts/conf/discovery/discovery.conf file to create exclude-type filters. The ability to create exclude-type filters simplifies the creation of the discovery filters.

Manual accept mode and system limit

You can set a manual accept mode option for a discovery filter to redirect any discovered systems that match the filter to the IP Manager's Pending Devices list. That way, you can view the systems on the Pending Devices list and then select which of them should be added to the modeled topology. The procedure to specify the manual accept mode option is described in the VMware Smart Assurance IP Manager User Guide.

In addition, you can set a system limit option for a discovery filter to limit how many discovered systems that match the filter are added to the topology. The procedure to specify the system limit option is described in the VMware Smart Assurance IP Manager User Guide.

IPv4 and SNMPv1/v2c support

Autodiscovery and discovery filters are applicable to the discovery of SNMPv1 or v2c systems in IPv4 networks. Manual discovery and seed files are applicable to the discovery of SNMPv1, v2c, or v3 systems in IPv4 or IPv6 networks.

Discovery filter fields

Since the IP Manager discovery is based on candidate systems that are represented as IP addresses, discovery filters are based on IP addresses. All that is known about a candidate system at the beginning of the autodiscovery process is its IP address.

Core fields

You use five fields which are referred to as core fields to specify a discovery filter. The five core fields are identified in Core fields of a discovery filter.

Table 5-13. Core fields of a discovery filter

Discovery filter fields	Discovery filter example
1 The System Name and System Description fields are case-sensitive. 2 The asterisk (*) wildcard character is described in Wildcard patterns	
IP Address Range	*.*.*.* (matches any IP address) 2
System Name 1	* (matches any system name) 2
System Description 1	* (matches and system description) 2
SystemOID	* (matches any system object ID) 2
System Types	Switch

When a candidate system's IP address matches an IP address of a discovery filter, the IP Manager uses the candidate system's IP address and the default values of the following options to poll the candidate system:

- SNMP port number (161 by default)
- SNMP version (AUTODETECT by default)
- Access mode (ICMPSNMP by default)
- Read community string (*public* by default)

  These options are described in the VMware Smart Assurance IP Manager Reference Guide. The polling of a candidate system is shown in Core fields.

Extended options

In addition to the five core fields of a discovery filter, you can specify additional information for a filter to control the polling of any system that matches an IP address of the filter. The additional fields, which are referred to as extended options, are identified in Discovery filter that contains extended options.

Table 5-14. Discovery filter that contains extended options

Discovery filter fields	Discovery filter example
1 SNMPv3 is not currently supported for autodiscovery.	
Core fields	
IP Address Range	*.*.*.*
System Name	*
System Description	*
SystemOID	*
System Types	Router
Extended option fields	
SNMP Port	165
SNMP Version 1	AUTODETECT
Access Mode	ICMPSNMP
Community String	public1

When a candidate system's IP address matches an IP address of an extended discovery filter, the IP Manager uses the candidate system's IP address and the values of the following user-specified extended options of the filter to poll the candidate system:

- SNMP Port

  Enables you to specify an alternate port on which systems receive SNMP requests. The default value is 0, which indicates that the IP Manager uses the port number value (161) of the defaultSNMPPort parameter in the discovery.conf file.

- SNMP Version

  Enables you to control the version of SNMP that the IP Manager uses to communicate with the system. Valid values for autodiscovery are V1, V2C, and AUTODETECT. The default is AUTODETECT, which means that discovery tries SNMPv2c first, and then SNMPv1 if necessary.

  If you specify an explicit SNMP version, V2C or V1, the IP Manager tries only the SNMP version that you specified. If the explicit version is different from the actual version that is supported by the system, the IP Manager generates a DiscoveryError with a message of "No response from SNMP Agent."

- Access Mode

  Determines which protocols are used to both discover and monitor the system and its components. Valid values for autodiscovery are ICMPSNMP and ICMPONLY. The default is ICMPSNMP, which means that both ICMP and SNMP protocols are used to discover and manage the system.

  When ICMPONLY is selected, discovery does not send SNMP polls to the system and cannot, therefore, discover the system and its components. Instead, the system is added to the modeled topology as a Host, and the system's SNMP agent is added to the topology as an SNMPAgent. The host is monitored only for connectivity by sending ICMP polls to the host.

  Note the following:

  - Even if an ICMPONLY system does not have an SNMP agent, an SNMPAgent object is created for the system. The SysObjectID attribute of the SNMPAgent object is empty.
  - When an ICMPONLY system has multiple IP (v4 or v6) addresses that resolve to the same name, all IP addresses are retained. You must manually delete any IP address that no longer belongs to the ICMPONLY system.
  - An additional Access Mode option, SNMPONLY, is available to a system that is imported from a seed file, or is added to the topology by using the **Add Agent** command.

- Community String

  Enables you to specify one or more read community strings that should be used to contact systems that match this filter. If no community strings are specified, autodiscovery uses the default read community string. Read community strings provides information about default community strings.

Discovery filter considerations

Discovery filters are inclusive-type filters, which mean that if you do not create any filters, no candidate systems are accepted. A system is accepted only if it matches a filter.

Consider the following points before creating discovery filters:

- When creating a filter, you must specify a value for each of its five core fields. The procedure for creating a discovery filter is given in the VMware Smart Assurance IP Manager User Guide.
- You can create multiple filters and arrange them in a high-to-low priority. A candidate system is compared against the highest priority filter first.
- A candidate system is subjected to two levels of filter matching, as described in Discovery filter matching.
- When a candidate system matches all five core fields of a filter and passes any additional autodiscovery safeguards, the system bypasses any lower priority discovery filters and becomes an object in the topology.
- To exclude a specific candidate system, specify the system in the ipExcludeList in the discovery.conf file.

- Seed systems, as listed in a seed file or entered through the **Add Agent** command, are not subject to the discovery filters.

  For systems that match a discovery filter and are added to the topology, they too are probed for the IP addresses of their neighboring systems. The autodiscovery cycle continues until no more new IP addresses match the discovery filters.

Discovery filter matching

A candidate system is subjected to two levels of filter matching:

- First-level filter matching

  Matching of IP address in the first core field of a filter, to determine whether the system is eligible for discovery.

- Second-level filter matching

  Matching of other criteria in the remaining four core fields of a filter after the candidate system passes ICMP and SNMP polling, to determine whether the system should be added to the modeled topology.

First-level filter matching

If a candidate system does not match the IP address of any discovery filter, the IP Manager discards the system.

For a candidate system that matches the IP address of a discovery filter, the IP Manager uses the SNMP port, SNMP version, access mode, and read community string values that are associated with the filter to poll the candidate system. For a non-extended filter, the polling values are identified in Core fields. For an extended filter, the polling values are identified in Extended options.

If the polling of the candidate system fails, the IP Manager either discards the system or places the system on the Pending Devices list. If the ICMP polling fails after the allotted number of ICMP retries, the IP Manager discards the candidate system. If the SNMP polling fails after the allotted number of SNMP retries, the IP Manager responds in one of two ways, depending on the value of the ShowPendingNONSNMP parameter in the discovery.conf file:

- If ShowPendingNONSNMP is FALSE (default), the IP Manager discards the candidate system.
- If ShowPendingNONSNMP is TRUE, the IP Manager places the candidate system on the Pending Devices list.

  The ShowPendingNONSNMP parameter is described in the VMware Smart Assurance IP Manager Reference Guide.

Second-level filter matching

If the polling of the candidate system is successful, the IP Manager reads the SNMP response to learn the system name (sysName), system description (sysDescr), system OID (sysObjectID), and other system-related information of the candidate system.

The IP Manager then performs the following tasks to determine whether the candidate system matches the filter:

- Compares the retrieved system description against the System Description field in the filter.
- Compares the retrieved system OID against the SystemOID field in the filter.
- Maps the retrieved system OID to an OID in the oid2type files to identify the system's type, and then compares that type against the "System Types" field in the filter.
- Passes the candidate system's IP address to the local name resolution service to resolve the address to a name, and then compares that name against the "System Name" field in the filter.

  The filter matching is shown in Phase 2: Determine the destination of the candidate system.

  If the candidate system matches the filter, the system enters Phase 3 of the discovery process. If the candidate system does not match the filter, the system is compared against the core fields of the next lower priority filter.

  Whether the candidate system matches this filter or another filter in the stack of discovery filters, the IP Manager assigns to the system the SNMP port, SNMP version, access mode, and read community string values that the IP Manager used to successfully poll the system.

  If a candidate system does not match the core fields of any discovery filter, the IP Manager discards the system.

Examples of discovery filters

A discovery filter can be broad to discover a broad range of systems, or restrictive to limit the number or type of discovered systems.

For a filter's IP address range, system name, system description, and system OID fields, you can use wildcards to create a matching pattern. As an example, in Examples of discovery filters, the asterisk (*) in the IP address range, system name, system description, and system OID fields is a wildcard character that matches any arbitrary string of characters, including an empty string. The discovery filter in Examples of discovery filters will match any candidate system that is a Switch. Wildcard patterns discusses wildcard syntax.

The following examples illustrate different types of discovery filters.

Filtering a range of IP addresses

To exclude a particular range of systems, for example, 22 in the second octet of an IP address range, use a filter that includes every other possibility, as shown in Filtering by IP address. For the System Types field, select all system types.

Table 5-15. Filtering by IP address

Discovery filter fields	Value
IP Address Range	172.<16-21>.55.* | 172.<23-31>.55.*
System Name	*

Table 5-15. Filtering by IP address (Continued)

Discovery filter fields	Value
System Description	*
SystemOID	*
System Types	<select all system types>

In general, you should use the ipExcludeList parameter in the discovery.conf file, not a discovery filter, to exclude specific systems.

Filtering by system type, name, and description

To narrow the range of a filter, specify criteria for multiple fields, as shown in Filtering by system type, name, and description. Only those systems that are routers, within a specified IP address range, whose system name begins with *ny*, and whose system description begins with *Cisco,* match this filter.

Table 5-16. Filtering by system type, name, and description

Discovery filter fields	Value
IP Address Range	172.21.55.*
System Name	ny*
System Description	Cisco*
SystemOID	*
System Types	Router

Filtering with manual accept mode

To manually accept each candidate system, enable manual accept mode, which is described in the VMware Smart Assurance IP Manager User Guide. In the example that is shown in Filtering with accept mode, routers and switches that match Filter 1 are automatically added to the topology. Hosts, however, that match Filter 2 are automatically added to the Pending Devices list.

Table 5-17. Filtering with accept mode

Discovery filter fields	Filter 1 values	Filter 2 values
IP Address Range	172.21.55.*	172.21.55.*
System Name	*	*
System Description	*	*
SystemOID	*	*
System Types	RoutersSwitches	Hosts
Ask before adding new systems	Unchecked	Checked

Filtering by system OID

To limit discovered systems by vendor, you can filter by the enterprise OID. The filter that is shown in Filtering by OID matches only systems of type Switch that have an OID of .1.3.6.1.4.9, which is the enterprise OID for systems that are made by Cisco Systems, Inc.

Table 5-18. Filtering by OID

Discovery filter field	Value
IP Address Range	*.*.*.*
System Name	*
System Description	*
SystemOID	.1.3.6.1.4.1.9.*
System Types	Switch

Discovery results

When the IP Manager attempts to discover a candidate system, one of the following occurs:

- Discovery successfully completes, and the system and its components are added to the modeled topology.
- Discovery starts but communication is lost between the IP Manager and the system.
- Discovery starts but the system does not respond to the discovery request.
- Discovery starts but one of the discovery probes encounters an error during the discovery process.
- Discovery starts but cannot complete because of an insufficient number of volume licenses.
- Discovery is unable to start or successfully complete.

Successful discovery

When a candidate system is successfully discovered, the IP Manager creates an object in its repository for the system and for each of the system's components.

Unsuccessful discovery

If the discovery of a candidate system does not complete, the IP Manager places the system on the Pending Devices list. Refer Pending Devices list for information about the Pending Devices list and the Discovery Progress window in which it appears.

Unsuccessful rediscovery

If the rediscovery of a system does not complete, the IP Manager creates a DiscoveryError notification for the system, and that notification appears in the Notification Log Console view of the Global Console. Rediscovering a system is the discovery of a system that is already in the IP Manager repository.

Discovery error scenarios and results

Discovery error scenarios and resultsdescribes different discovery error scenarios and results. Initial discovery refers to an attempt to discover a system that has not been previously discovered, and thus is not in the modeled topology. Rediscovery refers to an attempt to discover a system that is already in the topology.

Table 5-19. Discovery error scenarios and results

Discovery or rediscovery	Condition	Result
Discovery (initial discovery)	Cannot resolve hostname to IP address.	Add system to Pending Devices list. *System hostname does not resolve to an IP address* provides more information.
	No response from ICMP ping.	For a seed system (manual discovery), add system to the Pending Devices list as UNDISCOVERED. For a system that is found through autodiscovery, discard system.
	No response from SNMP agent.	For a seed system (manual discovery), add the system to the Pending Devices list as UNDISCOVERED. For a system that is found through autodiscovery, and ShowPendingNONSNMP set to TRUE, add the system to the Pending Devices list as UNDISCOVERED. For a system that is found through autodiscovery, and ShowPendingNONSNMP set to FALSE, discard the system. *System does not support SNMP* provides more information.
	ICMP response and SNMP response; current system limit exceeded.	For a seed system (manual discovery) or a system that is found through autodiscovery, display System limit exceeded message in the Discovery Progress window and add the system to the Pending Devices list as UNDISCOVERED.
	ICMP response and SNMP response; manual accept mode enabled for discovery filter.	For a system that matches the discovery filter, add the system to the Pending Devices list as THROTTLED.
	Insufficient number of volume licenses, which prevents complete discovery of systems.	Stop discovery process and send an OutOfLicense notification. Show “Out of license” message in the Discovery Progress window. Add any new discovery candidate systems to the Pending Devices list as UNDISCOVERED.
	Discovery completes successfully.	Classify system in accordance to its sysObjectID and add to topology. If the OID is not recognized, classify system as a Node.

Table 5-19. Discovery error scenarios and results (Continued)

Discovery or rediscovery	Condition	Result
Rediscovery	Receive no response from SNMP agent: SNMP request times out.	Send notification, typically Agent NotResponding, which is based on the correlation analysis, and send a DiscoveryError notification.
	SNMP agent encounters a loop, which prevents the system from being discovered.	Classify system according to its sysObjectID, add to topology, and send a DiscoveryError notification.
	No response from ICMP ping.	Send notification, typically Down, and send a DiscoveryError notification.
	Discovery not able to contact SNMP agent at known address.	Send a DiscoveryError notification.
	SNMP agent fails authentication.	Send a DiscoveryError notification.

Discovery errors

An error that occurs during the discovery of a candidate system results in the placing of the system on the Pending Devices list. Each system entry on the Pending Devices list has an accompanying comment that describes the discovery error.

In most cases, the IP Manager writes discovery errors to its log file. The log file is named <IP Manager instance name>_<locale>_<character encoding>.log (for example, INCHARGE-AM_en_US_UTF-8.log) and is located in the BASEDIR/smarts/local/logs directory in the IP Management Suite installation area.

In some cases, the IP Manager prints a message, in the Discovery Progress window just below the Discovery Status section, that explains the discovery error.

The following is a representative and not a complete list of discovery errors:

- System hostname does not resolve to an IP address
- SNMP agent violates SNMP protocol
- System does not support SNMP
- Incorrect read community string
- Insufficient number of volume licenses

System hostname does not resolve to an IP address

For a discovery candidate system that is identified by its hostname, the hostname might resolve to multiple IP addresses. If the IP Manager cannot reach the system by using one of the first several IP addresses that the name resolution service finds, the IP Manager does not discover the system. Instead, the system is placed on the Pending Devices list, and subsequent attempts to discover it will also fail.

Discovery cannot resolve the hostname for a system when one of the following is true:

- The name resolution service returns more IP addresses than the operating system allows.

- None of the IP addresses that are returned by the operating system result in the IP Manager's reaching the system's SNMP agent.

  If you are using a seed file, you can solve the hostname resolution problem by simply replacing a system's hostname with an accessible IP address of the system's SNMP agent. If making this replacement is impractical because your seed file is generated automatically, read the following discussions of name resolution sources.

UNIX name resolution sources

All supported UNIX systems return up to 35 addresses for a given name. To avoid the name resolution problem, you must configure your system so that at least one usable IP address is among the 35 addresses that are returned.

UNIX systems translate names by using one of three sources: DNS, NIS or NIS+, or a hosts file. The sources that are supported on a given system are specified by a host configuration file.

Note The IP Manager relies on the host operating system for name resolution. If a system is not configured for DNS, then there is no way to make the IP Manager use DNS other than making the system that the domain is running on use DNS.

The most controllable source is the hosts file, /etc/hosts. All UNIX implementations that VMware Corporation personnel have tested will return up to the first 35 translations in order of appearance in the file.

DNS translations are more difficult to control. First, the translations are often updated automatically. Second, the translations are sorted automatically, based on both internally defined and externally specified criteria.

VMware Corporation personnel have not been able to determine with certainty how translations that are made through NIS or NIS+ are ordered.

SNMP agent violates SNMP protocol

The IP Manager implementation of the SNMP protocol is strict in that the implementation only sends or receives SNMP messages that conform to the SNMP standard. Unfortunately, not all SNMP implementations are as strict. The result is that the IP Manager might receive non-conforming SNMP messages. In some cases, the IP Manager can successfully interpret and handle non-conforming messages.

One such case is when an SNMP agent returns the values of an SNMP request in the wrong order. For example, the IP Manager requests the values for var1, var2, and var3, and the SNMP agent returns var1, var3, and var2.

The IP Manager writes a message to its log file:

```
SWFE-W-Agent violates lexicographic ordering in response to GET_NEXT: Agent:172.16.1.94, First OID:
    .1.3.6.1.2.1.17.4.3.1.2.0.48.171.12.15.223
```

Although this problem does not hinder discovery, you might want to contact the system vendor to see if a fix is available for the vendor's implementation of the SNMP protocol.

By default, the IP Manager is configured to accept additional non-conforming SNMP messages because, in the BASEDIR/smarts/local/conf/runcmd_env.sh file, the value of the SM_SNMP_BUG_COMPATIBLE environment variable is set to TRUE. The VMware Smart Assurance System Administration Guide provides information about environment variables.

System does not support SNMP

Certain systems, such as desktop computers or printers, do not usually support SNMP. The discovery process assumes that a system supports SNMP. If a system does not support SNMP, the system is either not discovered or is placed on the Pending Devices list with a description that the SNMP request failed.

By default, systems that are found by autodiscovery and that do not support SNMP are removed from the discovery queue. Systems that are imported from a seed file, or added through the Add Agent command, that do not support SNMP are always added to the Pending Devices list.

If you want to include autodiscovered, non-SNMP systems in the modeled topology, set the value of the ShowPendingNONSNMP parameter to TRUE in the discovery.conf file.

When the value of ShowPendingNONSNMP is set to TRUE, candidate systems that do not support SNMP are placed on the Pending Devices list. *Managing systems on the Pending Devices list* provides information about managing systems on the Pending Devices list.

When autodiscovery is enabled, setting ShowPendingNONSNMP to TRUE can result in a large number of non-SNMP systems in the Pending Devices list.

