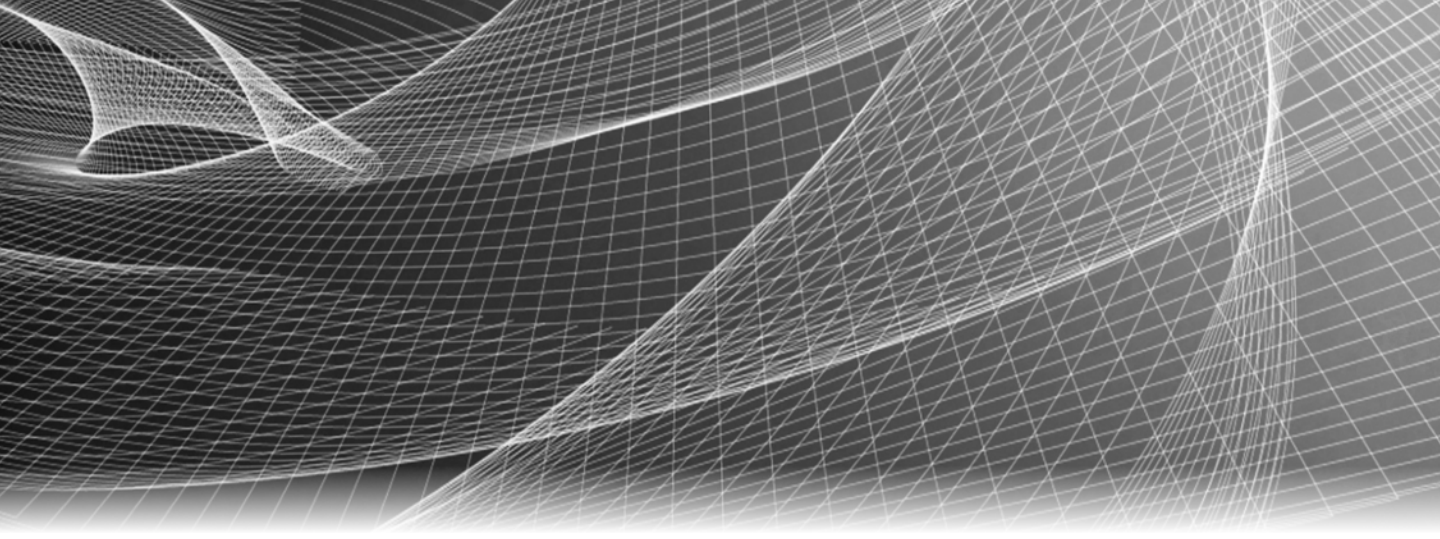

EMC® Ionix™ ControlCenter®

6.1

Integration Packages Product Guide

P/N 300-006-351
REV 09

EMC²

Published August 2012

For the most up-to-date regulatory document for your product line, go to the technical documentation and advisories section on the EMC online support.

CONTENTS

Index

FIGURES

TABLES

PREFACE

As part of an effort to improve its product lines, EMC 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 EMC representative if a product does not function properly or does not function as described in this document.

Note: This document was accurate at publication time. New versions of this document might be released on the EMC Online Support (http://support.emc.com). Check the EMC online support website to ensure that you are using the latest version of this document.

Purpose

This document describes how to configure and use the EMC Ionix ControlCenter Integration Packages.

Audience

This guide is part of the Ionix ControlCenter documentation set and is intended for system administrators who set up, configure, and manage the Ionix ControlCenter Integration Packages.

Conventions used in this document

EMC uses the following conventions for special notices:

⚠WARNING

WARNING indicates a hazardous situation which, if not avoided, could result in death or serious injury.

NOTICE

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

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

IMPORTANT

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

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, and utilities • URLs, pathnames, filenames, directory names, computer names, links, groups, service keys, file systems, and notifications
Bold	Used in running (nonprocedural) text for names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, and man pages Used in procedures for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • What the user specifically selects, clicks, presses, or types
Italic	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis, for example, a new term • Variables
`Courier`	Used for: • System output, such as an error message or script • URLs, complete paths, filenames, prompts, and syntax when shown outside of running text
`Courier bold`	Used for specific user input, such as commands
`Courier italic`	Used in procedures for: • Variables on the command line • User input variables
< >	Angle brackets enclose parameter or variable values supplied by the user
[ ]	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

EMC support, product, and licensing information can be obtained as follows:

Product information. For documentation, release notes, software updates, or information about EMC products, licensing, and service, go to the EMC Online Support (registration required) at:

http://support.EMC.com

Technical support — For technical support, go to EMC Online Support and select SUPPORT BY PRODUCT. On the Support by Product page, you will see several options, including one to create a service request. Note that to open a service request, you must have a valid support agreement. Contact your EMC 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 opinions of this document to:

techpubcomments@emc.com

CHAPTER 1
Introduction

This chapter introduces the Ionix ControlCenter Integration Packages software as part of the Ionix ControlCenter suite of products.

Topics include:

Ionix ControlCenter Integration Packages

The **Ionix ControlCenter Integration Packages** software is one of the Ionix ControlCenter family of products. The software integrates Ionix ControlCenter with industry-leading **Enterprise Management Frameworks (EMFs)**.

Each Integration Package expands the functionality of a particular Enterprise Management Framework (EMF) by providing support for storage resources that are managed by Ionix ControlCenter software.

Active Integration

In *Active Integration*, the EMF (SNMP) management product *actively solicits* SNMP traps from client SNMP agents via SNMP GET Requests.

Active frameworks use a centralized console or server to collect information from client SNMP agents and provide intelligent processing for any data collected. The Ionix ControlCenter Integration Package *provides an application* that resides with the EMF console or server to interpret the SNMP traps issued by Ionix ControlCenter.

Ionix ControlCenter currently qualifies active integration for *BMC Patrol 7* only. Files for integration with other active EMFs are provided with the Ionix ControlCenter Integration Packages software, but they have not yet been qualified by EMC and therefore there may be issues in using them with Ionix ControlCenter.

Passive Integration

In *Passive Integration*, the EMF (SNMP) management product *never solicits* SNMP traps, it waits passively, processing only those SNMP traps sent to it by SNMP agents.

In *Passive Integration*, the EMF (SNMP) management product relies on intelligent EMF agents to send messages to a central EMF console or server. The Ionix ControlCenter Integration Package *provides the files needed to configure the intelligent EMF agents* to interpret SNMP traps issued by Ionix ControlCenter. The intelligent EMF agents then forward the traps as framework-specific messages to the EMF console or server.

Ionix ControlCenter currently qualifies passive integration for *MOM 2005 and SCOM 2007* only. Files for integration with other passive EMFs are provided with the Ionix ControlCenter Integration Packages software, but they have not yet been qualified by EMC and therefore there may be issues in using them with Ionix ControlCenter.

Integration Packages

The Ionix ControlCenter Integration Packages software contains a number of integration packages for several EMFs. *Note that not all of these integration packages have been qualified by EMC for use with ControlCenter.* Therefore, although provided with the Integration Packages software, you may encounter issues when using non-qualified EMFs with Ionix ControlCenter.

Note: Any Enterprise Management Framework (EMF) that supports standard SNMP protocol can receive Ionix ControlCenter alerts. See the *"Integration Packages Applications Support"* Table in the *EMC Ionix ControlCenter 6.1 Support Matrix* for the most up to date list of qualified EMFs. The Support Matrix is available through the EMC Online Support.

Table 1 Ionix ControlCenter 6.1 Integration Packages Summary

Enterprise Management Framework	Integration Type	Download Folder	EMC Qualified
BMC PATROL 7	**Active**	**PATROL**	**YES**
CA Unicenter	Active	Win32	No
HP OpenView NNM / VPO / OVO	Active	OpenView	No
Tivoli NetView	Active	Win32	No
MOM 2005	**Passive**	**MOM**	**YES**
SCOM 2007	**Passive**	**MOM**	**YES**
BMC Patrol Enterprise Manager	Passive	BMC	No
Netcool OMNIbus	Passive	Netcool	No
Tivoli TEC	Passive	Tivoli	No

Components

The components of an integration package vary depending on the EMF.

Files included in integration packages perform various functions including but not limited to:

- Application configuration files that help you configure the Integration Packages application.

- Event configuration files that define trap formats for frameworks.
- Framework-specific files that contain symbols and graphics.

Ionix ControlCenter Software

Ionix ControlCenter is designed to provide centralized control for an entire distributed storage environment. It is a powerful, flexible, unified framework and suite of tools that provides end-to-end management of storage networks, storage devices, and other storage resources.

Ionix ControlCenter provides a centrally managed, single point of control for resources throughout the entire storage environment. From the graphical front end, Ionix ControlCenter lets you manage:

- Connectivity components — Such as Fibre Channel switches and hubs.
- Storage components — Such as EMC Symmetrix® arrays.
- Host components — Such as logical volume managers, file systems, databases, and backup applications.

Every physical and logical element that Ionix ControlCenter manages is known as a *managed object*. From a Console anywhere on the network, Ionix ControlCenter shows a consolidated view of the storage environment. You can monitor the health of, track the status of, report on, and control each managed object.

CHAPTER 2
Features and Functions

This chapter describes the architeture of integration packages as well as Active and Passive integration.

Topics include:

Integration Packages Architecture

The Integration Packages architecture consist of one or more software components that expand the functionality of Enterprise Management Frameworks (EMFs) by providing support for objects managed by Ionix ControlCenter. The Integration Packages software can launch specific object management tools that reside on the same host, or launch a Web browser to access Web-enabled management tools. The Integration Packages provide both active and passive integration.

Components

The Integration Packages consist of software components that vary depending on the framework's software and the integration type. The Integration Package for each framework product provides components including:

- Integration Gateway
- Configuration Files
- Application Files
- Framework-specific Files

Integration Gateway

The three basic functions during active integration are discovery, event processing, and polling. "Active Integration" on page 21 provides more information. For discovery and event processing, the Integration Packages rely on an Integration Gateway that provides an SNMP interface on the front end, including an SNMP Management Information Base (MIB) for discovery information, and SNMP traps for the events. Therefore, the protocol between the Integration Packages and the Integration Gateway is SNMP.

EMC Ionix ControlCenter 6.1 Planning and Installation Guide, Volume 1 describes how to install the Integration Gateway. After installing the Integration Gateway, configure ControlCenter as follows to have SNMP traps sent to the frameworks software:

1. Define trap destination address.
2. Create a management policy to use SNMP.
3. Modify alerts to use the management policy.

Appendix Appendix A, "MIBs and Traps" provides more information about traps used by the Integrations Packages.

The Integration Gateway uses registered UDP port 1273, instead of the standard SNMP port 161. This way the port does not conflict with any other SNMP agent that may reside on the host with the Integration Gateway.

Figure 1 on page 21 shows the architecture of the Integration Packages.

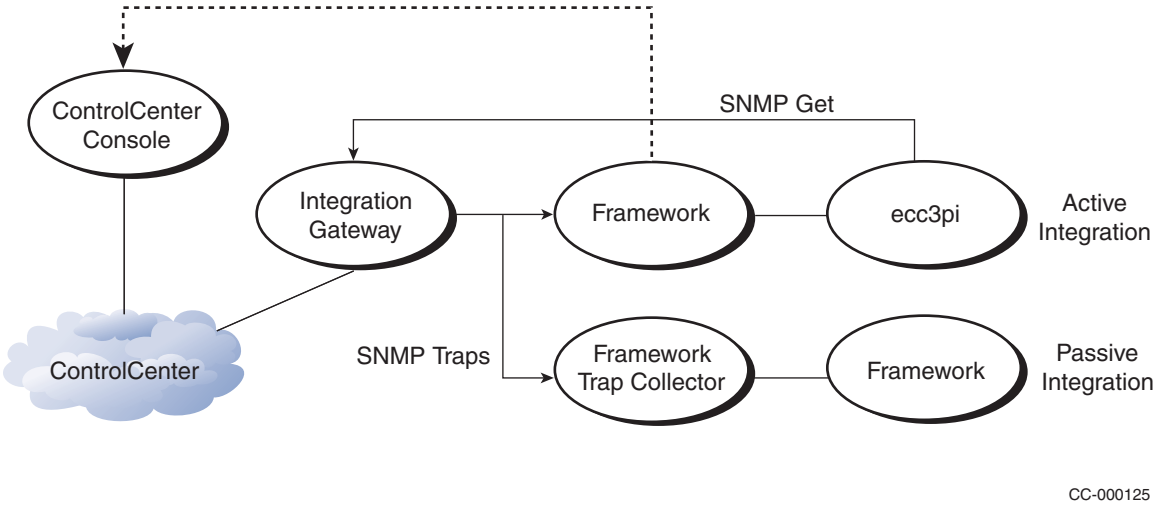

Figure 1 ControlCenter Integration Packages Architecture

Active Integration

Active integration with frameworks software involves three basic functions:

- Discovery — ControlCenter managed objects are added to the framework topology map.
- Event processing — ControlCenter events are received and processed.
- Polling— EMC Views and their managed objects receive periodic status checks.

The Integration Gateway provides an SNMP interface to obtain information about the objects that are managed by ControlCenter.

Integration Packages software resides with the frameworks. This software retrieves and receives information from ControlCenter and uses the framework's API to display this information.

Note: Active integration is qualified for BMC Patrol only. Files for active integration with other frameworks are available, but have not yet been qualified by EMC.

Discovery

In active integration, the frameworks provide basic IP discovery of the physical hosts and other infrastructure components (objects). ControlCenter manages a variety of storage-related network entities including storage systems and other logical elements that cannot be discovered by the frameworks without an active integration.

An active integration queries the Integration Gateway for a set of managed objects and adds an EMC View icon to the framework topology. An EMC View submap contains ControlCenter server icons and icons for various Console groups by default. "Managed Objects" on page 22 provides more information.

With display mode set to Object View, all managed objects appear in the EMC View submap within the framework console. Appendix Appendix B, "Configuration Settings" describes how to change this type of display mode.

In addition, active integration displays group object (icons) for the ControlCenter Console group that may contain one or more objects or other groups. Included are standard static groups that include managed objects for:

- Storage
- Hosts
- Connectivity

User-defined group icons also appear on the Console. Only *shared* groups appear in the top-level view. A folder labeled *All Objects* also appears that contains all managed objects defined in the MIB.

Managed Objects

ControlCenter manages a wide variety of network-related entities including storage systems, switches, hubs, and other logical elements such as databases and file systems. ControlCenter also manages container elements such as groups and fibre zones. All of the entities are stored as managed objects within a relational model in ControlCenter. "Discovery" on page 22 provides more information.

The Integration Gateway exposes only those objects that allow the Integration Packages to provide the functionality mentioned earlier. This means that only objects at a certain level of abstraction are exposed through the SNMP MIB.

For example, if an Integration Gateway exposes a Symmetrix system, but not the subcomponents such as directors or devices, an icon appears for the system. Director and device events are logged against the Symmetrix system and cause its icon to change color. The event contains enough information to identify the specific fault.

The following list of objects can exist within ControlCenter. These objects can be exposed through an SNMP MIB.Symmetrix

- EMC Celerra®
- EMC CLARiiON®
- Third-party storage
- Switches
- Groups
- Hosts
- EMC Ionix ControlCenter Server

Automatic Discovery

When you start the Integration Gateway, it sends a `coldStart` trap to the frameworks. The frameworks then invoke the Integration Package to process the trap.

- If the gateway does not exist in the topology, it is added to the topology and a full discovery of its managed objects is performed.
- If the gateway already exists in the topology, the Integration Packages refreshes the gateway's status and that of its managed objects.

Integration Packages also adds gateway and object icons to the topology in response to status change traps and delete traps. When you receive one of these traps, Integration Packages adds either the gateway or object icon, if it does not already exist. The object status is then set.

Note: Integration Packages adds only the Integration gateway and the object. You need to refresh the gateway or wait until the next polling cycle to refresh *all* objects for the gateway.

Integration Packages has a polling mechanism that periodically refreshes the status of all gateways and their managed objects in the topology. Figure 2 on page 24 shows a sample topology window.

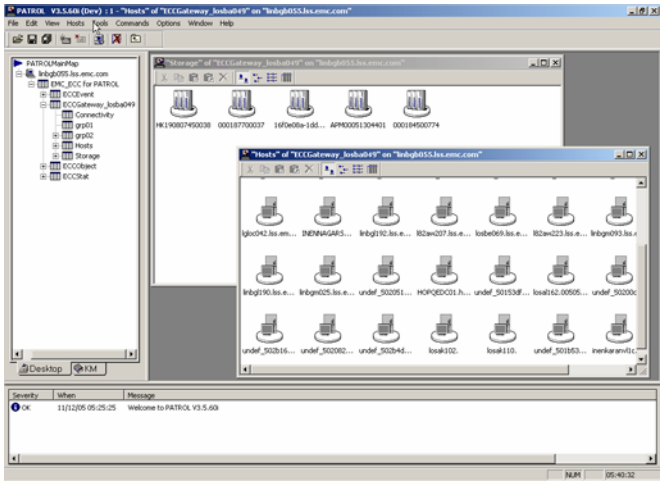

Figure 2 Topology window example

Manual Discovery

You can also manually add a gateway icon to the topology. After you add the icon, you can force a discovery of the gateway's managed objects by selecting the refresh function.

If you do not select refresh for the manually added gateway icon, then the managed objects are discovered automatically during the next refresh cycle.

To manually add an Integration Gateway, enter a label for the icon, and then enter the IP address and port number of the gateway. Refer to the chapters on each Integration Package for exact procedures on how to add a gateway.

Refresh

When you perform a refresh for an Integration Gateway, the Integration Packages:

- Perform a discovery and rebuilds the object view in its submaps.
- Add any new objects.
- Set any unknown or delete objects to unknown status.

If you refresh a single object, the Integration Packages retrieve the current object status from the gateway and updates the object.

Event Processing

The Integration Packages receive events from the Integration Gateway in the form of SNMP traps. These traps are:

- Status change trap
- Object delete trap
- Event trap

The frameworks receive all traps and save them within their event Consoles and logs. ControlCenter Integration Gateway related traps are forwarded to the Integration application for processing.

Status Change Trap

Status change traps indicate a change in the status of managed objects. The Integration application uses these traps to change the status of an object and the object's icon by using an icon color scheme similar to that of the ControlCenter Console. Colors are framework-dependent and may be different for each product. Table 2 on page 25 shows trap status and corresponding object color.

Table 2 Trap Status and Object Icon Color Correspondences

Trap Status	Object Icon Color
Normal	Green
Warning	Yellow
Critical	Red
Unknown	Blue

Object Delete Trap

When ControlCenter no longer manages an object, the Integration Gateway sends a delete trap. This trap indicates that the ControlCenter administrator removed an object from the server. When the Integration application receives a delete trap, it changes the object icon status to unknown. This gives the framework administrator an opportunity to investigate the event, and then manually delete the icon.

Event Trap

The Integration Gateway uses event traps to send messages to the frameworks. The messages can be events, alarms, or alerts associated with a managed object. An example is an environmental alarm for a Symmetrix system.

If an event causes a status change within the managed object, the Integration Gateway sends a corresponding status change trap.

Polling

The Integration Packages provide a polling function that periodically checks the status of the EMC Views and their managed objects. Polling parameters include:

- Frequency
- Retry count
- Refresh rate

You set these values in the file ecc3pi.ini.

Frequency

When the Integration Package starts, it refreshes all EMC Views in the topology, and then pings each Integration Gateway on every poll cycle. If an Integration Gateway continues to respond, its status remains the same. If an Integration Gateway responds but did not respond to the previous ping, or if you add a new EMC View after the previous poll, then the EMC View is refreshed as described in "Discovery" on page 22.

Retry Count

If an Integration Gateway does not respond to a poll, the Integration Package sets all of its managed objects to unknown status and sets the EMC View status to critical. The Integration Package assumes that the Integration Gateway is temporarily unavailable, and will repoll that Integration Gateway every minute up to a specified number of retries. (The default retry count is four.) The Integration Package returns to its normal polling rate after it performs the specified number of retries or if the Integration Gateway responds during one of the polling cycles.

Refresh Rate

The Integration Package periodically refreshes all EMC Views based on a specified refresh rate. Normally, traps cause updates to an EMC View and its managed objects. This refresh cycle provides a way for you to reestablish a status baseline in case traps get lost.

The refresh rate is normalized to start at midnight. The default refresh rate is 480 minutes (every 8 hours). This causes a refresh at 12:00 A.M., 8:00 A.M., and 4:00 P.M. When you start the Integration application, it performs the next refresh at the normalized time. For example, if you start the Integration Package at 5:00 A.M., with a 480 minute refresh rate, the next refresh occurs at 8:00 A.M. The refresh cycle is reset at midnight.

User Interface

The Integration Packages provide the same functionality across all qualified frameworks. However, differences between the frameworks and their APIs necessitate slight changes in the actual implementation.

Passive Integration

For system management products that rely on intelligent agents to forward ControlCenter messages and events to the frameworks Console, Integration Packages provide configuration files and settings that define traps and formats.

Integration Packages provide the necessary interface to display ControlCenter events on the framework Console. The Integration Gateway sends traps to the framework agent, which interprets and formats the event based on the Integration Packages files, and sends a message to the framework Console. You can then use these events to trigger other actions such as:

- Trouble ticketing
- Paging
- Email

CHAPTER 3
Integrating with BMC PATROL 7

This chapter describes how to install, configure, and operate the Integration Packages software for BMC PATROL 7.

Topics include:

Overview

BMC PATROL 7 is an agent-based monitoring technology that is widely used in multiple industry sectors. The BMC PATROL 7 product includes a central console which displays information from BMC PATROL 7 agents. BMC PATROL 7 agents employ Knowledge Modules (KMs) which are essentially sub-agents that monitor specific objects, devices, or applications. KMs maintain a namespace of information for each object, device, or application they are responsible for. This information is then processed and presented on the BMC PATROL 7 central console.

Integration Package for BMC PATROL 7

The Ionix ControlCenter integration package for BMC PATROL 7 includes a Knowledge Module (KM). The ControlCenter KM for BMC PATROL 7 integrates ControlCenter managed objects and generated events into BMC PATROL 7 to provide a unified approach for enterprise system management. Once data for an object, device, or application is inserted into it's KM namespace, it can be made available to other BMC products that interface with PATROL 7.

The ControlCenter KM for BMC PATROL 7 enables you to:

- Manage valuable EMC storage components as part of the global unified enterprise management suite.
- Use the BMC PATROL 7 software look and feel in combination with ControlCenter functionality.

Components

The ControlCenter KM for PATROL 7 incorporates ControlCenter objects and events into PATROL 7. ControlCenter contains an Integration Gateway that exposes ControlCenter managed objects using an SNMP mib, and creates SNMP traps in response to ControlCenter events and alerts.

The ControlCenter KM for PATROL 7 gets object information and events from the gateway and maintains that data within PATROL 7.

Distributed Files

The Integration Package includes the following files for PATROL 7:

- EMC_ECC_KM.tar

- ControlCenter KM — Process ControlCenter information into PATROL 7.
- ecc3pi — Trap receiver daemon that passes ControlCenter information to the KM.
- parser — Parser that translates information received from ecc3pi.
- Various icons and miscellaneous files for PATROL 7.

◆ README.txt — Instructions to configure your system for the ControlCenter integration.

Installing and Configuring for BMC PATROL 7

You must have the Ionix ControlCenter 6.1 Integration Packages installed on your system as well as BMC PATROL 7. The installation of these products is not discussed in this manual. The following documents provide more information:

◆ *EMC Ionix ControlCenter 6.1 Planning and Installation Guide, Volume 1*

◆ *BMC PATROL 7 Installation and Configuration Guide*

Configuring for BMC PATROL 7

Figure 3 on page 31 illustrates the configuration of the ControlCenter KM for PATROL 7. The ControlCenter KM, including the `ecc3pi` adapter, are packaged as a standard PATROL 7 KM. This KM can be installed directly to an agent or through the PATROL 7 distribution server.