Incorrect read community string

For errors that concern an incorrect read community string, you should examine the read community string and correct the string if necessary. In the Notification Log Console, you can determine which community string is being used by double-clicking the notification and viewing the Details tab in the Notification Properties dialog box:

For manual discovery, a read community string can be specified in the seed file or in the Add Agent dialog box for each seed system, or one or more read community strings can be specified in the Topology tab of the Domain Manager Administration Console. "Adding and removing read community strings" on page 152 provides information about how to change community strings.

For autodiscovery, one or more read community strings can be specified in a discovery filter or in the Topology tab of the Domain Manager Administration Console.

For Cisco devices, do not configure the community string with a "@" character.

After correcting the read community string, add the system to the topology with the Add Agent command. You do not have to restart autodiscovery or re-import the entire seed file just to discover this one system.

Insufficient number of volume licenses

A license file for the IP Management Suite may contain three types of licensing:

- Application licensing
- Feature licensing
- Volume licensing

  Volume licensing requires a license for each discovered system.

  Each IP Manager in the IP Manager deployment will retrieve blocks of volume licenses from the license server as needed when discovering systems. If an IP Manager discovers more systems than it is licensed to discover, it will generate the OutOfLicense notification to indicate that no more volume licenses are available and that no additional systems will be discovered.

  When the OutOfLicense is notified, the IP Manager will complete the discovery of any system and its components in the midst of being discovered, which includes all virtual routers that are configured for a router or a switch that implements virtual routers. When a router or switch that is implementing virtual routers is discovered, the chassis-based router/switch is counted as a system, and each virtual router is counted as a system. Any new discovery candidate systems are added to the Pending Devices list in the UNDISCOVERED state, and the message “Out of license” appears below the Discovery Status section of the Discovery Progress window.

  If you receive an OutOfLicense notification, request for a new license file that has a greater number of volume licenses. The
  VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers provides the procedure to install a license file and the VMware Smart Assurance System Administration Guide provides more information about license management.

DiscoveryError notifications

Discovery errors are generated for systems that support SNMP. When an error is encountered during the rediscovery of such a system, the IP Manager notifies a DiscoveryError. If the AccessMode for a system is ICMPONLY and the system is not reachable by ICMP during discovery, the IP Manager does not notify a DiscoveryError.

To obtain more information about a DiscoveryError notification, attach the Global Console to the Global Manager and double-click the notification in the Notification Log Console view to open the Notification Properties dialog box. Select the Details tab to see the DiscoveryErrorInfo attribute. A Details tab appears for the IP Manager that notified the event.

The following is a list of errors that result in a DiscoveryError notification:

- SNMP request times out
- SNMP agent loops
- System Down

- Qualified access address not found
- System previously discovered fails authentication
- Explanation modes differ

SNMP request times out

A common error is that an SNMP request times out before it completes. There are a number of reasons why this error might occur:

- The system might be busy and therefore drops SNMP packets.
- The network might be busy and therefore drops or delays SNMP packets.
- A large bridge forwarding table might be causing the SNMP agent not to respond in time. Discovery uses the bridge forwarding table to determine connectivity.

  If the error occurs because the system or network is busy, you should rediscover the system when the system or network is less busy.

  If you notice an excessive number of timeouts during the discovery process, you should consider increasing the timeout values for discovery polling.

SNMP agent loops

If a system has an SNMP agent with an improper MIB implementation, SNMP requests from the discovery process might loop over a table. Looping over a table is a violation of the standard protocol and prevents the discovery process from retrieving the necessary information to fully discover a system.

When the IP Manager encounters such a problem, it notifies a DiscoveryError and writes an error message to its log file. Both the DiscoveryError notification and the error message identify the OID where the agent loop occurred. As an example, the following error message identifies “.1.3.6.1.4.1.2021.11.11” as the OID where an agent loop occurred:

```
[25-Jun-2009 16:57:54+047ms EDT] t@60 Discovery #9
SWFE-W-ELOOP-Agent loops in response to GET_NEXT: Agent: 10.1.8.195, First
    OID: .1.3.6.1.4.1.2021.11.11
```

You can also invoke the **sm_snmp** command or the **sm_snmpwalk** command from the BASEDIR/smarts/bin directory to identify the location of an agent loop. The **sm_snmp** command is used to browse through a specific portion of the MIB, and the **sm_snmpwalk** command is used to browse through the entire MIB.

Here is an example of using the **sm_snmp** or **sm_snmpwalk** command to identify the location of an agent loop:

```
sm_snmp -d 10.1.8.195 get .1.3.6.1.4.1.2021.11.11
sm_snmpwalk 10.1.8.195
```

You should see an error message similar to the following:

```
SNMP-E-Agent 10.1.8.195, port 161 loops in response to
   Get(.1.3.6.1.4.1.2021.11.11)
```

The **sm_snmp** command is described in the HTML pages that are located in the BASEDIR/smarts/doc/html/usage directory of any Domain Manager installation area.

Upon encountering an agent loop, the discovery process continues with the next probe until the discovery process completes.

If the error is the result of a problem with the system's MIB, you should contact the system vendor to see if an update is available.

System Down

In some cases, discovery fails because the system is Down (unreachable). When this situation occurs, the IP Manager notifies a DiscoveryError and lists the DiscoveryError as an impact of the Down problem. The value of the DiscoveryErrorInfo attribute is "No response from Ping."

Qualified access address not found

This error occurs during rediscovery of a system when no IP address object exists in the topology for an SNMP agent, or each IP address that is associated with the SNMP agent is one of the following:

- An unmanaged IP address
- An IP address within the range of the ipExcludeList parameter in the discovery.conf file
- An IP address that is associated with an interface that is either in BACKUP or ONDEMAND mode

  Because the IP Manager has no valid IP address to access the SNMP agent, it notifies a DiscoveryError.

System previously discovered fails authentication

If, during a rediscovery, the SNMP agent of a system fails authentication, the IP Manager notifies a DiscoveryError. In the Notification Log Console, you can determine the reason for the error by double-clicking the DiscoveryError notification and viewing the Details tab in the Notification Properties dialog box. Alternatively, in the Topology Browser Console, you can determine the reason for the error by double-clicking the name of the failed system and viewing the Attributes tab in the system's properties dialog box.

For an SNMPv1 or v2c agent, the most likely reason for this error is that the read community string for the agent has changed. *Incorrect read community string* provides information about errors that concern an incorrect read community string.

For an SNMPv3 agent, the most likely reason for this error is that the authentication password, authentication protocol, or username for the agent has changed. " *Seed file format* provides information about SNMPv3 options.

Pending Devices list

The Pending Devices list, which appears in the Discovery Progress window, lists systems that are waiting to be discovered by the IP Manager. The IP Manager attempts to discover the systems on the Pending Devices list during the next pending discovery interval, or when a pending discovery is manually invoked. The VMware Smart Assurance IP Manager User Guideprovides information about the pending discovery interval and pending discovery.

To view the Pending Devices list, attach the Domain Manager Administration Console to the IP Manager and select Topology >Show Discovery Progress.

To view the Pending Devices list, attach the Domain Manager Administration Console to the IP Managerand select Topology >Show Discovery Progress. Pending Devices list that contains multiple pending systemsis an example of a Pending Devices list.

1 Pending

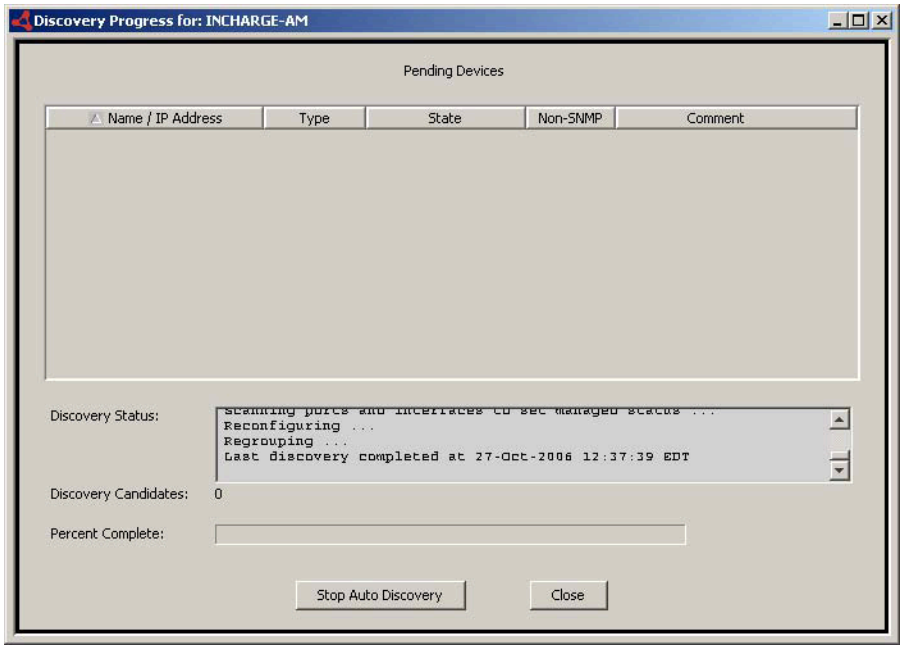
2 Devices

3 list

4 Discovery

5 Status

6 messages



Name / IP Address	Type	State	Non-SNMP	Comment
172.16.1.109	Host	UNDISCOVERED	☐	No response from Ping
172.16.1.14	Host	THROTTLED	☐	Hardware: x86 Family 6 Model 11...
172.16.1.15	Host	UNDISCOVERED	☐	No response from Ping
172.16.1.66	Host	UNDISCOVERED	☐	No response from Ping
172.16.19.222	Host	THROTTLED	☐	Hardware: x86 Family 6 Model 8 ...
172.16.19.224	Host	THROTTLED	☐	Hardware: x86 Family 6 Model 8 ...
172.16.193.180	Host	THROTTLED	☐	
172.16.193.218	Host	THROTTLED	☐	
172.31.254.170	Host	THROTTLED	☐	
172.31.254.171	Host	THROTTLED	☐	
BOOM	Host	UNDISCOVERED	☐	No response from SNMP agent

Figure 5-19. Pending Devices list that contains multiple pending systems

A system is added to the Pending Devices list when one of the following occurs:

- The system does not respond to ICMP polls during a discovery.
- The system does not respond to SNMP polls during a discovery.
- The system is processed by autodiscovery, and the manual accept mode is enabled. The VMware Smart Assurance IP Manager User Guideprovides information about manual accept mode.
- The number of discovered systems exceeds the current system limit. The VMware Smart Assurance IP Manager User Guideprovides information about setting the current system limit.
- The number of discovered systems exceeds the number of volume licenses that are available to the IP Manager. *Insufficient number of volume licenses* provides information about volume licenses and the OutOfLicense notification.
- The system does not support SNMP, which means that the system responded to ICMP polls but the discovery process could not contact an SNMP agent for the system.

- The IP Manager receives an SNMP trap that indicates that the configuration of the system might have changed. A system is placed on the Pending Devices list when one of the following traps is received:
  - coldStart (RFC 1215 MIB)
  - warmStart (RFC 1215 MIB)
  - a3ComSysModuleCardInsertEvent (3COM CB9000-MIB)
  - cefcFRUInserted (CISCO-ENTITY-FRU-CONTROL-MIB)
  - cefcFRURemoved (CISCO-ENTITY-FRU-CONTROL-MIB)
  - ciscoConfigManEvent (CISCO-CONFIG-MAN-MIB)
  - sysConfigChangeTrap (CISCO-STACK-MIB)

Information provided by the Pending Devices list

The Pending Devices list contains five columns that provide the following information for each system:

- Hostname or IP address of the system that could not be discovered.
- Type, which is one of the system types
- State is either UNDISCOVERED or THROTTLED, as described in Discovery states.
- Non-SNMP is a checkbox for indicating that the system does not support SNMP.
- Comment describes why the system is on the Pending Devices list, or contains the system's SysDescr value of the system if the current system limit has been exceeded.

  Discovery states describes the discovery states and the actions you can take to resolve the situation.

Table 5-20. Discovery states

Discovery state	Description	Available actions
UNDISCOVERED	Indicates that the system was not successfully discovered. Another attempt to discover the system will occur when the Pending Devices list is processed during the next pending discovery interval.	■ Designate the system as one that does not support SNMP. The system will be reclassified as a Host during the next pending discovery interval. ■ Rediscover the system. ■ Remove the system.
THROTTLED	Indicates that the system was autodiscovered with manual accept mode enabled, and placed on the Pending Devices list.	■ Accept the system and it will be discovered at the next full discovery, pending discovery, or rediscovery. ■ Remove the system.

Managing systems on the Pending Devices list

Depending on a system's state, you can invoke one of the following four actions on a system on the Pending Devices list. Right-click the system to see the available actions:

- Accept

  Pertains to systems in the THROTTLED state, which means that the system matched an autodiscovery filter that is configured for manual accept mode. If you accept the system, the IP Manager changes the state of the system to UNDISCOVERED.

- Remove

  Available for all systems on the Pending Devices list. Selecting this action removes the system from the Pending Devices list. If you remove a system, the IP Manager prompts you to confirm the removal.

- Mark Non-SNMP

  Available for all systems on the Pending Devices list. When you designate a system as Non-SNMP, the IP Manager reclassifies the system as a Host and adds it to the topology at the next discovery interval *assuming* that the system responds to the IP Manager's ICMP request. Systems that are marked as non-SNMP are treated similarly to systems that are marked as ICMPONLY.

  Once a system has been marked Non-SNMP in the Pending Devices list, you cannot change its status. You must rediscover the system manually by using the **Add Agent** command or by importing the system from a seed file.

- Rediscover

  Initiates a discovery of the system.

Fine-tuning parameter settings in discovery.conf

The discovery.conf file contains configuration parameters that you can edit to improve performance during topology discovery. The VMware Smart Assurance IP Manager Reference Guide describes these parameters and provides instructions on modifying them. The VMware Smart Assurance IP Manager User Guide provides helpful recommendations for setting the ICMP and SNMP polling configurations.

About certifying managed devices

After your network has been discovered, you might want to keep in your topology certain managed devices that are not yet certified by VMware Corporation. Because no certification is available for these devices, the IP Manager discovery process uses generic SNMP MIB-II drivers to discover the devices, and models them as instances of the Node class. You can, however, add such devices to a field certification file so that the IP Manager properly recognizes these devices.

You can add managed devices to the field certification file, oid2type_Field.conf, which is located in the BASEDIR/smarts/conf/discovery directory. User-made changes to this file will not be overwritten by subsequent IP Management Suite Service Packs.

Field-based certifications are intended for certifying simple devices, or for certifying devices that are clones of devices that have already been certified by VMware Corporation. Only VMware field engineers, partners, or knowledgeable customers should add device entries to the oid2type_Field.conf file.

Upon reading this section, you will know how to create field-based certifications. You will also know how to read and parse the certifications in any of the oid2type configuration files that are shipped with the IP Management Suite. All oid2type files are located in the BASEDIR/smarts/conf/discovery directory.

Syntax of device entries in oid2type_Field.conf

The syntax of device entries in the oid2type_Field.conf file is the same as the syntax of device entries in any of the oid2type files, except that the CERTIFICATION level is set to TEMPLATE instead of CERTIFIED or VALIDATED. Here is the syntax of a device entry in the oid2type_Field.conf file:

```
# One or more commented-out lines that identify the device
<sysObjID> {
    TYPE = <unitary computer system class>
    VENDOR = <manufacturer>
    MODEL = <model name>
    CERTIFICATION = TEMPLATE
    <discovery driver keyword> = <driver parameter name>
    .
    .
    .
    <discovery driver keyword> = <driver parameter name>
INSTRUMENTATION:
    <instrumentation keyword> = <instrumentation set>
    .
    .
    .
    <instrumentation keyword> = <instrumentation set>
}
```

Each entry contains the system object ID (OID) of a device, followed by several fields that are enclosed in a pair of braces ({ }). The fields describe the device.

A device entry maps the system object ID of a device to a list of existing discovery drivers that the IP Manager should use to discover the device, and to a list of existing instrumentation sets that the IP Manager should use to monitor the device.

Device entry examples

The following three example device entries field-certify a Cisco router, a VisualNetworks probe, and an Eastern-Research switch:

```
# Cisco Universal Broadband Router 924
.1.3.6.1.4.1.9.1.255 {
    TYPE = Router
    VENDOR = Cisco
    MODEL = UBR 924
    CERTIFICATION = TEMPLATE
    CONT = Cisco-Router
    NEIGHBOR = Cisco-Cdp
    HEALTH = Cisco-Router
INSTRUMENTATION:
    Card-Fault = Default
    Environment = CiscoEnvMon:DeviceID
    CPU/Memory = CiscoRouter:DeviceID
    Interface-Fault = MIB2
    Interface-Performance = CiscoRouter
    Interface-Ethernet-Performance = CiscoRouter_Ethernet
}
# VisualNetworks Visual UpTime Select T1 CSU
.1.3.6.1.4.1.1246.1.1.109 {
    TYPE = Probe
    VENDOR = VisualNetworks
    MODEL = Visual UpTime Select T1 CSU
    CERTIFICATION = TEMPLATE
    CONT = Generic-MIB2
INSTRUMENTATION:
    Interface-Fault = MIB2
    Interface-Performance = MIB2
}
# Eastern-Research DNX-11 Switch
 .1.3.6.1.4.1.644.2.4 {
    TYPE = Switch
    VENDOR = Eastern-Research
    MODEL = DNX-11
    CERTIFICATION = TEMPLATE
    CONT = Generic-Switch
INSTRUMENTATION:
    Interface-Fault = MIB2
    Interface-Performance = MIB2
    Port-Fault = MIB2
    Port-Performance = MIB2
}
```

Device entry field descriptions

Chapter 5, Discovery, in the VMware Smart Assurance IP Manager Reference Guide describes the device entry fields for a field-based certification.

Note the following about device entry field descriptions in oid2type_Field.conf:

- The discovery driver keywords are listed in the ic-oid2type.asl file, which is located in the BASEDIR/smarts/rules/discovery directory.
- For each discovery driver keyword in a device entry, the IP Manager searches the ic-oid2type.asl file to find the prefix value that is needed to identify the name of the appropriate discovery driver. The naming format for a discovery driver is:

  ```
  <prefix>-<param-name>-Driver
  ```

  The driver class is either GA_Driver or GA_CompoundDriver.
- Once the appropriate discovery driver is identified, the IP Manager finds the discovery driver in the .import files, which are located with the oid2type files in the BASEDIR/smarts/conf/discovery directory.

  The discovery drivers that are associated with the discovery driver keywords are described in detail in “Phase 3: Probe the system added to the topology” on page 38.

  The discovery drivers that are associated with the ADNR and DSCVDATA keywords work together to gather MIB-II-like interface and IP data within a device’s vendor-specific MIBs. The gathered data is used to determine whether the target device is a “match” to another device.

  Although a “VRRP” keyword exists to point to vendor-specific VRRP drivers, no comparable keyword, say “HSRP,” exists to point to vendor-specific HSRP drivers. Reason: HSRP is a Cisco-proprietary protocol that runs only on Cisco devices. The HSRP driver, named Cisco-HSRPGroup-Driver, is defined in the DISCOVERY_CISCO.import file.

  Many of the discovery driver and instrumentation keywords have MIB-II counterparts that can be useful in field certifications. Here are some examples:

- CONT = Generic-MIB2
- HOSTRESRCS = MIB2
- Interface-Fault = MIB2
- Interface-Performance = MIB2
- Interface-Ethernet-Performance = MIB2
- Port-Fault = MIB2
- Port-Performance = MIB2
- Port-Ethernet-Performance = MIB2

  For a device entry that specifies MIB-II discovery and instrumentation, the IP Manager will use generic SNMP MIB-II drivers and instrumentation to discover and monitor the device.