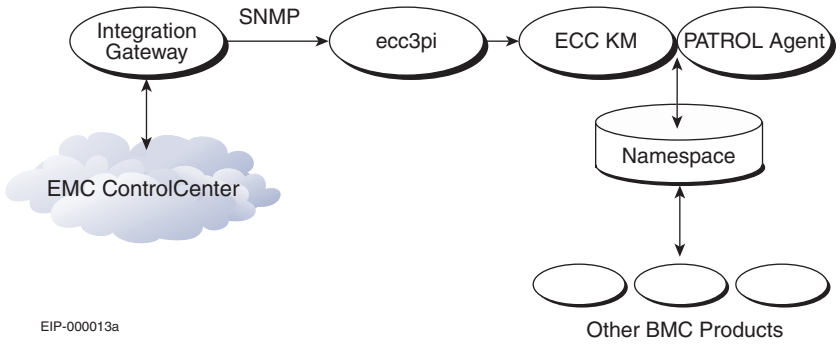

Figure 3 BMC PATROL 7 Integration Configuration

Configuring the Integration Gateway

The ControlCenter KM receives traps on a port other than the standard SNMP port 162 so as not to conflict with other SNMP trap daemons. You must configure the ControlCenter Integration Gateway to send traps to this port by editing the Integration Gateway `.ini` file.

Configuring the CNG.ini/CSG.ini

1. On Gateway Agent Host, modify or enter the following line in exec\CNG610\CNG.ini on Windows or %ECC_INSTALL_ROOT%/exec/CSG610/CSG.ini on Solaris.

2. `udp_port_number=1273`

   1273 by default, change if the port is being used by other process but it should be same as port *‹pppp›* mentioned in step 1 on page 32 .

3. Modify or enter the following line:

   `trap_client_registration=<xxx.xxx.xxx.xxx>,<nnnn>,10,ACTIVE`

   where:

   - `<xxx.xxx.xxx.xxx>` is the IP address of the BMC PATROL 7 agent host.
   - *‹nnnn›* is SNMP port, which should be same as snmp_port mentioned in step 2 on page 33 .

   The default SNMP port is 162.

   For example:

   `trap_client_registration=172.23.154.166,1333,10,ACTIVE`

4. Save the changes and restart the Gateway agent.

Configuring the ecc3pi.ini File

The default ecc3pi.ini file provided with the ControlCenter KM for PATROL 7 must be modified as follows:

1. Modify or enter the following line in Program Files\BMC Software\Patrol3\bin\ecc3pi.ini file:

   `poll_gateway=<xxx.xxx.xxx.xxx>:<pppp>`

   Where:

- <xxx.xxx.xxx.xxx> is the IP address of the Gateway agent host.
- <pppp> is UDP port, which should be same as udp_port_number in Gateway configuration (CNG/CSG.ini) file as mentioned in step 2 on page 32 .

For example:

```
poll_gateway=10.10.10.10:1444
```

2. Modify or enter the following line in Program Files\BMC Software\Patrol3\bin\ecc3pi.ini file:

```
get_community=public
snmp_community=public
snmp_timeout=500
snmp_retries=3
snmp_port=<nnnn>
```

Where:

- <nnnn> is SNMP port on Gateway Host, which should be same as udp_port_number in Gateway configuration (CNG/CSG.ini) file as mentioned in step 3 on page 32 .

For example:

```
snmp_port=1333
```

3. Stop the **PatrolAgent** service from Services.
4. Ensure that the **ecc3pi** process is not running.
5. Restart the **PatrolAgent** service from Services.

Installing the ControlCenter KM for BMC PATROL 7

This section provides procedures for installing the integration package either locally on a host or through a distribution server.

Note: The ControlCenter KM for PATROL 7 is in the form of a `.tar` file.

Direct Install Procedures

To install the ControlCenter KM for PATROL 7 on a local host:

1. Before you begin this installation, uninstall any previous versions of the PATROL 7 Integration Package, and then check that the PATROL 7 agent is operating properly.
2. Download the BMC Software Common Install engine (version 7.4.40) that is required for your platform from the BMC software website or FTP site. Each install engine is contained in an electronic product download (EPD) that consist of a self-extracting executable for Windows systems or a TAR (`.tar`) file for UNIX systems.

   Note: When uncompressing the `EMC_ECC_KM.tar` file by using WinZip version 9.0. The WinZip version 7.0 is unable to uncompress the `EMC_ECC_KM.tar` file correctly.

3. Extract the downloaded install engine to a local directory on the system where the installation is performed. The contents of the extracted file should resemble the following directory structure:

   ```
   ./bmc_products
      /Index
      /Install
      /Products
   ```

4. Move the `EMC_ECC_KM.tar` file to the directory in step 3 and extract it. The files and directories extracted from this `.tar` file overlays the install engine's `Index` and `Products` directories.
5. Start the installation by running the appropriate executable:
   - Microsoft Windows operating systems: `setup.exe`
   - UNIX operating systems: `setup.sh`

   Note: The Common Install engine requires a web browser to render its installation wizard. If your computer does not have a supported web browser installed, you need to start the installation in *Server Only* mode and then connect to the install engine's web server from a machine that does have a supported web browser. Start the installation in *Server Only* mode with `setup.exe -serveronly` or `setup.sh -serveronly` and follow the on-screen instructions.

6. Follow the instructions presented in the web browser-based install wizard, as shown in Tablex:
   a. From the Welcome to the Installation Utility screen: Click **Next**.

b. From the Review License Agreement screen: Select **Accept** and click **Next**.

c. From the Selection Installation option screen: Select **Install products on this computer now** and click **Next**.

d. From the Select Type of Installation screen: Select **Typical** and click **Next**.

e. From the Specify Installation Directory screen: Enter or browse to the BMC Software Product Installation directory where PATROL 7 is installed.

f. Select one or more roles for the computer on which you are planning to install the ControlCenter KM and click **Next**.

Note: A given system role may not install all of the ControlCenter KM files. Table 3 on page 36 and Table 4 on page 38 provide more information.

- If you previously installed the PATROL 7 Classic Console, select the **Console Systems** role.
- If you previously installed the PATROL 7 agent, select the **Managed System** role.
- If you previously installed the PATROL 7 Console Server, select the **Common Services** role.

g. Select the **Products and Components to Install** screen: Expand the Ionix ControlCenter folder, select **PATROL Knowledge Module for Ionix ControlCenter product**, and click **Next**.

h. From the Review Selections and Install screen: Click **Start install**.

Note: For classic PATROL 7 the ControlCenter KM files are dropped relative to the `$PATROL_HOME`. Therefore, the `$PATROL_HOME` location must be known before the files are extracted.

---

7. Restart the Patrol 7 Agent once the ControlCenter KM files are installed.

Direct Installation Examples

This section provides examples of the extracted ControlCenter KM files for both UNIX and Windows installations.

UNIX Directory Structure

Example directories include:

$PATROL_HOME = /opt/bmc/Patrol7/Solaris28-sun4

$PATROL_ROOT = /opt/bmc/Patrol7

Table 3 on page 36 lists examples of extracted ControlCenter KM files for UNIX.

Windows Directory Structure

Example directories include:

%PATROL_ROOT% = C:\Program Files\BMC Software\Patrol7

```
%PATROL_ROOT% = C:\Program Files\BMC Software\Patrol7
```

Table 4 on page 38 lists examples of extracted ControlCenter KM files for Windows.

Table 3 Examples of Extracted ControlCenter KM Files—for UNIX (page 1 of 2)

Directories and Files	System Role	Notes
$PATROL_HOME/bin/ ecc3pi eccprx	Managed System	Installs on PATROL 7 Agent hosts.
$PATROL_HOME/lib/knowledge/ EMC_ECC.kml EMC_ECC.km EMC_ECC_OBJ.km EMC_ECC_EVENT.km EMC_ECC_STAT.km EMC_ECC_VIEW.km EMC_ECC_GROUP.km EMC_ECC_HOST.km EMC_ECC_ENTERPRISE_DISK.km EMC_ECC_CONNECTIVITY.km	Managed System, Console Systems	Installs on PATROL 7 Agent or PATROL 7 Classic Console hosts.

Table 3 Examples of Extracted ControlCenter KM Files—for UNIX (page 2 of 2)

Directories and Files	System Role	Notes
$PATROL_HOME/lib/psl/ emc_ecc_cmn.lib emc_ecc_usr.lib	Managed System, Console Systems	Installs on PATROL 7 Agent or PATROL 7 Classic Console hosts.
$PATROL_HOME/lib/images/ emc_ecc_entdisk_ok.msk emc_ecc_entdisk_ok.xpm emc_ecc_group_ok.msk emc_ecc_group_ok.xpm emc_ecc_host_ok.msk emc_ecc_host_ok.xpm emc_ecc_ok.msk emc_ecc_ok.xpm emc_ecc_stat_ok.msk emc_ecc_stat_ok.xpm emc_ecc_switch_ok.msk emc_ecc_switch_ok.xpm emc_ecc_view_ok.msk emc_ecc_view_ok.xpm	Console Systems	Installs on PATROL 7 Classic Console hosts.
$PATROL_ROOT/lib/knowledge/EMC_ECC_1_2_00/ package.mof resource.mk4	Common Services	Installs on PATROL 7 Console Server hosts.

Table 4 Examples of Extracted ControlCenter KM Files—for Windows (page 1 of 2)

Directories and Files	System Role	Notes
%PATROL_HOME%\Windows_NT-x86\bin\ ecc3pi.exe eccprx.exe eccpsk.exe	Managed System	Installs on PATROL 7 Agent hosts.
%PATROL_HOME%\lib\knowledge\ EMC_ECC.kml EMC_ECC.km EMC_ECC_OBJ.km EMC_ECC_EVENT.km EMC_ECC_STAT.km EMC_ECC_VIEW.km EMC_ECC_GROUP.km EMC_ECC_HOST.km EMC_ECC_ENTERPRISE_DISK.km EMC_ECC_CONNECTIVITY.km	Managed System, Console Systems	Installs on PATROL 7 Agent or PATROL 7 Classic Console hosts.
%PATROL_HOME%\lib\psl\ emc_ecc_cmn.lib emc_ecc_usr.lib	Managed System, Console Systems	Installs on PATROL 7 Agent or PATROL 7 Classic Console hosts.

Table 4 Examples of Extracted ControlCenter KM Files—for Windows (page 2 of 2)

Directories and Files	System Role	Notes
%PATROL_HOME%\lib\images\ emc_ecc_entdisk_ok.bmk emc_ecc_entdisk_ok.bmp emc_ecc_group_ok.bmk emc_ecc_group_ok.bmp emc_ecc_host_ok.bmk emc_ecc_host_ok.bmp emc_ecc_ok.bmk emc_ecc_ok.bmp emc_ecc_stat_ok.bmk emc_ecc_stat_ok.bmp emc_ecc_switch_ok.bmk emc_ecc_switch_ok.bmp emc_ecc_view_ok.bmk emc_ecc_view_ok.bmp	Console Systems	Installs on PATROL 7 Classic Console hosts.
%PATROL_HOME%\Windows_NT-x86\bin\ ecc3pi.exe eccprx.exe eccpsk.exe	Managed System	Installs on PATROL 7 Agent hosts.
%PATROL_ROOT%\lib\knowledge\EMC_ECC_1_2_00\ package.mof resource.mk4	Common Services	Installs on PATROL 7 Console Server hosts.

Distribution Server Installation

Use the following procedure to install the Integration Package by using a distribution server.

Preconditions for KM Deployment

The BMC documentation provides further details. Ensure that the following preconditions are met before beginning KM distribution:

- BMC Distribution Manager was installed.
- BMC Distribution Manager was configured for deployment to systems where KM is installed:

- Connection accounts and privileged accounts were created for the systems targeted for deployment.
- Distribution profiles were created for each system type (Windows and UNIX).
- System groups were created that contain the systems and hosts where the knowledge module is deployed.

◆ BMC Distribution Client was deployed to systems.

◆ Any of the following are installed on the systems:

- PATROL 3.5 Agent, PATROL 3.5 Classic Console
- PATROL 7 Central Operator for Windows
- PATROL 7 RT Server
- PATROL 7 Console Server

KM Distribution Instructions

To distribute KM:

1. Extract the `EMC_ECC_KM.tar` file to a local directory on the Distribution Manager (for example: `C:\ecc_km`).

   Note: When uncompressing `EMC_ECC_KM.tar` use WinZip version 9.0. WinZip version 7.0 is unable to uncompress the `EMC_ECC_KM.tar` file correctly.