Device entries for oid2type_Field.conf

Field certification is useful when a new device uses the same discovery drivers and instrumentation as an existing VMware-certified device. For example, when the new device, say a Cisco 2503, is a variant of an VMware-certified device, say Cisco 2502.

You accomplish a field certification by fully or partially reusing the discovery driver parameters and instrumentation parameters for an existing VMware-tested certification. Typically, you use a trial-and-error approach to find the most suitable combination of driver and instrumentation parameters for the device that you want to certify.

Except for very simple devices, you should consult with VMware Customer Support when creating device entries in the oid2type_Field.conf file.

“Adding device entries to oid2type_Field.conf” section in Chapter 3, Discovery, in the VMware Smart Assurance IP Manager User Guide, provides information on how to add device entries to the oid2type_Field.conf file.

Discovery driver parameters

For field certification of a new device, you specify the discovery driver parameters that will enable the IP Manager to properly represent the device in its modeled topology.

If, for example, you want to discover a new Cisco router, look in the oid2type.conf or oid2type_Cisco.conf file for a Cisco router that is a variant of the new Cisco router, and then use the discovery drivers for that router to discover the new router.

Also, here is some information about discovery drivers that you might find helpful when specifying discovery drivers for a new device:

- A discovery driver is either a GA_Driver or a GA_CompoundDriver and is defined in a .import file. A GA_CompoundDriver lists a series of GA_CompoundDrivers or GA_Drivers or both that the IP Manager will run in the order that the drivers are specified. A GA_Driver specifies a ruleset and possibly a front end.
- At startup, the IP Manager loads into memory all of the drivers in the .import files and all of the certifications in the oid2type files. For this reason, a GA_CompoundDriver can refer to a driver that is located in a different .import file.

Instrumentation parameters

For field certification of a new device, you also specify the instrumentation parameters that will enable the IP Manager to correctly monitor the device for connectivity and performance analysis.

If, for example, you want to monitor the CPU on a new Cisco router, look in the oid2type.conf or oid2type_Cisco.conf file for a Cisco router that is a variant of the new Cisco router, and then use the CPU/memory instrumentation for that router to monitor the CPU on the new router.

Submit certification request to VMware

Ultimately, for a field-certified device, you should submit a device certification request to VMware Corporation so that the device will be certified in future releases of the IP Management Suite. VMware Corporation will use specialized tools to make the certification more accurate, and will apply standard quality assurance testing to the certification.

To submit a device certification request, go to the VMware online support website at support.vmware.com and select Support. On the Support page, you will see several options, including one for making a service request.

To open a service request, you must have a valid support agreement. Contact your VMware sales representative for details about obtaining a valid support agreement or with questions about your account.

Discovery hook scripts

In addition to the steps that are performed automatically during the discovery process, you can use discovery hook scripts to specify additional steps to occur at various stages of the discovery. For example, you could use a postprocessing script to manage or unmanage IP objects during the postprocessing phase of the discovery. The discovery hook scripts are written in the VMware Smart Assurance proprietary Adapter Scripting Language (ASL).

When discovery hook scripts run

The discovery hook scripts are identified in the DISCOVERY_CUSTOM.import and POST-DISCOVERY_CUSTOM.import files, which are located in the BASEDIR/smarts/conf/discovery directory in the IP Manager installation area.

Each time the IP Manager performs a discovery cycle, it runs the main discovery drivers plus the discovery hook scripts that are identified in the DISCOVERY_CUSTOM.import file. The discovery may be an initial discovery, a full discovery, or a pending discovery.

By default, running the discovery hook scripts has no effect on the discovery process because the scripts are just template files, which mean that you must open the scripts with an editor and add the code that performs the actual processing.

How discovery hook scripts identify systems for processing

The VMware Smart Assurance IP Manager Reference Guide shows how the custom-end-system.asl script identifies systems for processing. The discovery process automatically assigns the following value to the AgentName variable:

Name of SNMP agent that is hosted by the system that is currently being discovered

The custom-end-system.asl script uses the value of AgentName to obtain the name of the system by invoking the getSystem() operation on the SNMP agent object.

The custom-*start*-system.asl script uses the same method to identify systems for processing.

Discovery hook scripts and ICMPSNMP and ICMPONLY systems

If during the initial polling phase a system is assigned an ICMPSNMP access mode, the discovery process calls the custom-end-system.asl script. In contrast, if during the initial polling phase a system is assigned an ICMPONLY access mode, the discovery process does not call the custom-end-system.asl script. For an ICMPONLY system, the discovery process schedules the system for postprocessing right after the initial polling phase because additional probing is not applied to an ICMPONLY system.

Thus, any custom code that you add to the custom-end-system.asl script is applied to ICMPSNMP systems but not ICMPONLY systems. If you want to customize the discovery of both ICMPSNMP and ICMPONLY systems, you should add your custom code to the custom-start-*post*.asl script, the custom-end-*post*.asl script, or both.

Discovery methods describes the discovery process and identifies the topology that is discovered by the IP Manager.

“Modifying discovery hook scripts” section in Chapter 3, Discovery, in the VMware Smart Assurance IP Manager User Guide, provides information on how to edit a discovery hook script to add the ASL convenience patterns, the parsing rules, and the processing rules that perform the actual processing.

Automatic and manual discovery

Automatic discovery is scheduled under the control of an ICF-TopologyManager program that is running in the IP Manager. The ICF-TopologyManager controls all aspects of discovery, including the scheduling of automatic discovery and the execution of manually invoked discovery requests.

After the initial discovery by an IP Manager, you can schedule automatic discovery at a regular interval and manually invoke discovery at any time. You perform these tasks through a Domain Manager Administration Console that is attached to the IP Manager, or from the command line by using the **dmctl** utility.

Discovery methods

For both automatic and manual discovery, the following two methods of discovery are available:

- Full discovery
- Pending discovery

  Manual rediscovery of individual systems is also available.

Full discovery

During the initial full discovery, the IP Manager populates its repository with the topology that is discovered in the managed network environment. During any subsequent full discovery, the IP Manager attempts to rediscover the topology in its repository.

Rediscovering the topology reveals any changes in the configuration or connectivity of the managed systems and keeps the topology current. For example, if a card is added to a switch, the IP Manager must rediscover the switch so that it can add the new card to the topology. And if new systems have been added to the network and autodiscovery is enabled, the IP Manager will discover the new systems during the rediscovery.

Pending discovery

During a pending discovery, the IP Manager attempts to discover the systems on the Pending Devices list. The Pending Devices list contains the names of the systems that could not be discovered by the IP Manager.

You can view the Pending Devices list at any time by selecting Topology > Show Discovery Progress from the Domain Manager Administration Console. Pending Devices list describes the various aspects of the Pending Devices list.

Discovery considerations

Initiating a full discovery for a large topology can take a considerable amount of time, ranging from 1 hour to 2 or more hours, depending on the actual size of the topology. Keep this in mind when scheduling full discovery for the IP Manager.

Also, consider scheduling automatic full discovery during off hours (5:00 in the morning, for example). The VMware Smart Assurance IP Manager Deployment Guide recommends scheduling full discovery at least weekly and pending discovery at least daily. Typically, the duration of a pending discovery is much shorter than the duration of a full discovery.

Running the discovery more often will provide a more accurate network topology, but will also cost more in terms of network and system resources.

Scheduling automatic discovery

You can schedule full discovery and pending discovery intervals:

- Scheduling a full discovery interval determines how often the IP Manager attempts to rediscover the existing systems in its repository.
- Scheduling a pending discovery interval determines how often the IP Manager attempts to discover the systems on the Pending Devices list.

  By default, the full discovery interval is enabled and set to 1 day, and the pending discovery interval is enabled and set to 6 hours. The parameters that control scheduled discovery are located on the Topology tab page of the Domain Manager Administration Console.

  The full discovery interval should not be set to less than 60 minutes, and the pending discovery interval should not be set to less than 5 minutes. Violating either of these restrictions will result in the printing of an exception by the Domain Manager Administration Console. For example:

Apply changes failed: Invalid full discovery interval

The VMware Smart Assurance IP Manager User Guide provides information on how to configure the full and pending discovery intervals.

Read community strings

You can specify the read community strings that are used by discovery in several different places:

- Discovery filter
- Seed file
- **Add Agent** command

  If read community strings are specified for a discovery filter, the IP Manager uses those community strings when polling the systems that match the discovery filter. The VMware Smart Assurance IP Manager User Guide provides information about specifying read community strings for a discovery filter.

  If no read community string is specified in a discovery filter, or no community string is specified in the seed file or through the **Add Agent** command, or the specified community string is not correct, the IP Manager selects a community string in the following manner:

- Tries the community strings that appear in the Read Community Strings window (Topology tab) of the Domain Manager Administration Console.
- If no community strings appear in the Read Community Strings window, reads the default read community string in the discovery.conf file.
- If no read community strings are found for a system, places the system on the Pending Devices list during an initial discovery attempt, or notifies a DiscoveryError during a rediscovery attempt.

  Initial discovery and rediscovery errors are discussed in Discovery error scenarios and results.

About managing and unmanaging topology objects

The IP Manager monitors managed objects but not unmanaged objects. The IP Manager uses ICMP and SNMP, unless a different access mode is specified, to monitor a managed object for the purpose of determining the object's status and connectivity. An unmanaged object is not polled or rediscovered by the IP Manager, but remains present in the modeled topology.

A managed object is an object whose IsManaged attribute has a value of TRUE. IsManaged is a Boolean attribute: TRUE when an object is managed, and FALSE when an object is unmanaged.

The **Manage** and **Unmanage** commands enable you to control whether an object is managed. Unmanaging a system might be useful when, for example, a switch or a card is taken offline for maintenance.

Types of objects that can be managed or unmanaged

You can set the management state for the following types of objects:

- Systems such as routers, switches, and hosts
- Physical packages such as cards
- Network adapters such as ports and interfaces
- Protocol endpoints such as IP interfaces and STP nodes
- Management agents such as SNMP agents
- Logical devices such as fans, processors, temperature and voltage sensors, and file systems
- Redundancy groups such as redundancy groups that represent two or more cards or two or more network connections

  The ability to unmanage IP interfaces enables you to develop flexible management policies. When you unmanage an IP interface, the underlying physical interface continues to be managed because the status of the IP interface is determined by using ICMP, and the status of the physical interface is determined by using SNMP.

  You can unmanage the IP interfaces on systems where ICMP pings are not allowed because, for example, of a firewall. When you unmanage these IP interfaces, the IP Manager no longer polls them through ICMP. More important, however, is that the IP Manager still manages the underlying physical interfaces through SNMP.

Rules that govern manage and unmanage operations

The **Manage** and **Unmanage** commands are recursive. When you unmanage an object, all the management agent services and other objects that are functionally dependent on that object become unmanaged. For example, if you unmanage a switch, all of the ports and cards that belong to the switch are unmanaged.

The following list of rules determines whether an object can be managed or unmanaged and explains the recursive nature of the **Manage** and **Unmanage** commands:

- When you unmanage a system, a management agent service that is hosted by or is a part of that system also becomes unmanaged. For example, a system, such as a router, hosts an SNMP agent service. If you unmanage a router that hosts an SNMP agent service, the SNMP agent service is also unmanaged.
- When you unmanage a system, any logical device that is a part of that system also becomes unmanaged. Logical devices include objects such as fans, temperature and voltage sensors, processors, and file systems. Ports and interfaces also become unmanaged. If you unmanage a system, such as a router, the interfaces and sensors that are part of that router are not managed, nor can they be managed while the parent object is unmanaged. Likewise, if you manage a system, all of its components are managed except those that are explicitly unmanaged.

- When you add a logical device or service to an unmanaged system, that logical device or service automatically becomes unmanaged. For example, if an interface is added to an unmanaged host, the new interface is also unmanaged.
- When you unmanage a service, the protocol endpoint that is associated with that service becomes unmanaged.
- When you unmanage a logical device or a protocol endpoint, any other protocol endpoint that is layered over that logical device or protocol endpoint becomes unmanaged. For example, if you unmanage an interface, the subinterfaces or IP addresses that are layered over that interface are also unmanaged.
- When you unmanage a card, the managed state of any sub-cards, network adapters that are realized by the card, or systems that are packaged by the card remain managed. As such, monitoring and analysis of these items continue.

Methods to control the management state of objects

You can use any of the following methods to control the management state of objects:

- Set the management state of individual objects by attaching the Global Console to the IP Manager and by using the **Manage** and **Unmanage** commands.

  The VMware Smart Assurance IP Manager User Guide provides information on how to set the management state of an object.
- Use the discovery hook scripts that are provided with the IP Manager to set the management state for IPs, devices, interfaces, ports, and other objects.

  For example, if there are groups of IP addresses that are not normally reachable by using ICMP polling, specify the IP ranges or matching criteria in the custom-end-post.asl file to ensure that the IP Manager will set the management state of these IP addresses to unmanaged. The IP Manager will not ping IP addresses that are unmanaged.

  The VMware Smart Assurance IP Manager User Guide provides information on how to use discovery hook scripts.
- Use the Interface Management Policy that is provided with the IP Manager to set the management state for interfaces.

  The Interface Management Policy, which is accessible through a Polling and Thresholds Console, enables you to set a system-level policy to unmanage interfaces. By default, this policy is not active.

  The VMware Smart Assurance IP Manager User Guide describes the Interface Management Policy and provides instructions on changing its setting.
- Create IP tag filter groups to not only tag specific IP addresses on a discovered system, but to also set those IP addresses to managed or unmanaged when the system is discovered.

  The VMware Smart Assurance IP Manager User Guide describes the IP tagging feature and provides instructions on creating IP tag filter groups.

Persistent management state

The IP Manager retains the management state of ports, interfaces, IP addresses, and SNMP agents in the following situations:

- After a system has been recertified
- If the name format has been changed, as described in How discovery names a system

  If the DNS configuration has been changed

Incremental reconfigure

If new devices are added to the topology, a full reconfiguration happens which is time consuming. An incremental reconfigure happens only for the devices added, and is faster.

To enable incremental reconfigure, set *ThresholdOfChangedUCSForIncReconfig 10* in the tpmgr-param.conf file. The value of this flag controls the maximum number of devices for which incremental reconfigure will run. For example, if the value is 10, incremental reconfigure will run if the number of the last discovered device is less then 10, otherwise a bulk reconfigure will run.

By default, *ThresholdOfChangedUCSForIncReconfig* is set to 10 in tpmgr-param.conf. This means that for a discovery containing less than 10 systems, the reconfigure started and reconfigure finished messages will not be logged in the domain log. However, reconfigure will still take place. In order to observe the incremental reconfigure messages, set *DebugReconfigure TRUE* in tpmgr-param.conf.

To make the required change in tpmgr-param.conf, type:

<BASEDIR>/IP/smarts/bin/sm_edit conf/discovery/tpmgr-param.conf

To restart the IP domain or reload tpmgr-param.conf for the change to take affect, type:

<BASEDIR>/IP/smarts/bin/sm_tpmgr -s <IP 8.1.X domain> --load-conf=tpmgr-param.conf

To verify the *ThresholdOfChangedUCSForIncReconfig* setting, type:

<BASEDIR>/IP/smarts/bin/dmctl -s <IP 8.1.X domain> invoke ICF_TopologyManager::ICF-TopologyManager findParameter ThresholdOfChangedUCSForIncReconfig

To make the necessary change without restarting the IP domain, type:

<BASEDIR>/IP/smarts/bin/dmctl -s <IP 8.1.X domain> invoke ICF_TopologyManager::ICF-TopologyManager insertParameter ThresholdOfChangedUCSForIncReconfig <value>

To verify the debug setting, type:

<BASEDIR>/IP/smarts/bin/dmctl -s <IP 8.1.X domain> invoke ICF_TopologyManager::ICF-TopologyManager findParameter DebugReconfigure

To make the necessary change without restarting the IP domain, type:

<BASEDIR>/IP/smarts/bin/dmctl -s <IP 8.1.X domain> invoke ICF_TopologyManager::ICF-TopologyManager insertParameter DebugReconfigure TRUE

Handling duplicate IP address errors

In a networked environment, the same IP addresses can be reused across multiple systems that are running in separate networks. These IP addresses are known as overlapping or duplicate IP addresses. Overlapping IP addresses do not create problems across different networks; however, when a service assurance application for a centrally managed network attempts to discover the overlapping IP addresses, IP address identity problems arise.

By default, and assuming that IP tagging is not in effect, the IP Manager will respond as follows when it discovers an overlapping IP address:

- Classifies the IP address as a DuplicateIP.
- Suspends IP-related analysis for the address.
- Creates a DuplicateIP object.
- Generates a DuplicateIP notification.

  With the IP tagging feature, the IP Manager is able to distinguish between overlapping addresses belonging to different customers by adding a distinguishing tag to their names. That is, instead of using the standard naming scheme of “IP-<IP address>” to name an overlapping IP address instance, the IP Manager uses the “IP-<IP address>/<tag>” naming scheme to name the instance.

  For a similar scenario in an IPv6 network, the IP Manager classifies the IPv6 address as a DuplicateIP, suspends IPv6-related analysis for the address, creates a DuplicateIPv6 object, and generates a DuplicateIPv6 notification.

  By default, the IP Manager does not poll an address that is associated with a DuplicateIP or DuplicateIPv6 object. The reason is that the IP Manager has no way to determine which of the systems that are bound to that address is responding to the ICMP or SNMP polls.

How to prevent a duplicate IP address error

As an alternative to using the IP tagging feature to correct a duplicate IP address problem, consider setting the EnableUnmanageAdminDown parameter to TRUE in the tpmgr-param.conf file to prevent the problem from occurring. When EnableUnmanageAdminDown = TRUE, the IP Manager will not discover any IP address that is associated with an administratively down interface.

This discovery behavior allows for a situation in which two devices are sharing a management IP address for the following reason:

The currently active device is to be replaced by the administratively down device.

In this situation, the IP Manager will discover and create an IP object for the currently active device, but will not discover an IP object for the administratively down device. Thus, no DuplicateIP object is created, and the IP Manager can use the IP object to poll the currently active device. When the replacement device becomes active, and the currently active device becomes administratively down, the polling will continue.

Redundancy groups

A redundancy group is a logical entity that contains two or more objects that participate in a redundant configuration. For example, a remote site that is accessed through two routers may be modeled as a redundancy group that contains the two routers. If one router fails, the redundancy is at risk. If both fail, all capability is lost.

ECIM-R supports redundancy groups of cards, network adapters, network connections, protocol endpoints, and unitary computer systems.

The VMware Smart Assurance IP Manager User Guide provides information about creating redundancy groups.

HSRP groups

An HSRP group is a redundancy group of routers that use a proprietary protocol from Cisco called Hot Standby Routing Protocol. The group provides router backup in the event of a router system failure or an interface failure.

In an HSRP group, several routers that are connected to the same segment of an Ethernet, FDDI, or token-ring network work together to present the appearance of a single virtual router on the LAN. Because the routers share the same IP and MAC addresses, the hosts on the LAN are able to continue forwarding packets to a consistent IP and MAC address in the event of a router failure. The process of transferring the routing responsibilities from one router to another is transparent to the user.

The VMware Smart Assurance IP Manager User Guide provides information about creating HSRP groups.