2. Import the ControlCenter KM Components into the Distribution Manager Repository with the following steps.

   a. Select the **Components** tab on the main Distribution Manager screen.

   b. Select the **Components** link and click **Import**. The Import components dialog box displays.

   c. Enter the full path to the `bmc_products` directory created when you extracted the `.tar` file (for example: `C:\ecc_km\bmc_products`). Or, click **Browse** and select the directory. Then, click **Next**. A tree of components that can be imported displays.

   d. Expand the Ionix ControlCenter folder and select the checkbox for the **PATROL Knowledge Module for Ionix ControlCenter (1.2.00)** component. Then, click **OK**.

e. Click **Import** on the following screen to perform the import of the selected component. When the import completes, click **Close**.

3. Set up a collection that contains the ControlCenter KM for each system type as follows:

   a. Select the **Collections** tab on the main Distribution Manager screen.

   b. Click the **Selections** link and click **Add**. The Add collection dialog box displays.

   c. Enter a Collection name (for example, **`ControlCenter KM`**) and click **Add**. A new link appears under the Collections list with this name.

4. Configure the newly added collection as follows:

   a. Click the link with the collection name you specified. This presents a collection screen in the right panel.

   b. Select the **Components** tab on the collection screen. This displays the list of components selected for this collection. This list is empty. Click **Add**. This presents a tree of all components that have been imported into the Distribution Manager.

   c. On the component tree expand the Ionix ControlCenter folder, check the **PATROL Knowledge Module for Ionix ControlCenter (1.2.00)**, and then click **OK**. This adds the component to the collection's component list.

   d. Select the **Configurations** tab on the collection screen to display the collection Configurations screen.

   e. Click **Add** to open the Add configuration screen.

   f. Enter a configuration name and description then click **Next**.

   g. Click **Next** on the following Instructions screen to begin answering questions that are specific to the install of the ControlCenter KM.

      Note: There are no uninstall questions for this knowledge module.

   h. Enter the PATROL 3.x Product Directory and click **Next**. For example, if you installed the PATROL 3.5 Agent to the `C:\Program Files\BMC Software\Patrol3` directory, enter **`Patrol3`**.

5. Schedule the ControlCenter KM distribution as follows:

   a. Select the **Distribution** tab on the Distribution Manager.

b. Click **Add**. The Add Distribution Set dialog box displays.

   Enter a **Distribution set** name (for example, Ionix ControlCenter Distribution) and click **Add**. This causes the newly added distribution set properties to display in the right panel.

c. Select the **Distribution's Items** tab in the right panel.

d. Click **Add Item** to display the Add distribution Item dialog box.

e. Select the name of the **collection** you created from the Collection drop-down list box.

f. Select the name of the **configuration** you created from the Configuration drop-down list box.

g. Select the name of the **System group** where you want to deploy your collection or configuration.

   Click **Add** to finish creating the distribution set.

h. Select the **Distribution's Schedule** tab in the right panel to display the Schedule screen.

i. Click **Add Schedule**.

j. Select either **Distribute immediately** or **Schedule distribution** and enter start date and start time information for when to distribute.

k. Click **Install**.

l. Click **Add** to complete the scheduling of the distribution.

6. Review deployment status and history by selecting the Distribution Manager's **Reports** tab.

   a. Select the **Active** tab to view the status of scheduled deployments currently under way.

   b. Select a distribution name from the **Distribution** drop-down list box to view that active distribution's status. The status of the distribution for each host is displayed.

   c. Select the **History** tab to view the status of distributions that have completed.

   d. Click **installation.log** links to view the installation log file for specific hosts.

Uninstalling the Integration Package for BMC PATROL 7

The uninstall can be handled by the BMC Common Install engine or Distribution Server based on how the ControlCenter KM was installed.

Alternatively, the files listed in the direct installation example may be manually removed or an uninstall script may be written to remove the ControlCenter KM package.

Using the Ionix ControlCenter KM for BMC PATROL 7

The ControlCenter KM for BMC PATROL 7 enables you to view EMC objects and events from a PATROL 7 console. You can then define appropriate actions and responses within PATROL 7 for these events. ControlCenter information can also be forwarded from PATROL 7 to other BMC products.

ControlCenter Objects in BMC PATROL 7

ControlCenter objects are instantiated within PATROL 7 and displayed in the PATROL 7 console.

The objects include:

```
EMC_ECC
EMC_ECC_VIEW
EMC_ECC_GROUP
EMC_ECC_OBJECT
```

The `EMC_ECC` top-level object acts as a container for the discovered Ionix ControlCenter objects. The `EMC_ECC_VIEW` represents each gateway object. Contained within any gateway object are the discovered Ionix ControlCenter groups (both preset and any user-defined). Each Ionix ControlCenter group contains the top-level objects which are designated to that group.

Figure 4 on page 44 shows the ControlCenter KM configured to collect data and event objects from two Gateways (`losae039` and `losav139`). The ControlCenter default groups of Hosts, Storage, Connectivity and also Connectivity, as well as several use-defined groups, can be seen under the `losav139` gateway object.

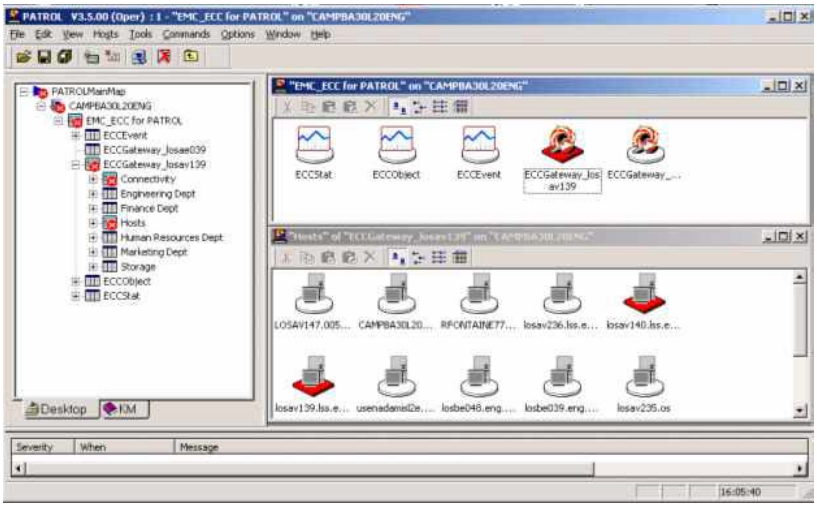

Figure 4 Configuration Collecting Data and Events from Two Gateways

Ionix ControlCenter Events in BMC PATROL 7

As ControlCenter alerts and events are received by the ControlCenter KM, they are associated with the respective object. Events can be viewed through PATROL 7 by selecting the event icon. Figure 5 on page 45 shows an example:.

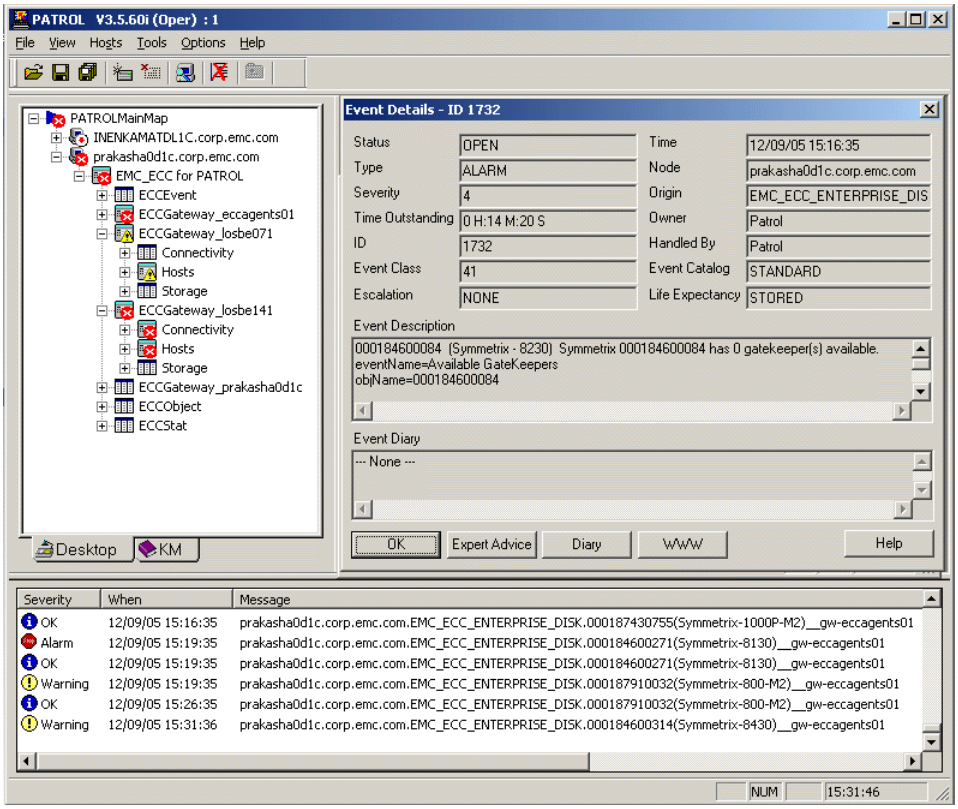

Figure 5 BMC PATROL 7 Event Details

The following naming conventions are used:

```
<hostname>.EMC_ECC_HOST.<OSHostName>(Product or OS type)_gw_<ECC GatewayName>
<hostname>.EMC_ECC_ENTERPRISE_DISK.<arraySerialNumber>(Product Type)_gw_<ECC
GatewayName>
<hostname>.EMC_ECC_CONNECTIVITY.<switchSerialNumber>(Product Type)_gw_<ECC
GatewayName>
```

Where `<hostname>` is the host on which the PATROL 7 agent is running.

Figure 6 on page 46 provides an example of the PATROL 7 Event Manager window that details the ControlCenter alerts.

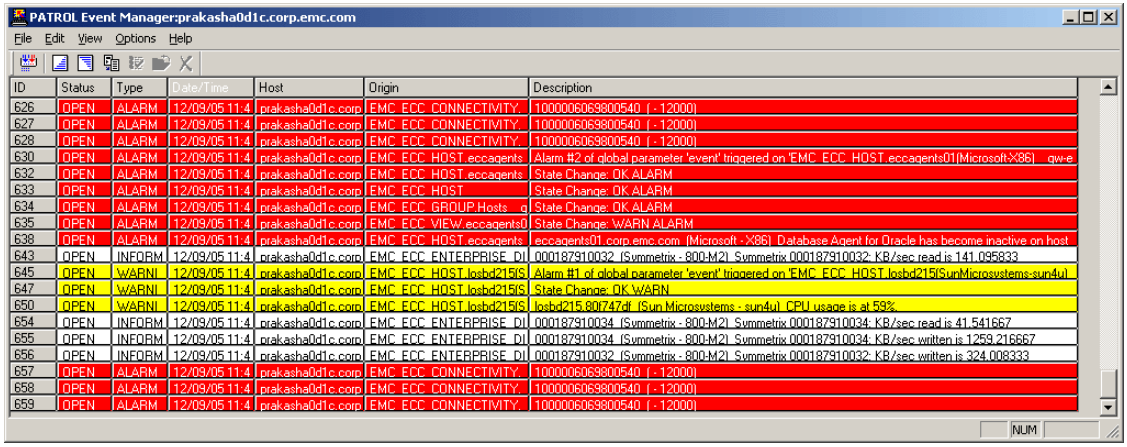

Figure 6 BMC PATROL 7 Event Manager

Define Gateways

The Gateway Setup configuration dialog box allows the user to define Ionix ControlCenter Gateways, which are used to collect data and receive events. Figure 7 on page 46 provides an example.

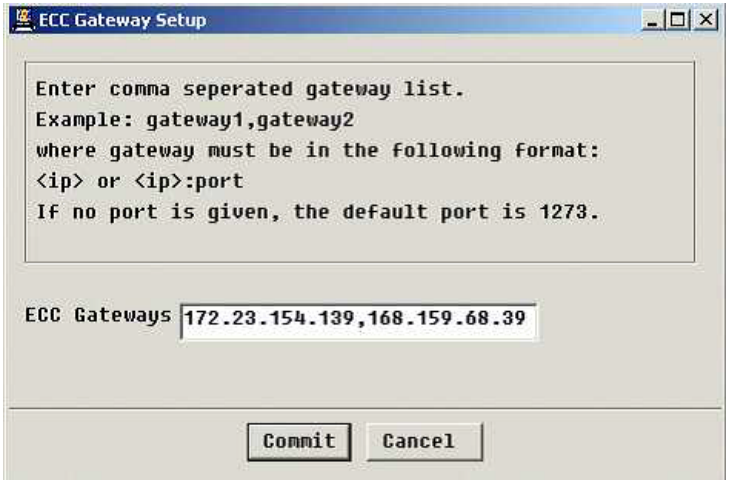

Figure 7 Ionix ControlCenter Gateway Setup

Filters

The user can define certain filters which limit the types of ControlCenter objects that get exposed to PATROL 7. Only those specified objects and their associated alerts appear in PATROL 7. You can specify to filter objects by Gateway, Group(s), or individual object(s).

The **Gateway Include List** configuration dialog box allows you to define Ionix ControlCenter Gateways , which are exposed to PATROL 7 as shown in Figure 8 on page 47.

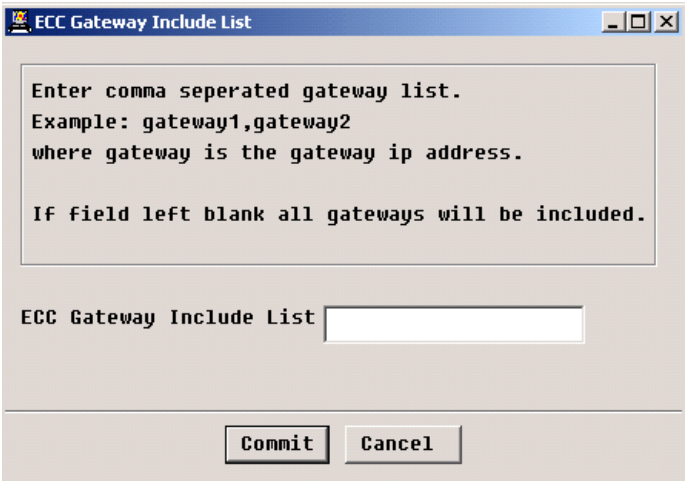

Figure 8 Ionix ControlCenter Gateway Include List

The **Group Include List** configuration dialog box allows you to define which Ionix ControlCenter Groups are exposed to PATROL 7, as shown in Figure 9 on page 48.

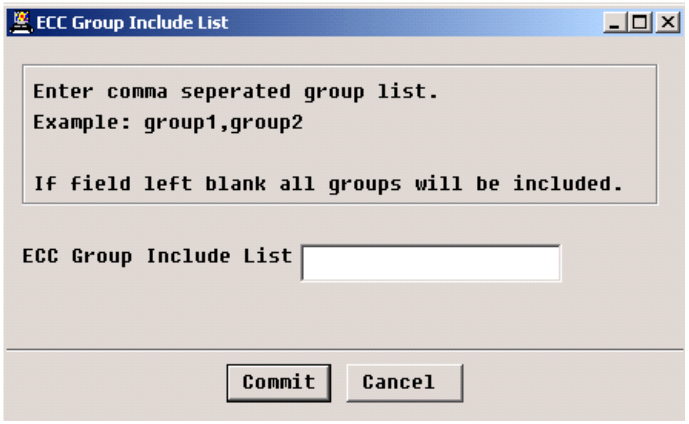

Figure 9 Ionix ControlCenter Group Include List

The **Object Include List** configuration dialog box allows the user to define which Ionix ControlCenter Objects are exposed to PATROL 7. Figure 10 on page 48 provides an example.

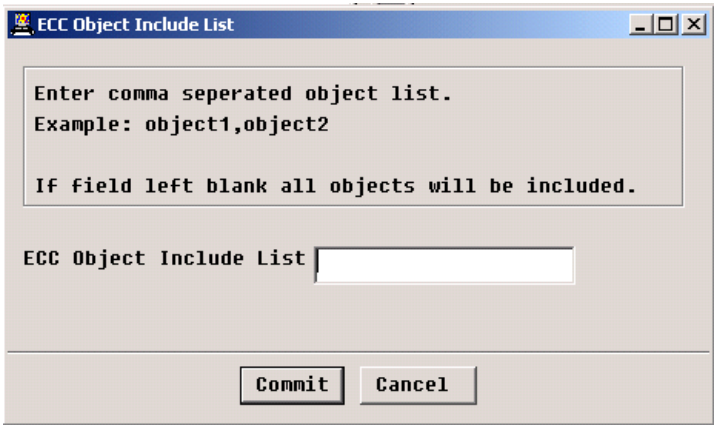

Figure 10 Ionix ControlCenter Object Include List

CHAPTER 4
Integrating with MOM and SCOM

This chapter describes how to install, configure, and operate Integration Packages with Microsoft Operations Manager 2005 (MOM 2005) and System Center Operations Manager 2007 (SCOM 2007).

Topics include:

Overview

The Integration Package for **Microsoft Operations Manager 2005 (MOM 2005)** and **System Center Operations Manager 2007 (SCOM 2007)** integrates ControlCenter-generated events into MOM and SCOM software to provide a unified approach for enterprise system management.

MOM 2005 and SCOM 2007 are essentially the same Enterprise Management Framework product, only the product name was changed between releases from MOM 2005 to SCOM 2007 and the format of the management pack in the integration packages folder is different for each:

- MOM 2005 uses ecc3pi_mom2005.**akm**
- SCOM 2007 uses ecc3pi_scom2007.**xml**

For the remainder of this chapter we will refer to MOM 2005 and SCOM 2007 together as MOM 2005 / SCOM 2007.

Both MOM 2005 and SCOM 2007 represent passive integration. Both are comprehensive server-monitoring solutions that provide a consolidated view of enterprise-wide events and status information. They collect events and messages from many different data sources and present a single view of the current state of managed systems.

Integration Package for MOM and SCOM

The ControlCenter Integration Package for MOM 2005 / SCOM 2007 enables you (as an Enterprise manager) to:

- Manage valuable EMC storage components as part of the global unified enterprise management suite.
- Use MOM 2005 / SCOM 2007 software look and feel in combination with ControlCenter functionality.

Components

The Integration Package for MOM 2005 / SCOM 2007 incorporates ControlCenter events into the MOM 2005 / SCOM 2007 Console. ControlCenter contains an Integration Gateway that creates SNMP traps in response to events and alerts generated by ControlCenter. The Integration Gateway can also be configured to write events to the Windows Event Log.

Distributed Files

Integration Package provides three files for MOM 2005 / SCOM 2007 integration:

- `ecc3pi_mom2005.akm` — This file contains the Ionix ControlCenter Management Pack for MOM 2005. You need to import this file through the MOM 2005 Administrators Console.
- `ecc3pi_scom2007.xml` — This file contains the Ionix ControlCenter Management Pack for SCOM 2007. You need to import this file through the SCOM 2007 Administrators Console.
- `README.txt` — This file contains instructions to configure your system for the ControlCenter integration.

MOM 2005 and SCOM 2007 Components

The integration creates the following components within MOM 2005 / SCOM 2007:

- Public Views — EMC Alarms
- Computer Groups — EMC View
- Processing Rule Groups — EMC Rules

The rule group, EMC Rules, contains four event entries, one for each possible event (trap) written to the event log by the Integration Gateway. This rule group is associated with the EMC View Computer group.

Installing and Configuring for MOM 2005 and SCOM 2007

You must have ControlCenter Integration Packages installed on your system as well as MOM 2005 / SCOM 2007. The ControlCenter Integration Gateway (along with a MOM 2005 / SCOM 2007 agent) can be installed on a Windows host, or it can be installed on the MOM 2005 / SCOM 2007 Administrators Console host.

The installation of these products is not discussed in this manual. The following documents provide more information:

- *EMC Ionix ControlCenter 6.1 Planning and Installation Guide, Volume 1*
- *Microsoft Operations Manager 2005 Installation Guide*
- *System Center Operations Manager 2007 Instalation Guide*

Configuring for MOM 2005 and SCOM 2007

The ControlCenter Management Pack for MOM 2005 / SCOM 2007 contains the rules that are used to recognize ControlCenter events. These rules are processed by either the MOM 2005 / SCOM 2007 consolidator or a MOM 2005 / SCOM 2007 agent, depending on where you install the Integration Gateway.

Configuring the Integration Gateway

You must configure the ControlCenter Integration Gateway to write events to the Windows Event Log. This is done by editing the Integration Gateway `INI` file (`CNG.ini`). To edit the gateway ini file:

1. Add the following as a new entry under the existing `[EccGatewayConfig]` section in the `CNG.ini` file:

   [EccGatewayConfig]
   Nt_EventLog_Key = EMC_Alarms

2. Restart the Integration Gateway.

This adds the following registry key entry on the host, which MOM looks for to determine if the host is running the Gateway:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLo
  g\Application\EMC_Alarms
```

Installing the Integration Package for MOM and SCOM

The Integration Package for MOM 2005 / SCOM 2007 is in the form of a MOM 2005 / SCOM 2007 Management Pack. Before you begin this installation, uninstall any previous versions of the MOM 2005 / SCOM 2007 Integration Package, and then check that MOM 2005 / SCOM 2007 is operating properly.

The Integration Packages is a client application installed from the Ionix ControlCenter Server. *EMC Ionix ControlCenter 6.1 Planning and Installation Guide, Volume 1* includes information about installing Integration Packages.

The Integration Package for MOM 2005 / SCOM 2007 software is located in the MOM directory.

To install the Integration Package for MOM 2005 / SCOM 2007:

1. Start the MOM 2005 / SCOM 2007 Administrator Console.

2. Right-click **Rules**, **Processing Rule Groups**, and then select **Import Management Pack**. The Import Management dialog box appears.
3. Ensure that the option "Replace existing rule groups with those of imported Management Pack." is selected.
4. Select `ecc3pi_mom2005.akm or ecc3pi_scom2007.xml` and then click **Import**.
5. Rescan either all hosts, or only the specific host where you installed the Integration Gateway.
6. Restart MOM 2005 / SCOM 2007 Console.

Uninstalling the Integration Package for MOM and SCOM

MOM 2005 / SCOM 2007 does not currently facilitate removal of an installed Management Pack. Therefore, the individual elements of the Integration Packages must be deleted manually. To uninstall the Integration Package for MOM 2005 / SCOM 2007:

- Start the MOM 2005 / SCOM 2007 Console.
- Manually delete the EMC integration components:
  - Public Views — EMC Alarms
  - Computer Groups — EMC View
  - Processing Rule Groups — EMC Rules

Using the Integration Package for MOM 2005 and SCOM 2007

The Integration Package for MOM 2005 / SCOM 2007 enables you to view EMC events from the MOM 2005 / SCOM 2007 console. You can then define appropriate actions and responses within MOM 2005 / SCOM 2007 for these events. EMC events are displayed in the EMC Alarms public view.

EMC events are also displayed within the EMC View computer group. This group shows the Windows host where you installed the Integration Gateway.

Double-click the EMC View group to display the Integration Gateway host, and then double-click the host to show the specific events. Double-click the event to view the specific details.

APPENDIX A
MIBs and Traps

This appendix discusses the MIBs and Traps used by the ControlCenter Integration Packages.

Topics include:

Ionix ControlCenter MIBs

ControlCenter sends traps that are defined by the *FibreAlliance MIB* (FCMGMT-MIB) and the *EMC Gateway MIB* (EMCGATEWAY-MIB).

Both MIBs need to be loaded in order for SNMP traps to be received and decoded correctly.

1. Download the **FCMGMT-MIB** from ByteSphere
2. Copy and paste the text of the **EMC Gateway MIB** (version 1.01) into a text file. *(See EMC Gateway MIB below).*
3. Load both MIBs into your framework or trap receiver application.

EMC Gateway MIB

```
------------------------------------------------------

-- Copyright (c)2000,2001,2006,2007,2008
-- All rights reserved by

--

-- EMC Corporation

-- 171 South Street

-- Hopkinton, MA 01748 USA

--

-- This software is furnished under a license and may be -- used and
copied inclusion of the above copyright
-- notice. This software or  any  other only  in
-- accordance  with  the  terms  of such  license and with
-- the copies thereof may not be provided or otherwise
-- made available to any other person. No title to and
-- ownership of  the  software  is  hereby transferred.