VRRP groups

A VRRP group is identical to an HSRP group except that the VRRP group uses an industry standard protocol called Virtual Router Redundancy Protocol.

The VMware Smart Assurance IP Manager User Guide provides information about creating VRRP groups.

ICMP

Internet Control Message Protocol is tightly integrated with IP. ICMP messages, which are delivered in IP packets, are used for out-of-band messages that are related to network operation. Some of ICMP's functions are to announce network errors, to announce network congestion, to announce timeouts, and to assist in troubleshooting network problems.

ICMP supports an Echo function that sends a packet on a round-trip between two hosts. Ping, which is a common network management tool that is based on the Echo function, transmits a series of packets to measure average round-trip times and to compute loss percentages. Ping is central to the IP Manager ICMP polling.

ICMP IP version support

The IP Manager supports both IPv4 ICMP polling and IPv6 ICMP polling. An IPv4 ICMP poller sends ICMP messages for use with IP version 4. An IPv6 ICMP poller sends ICMP messages for use with IP version 6.

All IP Manager products support IPv4 and IPv6 ICMP polling.

ICMP operation

The IP Manager uses ICMP polling for two purposes:

- To determine whether a system is reachable
- To monitor a system after the system has been discovered

SNMP

The Simple Network Management Protocol is used by network management applications for discovery, fault management, and performance management. SNMP is an application-layer protocol that exchanges management information between a network management application, such as the IP Manager, and one or more managed network objects, called agents.

SNMP, itself, does not provide network management. What SNMP does provide is a framework, or infrastructure, on which network management applications can be built.

SNMP key components

SNMP-managed network illustrates the SNMP key components: management station, agent, management information base, and SNMP protocol.

1 Agent

2 SNMP Agent

3 MIB

4 SNMP Agent

5 MIB

6 SNMP Agent

7 MIB

8 Agent

9 Agent

10 IP Network

11 MIB

12 Management

13 Applications

14 SNMP Manager

15 Management Station

16 SNMP

17 SNMP

18 SNMP

19 SNMP

Figure 5-20. SNMP-managed network

Each SNMP agent or SNMP manager maintains a MIB. An agent MIB contains information about the network system managed by the SNMP agent, and about the agent itself. A manager MIB contains network management information that is extracted by the SNMP manager from each agent MIB.

SNMP basic operations

SNMP includes the following basic operations:

- GET

  Enables the SNMP manager to retrieve (read) the value of MIB objects at the SNMP agent.

- SET

  Enables the SNMP manager to set (write) the value of MIB objects at the SNMP agent.

- NOTIFY

  Enables an SNMP agent to asynchronously notify the SNMP manager of significant but unsolicited events (traps).

- INFORM

  Enables an SNMP manager to asynchronously send an alert (similar to a trap) to another SNMP manager.

SNMP versions

Currently, there are three versions of the SNMP protocol: v1, v2c, and v3. SNMPv1 and v2c offer rudimentary authentication and authorization schemes, whereas SNMPv3 offers robust authentication and authorization schemes plus data privacy.

SNMPv1 security

SNMPv1 security consists of a pairing of an SNMP agent with some arbitrary set of SNMP managers to form an SNMP community. Each SNMP community is given a name called the community name or community string for the community. Common community names are *public* (read-only), *private* (read-write), and *trap*. The permissions (read-only, read-write) for a community name indicate the read-write permissions of an SNMP manager when using that community name to access an agent's MIB. Community names can be thought of as passwords to SNMPv1 agents.

SNMPv2c security

Like SNMPv1, SNMPv2c uses the notion of communities to establish trust between SNMP managers and SNMP agents. Unlike SNMPv1, SNMPv2c uses an updated version of the structure of management information (SMI), which extends the MIB object tree, allows several new data types, and makes a number of other changes. SNMPv2c is more efficient than SNMPv1 and has better error-handling capabilities.

SNMPv3 security

SNMPv3 provides integrity, authenticity, data privacy, and access control for SNMP messages exchanged between an SNMP manager and the managed SNMP agents. Unlike the community-based administrative model of SNMPv1 and V2C, SNMPv3 unambiguously identifies the source and destination of each SNMP message. And instead of using community names to establish trust between SNMP managers and SNMP agents, SNMPv3 uses the following security-related services to establish trust:

- Authentication

  The source includes information in each sent message that identifies the source as authentic, and performs the required functions to ensure message integrity. A typical authentication scheme requires that the source and destination parties share the same authentication key.

- Privacy

  Messages are encrypted to achieve privacy. The encryption is done in such a way that only the intended destination can perform the decryption. A typical privacy scheme requires that the source and destination parties share the same privacy key.

- Access control

  Both the source and destination play a part in access control. Each destination may have a distinct access policy for each potential source, which gives an administrator considerable flexibility in setting up an SNMP management system and assigning various levels of authorization to different users.

SNMP version support

The IP Manager fully supports SNMPv1 and v2c. It also provides the following support for SNMPv3:

- Supports SNMPv3 authentication, data encryption, and access control
- Supports manual discovery but not autodiscovery of candidate systems having SNMPv3 agents

- Authenticates and decrypts SNMPv3 traps

  As further clarification of autodiscovery support, the IP Manager supports the autodiscovery of SNMPv1 or v2c systems in IPv4 networks.

SNMP IP version support

The IP Manager supports both IPv4 SNMP polling and IPv6 SNMP polling. An IPv4 SNMP poller sends SNMP messages for use with IP version 4. An IPv6 SNMP poller sends SNMP messages for use with IP version 6.

All IP Manager products support IPv4 and IPv6 SNMP polling.

About ICMP and SNMP polling parameters

The parameters for controlling IP Manager ICMP and SNMP polling during discovery are contained in the discovery.conf file and described in the VMware Smart Assurance IP Manager Reference Guide. The guide also describes the parameters for controlling ICMP and SNMP polling for correlation analysis which are accessed through the Polling and Thresholds Console.

Availability Manager

6

This chapter includes the following topics:

- Network notifications
- Network topology
- Network containment
- Root-cause problems, exceptions, and events
- Trap analysis
- Redundancy groups

Network notifications

IP Performance Manager and the IP Availability Manager pass the results of its analysis in the form of notifications to the Global Manager. The Global Manager, in turn, passes the notifications to the Global Console, where they are presented in a tabular form in the Notification Log Console.

Each notification consists of a set of attributes that describe properties of the notification, such as Class, Name, Event, Source, and Impact. The value in the Name attribute is the display name of the notification; VMware Smart Assurance IP Manager Reference Guide describes the naming formats used to create network object names. The numeric value for the Impact attribute indicates the effect of this notification on other objects in the managed network; the larger the numeric value, the larger the effect.

The severity of a notification is reflected by its color, where red indicates the highest severity level.

Viewing network notifications

You can view the notifications through the Global Console in two basic ways:

- As tabular entries in a Notification Log Console view

  Included with the Notification Log Console view (or any other view where you can see notifications) is a Notification Properties dialog box for viewing detailed information about an individual notification.

- As color-coded severity bar icons in a Map Console view

  In a map, a color-coded severity bar icon appears underneath nodes (graphical representations of topology objects) that are affected by abnormal conditions denoted in notifications.

Depending on the type of event and information available for display, the Notification Properties dialog box displays up to seven tabs. The tabs are: General, Audit Log, Codebook, Impact, Details, Caused By, and Aggregates.

Note You view the details for a network notification by clicking the Details tab in the Notification Properties dialog box. The Details tab identifies the Domain Manager source of the notification and the notification's attributes and corresponding values. If a notification has multiple sources, one Details tab appears for each source.

The VMware Smart Assurance IP Manager User Guide provides additional information on the Notification Properties dialog box.

Network topology

The Global Manager imports instances of the following topology object types from IP Availability Manager. The imported topology includes systems, physical packages, device connections, logical links, collections, redundancy groups, and duplicate IPs (DuplicateIP and DuplicateIPv6 objects)—the duplicate IPs are not shown in the list that follows:

Note Bullet list items beginning with a dash indicate class hierarchy; for example, FileServer is a subclass of Host.

- Systems:
  - Bridge
  - CallServer
  - Host:
    - FileServer
  - Hub
  - LoadBalancer
  - MediaGateway
  - Node
  - Probe
  - Router:
    - Firewall
    - Multilayer Switch Feature Card (MSFC)
    - Router Switch Feature Card (RSFC)
    - Router Switch Module (RSM)
  - Switch
  - TerminalServer
- Physical packages:
  - NASChassis
- Device connections:
  - NetworkConnection:
    - Cable
    - TrunkCable
- Logical links:
  - IPNetwork
  - IPv6Network
  - VLAN
- Collections:
  - Partition
- Redundancy groups:
  - SystemRedundancyGroup
  - NetworkAdapterRedundancyGroup
  - NetworkConnectionRedundancyGroup

The Global Manager combines this topology with the topologies received from the other Domain Managers connected to the Global Manager. In some cases, depending on the other Domain Managers, *combining* means migrating certain object types imported from IP Availability Manager with certain object types imported from the other Domain Managers.

Network topology in maps

The Global Console presents the topology information in a variety of dynamically updated formats that show the status of the network objects and their many relationships. One of those formats is the topology map, which is a graphical representation of the topology.

Network topology map pop-up menus

Two types of pop-up menus are available in a network topology map:

- Right-click a selected node in a map

  Opens a pop-up menu that enables you to select any of the map types available to the selected node (for example, Physical Connectivity, VLAN Connectivity, Group Membership), to display notification properties (if any) for the selected node, and so on.

  For a node marked with a + (plus) sign, the Expand Node menu option allows you to expand the node to see all nodes that are physically or logically connected to the node.

- Right-click an open space in a map

  Opens a pop-up menu that enables you to display a circular layout of the map, to manually position nodes in the map, and so on. The Reset map menu option enables you to reset changes made to the map or to return to the saved map display.

  In the topology tree of a Map Console, a pop-up menu is also available when you right-click an object. This menu enables you to select any of the map types available to the selected object, to display notification properties (if any) for the selected object, and so on.

  The VMware Smart Assurance Service Assurance Manager Operator Guide provides detailed descriptions of map menus and pop-up menus available for the Map Console.

Network topology map graphical representations

Network topology maps contain objects that may include IP networks, VLANs, routers, switches, hosts, and various types of connections such as cables and trunk cables. In a map display, a node is a graphical representation of a topology object, and an edge is a graphical representation of a relationship or connection between objects.

Note that your system administrator may replace the standard map icons with other map icons that are preferred by your organization. In that case, use Map > Map Legend to see the definitions of your map icons.

Network topology map types

The following map types are available for viewing the network topology:

- Physical Connectivity
- IP Network Connectivity
- IP Network Membership
- VLAN Connectivity
- VLAN Membership
- Group Maps, such as a Group Membership Map

Object type	Available map types
Bridge Host Hub MSFC Probe Router RSM RSFC Switch TerminalServer Node	Physical Connectivity IP Network Connectivity VLAN Connectivity
IPNetwork	Physical Connectivity IP Network Membership IP Network Connectivity
VLAN	Physical Connectivity VLAN Membership VLAN Connectivity
Group	Group Physical Connectivity Group IP Network Connectivity Group VLAN Connectivity Group Membership

Physical Connectivity map

The Physical Connectivity map is the default map that appears when you select a system object (switch, router, hub, bridge, host) in the Topology tab of the Map Console. This map displays Layer 2 physical connectivity between system objects, including cables and network connections. A Physical Connectivity map is available for a selected system object, for all system objects in a selected IP network, and for all system objects in a selected VLAN.

For a Physical Connectivity map, the edges appear as:

- Solid lines for direct connections such as cables.
- Jagged lines for network connections.

  If you display a Physical Connectivity map for a selected IP network or VLAN, the map displays the physical connectivity of the system objects *inside* the IP network or VLAN. An icon representing the IP network or VLAN does not appear.

  In a Physical Connectivity map, a hop is the distance between two system objects that are physically connected.

IP Network Connectivity map

The IP Network Connectivity map is the default map that appears when you select an IP network in the Topology tab of the Map Console. It displays Layer 3 connectivity between routers and IP networks. An IP Network Connectivity map is available for a selected switch, router, RSM, hub, host, or IP network.

For an IP Network Connectivity map, the edges appear as:

- Solid lines for IP addresses for objects on the connected IP network.
- Jagged lines for network connections.

  In an IP Network Connectivity map, a hop is a logical hop. It represents the distance between a device and the IP network.

IP Network Membership map

The IP Network Membership map displays all objects having an IP address on a selected IP network. The edges appear as solid lines for participation in the logical group and do not change color.

For an IP Network Membership map, the concept of a hop is not applicable.

VLAN Connectivity map

The VLAN Connectivity map is the default map that appears when you select a VLAN object in the Topology tab of the Map Console. It displays VLANs, the switches that participate in the VLAN, and the routers that enable connections between VLANs. A VLAN Connectivity map is available for a selected switch, router, RSM, hub, host, or VLAN.

For a VLAN Connectivity map, the edges appear as:

- Solid lines for devices connected to ports on a VLAN or switch.
- Jagged lines for network connections.

  In a VLAN Connectivity map, a hop is a logical hop. It represents the distance between an object (router or switch) and the VLAN.

  If a switch appears in a VLAN Connectivity map and no edge connects the switch to the VLAN, the switch cannot recognize the VLAN even though the switch has another port that participates in the VLAN.

VLAN Membership map

The VLAN Membership map displays all devices that are members of a selected VLAN. The edges appear as solid lines for participation in a selected VLAN and do not change color.

For a VLAN Membership map, the concept of a hop is not applicable.

Group maps

A group consists of one or more infrastructure devices, IP networks, VLANs, or subgroups. Groups are created by the administrator and might not be available for the session. The VMware Smart Assurance Service Assurance Manager Configuration Guide provides information about creating groups and importing data for groups.

Four types of group maps exist:

- Group Physical Connectivity map — This map displays physical connectivity between group objects. The solid line edges between groups represent one or more physical connections between objects of different groups. The edges do not change color to reflect their state. The number of hops cannot be changed.
- Group IP Network Connectivity map — In this map, an edge between two groups indicates that there are one or more IP networks common to the two groups. The group icon containing the IP network is displayed; no IP network icon is displayed. The number of hops cannot be changed.
- Group VLAN Connectivity map — In this map, an edge between two groups indicates that there are one or more VLANs common to the two groups. The group icon containing the VLAN is displayed; no VLAN icon is displayed. The number of hops cannot be changed.
- Group Membership — This map displays all the objects of a selected group or a subgroup. The edges do not appear. The concept of a hop is not applicable.

Network containment

An object's Containment view appears in a dialog box that organizes the information for the object into tab pages and tables.

The intent of the Containment view is to allow users to examine detailed component topology of an individual object. The Global Console retrieves the topology information from the Global Manager and from the underlying Domain Managers. If an underlying Domain Manager is not attached to the console, the console automatically establishes a connection to the Domain Manager by using the user's login ID.

Opening a network containment dialog box

To obtain containment information about an individual network object, you can use any of the following common methods to open a Containment dialog box for the object:

- In a network map tree, right-click the object and then select Containment in the pop-up menu.
- In a network map display, right-click the object and then select Containment in the pop-up menu.

- In a Topology Browser Console, right-click the object and then select Containment in the pop-up menu.
- In a Notification Log Console, double-click a notification for the object and click the Containment button in the Notification Properties dialog box.

  In an object's Containment view, the tabs that appear depend on which IP Managers are deployed, which feature licenses are installed, and which components are discovered.

  Note Containment is not available for user-defined groups and services.

  "Network containment tab pages" table in the VMware Smart Assurance IP Manager Reference Guide, provides information on Network classes, Containment tab pages, and the associated attributes.

Root-cause problems, exceptions, and events

IP Availability Manager diagnoses root-cause problems and detects (computes) events from status changes gathered from or issued by devices in the managed environment. It uses the events as symptoms to diagnose root-cause problems.

Note Events used as symptoms to diagnose a failure can also be perceived as impacts resulting from the failure.

IP Availability Manager also generates exceptions, or aggregate events, where an exception is a collection of problems or events associated with a particular managed system, IPv6Network, or VLAN. IP Availability Manager generates an exception whenever any of the composite problems or events for the system, IPv6Network, or VLAN occur. For example, IP Availability Manager will generate an exception for a router when a card on the router has a problem or the router is down.

The Global Manager imports all problems diagnosed by IP Availability Manager and certain key events detected by IP Availability Manager. It also imports the IPv6Network inaccessible exception. The problem notifications (one problem per notification), notifications (one event per notification), and IPv6Network exception notifications (one exception per notification) are displayed in the Global Console.

The VMware Smart Assurance IP Manager Reference Guide provides information on key attributes, diagnosed problems, generated exceptions, and detected events for all categories of network object types in the IP Availability Manager data model.

Trap analysis

A system or network adapter is considered to be unstable if it fluctuates too often between up and down states over a short period of time.

The IP Availability Manager monitors a system or network adapter's state through the SNMP traps it receives and determines when to generate an unstable notification based on a combination of fixed values and user-controlled settings. The IP Availability Manager also calculates a stable time in which to wait before clearing an Unstable notification.

IP Availability Manager monitors the following SNMP traps to determine a change in an object's state:

- coldStart and warmStart traps for a system
- linkUp or linkDown traps for a network adapter

  IP Availability Manager primarily uses linkDown traps for the analysis of unstable network adapters. However, for devices that only send linkUp traps, IP Availability Manager does use linkUp traps. When a device sends both linkUp and linkDown traps, IP Availability Manager uses linkDown traps for the unstable analysis.

Thresholds associated with unstable analysis

IP Availability Manager uses the following values to diagnose an object as unstable:

- Minimum Traps — The minimum number of Link/Restart traps received in order to conclude that the object is unstable. This variable is set by the LinkTrapThreshold parameter (contained in the Interface/Port Flapping setting) for network adapters and the RestartTrapThreshold parameter (contained in the Connectivity setting) for systems.
- Trap Window — The period within which the Minimum Traps must be received to declare the object unstable. This window is set by the LinkTrapWindow parameter (contained in the Interface/Port Flapping setting) for network adapters and the RestartTrapWindow parameter (contained in the Connectivity setting) for systems. After an object is declared unstable, IP Availability Manager computes the Stable Time.
- Stable Time — The amount of time that must elapse without additional Link/Restart traps before IP Availability Manager declares the object stable again. Stable Time depends on the length of time that the object was unstable. This is at least as large as the unstable length of time, and at least as large as the Trap Window. However, the stable time cannot be longer than 1 hour.

Example of unstable analysis

Unstable system/network adapter diagnosis illustrates how a system or network adapter is diagnosed as being unstable.

1 Up

2 Down

3 80 secs

4 idle

5 1st Trap

6 Last Trap

7 idle

8 waiting

9 Send

10 Unstable

11 Notification

12 Clear

13 Unstable

14 Notification

15 80 secs

16 unstable

Figure 6-1. Unstable system/network adapter diagnosis

In the example, assume that the Link/RestartTrapWindow parameter has a value of 30 seconds and that the Link/RestartTrapThreshold parameter has a value of 2.

IP Availability Manager performs the following actions:

1 As soon as IP Availability Manager receives a linkDown trap from a physical port or interface (or a coldStart/warmStart trap from a system), it begins counting.

2 When IP Availability Manager receives two or more traps within the 30-second Trap Window, it considers the network adapter or system to be unstable and sends an Unstable notification. The Minimum Traps variable (set by the Link/RestartTrapWindow parameter) determines the number of traps (2) that IP Availability Manager must receive within the Trap Window (set by the Link/RestartTrapWindow parameter) before IP Availability Manager considers an object unstable.

3 IP Availability Manager continues to receive traps for 80 seconds after the initial trap. This results in a Stable Time of 80 seconds.

   The Stable Time is the amount of time IP Availability Manager waits before it clears the Unstable notification. In our example, the Stable Time is set at 80 seconds since it is greater than the Trap Window (30 seconds) and less than one hour.

   As you can see, IP Availability Manager uses a relative measure to determine how long an object must be stable before it clears the Unstable notification. This measure is proportional to the amount of time an object is unstable. The longer an object is unstable the longer it must be stable before the Unstable notification is cleared. Because the object in our example remains unstable for 80 seconds, IP Availability Manager clears the Unstable notification no sooner than 80 seconds after it receives the final trap.

Unstable notifications for backup or dial-on-demand interfaces

By default, the Interface/Port Flapping (Unstable) setting is not associated with backup interfaces or with dial-on-demand interfaces. This means, by default, that IP Availability Manager will not generate Unstable notifications for the interfaces in those groups.

To enable diagnosis of Unstable notifications for backup interfaces or dial-on-demand interfaces:

1 Add the **Interface/Port Flapping** setting to the **Backup** or **Dial-on-Demand** groups.

2 Configure the **Interface/Port Flapping** setting's parameters accordingly.

3 Reconfigure your Polling and Thresholds groups by using the **Reconfigure** toolbar button on the **Domain Manager Administration Console**.

Combined settings and their notifications summarizes the notifications that may appear after you apply the Interface/Port Flapping setting to backup interfaces or dial-on-demand interfaces.

Table 6-1. Combined settings and their notifications

Settings	Resulting notifications
Interface/Port Flapping Backup Interface Support	■ The Down notification is not generated if the interface is down. ■ The BackupActivated notification is generated if the interface comes up. ■ The ExceededMaximumUptime notification is generated if the interface stays up too long. ■ The Unstable notification is generated if the interface or port is going up and down.
Interface/Port Flapping Dial-On-Demand Interface Support	■ The Down notification is not generated if the interface is down. ■ The ExceededMaximumUptime notification is generated if the interface stays up too long. ■ The Unstable notification is generated if the interface or port is going up and down.

Redundancy groups

A redundancy group consists of two or more objects of the same type that are configured in such a way as to provide backup resources for critical network objects. For example, two routers accessing the same host may be modeled as a redundancy group containing the two routers. If one router fails, the redundancy is “at risk.” If both fail, all capability is lost.

IP Availability Manager supports the creation and analysis of redundancy groups for the following objects:

- CardRedundancyGroup — Consists of two or more Card objects that participate in a redundant configuration.
- NetworkAdapterRedundancyGroup — Consists of two or more Port or Interface objects that participate in a redundant configuration.
- NetworkConnectionRedundancyGroup — Consists of two or more NetworkConnection, Cable, or TrunkCable objects that participate in a redundant configuration. NetworkConnectionRedundancyGroups (NCRG) are simply reference classes, they are not created during discovery. There is only one relationshipset: ComposedOf/PartOf between members of the NetworkConnection class. If that is populated, the typical events (ReducedRedundancy, AtRisk, AllComponentsDown) work as expected.
- SystemRedundancyGroup — Consists of two or more system (Switch, Router, Hub, Bridge, Host) objects that participate in a redundant configuration.

- HSRPGroup — Consists of two or more Cisco devices that support the Hot Standby Router Protocol (HSRP) and are configured to ensure that user traffic recovers immediately and transparently from first hop failures in network edge devices or access circuits. The devices are connected to the same segment of a network and, using HSRP, work together to present the appearance of a single router on the LAN. The devices in an HSRPGroup share an IP address and a MAC (Layer 2) address.
- VRRPGroup — Consists of two or more VRRP enabled routers. The VRRPGroup is created for each Virtual Router Identifier (VRID) and is composed of a logical element defined per interface per VRRPGroup on a VRRP router called VRRPEndpoint. VRRPEndpoints belonging to the same VRRPGroup (that is, having the same VRID) have a common virtual IP and a Virtual MAC. The Virtual MAC is defined as xx-xx-xx-xx-xx-VRID.

Card redundancy groups

IP Availability Manager automatically creates CardRedundancyGroups during discovery for Cisco devices that support the CISCO-STACK-MIB. Each such group consists of two cards. You can view the card redundancy groups and their members by using the Domain Manager Administration Console.

In addition, you can create redundancy groups for cards that do not support the CISCO-STACK-MIB. The process for manually creating a card redundancy group is similar to that for creating a system redundancy group, described in the following section. First, create an instance of the CardRedundancyGroup class and then insert the cards that participate in the redundancy group into the ComposedOf relationship.

System redundancy groups

Redundancy groups for network adapters, network connections, and systems must be manually created by using the **dmctl** utility or an Adapter Scripting Language (ASL) script.

To create a redundancy group for one of these types of objects:

1. Create an instance of one of the redundancy group classes.
2. Insert the objects that participate in the redundancy group into the ComposedOf relationship of the redundancy group.

   The following example uses the **dmctl** utility to create an instance of a system redundancy group and inserts two members into the group:

   ```
   /opt/InCharge7/IP/smarts/bin/dmctl -s INCHARGE-AM-PM
   Server INCHARGE-AM-PM User: admin
   admin's Password: XXXXXXXX
   Domain Manager Control Program (V7.0.0.0) -- Type 'help' for a list of commands.
   Attached to 'INCHARGE-AM-PM'
   dmctl> create SystemRedundancyGroup::RouterRedundancyGroup
   dmctl> insert SystemRedundancyGroup::RouterRedundancyGroup:: ComposedOf Router::router1.smarts.com
   dmctl> insert SystemRedundancyGroup::RouterRedundancyGroup:: ComposedOf Router::router2.smarts.com
   ```

   Network adapter, network connection, and system redundancy groups can be viewed through the Domain Manager Administration Console or the Topology Browser view of the Global Console.

Network adapter redundancy groups

The status of a network adapter redundancy group changes when at least one adapter in the group is flapping or has a status of down, disabled, not present, or testing.

Note By default, a value of testing for the network adapter's status results in a Down notification. You can modify this behavior with the Testing Notification Mode.

In addition, you can create a network adapter redundancy group where the status calculation considers the state of the system containing the adapter. System status is useful when the group's members come from multiple systems. You determine the behavior by the topology you create. When all the network adapters in a group come from the same system, you can make the group PartOf a UnitaryComputerSystem. This prevents the unresponsive status of the system from affecting the status calculation.

The following examples use ASL code fragments to show several different network adapter redundancy groups.

In the first example, both network adapters are part of the host named host_1. Because of this, the redundancy group is inserted into the PartOf relationship with the host. If host_1 is unresponsive, IP Availability Manager does not generate any notifications about the redundancy group. If the group is not part of the system, IP Availability Manager would notify AllComponentsDown when host_1 is unresponsive.

Also, note the use of the LayeredOver relationship from the network adapter redundancy group to the system. This information is used to draw maps of network adapter redundancy groups in the Global Manager:

```
rg = create("NetworkAdapterRedundancyGroup",
   My-Redundancy-Group");
//
// Turns off system status calc
rg->PartOf += object("Host", "host_1");
//
// For service map
rg->LayeredOver += object("Host", "host_1");
//
rg->ComposedOf += object("Interface", "IF-host_1/1");
rg->ComposedOf += object("Interface", "IF-host_1/2");
```

In the next example, the network adapters are from different systems. Because of this, the redundancy group is not inserted into the PartOf relationship with any system. As a result, the status of this redundancy group is based on the status of the adapters and the responsive state of their containing systems:

```
rg = create("NetworkAdapterRedundancyGroup",
   My-Redundancy-Group");
//
rg->ComposedOf += object("Interface", "IF-host_1/1");
```

```
rg->ComposedOf += object("Interface", "IF-host_2/2");
//
// For service map
rg->LayeredOver += object("Host", "host_1");
rg-> LayeredOver += object("Host", "host_2");
```

In the next example, the network adapters are from a single system, but the system status is considered in the status calculation because the group is not a part of a system:

```
rg = create("NetworkAdapterRedundancyGroup",
   My-Redundancy-Group");
//
// For service map
rg->LayeredOver += object("Host", "host_3");
//
rg->ComposedOf += object("Interface", "IF-host_3/1");
rg->ComposedOf += object("Interface", "IF-host_3/2");
```

In the final example, the network adapters are from a single system but the system status is not included in the status calculation for the redundancy group. Again, this is because the group is inserted into the PartOf relationship with the system which consists of the network adapters.

```
rg = create("NetworkAdapterRedundancyGroup",
   My-Redundancy-Group");
//
// Turns off system status calc
rg->PartOf += object("Host", "host_4");
//
// For service map
rg->LayeredOver += object("Host", "host_4");
//
rg->ComposedOf += object("Interface", "IF-host_4/1");
rg->ComposedOf += object("Interface", "IF-host_4/2");
```

NIC teaming functionality

The primary purpose of the NIC teaming functionality is that when a teamed NIC is employed, the standard MIB-II tables only represent the logical (teamed) NIC. This inherently inhibits the creation of L2 connectivity between the Host, with the teamed NICs, and the adjacent switch or switches. The reason is that the same logical MAC is observed on multiple access ports. The NIC teaming functionality pulls the physical NIC information (including the BIM - Burned-in MAC) from the Enterprise MIBs and creates a NetworkAdapterRedundancyGroup to represent the logical grouping of the physical NICs. This allows L2 Cable connections to form normally with adjacent Switches.

Note NIC teaming is possible for a number of different media types. However, the primary focus is on attempting to solve the issue for Ethernet-media devices.

NIC teaming support is applicable specifically to device specific discovery. NIC teaming support occurs at three different points in IP Availability Manager which are as follows:

- IP Availability Manager initialization — This involves loading of the various meta-data to provide network access redundancy and allow rapid convergence when a failure occurs.
- oid2Type configuration — This involves defining the specific probes to be launched for the vendors that are supported.
- Discovery probes — This involves additional data collection handled through the normal additional discovery probes.

  Upon discovery of the device (host) that is configured for NIC teaming, a redundancy group namely, NetworkAdapterRedundancyGroup is created in the IP Availability Manager. A NetworkAdapterRedundancyGroup consists of two or more port or interface objects that participate in a redundant configuration (NIC teaming). The NARG (NetworkAdapterRedundancyGroup) can indeed consist of Ports or Interfaces, but the members of the teamed NIC will always be created as Interfaces:

  Relationship: The NetworkAdapterRedundancyGroup has a ComposedOf relationship with Interfaces and Interfaces have a PartOf relationship with NetworkAdapterRedundancyGroup.

Network adapter redundancy group for NIC Teaming

NIC teaming functionality provides network access redundancy and allows for rapid convergence when a failure occurs. NIC teaming is the process of grouping together several physical NICs (Network Interface Cards) into one single logical NIC, which can be used for network fault tolerance and/or transmission load balancing. In a team of two or more NICs, the secondary NIC assumes the primary or active role, if the primary NIC fails. However, failover will not occur unless the primary NIC is specified.

NIC teaming has two purposes:

- Fault Tolerance: By binding more than one physical NIC to a logical NIC, high availability is maximized. Even if one NIC fails, the network connection does not cease and continues to operate on other NICs.
- Load Balancing: Balancing the network traffic load on a server can enhance the functionality of the server and the network. Load balancing within a network interconnect controller (NIC) team enables distributing traffic amongst the members of the NIC team so that traffic is routed among all available paths. In most scenarios of Load Balancing, it is required that teamed connections usually terminate on the same remote device and that the device has an Etherchannel (or equivalent protocol) configured.

HSRPGroups

During discovery, IP Availability Manager automatically creates HSRPGroups for Cisco devices that support the CISCO-HSRP-MIB and are configured in an HSRPGroup. Each such group is composed of two or more routers and their interfaces.

You can view the HSRPGroups and their members by using the Domain Manager Administration Console and the Global Console. Analysis and monitoring of a HSRPGroup occurs automatically after the group is created and its member routers are inserted into the group.

VRRP Groups

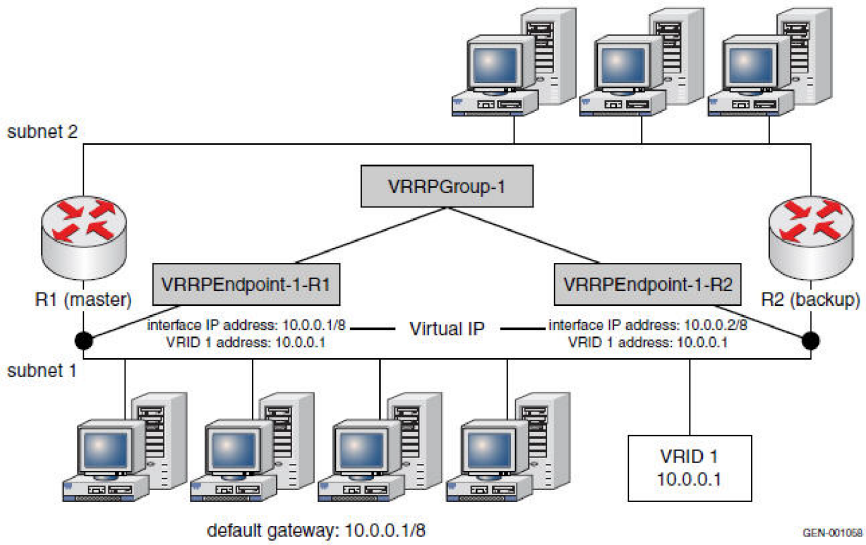
During discovery of the VRRP enabled routers, IP Availability Manager creates a new redundancy group called VRRPGroup. The VRRPGroup is created for each Virtual Router Identifier (VRID). It is composed of a logical element defined per interface per VRRPGroup on a VRRP router called VRRPEndpoint. VRRPEndpoints belonging to the same VRRPGroup (that is, having the same VRID) have a common virtual IP and a Virtual MAC. The Virtual MAC is defined as xx-xx-xx-xx-xx-VRID. The VRRP-enabled router having the highest priority is considered as the master router and all other routers are considered as backup routers.

Note Virtual IP of VRRP is derived from vrrpAssoIpAddr, which is part of Indexes in vrrpAssoIpAddrTable. If the vrrpAssoIpAddrTable does not contain values in the MIB, then VRRPGroup will not be created. If ipNetToMediaTable does not contain values with current Virtual IP for that VRRPGroup, then relationships like HostedBy, LayeredOver, ConnectedVia, Underlying, ConcreteDependency_Dependent will not be created.

Basic VRRP Configuration

Basic VRRP configuration illustrates the basic VRRP configuration.

Figure 6-2. Basic VRRP configuration



The VRRP configuration uses a single Virtual Router Identifier (VRID 1). Router R1 is the address owner and serves as the master router and router R2 serves as the backup router. The four end hosts on subnet 1 are configured to use 10.0.0.1/8 as the default router. IP address 10.0.0.1 is associated with VRID 1.

In this example, if R1 becomes unavailable, R2 takes over VRID 1 and its associated IP addresses. Packets sent to IP destinations outside the 10.x.x.x subnet using 10.0.0.1 as the router are then forwarded by R2. Even though R2 assumes the forwarding responsibilities of R1, it may or may not process any packet with destination address (DA) 10.0.0.1, depending on the accept-data configuration. When R1 becomes active again, it takes over as the master router and R2 reverts as the backup router.

VRRP supported devices

The following devices are supported for VRRP:

- All devices that implement RFC 3768
- Nortel devices
- Juniper ERX 700,1400 switches

Performance Manager and Server Performance Manager

7

This chapter includes the following topics:

- About IP Performance Manager
- Performance monitoring
- Performance analysis
- Performance notifications
- Before you start
- SNMP traps processed
- DisplayName and Name attributes

About IP Performance Manager

Knowing the health of the network objects in your network is crucial to ensuring optimum network performance. Constant monitoring, especially of critical ports in the network, is necessary to detect when performance deteriorates so that you can take appropriate corrective action.

IP Performance Manager monitors the performance of Layer 2 and Layer 3 network objects in switched and routed networks, and automates the analysis of faults using performance thresholds. Faults can occur in systems (switches, routers, hubs, bridges, and hosts) and in network adapters (ports and interfaces). When a performance threshold is exceeded, IP Performance Manager generates a system-level exception notification.

Note Thresholds can be set in the Polling and Thresholds tab in the Domain Manager Administration Console. The *VMware Smart Assurance IP Manager User Guide* provides information on how to set thresholds.

For example, when a particular network adapter exceeds the ErrorThreshold, IP Performance Manager notifies HighErrorRate fault for network adapter and generates an ErrorException for the system that contains the network adapter. In addition, if the network adapter is a member of a Virtual Local Area Network (VLAN), IP Performance Manager also generates the exception notification at the VLAN level.

Performance monitoring

After the initial discovery and modeling of the network, IP Performance Manager continuously monitors the status of the discovered systems using:

- Internet Control Message Protocol (ICMP) polling
- Simple Network Management Protocol (SNMP) polling

  IP Performance Manager also processes SNMP trap messages received from the network to obtain status information for the discovered systems.

  The VMware Smart Assurance IP Manager Reference Guide provides information about the MIBs polled and SNMP traps processed by the IP Performance Manager.

Performance analysis

IP Performance Manager retrieves fault-related, intra-system information from each network object and uses that information to conduct performance analysis.

Performance notifications

IP Performance Manager exports the results of its network performance analysis to the Global Manager in the form of notifications, which are displayed in the Notification Log Console view of the Global Console. The notifications identify IP Performance Manager by its domain name in the Source attribute and also the IP Performance Manager can be identified by its domain manager group. Users can double-click a notification to view detailed information about the notification.

IP Performance Manager creates three types of notifications: faults, events, and exceptions. Exceptions are also referred to as aggregate events.

- A fault indicates a performance problem detected by IP Performance Manager. Examples of faults are:
  - HighErrorRate for a network adapter
  - HighUtilization of a switch’s backplane
  - HighBufferMissRate on a system’s memory component
- An event indicates an abnormal condition detected by IP Performance Manager.

- An exception indicates that one or more faults are occurring in a particular system or VLAN.

  The VMware Smart Assurance IP Manager User Guide provides information about viewing the notifications, and the VMware Smart Assurance IP Manager Reference Guide provides descriptions and examples of the types of faults detected by IP Performance Manager.

  Note You view the impacts for a network problem notification by clicking the Impact tab in the Notification Properties dialog box. The Impact tab lists all of the affected objects for the notification. Because IP Performance Manager does not diagnose root-cause problems, it does not diagnose impacts. Therefore, the Notification Properties dialog box for notifications generated by IP Performance Manager does not contain the Impact tab.

Before you start

The *Network Objects and Their Failures* section in the VMware Smart Assurance IP Manager Reference Guide identifies the topology objects discovered and the events notified by the IP Performance Manager.

Console operations are controlled by certain console operation options such as Notification Properties, Expand Map Node, and Show Containment. The VMware Smart Assurance Service Assurance Manager Configuration Guide provides instructions on changing the console operations.

IP Performance Manager also instantiates device connection, collection, management agent, and redundancy group object types, but because IP Performance Manager does no analysis for these object types, they are not discussed here. The VMware Smart Assurance IP Manager Reference Guide describes all of the object types instantiated by IP Performance Manager.