-- The information in this software is subject to change
-- without notice and should not be construed as a
-- commitment by EMC Corporation.

--

-- EMC Corporation assumes no responsibility for the use
-- or reliability of this software.
```

```
-----------------------------------------------------------------
----------
--
-- EMC Gateway Management Information Base description
--
-----------------------------------------------------------------
----------
-----------------------------------------------------------------
----------
--
-- Author Identification
--
-- JDS      John D. Sullivan, EMC
-- JCE      John C. Ehn, EMC
--
-------------------------------------------------------
EMCGATEWAY-MIB
DEFINITIONS ::= BEGIN
   IMPORTS
   enterprises
        FROM RFC1155-SMI
   OBJECT-TYPE
        FROM RFC-1212
   TRAP-TYPE
        FROM RFC-1215
   connUnitId, connUnitEventId, connUnitEventType,
connUnitEventObject,
   connUnitEventSeverity, connUnitName, connUnitType,
   connUnitEventDescr, connUnitStatus, connUnitState
        FROM FCMGMT-MIB;
--Textual conventions for this MIB

   DisplayString ::=
     OCTET STRING
```

```
-- This data type is used to model textual information
-- taken from the NVT ASCII character set. By convention,
-- objects with this syntax are declared as having
--      SIZE (0..255)
   emc               OBJECT IDENTIFIER ::= { enterprises 1139 }
   eccGateway        OBJECT IDENTIFIER ::= { emc 3 }
   eccGatewayRevision OBJECT-TYPE
        SYNTAX DisplayString (SIZE (4))
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "This is the revision number for this MIB. The
            format of the revision value is as follows
                (0) = high order major revision number
                (1) = low order major revision number
                (2) = high order minor revision number
                (3) = low order minor revision number
            "
        ::= { eccGateway 1 }

    eccUnitStatusChange TRAP-TYPE
        ENTERPRISE eccGateway
        VARIABLES { connUnitStatus,
                    connUnitState,
                    connUnitName,
                    connUnitType }
        DESCRIPTION
            "The overall status of the ControlCenter-monitored
            device has changed.
            Recommended severity level (for filtering): alert"
        ::= 1
    eccUnitDeletedTrap TRAP-TYPE
        ENTERPRISE eccGateway
```

```
        VARIABLES { connUnitId,
                    connUnitName }
        DESCRIPTION
"A ControlCenter-monitored device has been deleted from the
Console.
            Recommended severity level (for filtering): warning"
        ::= 3
    eccUnitEventTrap TRAP-TYPE
        ENTERPRISE eccGateway
        VARIABLES { connUnitEventId,
                    connUnitEventType,
                    connUnitEventObject,
                    connUnitEventDescr,
                    connUnitEventSeverity,
                    connUnitName,
                    connUnitType }
        DESCRIPTION
            "An event has been generated by the
            ControlCenter-monitored device.
            Recommended severity level (for filtering): info"
        ::= 4
END
```

The FibreAlliance MIB and the EMC Gateway MIB

ControlCenter sends traps using the EMC Gateway Enterprise OID (.1.3.6.1.4.1.1139.3). The FibreAlliance MIB uses Enterprise OID .1.3.6.1.3.94. If the FibreAlliance MIB is loaded, it will only be able to decode traps that state the FibreAlliance Enterprise OID. To get around the issue, the EMC Gateway MIB provides a "bridge" so that traps using the EMC Gateway OID can be decoded correctly using the FibreAlliance MIB.

All traps use the following three pieces of information to identify themselves:

- **Enterprise OID**

The number that identifies a specific Management Information Base (MIB). A MIB is a text file that defines the format of the data coming from a specific type of device. If a device supports the method defined by FibreAlliance, for example, it is said to support the "FibreAlliance MIB".

- **Generic Trap Number**

  The basic classification of the trap.
  Most vendor-specific traps use "6".

- **Specific Trap Number**

  The Specific Trap Number refers to the actual number of the trap defined in the MIB. See the below TRAP-TYPE entries for examples.

Example:
A connUnitEventTrap from a FibreAlliance-compatible device, would be sent with Enterprise OID .1.3.6.1.3.94, Generic Trap 6, and Specific Trap 4 in the header. ControlCenter sends the same trap as Enterprise OID .1.3.6.1.4.1.1139.3, Generic Trap 6, and Specific Trap 4.

The Gateway MIB works around this problem by registering .1.3.6.1.4.1.1139.3 and re-creating the two trap-types used by ControlCenter, but by pointing the FibreAlliance MIB for the details. Because of this, both MIBs must be loaded.

connUnitTable

The Integration Gateway stores ControlCenter managed objects in the `connUnitTable` within the MIB, one entry per object. The Integration Packages read this MIB table as part of the discovery process.

SNMP Port

The Integration Gateway, by default, uses a different port number than the standard SNMP port 161. The Integration Gateway implements a standard MIB and SNMP. It uses a different port, so it does not conflict with any other SNMP agents that may reside on the host with the Integration Gateway.

The Integration Gateway uses registered port 1273 by default. This can be manually changed in the CNG.ini file by modifying the following line:

udp_port_number = 1273

Ionix ControlCenter Traps

The Integration Gateway supports four SNMP Traps:

- coldStart trap
- connUnitStatusChange trap
- connUnitDeletedTrap
- connUnitEventTrap

coldStart Trap

The Integration Gateway sends a `coldStart` trap to the frameworks when it starts and initializes. The Integration Packages software uses this trap to trigger the automatic discovery process. The trap contains the IP address of the Integration Gateway host.

The Integration Packages read the `connUnitTable` from the MIB by using SNMP `get` and `getnext` requests and add any ControlCenter managed objects that are stored in this table to the framework topology.

connUnitStatusChange Trap

The Integration Gateway sends the `connUnitStatusChange` trap when the state or status of a managed object changes. The trap identifies the object and its state and status. The Integration Packages software uses this information to set the status (icon color) for the object in the topology.

Note: If the object does not exist, the Integration Packages adds it to the topology.

connUnitDeleted Trap

The Integration Gateway sends the `connUnitDeletedTrap` when a managed object is deleted from the ControlCenter repository. The trap identifies the deleted object, and the Integration Packages use this information to set the status (icon color) of the object to unknown to indicate that the trap occurred. You must, however, manually delete the object from the framework topology.

connUnitEvent Trap

The Integration Gateway sends a `connUnitEventTrap` to report all other alerts and events. The trap identifies the object involved in the event, and includes the event description.

The event description contains the event details. The individual agents supply the event description. For example, the Symmetrix agent generates appropriate Symmetrix alerts.

Alerts sent to the Integration Gateway are part of your ControlCenter alert policies. With the ControlCenter Console, you can set alerts for objects and specify the alerts sent to the Integration Gateway that generate event traps.

The integration application only processes event traps of type:
`connUnitEventType=5 (topology)`

The Integration Gateway sends these traps for group objects, when an object is deleted from a group. A topology event also causes a refresh of the object. All other traps are logged in the frameworks event log.

Note: If an event causes a change in state or status for an object, the Integration Gateway also sends a `connUnitStatusChange` trap.

Ionix ControlCenter SNMP Trap Formats

eccUnitEventTrap

Generic Type 6, Specific Type 4
Notifies of a specific event that has occurred

- **Varbind 1 - connUnitEventId**
  32-bit Integer
  Internal Event ID incremented for each event
- **Varbind 2 - connUnitEventType**
  Integer
  Type of Event:

- unknown(1)
- other(2)
- status(3)
- configuration(4)
- topology(5)

- **Varbind 3 - connUnitEventObject**
  OID
  Identifier of the referred object
  This is always .0.0

- **Varbind 4 - connUnitEventDescr**
  Octet String
  Description of Event in one of two Formats:

  Agent Event
  Fields are surrounded by brackets ([])

  Field 1 - Alert ID
  Field 2 - Alert Name
  Field 3 - Alert Object
  Field 4 - Alert Description

  Server Event
  One field with description, no surrounding brackets

- **Varbind 5 - connUnitEventSeverity**
  Integer
  Severity of Event:
  - unknown(1)
  - emergency(2)
  - alert(3)
  - critical(4)
  - error(5)
  - warning(6)
  - notify(7)
  - info(8)
  - debug(9)
  - mark(10)

- **Varbind 6 - connUnitName**
  Octet String
  Name of ControlCenter object to which the event refers

- **Varbind 7 - connUnitType**
  Integer
  Type of Object:
  - unknown(1)
  - other(2)
  - hub(3)
  - switch(4)
  - gateway(5)
  - converter(6)
  - hba(7)
  - proxy-agent(8)
  - storage-device(9)
  - host(10)
  - storage-subsystem(11)
  - module(12)
  - swdriver(13)
  - storage-access-device(14)
  - wdm(15)
  - ups(16)
  - nas(17)

eccUnitDeletedTrap

Generic Type 6, Specific Type 3
Notifies that a top-level object has been deleted from the Console

- **Varbind 1 - connUnitId**
  Octet String
  A unique identifier for the object
- **Varbind 2 - connUnitName**
  Octet String
  The name of the ControlCenter object that has been deleted

eccUnitStatusTrap

Generic Type 6, Specific Type 1
Notifies a change in status of a top-level object

- **Varbind 1 - connUnitStatus**
  Integer
  Status of Object:

- unknown(1)
- unused(2)
- ok(3)
- warning(4)
- failed(5)

- **Varbind 2 - connUnitState**
  Integer
  State of Object:
  - unknown(1)
  - online(2)
  - offline(3)

- **Varbind 3 - connUnitName**
  Octet String
  Name of ControlCenter object to which the status change refers

- **Varbind 4 - connUnitType**
  Integer
  Type of Object:
  - unknown(1)
  - other(2)
  - hub(3)
  - switch(4)
  - gateway(5)
  - converter(6)
  - hba(7)
  - proxy-agent(8)
  - storage-device(9)
  - host(10)
  - storage-subsystem(11)
  - module(12)
  - swdriver(13)
  - storage-access-device(14)
  - wdm(15)
  - ups(16)
  - nas(17)

Example:
In HP OpenView Network Node Manager, you could modify the information passed to the Alerts display by using the **Event Configuration** dialog:

1. In the Event Configuration dialog, under Enterprises, click EMC_Gateway. A list of events should appear below.
2. Under Events for Enterprise EMC_Gateway, edit emcGW_event.
3. Go to the Event Message tab and define the Event Log Message field using the above reference information. For instance:

   Severity: $5 Alert: $4 MO: $6

   This will cause the alert to display the severity, the alert text, and the affected object, while ignoring the trap definition information.
4. Once finished, save the changes, and new traps should display as expected.

Event Severity Mapping

This section shows the severity mapping for events for each qualified framework. The tables in this section show the mapping between ControlCenter severity level values and Fibre Channel MIB SNMP values, and between the Fibre Channel MIB SNMP values and framework-specific values.

The Integration Gateway maps ControlCenter severity levels to Fibre Channel MIB values. Table 5 on page 66 shows the relationship between ControlCenter severity levels, `connUnitEventSeverity`, and `connUnitStatus`.

Table 5 Severity Mapping Reference Table

ControlCenter Severity Level	connUnitEventSeverity	connUnitStatus
Fatal	emergency (2)	failed (5)
Critical	critical (4)	failed (5)
Warning	warning (6)	warning (4)
Minor	notify (7)	warning (4)
Information Normal	info (8)	ok (3)

For the frameworks shown in Table 6 on page 67, the ControlCenter integration maps `connUnitStatusChanged` traps to icon status based on the `connUnitStatus` variable within the trap.

Table 6 connUnitStatus Mapping Reference Table

connUnitStatus	HP OpenView NNM	Tivoli NetView	CA Unicenter
Unknown	Unknown	Unknown	Unknown
Unused	Unknown	Unknown	Unknown
OK	Normal	Normal	OK
Warning	Marginal	Marginal	Warning
Failed	Critical	Critical	Failed

For the frameworks shown in Table 7 on page 67, the ControlCenter integration maps the Gateway trap type to a severity level in the framework event display, and ignores the status or severity variables within the trap.

Table 7 Gateway to Framework Event Mapping Reference Table

Gateway Trap Type	PEM	MOM / SCOM	TEC	OVO
coldStart	Informatio n	Information	Minor	Normal
connUnitStatusChang ed	Minor	Warning	Warning	Minor
connUnitDeleted	Minor	Warning	Warning	Minor
connUnitEvent	Warning	Error	Minor	Warning

The BMC PATROL integration maps `connUnitEvent` severity to icon status and ignores other traps as shown in Table 8 on page 67.

Table 8 BMC PATROL Mapping Reference Table

connUnitEventSeverity	BMC PATROL
emergency (2)	Alarm
critical (4)	Alarm

Table 8 BMC PATROL Mapping Reference Table

connUnitEventSeverity	BMC PATROL
warning (6)	Warning
notify (7)	Warning
info (8)	Information

APPENDIX B
Configuration Settings

This appendix discusses the configuration settings used by the ecc3pi integration application for Enterprise Management Frameworks (EMFs).

Topics include:

The ecc3pi Application Settings

The integration application, `ecc3pi`, creates an `.ini` file of configuration settings in the framework `bin` directory. The default settings are appropriate in most situations, however, you can modify them.

The configuration settings contain three groups:

- Polling
- SNMP
- Repository (for CA only)

In addition, these global settings appear in the `.ini` file:

- ECC — Console mode is an EMC_View that shows groups including the **All Objects** group.
- MIB — Object mode is an EMC_View that shows all managed objects.

Polling Frequency

The integration application constantly polls the Integration Gateway for status information. There are polling frequency parameters that control the polling cycle that are defined in `ecc3pi`. They are:

- Heartbeat
- Retry
- Refresh

The integration application polls the gateway at a default specified heartbeat interval. If the gateway does not respond, the integration application repolls the gateway every minute up to the specified retry limit.

default polling frequency is every 15 minutes and retry limit is 4.

The periodic polling checks the status of the Integration Gateway. The periodic refresh triggers a rediscovery of the ControlCenter managed objects. You can use these features to resynchronize the framework in the event that traps get lost.

The refresh default time is every eight hours.

Note: The default refresh normalization time is midnight.

SNMP GET Request

The integration application sends SNMP GET requests to the gateway as part of polling, discovery, and event processing.

These parameters control the request retry frequency.

- The heartbeat timeout is the quantity of time that the integration application waits for a response from a periodic Integration Gateway poll. The heartbeat default is three seconds.
- The data timeout is the time the integration application waits for a response for a request for object information during a discovery. The data timeout defeat is five seconds.
- Both requests use the same number of specified retries. The default retry number is three.

Repository (CA Unicenter)

The Integration Packages require a repository name, and a username and password, to connect to the CA repository. You can manually edit the `.ini` file to add your username and password.

If you specify a username and password, then you are not prompted for this information when you start the Integration Package for CA Unicenter.

Note: This may be a security concern since the username and password are saved in plain text. If this is a security concern, or if you do not want automatic login, then delete these entries. You are then prompted for this information each time you start the integration.

Configure Ionix ControlCenter Server to Enable Status Traps

Active integrations like NNM, Unicenter, and Tivoli NetView (Windows) process connUnitStatusChange traps to display color changes on objects. Passive integrations are usually configured to suppress these traps since this helps to eliminate unnecessary traps, traffic and so on.

To control this feature, edit the following file on the Ionix ControlCenter Server:

```
<install_directory>\ecc_inf\data\<servername>\class\SnmpEventPlugi
   n.properties
```

Edit the following line (`true` is the default— the alert is sent, changing to `false`—the alert is not sent):

`enableStatusListener=true (or false)`

Sample ecc3pi.ini Configuration File

This section provides a sample `ecc3pi.ini` file.

Note: For BMC PATROL KM, replace **`get_community=public`** with **`snmp_community=public`** in the default `ecc3pi.ini` provided.

```
; ecc3pi.ini
; EMC Integration Packages configuration settings
[Global Settings]
; Allows the user to change the display mode within the framework.
; valid choices are MIB and ECC. The system default is ECC
display_mode=ECC
[Polling Attributes]
; Specify the heartbeat polling frequency for EMC gateways, in minutes.
; A value of zero disables polling.
; Specify the retry count for non-responding EMC gateways
; before returning to normal polling frequency.
; Specify the periodic refresh of EMC gateways to discover new Managed
   Objects, in minutes.
; Recommended defaults:
;  heartbeat=15
;  retry=4
;  refresh=480
heartbeat=15
retry=4
refresh=480
[SNMP Attributes]
; Specify the SNMP heartbeat polling timeout for EMC gateways, in seconds.
; Specify the SNMP data request timeout for EMC gateways, in seconds
; Specify the SNMP request retry count
; Specify the SNMP GET community string
; Recommended defaults:
;  heartbeat=3
;  data=5
;  retry=3
;  get_community=public
heartbeat=3
data=5
retry=3
get_community=public
; Parameter is for CA/Unicenter ONLY.
; Specify CA/Unicenter Repository information as:
; repository name, user name, password
[Repository]
repository=FONTAINEXXXXB
username=
password=
```

APPENDIX C
Configuring ControlCenter to Send Alerts

This appendix discusses how to configure ControlCenter to accept multiple trap destinations.

Topics include:

Multiple Trap Destinations

When you install the Integration Gateway, you specify the framework IP address as the initial default trap destination. If you need more than one trap destination, you need to manually edit the `CNG.ini` file. by performing this procedure, as follows:

1. Open the `CNG.ini` file
2. Add the following line:
   `trap_client_registration(ip,port,severity,state)`

In addition, you can filter traps on a severity level as described in `FcEventSeverity`. For example, if you do not want low severity alerts to trigger traps, and you want only critical and fatal alerts to be sent to the framework, then you can modify the severity parameter from 10 (all messages) to 4. Each trap destination can be configured to receive traps from a specified severity level and up.

The gateway then sends traps to all specified destinations.

Trap Variables

In addition to the variables defined in the `connUnitStatusChange` and `connUnitEventTrap` of the `fcmgmt.mib`, the Integration Gateway adds variables to the trap when it is sent. You can use these variables for further processing on the framework side for tasks such as parsing and event log filtering.

Table 9 on page 77 describes `connUnitStatusChange` trap variables:

Table 9 connUnitStatusChange (Trap 1) Variables

Variable Number	Variables sent from Integration Gateway	Values	Description
	Date/Time		Timestamp of the trap
	Source		IP/Hostname of Integration Gateway
1	connUnitStatus	Unknown (1) Ok (3) Warning (4) Failed (5)	New status of object as defined in MIB
2	connUnitState	Unknown (1) Online (2) Offline (3)	New state of object as defined in MIB
3 [1]	connUnitName		Top-level source of the alert
4 [a]	FcUnitType	Other (2) Switch) (4) Host (10) Storage-subsystem (11)	Type of the top-level source as defined in the MIB

1. Additional variable.

Table 10 on page 78 describes `connUnitEventTrap` variables:

Table 10 connUnitEventTrap (Trap 4) Variables

Variable Number	Variables sent from Integration Gateway	Values	Description
	Date/Time		Timestamp of trap
	Source		IP/Hostname of Integration Gateway
1	connUnitEventId	Number	Sequential number for each event per connUnitEntry
2	connUnitEventType	Status (3)	Fixed value
3	connUnitEventObject	`.ccitt.zeroDotZero`	
4	connUnitEventDescr	Alert ID Alert Name Alert Object Alert Description	Unique identifier Name of alert Name for source object Detailed description
5 [1]	FcEventSeverity	Unknown (1) Fatal (2) Critical (4) Warning (6) Minor (7) Information (8)	This is the severity from ControlCenter and can be mapped to framework templates or events
6 [a]	connUnitName		Top-level source of the alert
7 [a]	FcUnitType	Other (2) Switch (4) Host (10) Storage-subsystem (11)	Type of the top-level source as defined in the MIB

1. Additional variable.

APPENDIX D
Framework Integration Examples

This appendix presents Enterprise Management Framework integration examples.

Topics include:

Netcool/OMNIbus

Netcool needs only one configuration file, ecc3pi_nco.rules, which will be merged with the trapd.rules file on the host where the Trapd probe resides.

The following example shows modifications around formatting and display, severity mapping between ControlCenter and Netcool, as well as filtering.

```
#######################################
#                                     #
# Start EMC ControlCenter trap rules #
#                                     #
#######################################
#
#############################
# match CC enterprise ID  #
#############################


if (match($enterprise, ".1.3.6.1.4.1.1139.3"))


{



#############################################################
#
# match and assign Node field to the variables that CC
# sends in the traps... Note: for specific trap "1",
# the Node field comes through as variable $3.  For specific
# trap "4" events, the Node comes through as variable $6.
# Variable $6 can come through with the domain suffix
# appended, the extract statement pulls any appended text
```

```
# from the Node assignments.

#################################################################
#


if (match($specific-trap, "1")){@Node = $3}
else if(match($specific-trap, "4"))
{
if(regmatch($6, "([0-9A-Za-z]+)\.[0-9A-Za-z].*"))
        {

$nodename=extract($6,"([0-9A-Za-z]+)\.[0-9A-Za-z].*")
        @Node = $nodename
        }
else    { @Node = $6 }
}
############################
# Assign basic fields
############################
@Agent = "ECC"
@AlertGroup = "Enterprise"
@Summary = "EMC trap received"
if (match($generic-trap, "0"))
{
@Summary = "EMC Gateway started on " + $PeerIPaddress
@Severity =  0
}
######################
```

```
# Status Trap Filter  #
########################
if (match($specific-trap, "1"))
{
############################
# Assign groups to alerts
############################
if(nmatch($4, "4")){@AlertGroup = "switch"}
        else if(nmatch($4, "10")){@AlertGroup = "host"}
        else if(nmatch($4, "11")){@AlertGroup =
"storage-subsystem"}
        else {@AlertGroup = "other"}
############################
# Assign state change values
############################
if(nmatch($2, "2")){@AlertKey = "Online"}
else if (nmatch($2, "3")){@AlertKey = "Offline"}
else {@AlertKey = $2}
############################
# Assign Summary field
############################
@Summary = @AlertGroup + " is " + @AlertKey
############################
# Assign Unique Identifier
############################
@Identifier = @Node + ":" + @AlertGroup
########################################
```

```
# Discard all state change alerts
# EXCEPT the 'OK' 'ONLINE' combination
########################################
if(nmatch($1, "3")){@Severity = 0}
else {discard}
}
else if (match($specific-trap, "3"))
{
@Summary = "EMC Delete trap: " + $OID1 + " " + $1
@Severity = 0
}
######################
# Event trap filter  #
######################
else if (match($specific-trap, "4" ))
{
############################
# Assign groups to alerts
############################
if(nmatch($7, "4")){@AlertGroup  = "switch"}
else if(nmatch($7, "10")){@AlertGroup = "host"}
else if(nmatch($7, "11")){@AlertGroup = "storage-subsystem"}
else {@AlertGroup = "other"}
##################################
# Extract and Assign Summary field
##################################
$summary=extract($4, ".*Description: (.*)\].*")
```

```
@Summary=$summary
##########################################
# Extract and Assign Sub-Groups to alerts
##########################################
$alertkey=extract($4, ".*Alert Name: (.*)\].*")
@AlertKey=$alertkey
####################
# Assign Severity
####################
if (nmatch($5, "2")){@Severity = 5}
else if (nmatch($5, "3")){@Severity = 4}
else if (nmatch($5, "4")){@Severity = 5}
else if (nmatch($5, "5")){@Severity = 4}
else if (nmatch($5, "6")){@Severity = 3}
else if (nmatch($5, "7")){@Severity = 3}
else if (nmatch($5, "8")){@Severity = 0}
else {@Severity = 1}
########################################
# Extract and Assign Unique Identifier
########################################
$identifier=extract($4, ".*Alert ID:(.*)\].*")
@Identifier=@Node + ":" + $identifier
}
}
```

Tivoli TEC

To leverage the additional trap variables sent from the Integration Gateway Agent, the oid and cds files needed some modifications. Because TEC is a passive integration the status and delete trap were suppressed.