All of the object types instantiated by IP Performance Manager are available to a Global Console attached to IP Performance Manager, and a subset are available to a Global Console attached to the Global Manager. The VMware Smart Assurance IP Manager Reference Guide provides a list of the object types imported by the Global Manager.

SNMP traps processed

IP Performance Manager processes data found in the following fields of each SNMP trap message:

- Enterprise (the sysObjectID of the agent/object)
- Generic Trap Identifier
- Specific Trap Identifier
- Variable-Bindings
- IP address of the SNMP agent

  IP Performance Manager uses trap messages for two purposes:

- Discovery

- Analysis

SNMP traps processed for discovery

A trap message processed for discovery purposes conveys one of the following conditions for a device in the managed network:

- A device, not yet in the modeled topology, was started.
- A device, already in the modeled topology, was equipped with an additional field replaceable unit (FRU) or had an FRU removed; for example, a card was inserted into the device.
- A device, already in the modeled topology, had a TCP connection closed by a remote application; for example, a Telnet session to the device was terminated.

SNMP traps processed for analysis

A trap message processed for analysis purposes pertains to a device that has been discovered and therefore is already in the modeled topology. IP Performance Manager extracts the information from the trap message and updates the appropriate device attributes.

Note IP Performance Manager uses trap messages and the results obtained from periodic SNMP polling to detect network degradations that adversely affect network performance.

Chapter 3, MIBs Polled and SNMP Traps Processed, in the
VMware Smart Assurance IP Manager Reference Guide, provides more information.

DisplayName and Name attributes

Each network object has a unique name, which is defined by the DisplayName and Name attributes for the object.

Naming convention for network objects

For a discovered system, IP Performance Manager uses a consistent naming convention to assign the system and each of its components a unique name. The name assigned to a system is determined by the setting of the NameFormat parameter in the name-resolver.conf file, which is located in the BASEDIR/smarts/conf/discovery directory. The assigned name becomes the value of the system's Name attribute. Refer How discovery names a system for information about the name resolution process used to name a system.

Note The values of a system's Name attribute and DisplayName attribute may be different, depending on the setting of the DisplayNameFormat parameter in the discovery.conf file. The
VMware Smart Assurance IP Manager Reference Guide describes the configuration parameters in the discovery.conf file.

To name a discovered network object, IP Performance Manager uses the appropriate naming format. The naming formats are the default naming conventions. If no naming format is listed for the Display Name, the naming format is the same as the one listed for Name.

NAS

8

This chapter includes the following topics:

- Introduction to network-attached storage
- The VMware Smart Assurance IP NAS solution
- NAS IP discovery
- Celerra devices for NAS Extension functionality
- Scenarios
- NAS notifications
- NAS topology map
- NAS containment

Introduction to network-attached storage

Network-attached storage along with storage area networks (SANs) are increasingly important as enterprise storage solutions. SANs are optimized for high-volume block-oriented data transfers, while NAS provides storage services for applications that access data at the file level. Network-attached storage systems include centrally managed servers and an operating system that is optimized for fast file access and heterogeneous file sharing over an IP network.

NAS provides facilities for high availability, rapid recovery, data protection, ease of management, and data backup and recovery. NAS is typically implemented for messaging, development applications, business and regulatory compliance, and general file sharing. NAS is flexible and extensible, allowing for the addition of storage devices to a network, as needed.

A NAS system includes a network server, file servers, and software for configuring and mapping file locations to the network-attached devices. Each NAS device is attached to a local area network (typically, an Ethernet network) and assigned an IP address. The storage system may be a separate back-end system connected either logically or physically to the NAS system.

A NAS system discovers Control Station as a host through SNMP and gets information on file servers using HTTP adapter. A logical entity called NASChassis will PackageSystem host and fileservers.

Overview of the IP Availability Manager Extension for NAS

The VMware Smart Assurance IP Availability Manager Extension for NAS (NAS Extension) is applied to an existing IP Availability Manager installation. After installation, the Availability Manager can discover NAS devices, and perform root cause and impact analysis on these devices.

The VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers provides information on installation and startup issues for the IP Availability Manager Extension for NAS. The guide also provides information on how to install and start the HTTPS Adapter service.

With the NAS Extension, the Availability Manager discovers and models network-attached storage devices as IP devices. Supported systems include VMware Celerra® systems (Data Movers and Control Stations), and Network Appliance (NetApp) Filers.

The Availability Manager with the NAS Extension identifies IP-related failures, diagnoses root cause problems, and calculates the impacts of failures.

To discover and monitor the NAS topology, the Availability Manager reads configuration information within the Celerra Network Server using the HTTPS Adapter, an XML/HTTPS discovery probe. The VMware Smart Assurance Global Console displays the topology, notifications, and maps related to both the IP and NAS elements.

The NAS Extension is implemented through the IP Availability Manager and Service Assurance Manager.

Certified NAS Devices

The following devices are certified for use with the IP Availability Manager Extension for NAS:

- Celerra CNS, NS series, and NSX, with NAS release 4.0 and later.

  Note To support NAS Extension management functionality, the Celerra Control Station must support XHMP service and have HTTPS enabled.

- NetApp: F700, F800 and F900 series with ONTAP 6.1—6.5

VMware Celerra devices

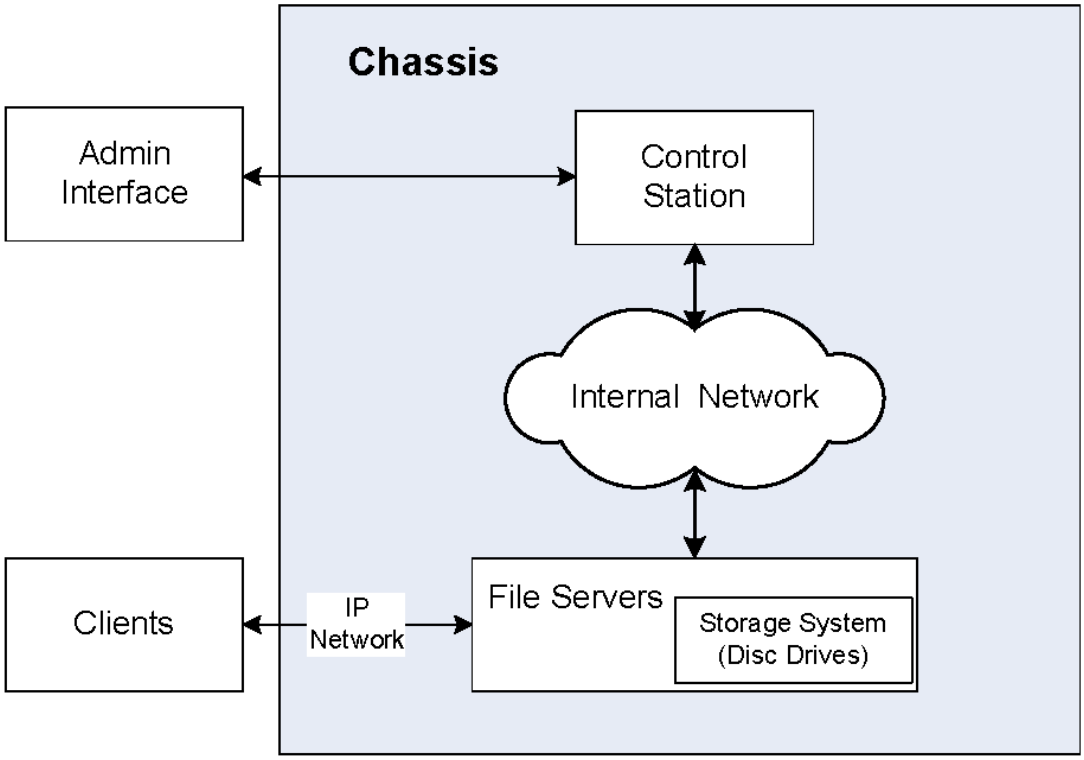
Celerra is a high performance IP storage platform. It delivers NAS (Network Attached Storage) and iSCSI capabilities to consolidate application storage and file servers in either an integrated configuration, or as a gateway connected to a CLARiiON or Symmetrix storage system.

A Celerra installation includes the following IP-related devices:

- Celerra Network Server
- Celerra Control Station.
- Celerra Data Movers

  NAS device relationships shows these relationships.

Figure 8-1. NAS device relationships



The Control Station includes an IP address that is exposed on the external network. The Data Movers also have IP addresses for external client connectivity.

The Data Movers and Control Station communicate with each other using Ethernet connections over the following supported file access protocols: CIFS, NFS, FTP, TFTP. The Availability Manager does not currently discover or monitor the internal network or the back-end storage systems.

Celerra Network Server

The network server (not a physical device), is represented by the NASChassis class. A NASChassis is a cabinet that holds one or more Data Movers and one or more Control Stations.

Celerra Control Station

The Control Station, represented by the Host class, is a dedicated management computer that monitors and controls cabinet components and allows access to the full functionality of the Celerra Network Server software. It contains utilities for installing and configuring the Celerra Network Server, maintaining the system, and monitoring system performance. The Control Station runs a set of programs that are collectively referred to as the Control Station software, and an VMware-customized version of Linux as its operating system. The Control Station provides the Celerra Manager, a web interface for Celerra administration.

Celerra Control Station is discovered as a Host by IP. Host (Celerra control system) is defined by *SystemsPackagedIn* relationship with NASChassis. Conversely, NASChassis has a *PackagesSystems* relationship with FileServer and Host.

Celerra File Servers or Data Mover

The Data Movers, represented by the FileServer class, are the Celerra components that transfer data between the storage system and the network client. A Celerra Network Server can have from one to 14 Data Movers.The Data Movers are managed via the Control Station.

A Data Mover can be active (also called primary) or, in a Celerra Network Server model that supports multiple Data Movers, can be a standby for other Data Movers.

Each Data Mover is independent; there is no communication between Data Movers. If a Data Mover fails, the Control Station manages the failover to the standby Data Mover, assuming a standby has been configured.

A Data Mover and the Control Station communicate with each other using both Ethernet and serial connections. A Data Mover communicates with the network using network interface cards (NICs), which may support 10/100/1000Base T Ethernet or optical Gigabit Ethernet.

A Data Mover communicates with the back-end data storage systems using Fibre Channel connections. In the Celerra Clustered Network Server, storage processing is provided by a component of the Data Mover, a Fibre Channel HBA card, that connects to the storage array through a Fibre Channel switch. In the Celerra NS series, storage processing is provided by an auxiliary storage processor enclosure containing two storage processors.

Celerra File Servers or Data Movers are discovered as its FileServers by IP. A FileServer is defined by *ComposedOf* relationship with FileSystem. Conversely, a FileSystem is defined by *PartOf* relationship with FileServer.

NetApp Filer

NetApp Filers, also represented by the FileServer Class, are the NetApp storage systems. These devices combine file system technology with a proprietary operating system that provides for management as well as optimized storage utilization.

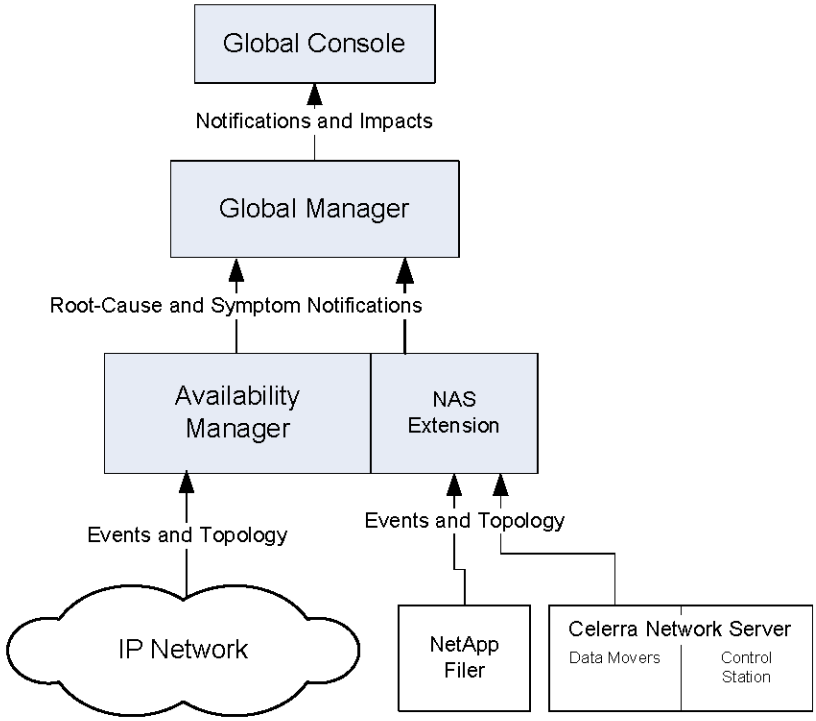
The VMware Smart Assurance IP NAS solution

The IP NAS solution architecture consists of the following components:

- IP Availability Manager — A Domain Manager that discovers, monitors, and analyzes failures in the underlying IP network. With the NAS Extension applied, the Availability Manager discovers and monitors NAS devices, correlates events, diagnoses root-cause problems, and calculates impacts.
- Global Manager — Abstracts and consolidates information from the Domain Managers and deduplicates notifications.
- Global Console — Provides display capability for notifications, maps, and other graphical information.

  Managed NAS architecture shows the components and the flow of information for the managed NAS architecture.

Figure 8-2. Managed NAS architecture



Monitoring

To discover the IP configuration of a NAS device, the Availability Manager uses the HTTPS Adapter, which initiates a process contained within the NAS Extension. This XML-based probe associates the Data Movers with the Celerra Network Server via the Control Station.

After the initial discovery of the NAS IP elements by the IP Manager, the Availability Manager continuously monitors the status of these elements by periodically sending ICMP and SNMP polls to the IP devices. NAS IP discovery provides more information. The Discovery methods provides more information about the full IP discovery process.

Analysis

The Availability Manager correlates root cause failures in the underlying IP transport network to problems in the NAS IP elements. For example, the Availability Manager indicates the NAS IP elements that have been impaired or impacted by the failure of a router in the IP transport network.

Notifications

The Availability Manager reports its analysis results to the Global Manager in the following types of notifications: root-cause problem notifications, symptomatic event notifications, and aggregate event notifications.

Relevant notifications are displayed in the Notification Log Console view of the Global Console. Root-cause failures diagnosed by the Availability Manager display a value of “Yes” for the value of the IsRoot attribute and identify the Availability Manager by its domain name in the Source attribute.

Impacts

The Availability Manager calculates the Impacts that the failures of one managed element may have on other managed elements. Impact information is included in notifications.

NAS IP discovery

The Availability Manager with the NAS Extension uses standard ICMP (Ping) and SNMP probes to determine device status and containment information. In addition, the NAS Extension uses an HTTPS Adapter to gather configuration information about Celerra Control Station devices, and some post-processing steps to complete the topology creation.

File Servers present in different networks can be discovered in different IP Managers. During post-processing the fileservers are related to the NASChassis.

Note The internal Celerra IP network (the IP network that supports communication between file servers and the control station hosts) is automatically excluded from discovery.

ICMP and SNMP probes

The ICMP probe determines whether a device is operational.

The SNMP probe uses the management, MIB-2, system information to determine the type of device. If the NAS Extension determines that the device is a Control Station, the Availability Manager launches the HTTPS Adapter probe. In addition, if the MIBs are included in the SNMP Agent of the device, the SNMP probe gathers containment information about the device’s interfaces and IP address.

HTTPS/XML discovery probe

The HTTPS Adapter gathers information about Celerra devices that are unavailable through the Availability Manager SNMP probes. When the Availability Manager encounters a NAS device during discovery, it launches a probe that queries the Control Station for information such as the associated FileServers and software version.

Discovery in the domain manager

IP Manager discovers the Control Station and gets the list of file servers and its details through HTTP adapter, this information is cached in the NASChassis class.

The Control Station objects such as, Host, fileserver, and NASChassis will be created in the IP Manager.

Note When the Host (Control Station) is discovered from IP Manager, both the Host (Control Station) and File Server (Data Mover) objects are moved to the pending list, as UNDISCOVERED although the host is present in the topology. When the Data Movers are discovered, all the NAS devices are removed from the pending list.

NAS post processing

Post-processing creates the NAS topology as follows:

- Sets the NASChassis object display name
- Sets IP Address, Software Version, Slots and Standby.
- Sets the FileServer object display name
- Sets the Host (Control Station) object display name
- Deletes obsolete NASChassis, FileServer and Host (Control Station) objects

  The IP Availability Manager associates Host objects (Control Stations) and FileServer objects (Celerra Data Movers) with the NASChassis object.

  In addition, the Availability Manager establishes relationships between these components and the underlying IP network.

  The NAS topology is stored in the Repository and displayed on the Global Console using the Topology Browser and various maps. For more information, refer to NAS notifications.

Celerra devices for NAS Extension functionality

To support NAS Extension functionality, you may need to configure the following:

- Control Station and Data Mover IP addresses
- Control Station usernames and passwords

Control Station and Data Mover IP addresses

The IP Availability Manager Extension for NAS requires access to the Control Station and Data Mover IP addresses that are exposed on the external network. You configure the IP addresses after the devices are cabled to the network switches.

For information about cabling the Celerra Network Server, refer to the *Celerra NS500/NS600/NS700/NS704 Integrated Configuration Version 5.4 Setup Guide*.

The *Configuring and Managing Celerra Networking* and *Configuring and Managing Celerra Network High Availability* technical modules provide more information about how to configure the external IP addresses once the cables are in place.

Control Station usernames and passwords

When using the HTTPS Adapter, the Availability Manager accesses each Celerra Control Station device. Some Control Station devices may restrict access to configured usernames and passwords. If the default username and password (*nasadmin*) are used, no configuration is required. For information about encryption and other security issues, refer to the VMware Smart Assurance System Administration Guide.

Scenarios

The Availability Manager builds a data model of the discovered NAS IP elements in the managed environment. The model represents the discovered elements as instances of the VMware Smart Assurance ICIM classes.

The NAS IP elements are represented by instances of the following ICIM classes:

- NASChassis
- FileServer
- Host (Celerra Control Station)

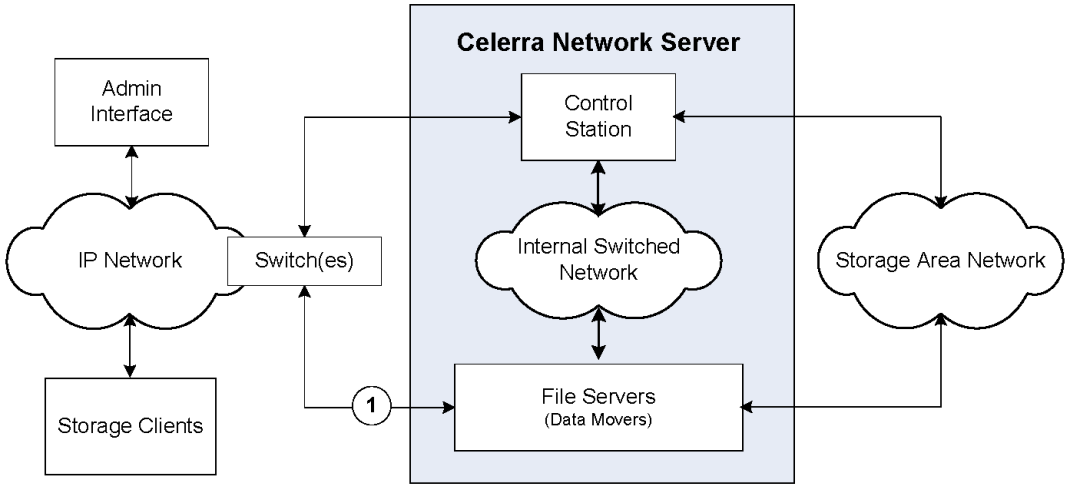
  During the discovery post-processing phase, the IP Manager creates the relationships and connections between the storage objects: Celerra Data Mover (FileServer object), NAS Network Server (NASChassis object), and Celerra Control Station (Host object).

  Note that, in addition, the Availability Manager discovers the Layer 2 and Layer 3 elements and relationships that are described in the VMware Smart Assurance IP Manager Reference Guide; for example, Cable, Card, Interface. The VMware Smart Assurance IP Manager Reference Guide also provides more information about the VMware Smart Assurance ICIM classes.

Scenario 1: External IP network cable linking to NAS device fails

External network cable failure to a file server shows a scenario where the external connection to a file server fails because a cable is down (1 in the figure). Because a file server may be equipped with a hot spare and failover may occur automatically, NAS functionality may not be affected. However, the operator needs to know that the cable has failed and which devices may have been affected.

Figure 8-3. External network cable failure to a file server



In this case, the following notifications may be generated:

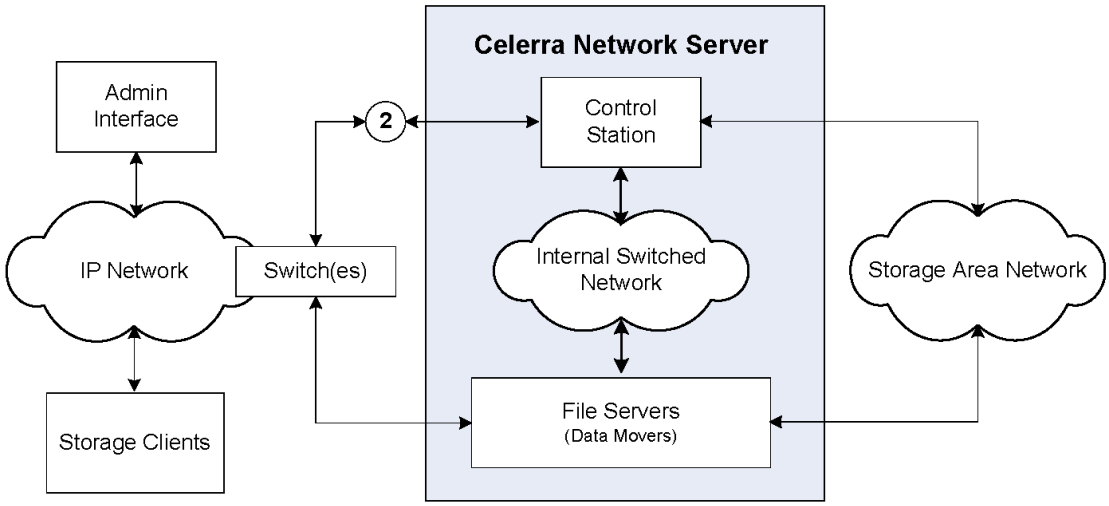
- Notifications: Cable Down, IP Network Switch Interface Down, FileServer Unresponsive (in the event that failover does not happen)
- Root Cause: Interface Down
- Impact: FileServer Down

Scenario 2: Connectivity loss on the Control Station external interface

When an external interface to the Control Station fails because of device failure or loss of connectivity to the IP network, the Control Station may fail over to another host. However, the operator needs to know that the interface has failed and which devices may have been affected.

External network cable failure to a control station shows a scenario where an external interface on the Control Station has failed (2 in the figure).

Figure 8-4. External network cable failure to a control station



In this case, the following notifications are generated:

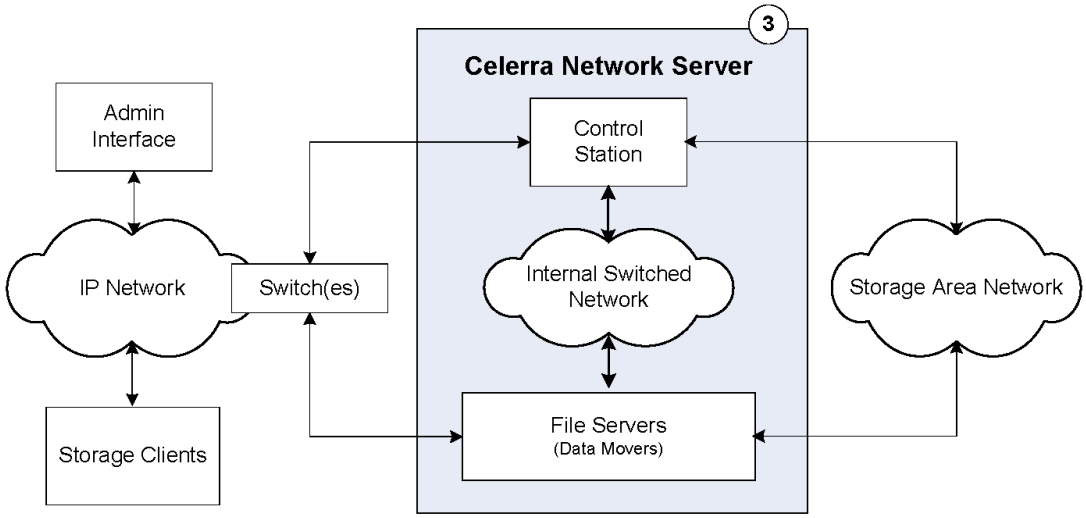
- Notifications: Host (Control Station) Unresponsive
- Root Cause: Interface Down
- Impact: Host (Control Station) Unresponsive

Scenario 3: Network Server fails due to a power supply failure

In the event that the power supply to the Network Server is interrupted, for example, because a circuit breaker is tripped, all of the systems packaged within the Network Server will fail.

Power Failure on the network server shows a scenario where the Network Server is powered off (#3 in the figure).

Figure 8-5. Power Failure on the network server



In this case, the following notifications are generated:

- Notifications: NASChassis Down, Host Unresponsive, FileServer Unresponsive
- Root Cause: NASChassis Down

  Impacts: FileServer Unresponsive, Host Unresponsive

NAS notifications

You can view the notifications, maps, and containment information by attaching the Global Console to the Global Manager. The Availability Manager reports NAS device notifications to the Global Manager. You can then view the notifications through the attached Global Console in two basic ways:

- As table entries in a Notification Log Console view
- As color-coded severity icons in a Map Console view

  The VMware Smart Assurance IP Manager User Guide provides information on how to obtain detailed information about an individual NAS device notification and the Notification Properties dialog box.

NAS topology map

The IP Availability Manager sends NAS device topology—instances of the NASChassis, Host and FileServer classes—to the Global Manager along with the standard IP topology.

The Global Console presents the topology information in a variety of dynamically updated formats that show the status of the NAS device and network elements and their many relationships. One of those formats is the topology map, which is a graphical representation of the topology.

The VMware Smart Assurance IP Manager User Guide provides information on how to open a NAS device topology map.

NAS device topology map graphical representations

NAS device Topology maps contain FileServer and Host (Control Station) elements, along with their relationships and connections. In a map display, a *node* is a graphical representation of an element, and an *edge* is a graphical representation of a relationship or connection between elements.

Default Node for NAS device topology maps identifies and describes the default node icon for FileServer that may appear in a NAS topology map.

Table 8-1. Default Node for NAS device topology maps

Icon / Visual indicator	Description
	FileServer, such as Celerra Data Mover or NetApp Filer

You can replace the standard map nodes with other map nodes that are preferred by your organization. For detailed information about maps, refer to the
VMware Smart Assurance Service Assurance Manager Operator Guide.

NAS Connectivity Map

The NAS Connectivity map shows logical connectivity between NAS FileServer elements and NAS Host elements. No physical connectivity details are included in this map. See NAS Connectivity Map for an example.

The initial layout of a NAS Connectivity map depends on the element that you select to launch the map. The edge marked with a + (plus) sign indicates that the edge can be expanded to show hidden MountPoints.

Selecting a NAS element results in an initial layout that centers around the selected element (also known as the map's focal object) and shows the relationships to all elements within one hop of the selected element. Expanding the expandable nodes and edges reveals additional relationships and elements and eventually ends in a fully expanded map.

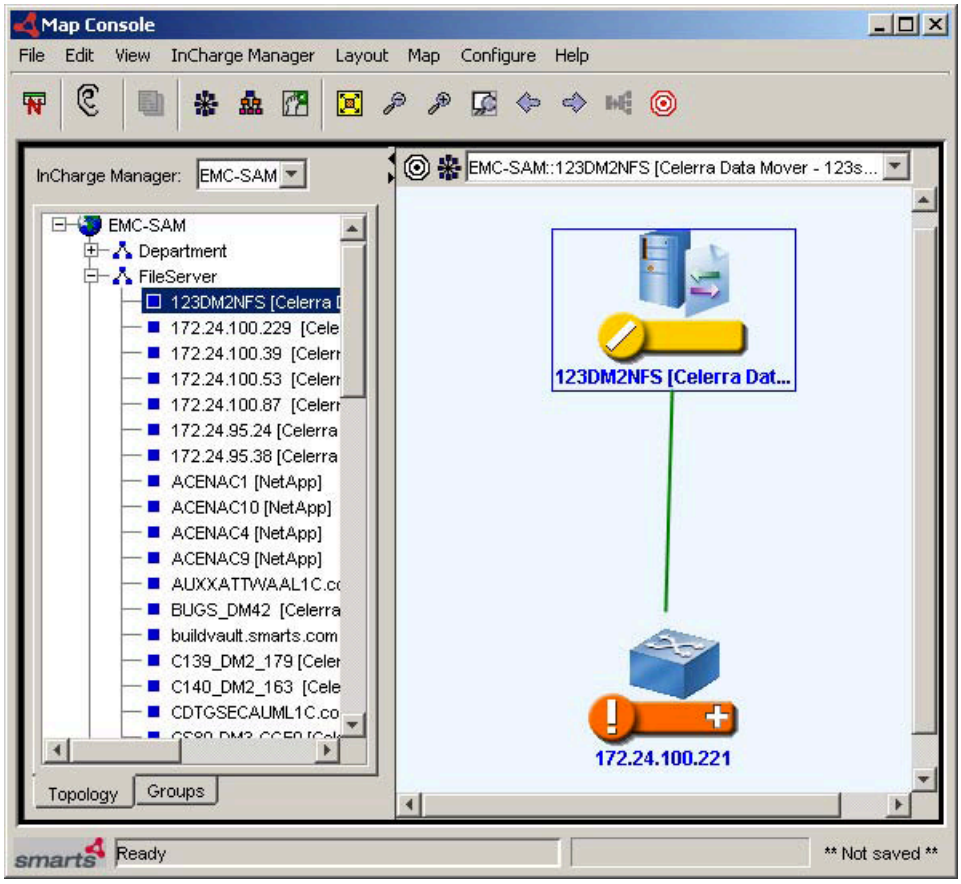
NAS device map types

The following Availability Manager map types show NAS devices.

- Physical Connectivity
- IP Network Connectivity
- NAS Connectivity
- SAN Connectivity

  Physical connectivity map with a NAS FileServer shows a Physical Connectivity map for a NAS FileServer. In this map, the FileServer (123DM2NFS, a Celerra Data Mover) is displayed with a major alert.

Figure 8-6. Physical connectivity map with a NAS FileServer



IP Network Connectivity Map with a NAS FileServer shows an IP Network Connectivity map with the same FileServer.

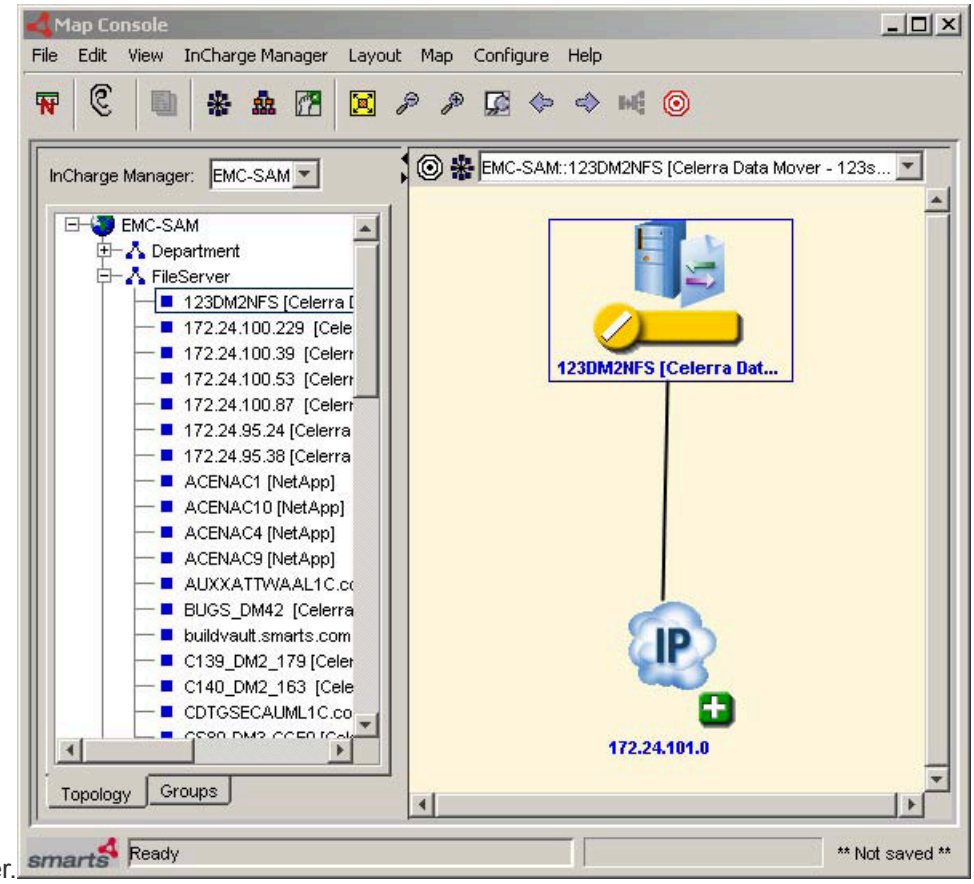


Figure 8-7. IP Network Connectivity Map with a NAS FileServer

NAS Connectivity Map with a NAS FileServer shows a NAS Connectivity map with a FileServer named server_2.

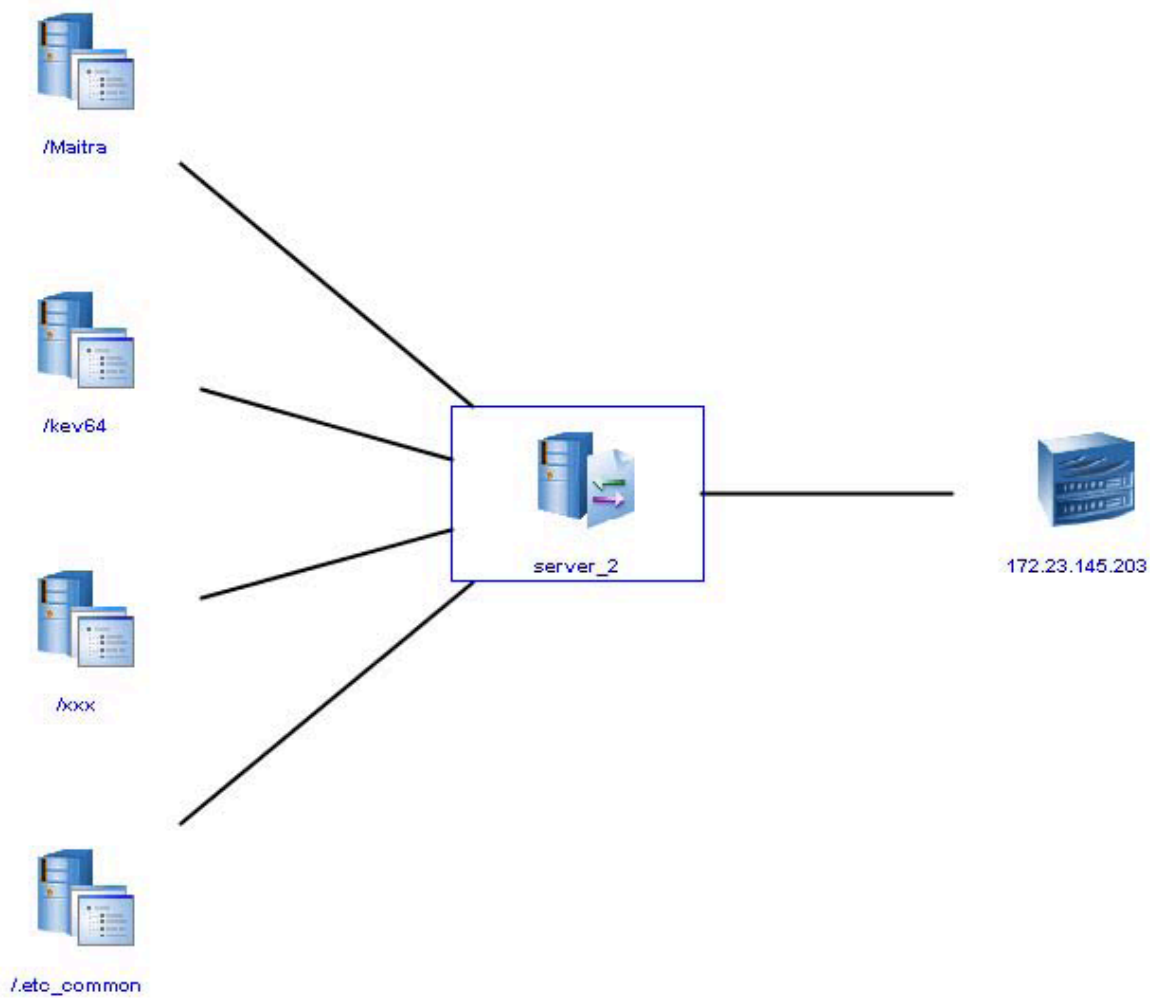


Figure 8-8. NAS Connectivity Map with a NAS FileServer

NAS containment

The Containment view shows topology information for each type of NAS element. An element's Containment view displays in a dialog box that organizes the information for the element into tab pages and tables.

In a Containment view, different information is available for different types of elements. The containment information is generated by the Availability Manager upon request by the Global Manager.

The VMware Smart Assurance IP Manager User Guide provides information on how to obtain containment information about an individual NAS device element and also on the NAS Containment dialog box.

Terminology

9

This section describes common terms used in the IP Manager.

This chapter includes the following topics:

- System and device
- Modeled topology
- Object
- Domain Manager
- Satellite Domain Manager
- Device connection
- Discovery
- Map Console
- NASChassis Element
- Edge (Map Console display component)
- IP Availability Manager
- IP Performance Manager
- Event
- Icon (Map Console display component)
- IP Manager
- Global Manager
- Notification
- Notification Log
- Repository
- Global Console
- Impact Analysis
- Logical link

- Partition
- Managed Element
- Root-Cause Analysis
- Topology Browser
- Adapter
- FileServer Element
- Certification
- Certification Matrix
- Codebook
- Correlation Model
- Correlator
- Broker
- Containment
- ICMP
- ICMP operation
- SNMP
- SNMP key components
- SNMP basic operations
- SNMP versions
- SNMPv1, SNMPv2c, and SNMPv3 security
- NASChassis
- FileServer
- Comparable terms
- Wildcard patterns

System and device

The term *system*, as used in this document, is a generic term representing a computer-based network entity, such as a switch, router, hub, bridge, or host. The term *device*, as used in this document, has essentially the same meaning as system except that *device* also conveys a sense of specific model, such as a specific model of switch, router, hub, bridge, or host.

Modeled topology

IP Availability Manager uses VMware Smart Assurance ICIM class models to create (within its repository) instances of the managed network systems (switches, routers, hubs, bridges, and hosts), their dependencies, their internal components, and their connections. The *modeled topology* mirrors the real topology in the managed network.

Object

The term “object” has a dual meaning: It simultaneously represents both:

- An VMware Smart Assurance object in the modeled topology.
- A physical or logical entity in the real topology. An VMware Smart Assurance object corresponds to a physical or logical entity in the real topology.

Domain Manager

A Domain Manager is a service assurance application that is associated with a particular type of information technology domain, such as networks, systems, applications, or application services. For IP Manager, the domain is IP network transport. A Domain Manager includes a Repository of the managed topology and a problem correlation model that monitors network elements, uses correlation technology to pinpoint the root causes of failures, and diagnoses the effects of failures on related elements. Domain Managers report results to the Global Manager for consolidation, and display through the Global Console.

Each Domain Manager is autonomous in the sense that it:

- Maintains its own data models, repository, and problem signatures.
- Monitors and analyzes the discovered objects in its own domain.

  VMware Smart Assurance IP Manager represents four VMware Smart Assurance Domain Managers:

- VMware Smart Assurance IP Availability Manager
- VMware Smart Assurance IP Performance Manager
- VMware Smart Assurance IP Server Performance Manager
- VMware Smart Assurance IP Availability Manager Extension for NAS

  IP Availability Manager and IP Performance Manager are examples of VMware Smart Assurance Domain Managers. You can choose to start IP Availability Manager (AM) only, IP Performance Manager (PM) only or a combination of both IP Availability Manager and IP Performance Manager (AM-PM).

  The VMware Smart Assurance Installation Guide for SAM, IP, ESM, MPLS, and NPM Managers provides the procedure to install services manually for the underlying servers used in the IP Management Suite.

Satellite Domain Manager

A satellite Domain Manager refers to any of the VMware Smart Assurance Domain Managers that import topology, CLI device-access objects (if needed), and status updates from the IP Availability Manager.

Satellite Domain Managers include:

- MPLS Manager
- Multicast Manager
- Network Protocol Manager for BGP
- Network Protocol Manager for EIGRP
- Network Protocol Manager for IS-IS
- Network Protocol Manager for OSPF VoIP Availability Manager
- VMware Smart Assurance VoIP Availability Manager

  IP Availability Manager Extension for NAS supplies additional event information to Storage Insight for Availability.

  Note A feature license controls whether IP Availability Manager Extension for NAS software is available for configuration. With this extension, IP Availability Manager can discover and monitor VMware Celerra®® NAS gateways and diagnose network-attached storage (NAS) connectivity problems.

Device connection

A device connection is a link between two network adapters. IP Availability Manager performs connectivity analysis on the following device connection object types:

- Cable — A connection between a port and an interface. For example, a cable connects a port on a switch to an interface on a router.
- TrunkCable — A connection between two ports. Switches are often trunked to connect multiple segments or to provide redundant pathways through the network.
- NetworkConnection — A connection between two interfaces. A network connection can be a logical connection or a physical connection. An example of a logical connection is when routers are connected through a virtual circuit and none of the intermediate network devices are included in the topology. An example of a physical connection is when routers are connected through a serial or point-to-point connection.

Discovery

VMware Smart Assurance discovery is the process of using VMware Smart Assurance object class models to create a representation of the managed topology within the repository, or database, of a Domain Manager. For the IP Manager, data is collected from the managed topology to create instances of the managed network systems, their relationships, their internal components, and their logical connections.

When a system is added to the managed network environment, the IP Manager performs discovery to determine the system's configuration and its relationships to other managed systems and components. When a system is removed from the managed network environment and deleted from the topology, the IP Manager removes the system and all of its components from the modeled topology.

Systems can be added to or deleted from the modeled topology at any time. Adding new systems does not impede the ability of the IP Manager to monitor systems that are already managed.

Map Console

In the Global Console, the Map Console displays graphical representations of physical and logical topologies including display of managed element icons and edges (connections between elements). Network topology maps for NAS devices include:

- Physical Connectivity Map
- IP Network Connectivity Map

NASChassis Element

In the VMware Smart Assurance model for the NAS Extension, the class that represents the Celerra Network Server.

In the Global Console for the NAS Extension, a NASChassis is a representation of a Celerra Network Server, which is the cabinet that contains NAS system elements including a Control Station, Data Movers, and other components that are not monitored by VMware Smart Assurance.

Edge (Map Console display component)

In a map view, an edge represents a connection between two elements. Note that a map edge is not to be confused with a device such as an edge router.

IP Availability Manager

VMware Smart Assurance IP Availability Manager is an example of an VMware Smart Assurance Domain Manager, where a Domain Manager is a service-assurance application associated with a particular type of information-technology domain (networks, systems, applications, and network application services). For IP Availability Manager, the domain is IP network transport connectivity. Each Domain Manager is autonomous in the sense that it maintains its own data models, repository, and problem signatures, and monitors and analyzes the discovered objects in its own domain.

IP Performance Manager

VMware Smart Assurance IP Performance Manager is an VMware Smart Assurance Domain Manager, where a Domain Manager is a service-assurance application associated with a particular type of information-technology domain (networks, systems, applications, application services). For IP Performance Manager, the domain is IP network transport performance. Each Domain Manager is autonomous in the sense that it maintains its own data models, repository, and problem signatures, and monitors and analyzes the discovered objects in its own domain.

IP Server Performance Manager is an enhanced version of IP Performance Manager and is available through a feature license. In addition to the standard capabilities of IP Performance Manager, IP Server Performance Manager can also discover and monitor host disks, host filesystems, and additional information about host processors and memory.

Event

A notable condition that occurs in a network device. The condition may be identified by a message sent from the network element, by a management station polling the element, or by an adapter or third-party software. A Domain Manager uses events, alerts, and other data to perform root-cause and impact analysis.

Icon (Map Console display component)

In a map view for the NAS Extension, an icon represents an entity such as a FileServer or Host element. In the map view, icons are connected by edges.

IP Manager

The IP Manager discovers Layer 2 and Layer 3 connectivity in multivendor, switched, and routed networks. It monitors and analyzes network connectivity and performance, and exports network topology, problems, and events to the VMware Smart Assurance Service Assurance Manager (Global Manager).

Global Manager

The Global Manager integrates the topology, problem, and event information that is imported from the IP Manager component servers and the satellite Domain Managers, and relates the information to services and customers. It also provides cross-domain and end-to-end impact analysis.

The Global Manager displays the topology, problem, and event information through the Global Console.

Notification

In the Notification Log of the Global Console, an VMware Smart Assurance message, issued by a Domain Manager, that indicates a problem or other significant event.

Notification Log

In the Global Console, a listing of current notifications, including comprehensive information about the element involved, impact on other elements, root-cause indication, severity, acknowledgment status, escalation information, and so on.

Repository

The VMware Smart Assurance in-memory, object-oriented data structure that stores information about the managed elements currently in a network, and the relationships between the elements.

Global Console

The Global Console provides a graphical user interface for configuration and administration of Global Manager, Adapter Platforms, Domain Managers, and externally running VMware Smart Assurance adapters, such as the SNMP Trap Adapter.

When the Global Console is attached to the Global Manager, you can browse the topology in various forms, including maps, and to view notifications about problems that impact availability and performance.

Impact Analysis

The process of analyzing the effect that root-cause problems have on other elements in the managed system. Impact analysis is performed by Domain Managers, such as the Availability Manager.

Logical link

A logical link models the path that connects service access points to each other. IP Availability Manager performs connectivity analysis on the following logical link object types:

IP Network - A logical link that allows any two IPv4 addresses that are “ConnectedVia” the IPv4Network to communicate with one another.

IPv6Network — A logical link that allows any two IPv6 addresses that are “ConnectedVia” the IPv6Network to communicate with one another.

VLAN — A logical link that allows any two ports (or MAC addresses) that are “PartOf” the VLAN to communicate with one another.

Partition

A partition is a group of managed devices formed by IP Availability Manager to help in root-cause analysis. IP Availability Manager builds a topology that describes managed devices in your network and their interconnections. If a device is modeled in the topology, there must be a path in the actual network that connects the device to the management station. Ideally, IP Availability Manager can also have representations of all the devices on that path.

However, some devices along the path might not actually be represented in your network topology. For example, those devices might not have SNMPAgents, or IP Availability Manager might not have access to their SNMPAgents, or access to the agents might be blocked by a firewall or some other administrative mechanism. Also, if you do not use autodiscovery, your seed file or information obtained from another Network Management System (NMS) might be incomplete. Or, you might have explicitly unmanaged or deleted some of the devices which are on that path.

IP Availability Manager's root-cause analysis algorithms need to determine the connectivity among managed devices. When the topology is incomplete, IP Availability Manager creates partitions to group all the related managed devices, which helps analysis to proceed.

Within a single partition, any two managed devices are indeed connected by a path. However, objects within distinct partitions have no path between them in the modeled topology, even though they are connected in the actual network.

Because many configurations produce large numbers of apparently isolated objects, usually hosts, IP Availability Manager only creates partitions with two or more devices.

You can assign names to the partitions in your network using the partition.conf file. The VMware Smart Assurance IP Manager User Guide provides more information about naming partitions.

Managed Element

An instance of a physical or logical network element that is monitored by a Domain Manager.

Root-Cause Analysis

The process of analyzing the cause and effect relationships among alarms, alerts, and events in complex systems in order to determine the authentic problem. Root-cause analysis is performed by Domain Managers.

Topology Browser

In the Global Console, the Topology Browser provides a complete inventory (actual and in representational form) of managed elements in the Repository of a Domain Manager, and the interconnections between those elements. The Topology Browser displays the topology in hierarchical form arranged by classes and instances.

Adapter

A program that integrates an VMware Smart Assurance Manager with external components. For the NAS Extension, an HTTPS Adapter queries the Celerra Control Station for internal topology information.

FileServer Element

In the VMware Smart Assurance model for the NAS Extension, the class that represents Data Movers and Filers. In the Global Console for the NAS Extension, a FileServer element is a representation of a Celerra Data Mover or NetApp Filer.

Certification

A label assigned to a network element (system, device, service or application) during the discovery process. The label is based on information that is contained in the oid2type*.conf files. Labels include:

CERTIFIED---Indicates that the element is discovered using standard mib-2 data and proprietary MIBs.

TEMPLATE---Indicates that the OID is recognized by a new entry in oid2type*.conf. Note that oid2type*.conf is a user-modifiable file with no default entries.

Certification Matrix

Listing of devices that VMware Smart Assurance certifies for management by an VMware Smart Assurance Manager.

Codebook

An VMware Smart Assurance data structure that associates problems with the symptoms they cause. This information is used for root-cause and impact analysis using the VMware Smart Assurance patented Codebook Correlation Technology.

Correlation Model

An object-oriented data model that describes the properties of an element, including the characteristic problems of the element, problem symptoms, and symptom propagation patterns to related elements.

Correlator

A general-purpose correlation analysis engine that performs filtering and model processing, along with root-cause and effects analysis. It is driven by three inputs: problem models, topology, and real-time data and events.

Broker

The VMware Smart Assurance Broker facilitates Global Console connections to the component applications in an VMware Smart Assurance system deployment.

When you start a Global Console process, the process connects to the Broker and launches a dialog box through which the user views and selects any component application that is registered with the Broker. After the user selects an application, the Global Console connects to the application and disconnects from the Broker. The Broker can support IPv4 and IPv6 protocols. If the component applications in your deployment are identified by IPv4 and IPv6 addresses, the Broker must reside on a dual-stack host so that it has both an IPv4 address and an IPv6 address. If all component applications in your deployment use the same IP protocol version, the Broker must reside on a host with an IP address that accepts that protocol version.

Containment

The Containment dialog box provides detailed information about a managed element. Containment information varies depending upon the type of element and the type of analysis being performed by the underlying domain. As examples, for a network connection, the containment information includes details about connection points; for a device managed by IP Performance Manager, containment information includes details about the device's memory. The Containment dialog box is available for selected notifications in the Notification Log Console, for selected instances in the Topology Browser Console, and for selected nodes in the Map Console. When an element is managed by multiple Domain Managers, the Containment dialog box includes all the containment information for each domain.

ICMP

Internet Control Message Protocol is tightly integrated with IP. ICMP messages, which are delivered in IP packets, are used for out-of-band messages that are related to network operation. Some of ICMP's functions are to announce network errors, to announce network congestion, to announce timeouts, and to assist in troubleshooting network problems.

ICMP supports an Echo function that sends a packet on a round-trip between two hosts. Ping, which is a common network management tool that is based on the Echo function, transmits a series of packets to measure average round-trip times and to compute loss percentages. Ping is central to the IP Manager ICMP polling.

The IP Manager supports both IPv4 ICMP polling and IPv6 ICMP polling. An IPv4 ICMP poller sends ICMP messages for use with IP version 4. An IPv6 ICMP poller sends ICMP messages for use with IP version 6. All IP Manager products support IPv4 and IPv6 ICMP polling.

ICMP operation

The IP Manager uses ICMP polling for two purposes:

- To determine whether a system is reachable
- To monitor a system after the system has been discovered

SNMP

The Simple Network Management Protocol is used by network management applications for discovery, fault management, and performance management. SNMP is an application-layer protocol that exchanges management information between a network management application, such as the IP Manager, and one or more managed network objects, called agents.

SNMP, itself, does not provide network management. What SNMP does provide is a framework, or infrastructure, on which network management applications can be built.

In this document, unless otherwise noted, SNMP works in conjunction with both IPv4 and IPv6 protocols.

SNMP key components

SNMP key components are the management station, agent, management information base, and the SNMP protocol. Each SNMP agent or SNMP manager maintains a MIB. An agent MIB contains information about the network system managed by the SNMP agent, and about the agent itself. A manager MIB contains network management information that is extracted by the SNMP manager from each agent MIB.

SNMP basic operations

SNMP includes the following basic operations:

- GET
- SET
- NOTIFY
- INFORM

SNMP versions

Currently, there are three versions of the SNMP protocol: v1, v2c, and v3. SNMPv1 and v2c offer rudimentary authentication and authorization schemes, whereas SNMPv3 offers robust authentication and authorization schemes plus data privacy.

SNMPv1, SNMPv2c, and SNMPv3 security

SNMPv1 security consists of a pairing of an SNMP agent with some arbitrary set of SNMP managers to form an SNMP community. Each SNMP community is given a name called the community name or community string for the community. Common community names are *public* (read-only), *private* (read-write), and *trap*. The permissions (read-only, read-write) for a community name indicate the read-write permissions of an SNMP manager when using that community name to access an agent's MIB. Community names can be thought of as passwords to SNMPv1 agents.

Like SNMPv1, SNMPv2c uses the notion of communities to establish trust between SNMP managers and SNMP agents. Unlike SNMPv1, SNMPv2c uses an updated version of the structure of management information (SMI), which extends the MIB object tree, allows several new data types, and makes a number of other changes. SNMPv2c is more efficient than SNMPv1 and has better error-handling capabilities.

SNMPv3 provides integrity, authenticity, data privacy, and access control for SNMP messages exchanged between an SNMP manager and the managed SNMP agents. Unlike the community-based administrative model of SNMPv1 and V2C, SNMPv3 unambiguously identifies the source and destination of each SNMP message. And instead of using community names to establish trust between SNMP managers and SNMP agents, SNMPv3 uses the following security-related services to establish trust:

- Authentication

  The source includes information in each sent message that identifies the source as authentic, and performs the required functions to ensure message integrity. A typical authentication scheme requires that the source and destination parties share the same authentication key.

- Privacy

  Messages are encrypted to achieve privacy. The encryption is done in such a way that only the intended destination can perform the decryption. A typical privacy scheme requires that the source and destination parties share the same privacy key.

- Access control

  Both the source and destination play a part in access control. Each destination may have a distinct access policy for each potential source, which gives an administrator considerable flexibility in setting up an SNMP management system and assigning various levels of authorization to different users.

NASChassis

A chassis is a physical package that encloses other elements; in this case, the enclosed elements are the Celerra Data Movers and Control Station.

NASChassis is instantiated for each Celerra Network Server (NASChassis object) found during the discovery process.

The NASChassis is related to the FileServer and Host objects through the PackagesSystems relationship. The NASChassis is also related to remote Control Station Host objects through the ConnectedTo relationship. The ConnectedTo relationship reflects inter-Celerra device communication.

FileServer

A file server (Celerra Data Mover or NetApp Filer) is responsible for central storage and management of data files so that other computers can access the files. File server configurations can vary greatly, from a networked group of storage arrays that includes many disk drives to a virtual entity on a computer host. In a large Celerra network, however, a file server may be included in a Celerra Network Server that contains a Control Station, or it might be associated with a remote Control Station on the same IP network, or with a Control Station on a network not monitored by VMware Smart Assurance. There is no concept of Control Station with the NetApp Filer, which includes its own management software.

A FileServer object is related to the NASChassis object by the PackagedIn relationship, and to the IP network through the ConnectedVia relationship.

Comparable terms

Comparable Terms lists some comparable terms used in VMware NAS Element Management and VMware ControlCenter.

Table 9-1. Comparable Terms

VMware Smart Assurance term	Comparable VMware NAS element management term	Comparable VMware ControlCenter term
Broker	N/A	Store
Domain Manager, such as the Availability Manager	Celerra Manager	ControlCenter Infrastructure including ControlCenter Server, Repository, and Stores
Edge (a Map Console term)	N/A	Link In the ControlCenter Console, the Relationship View and Topology View display *links* that represents connections between devices.
Global Console Web Console	N/A	ControlCenter Console and Web Console
Map	N/A	In the ControlCenter Console, the Path Details View and Relationship View
Topology Browser	Celerra Manager user interface	In the ControlCenter Console, the Topology View
Repository	Database	Repository

Wildcard patterns

A wildcard pattern is a series of characters that are matched against incoming character strings. You can use these patterns when you define pattern matching criteria.

Matching is done strictly from left to right, one character or basic wildcard pattern at a time. Characters that are not part of match constructs match themselves. The pattern and the incoming string must match completely. For example, the pattern abcd does not match the input abcde or abc. A compound wildcard pattern consists of one or more basic wildcard patterns separated by ampersand (&) or tilde (~) characters. A compound wildcard pattern is matched by attempting to match each of its component basic wildcard patterns against the entire input string. Basic and compound wildcard patterns are defined in the VMware Smart Assurance IP Manager Reference Guide.

If the first character of a compound wildcard pattern is an ampersand (&) or tilde (~) character, the compound is interpreted as if an asterisk (*) appeared at the beginning of the pattern. For example, the pattern ~*[0-9]* matches any string not containing any digits. A trailing instance of an ampersand character (&) can only match the empty string. A trailing instance of a tilde character (~) can be read as “except for the empty string.” Spaces are interpreted as characters and are subject to matching even if they are adjacent to operators like “&”. Special characters for compound wildcard patterns are summarized in the VMware Smart Assurance IP Manager Reference Guide.