```
############################################################
##
#  ecc3pi_tec.baroc
#
TEC_CLASS: EMC_Event ISA EVENT
  DEFINES {
    sub_source: default= "EMC";
    severity: default = WARNING;
    msg: default = "EMC Gateway trap received.";
    Description: STRING;
  };
END
TEC_CLASS: EMC_ECC_GW_start ISA EMC_Event;
# An EMC Gateway has started.
END
TEC_CLASS: EMC_ECC_GW_status ISA EMC_Event;
# The status of on EMC managed object has changed.
END
TEC_CLASS: EMC_ECC_GW_delete ISA EMC_Event;
# An EMC managed object has been deleted from the
ControlCenter database.
END
TEC_CLASS: EMC_ECC_event_informational ISA EMC_Event
```

```
# An EMC managed object has reported an event that may
indicate a fault.
     DEFINES {
        severity: default = INFORMATIONAL;
hostname: dup_detect=YES;
ECC_Event_ID: STRING, dup_detect=YES;
ECC_Event_Severity: STRING;
ECC_Description: STRING;
ECC_Detail: STRING;
     };
END
TEC_CLASS: EMC_ECC_event_warning ISA EMC_Event
# An EMC managed object has reported an event that may
indicate a fault.
     DEFINES {
        severity: default = WARNING;
hostname: dup_detect=YES;
ECC_Event_ID: STRING, dup_detect=YES;
ECC_Event_Severity: STRING;
ECC_Description: STRING;
ECC_Detail: STRING;
     };
END
TEC_CLASS: EMC_ECC_event_minor ISA EMC_Event
# An EMC managed object has reported an event that may
indicate a fault.
     DEFINES {
        severity: default = MINOR;
```

```
hostname: dup_detect=YES;

ECC_Event_ID: STRING, dup_detect=YES;

ECC_Event_Severity: STRING;

ECC_Description: STRING;

ECC_Detail: STRING;

     };

END

TEC_CLASS: EMC_ECC_event_critical ISA EMC_Event

# An EMC managed object has reported an event that may
indicate a fault.

     DEFINES {

        severity: default = CRITICAL;

hostname: dup_detect=YES;

ECC_Event_ID: STRING, dup_detect=YES;

ECC_Event_Severity: STRING;

ECC_Description: STRING;

ECC_Detail: STRING;

     };

END

TEC_CLASS: EMC_ECC_event_fatal ISA EMC_Event

# An EMC managed object has reported an event that may
indicate a fault.

     DEFINES {

        severity: default = FATAL;

hostname: dup_detect=YES;

ECC_Event_ID: STRING, dup_detect=YES;

ECC_Event_Severity: STRING;
```

```
ECC_Description: STRING;

ECC_Detail: STRING;

     };

END


##################################################################
##

#  ecc3pi_tec.cds

#  Status and Delete trap not used, severity added

##################################################################
##

#

#  Description: Default set of class definition statements
for #  the SNMP Trap TEC Adapter.

##################################################################
##

#  DEFAULT SLOT VALUES

##################################################################
##

# The forwarding_agent attribute is commented out so people
who

# upgrade to TEC 3.7 won't get PARSING FAILED when the event
with

# the new slot arrives at the TEC server.
#

##################################################################
##

MAP_DEFAULT

   source = SNMP;

   sub_source = NET;
```

```
#  forwarding_agent = $SOURCE_ADDR;
   origin = $AGENT_ADDR;
   adapter_host = $ADAPTER_HOST;
END


## Move this class definition before the standard SNMP Cold
Start Trap
CLASS emcGW_start
  SELECT
        1: ATTR(=,$ENTERPRISE) , VALUE(PREFIX,
"1.3.6.1.4.1.1139.3" );
        2: $TYPE = 0 ;
  MAP
        sub_source = "EMC";
        severity = MINOR;
        msg = PRINTF("EMC Gateway started on %s",
$AGENT_ADDR);
END
################################################################
#
#  EMC ControlCenter event classes
#
CLASS EMC_ECC_event_informational
   SELECT
     1: ATTR(=,$ENTERPRISE) , VALUE(PREFIX,
"1.3.6.1.4.1.1139.3" );
     2: $SPECIFIC = 4 ;
     3: ATTR(=, "connUnitEventId");
```

```
      4: ATTR(=, "connUnitEventType");
      5: ATTR(=, "connUnitEventObject");
      6: ATTR(=, "connUnitEventDescr");
      7: ATTR(=, "connUnitEventSeverity") , VALUE(=,8);
      8: ATTR(=, "connUnitName");
      9: ATTR(=, "connUnitType");
   FETCH

      1: SUBSTR($V6,1,22);
      2: SUBSTR($V6,23,220);
      3: SUBSTR($V6,23,43);
   MAP
      hostname = $V8;
      origin = $V8;
      msg = $F2;
      ECC_Event_ID = PRINTF("Type: %s", $F3);
      ECC_Event_Severity = $V7;
      ECC_Description = PRINTF("EMC: %s   UnitEventSeverity:
%s UnitName: %s", $F2, $V7, $V8);
      ECC_Detail = PRINTF("Slot V6: %s", $V6);
END

CLASS EMC_ECC_event_warning
   SELECT
      1: ATTR(=,$ENTERPRISE) , VALUE(PREFIX,
"1.3.6.1.4.1.1139.3" );
      2: $SPECIFIC = 4 ;
```

```
      3: ATTR(=, "connUnitEventId");
      4: ATTR(=, "connUnitEventType");
      5: ATTR(=, "connUnitEventObject");
      6: ATTR(=, "connUnitEventDescr");
      7: ATTR(=, "connUnitEventSeverity") , VALUE(=,6);
      8: ATTR(=, "connUnitName");
      9: ATTR(=, "connUnitType");
   FETCH
      1: SUBSTR($V6,1,22);
      2: SUBSTR($V6,23,220);
      3: SUBSTR($V6,23,43);
   MAP
      hostname = $V8;
      origin = $V8;
      msg = $F2;
      ECC_Event_ID = PRINTF("Type: %s", $F3);
      ECC_Event_Severity = $V7;
      ECC_Description = PRINTF("EMC: %s   UnitEventSeverity:
%s UnitName: %s", $F2, $V7, $V8);
      ECC_Detail = PRINTF("Slot V6: %s", $V6);
END


CLASS EMC_ECC_event_minor
   SELECT
      1: ATTR(=,$ENTERPRISE) , VALUE(PREFIX,
"1.3.6.1.4.1.1139.3" );
      2: $SPECIFIC = 4 ;
```

```
      3: ATTR(=, "connUnitEventId");
      4: ATTR(=, "connUnitEventType");
      5: ATTR(=, "connUnitEventObject");
      6: ATTR(=, "connUnitEventDescr");
      7: ATTR(=, "connUnitEventSeverity") , VALUE(=,7);
      8: ATTR(=, "connUnitName");
      9: ATTR(=, "connUnitType");
   FETCH
      1: SUBSTR($V6,1,22);
      2: SUBSTR($V6,23,220);
      3: SUBSTR($V6,23,43);
   MAP
      hostname = $V8;
      origin = $V8;
      msg = $F2;
      ECC_Event_ID = PRINTF("Type: %s", $F3);
      ECC_Event_Severity = $V7;
      ECC_Description = PRINTF("EMC: %s   UnitEventSeverity:
%s UnitName: %s", $F2, $V7, $V8);
      ECC_Detail = PRINTF("Slot V6: %s", $V6);
END


CLASS EMC_ECC_event_critical
   SELECT
      1: ATTR(=,$ENTERPRISE) , VALUE(PREFIX,
"1.3.6.1.4.1.1139.3" );
      2: $SPECIFIC = 4 ;
```

```
      3: ATTR(=, "connUnitEventId");
      4: ATTR(=, "connUnitEventType");
      5: ATTR(=, "connUnitEventObject");
      6: ATTR(=, "connUnitEventDescr");
      7: ATTR(=, "connUnitEventSeverity") , VALUE(=,4);
      8: ATTR(=, "connUnitName");
      9: ATTR(=, "connUnitType");
   FETCH
      1: SUBSTR($V6,1,22);
      2: SUBSTR($V6,23,220);
      3: SUBSTR($V6,23,43);
   MAP
      hostname = $V8;
      origin = $V8;
      msg = $F2;
      ECC_Event_ID = PRINTF("Type: %s", $F3);
      ECC_Event_Severity = $V7;
      ECC_Description = PRINTF("EMC: %s   UnitEventSeverity:
%s UnitName: %s", $F2, $V7, $V8);
      ECC_Detail = PRINTF("Slot V6: %s", $V6);
END


CLASS EMC_ECC_event_fatal
   SELECT
      1: ATTR(=,$ENTERPRISE) , VALUE(PREFIX,
"1.3.6.1.4.1.1139.3" );
      2: $SPECIFIC = 4 ;
```

```
      3: ATTR(=, "connUnitEventId");
      4: ATTR(=, "connUnitEventType");
      5: ATTR(=, "connUnitEventObject");
      6: ATTR(=, "connUnitEventDescr");
      7: ATTR(=, "connUnitEventSeverity") , VALUE(=,2);
      8: ATTR(=, "connUnitName");
      9: ATTR(=, "connUnitType");
   FETCH

      1: SUBSTR($V6,1,22);
      2: SUBSTR($V6,23,220);
      3: SUBSTR($V6,23,43);
   MAP
      hostname = $V8;
      origin = $V8;
      msg = $F2;
      ECC_Event_ID = PRINTF("Type: %s", $F3);
      ECC_Event_Severity = $V7;
      ECC_Description = PRINTF("EMC: %s   UnitEventSeverity:
%s UnitName: %s", $F2, $V7, $V8);
      ECC_Detail = PRINTF("Slot V6: %s", $V6);
END

# ecc3pi_tec.oid
#"experimental"         "1.3.6.1.3"
#"fcmgmt"               "1.3.6.1.3.94"
"connUnitId"            "1.3.6.1.3.94.1.6.1.1"
```

```
"connUnitType"       "1.3.6.1.3.94.1.6.1.3"
"connUnitState"         "1.3.6.1.3.94.1.6.1.5"
"connUnitStatus"        "1.3.6.1.3.94.1.6.1.6"
"connUnitName"   "1.3.6.1.3.94.1.6.1.20"
"connUnitEventId"       "1.3.6.1.3.94.1.11.1.3"
"connUnitEventSeverity" "1.3.6.1.3.94.1.11.1.6"
"connUnitEventType"     "1.3.6.1.3.94.1.11.1.7"
"connUnitEventObject"   "1.3.6.1.3.94.1.11.1.8"
"connUnitEventDescr"    "1.3.6.1.3.94.1.11.1.9"
```

HP ITO/VPO/OVO

Over the years, HP renamed the OpenView Operations (OVO) product many times. The initial name with release 2 was OPC; release 5.x was called IT/Operations (ITO); release 6.x was known as VantagePoint Operations (VPO); and as of release 7.x, HP renamed it to OpenView Operations (OVO). These names still appear inside the product in different variations.

NNM is always installed as part of OVO, but not all customers use the NNM screens and functionality.

The installation script asks whether to install software for NNM, OVO, or both. Depending on the selection, only part of the Integration Packages software will be loaded. Even if the customer is not using NNM, it might be best to install both.

Before running the install script, you can modify the trap.dat file to assign severity levels to different traps. Then, for example, if ControlCenter triggers with a Critical error, the error would appear with the severity you defined in the trap.dat (such as Warning).

```
SYNTAX_VERSION 3


SNMP "EMC_Traps"
DESCRIPTION "Message Conditions for SNMP Trap Interception"
SEVERITY Normal
APPLICATION "EMC ControlCenter"
MSGGRP "EMC_Alarms"
MSGCONDITIONS
  DESCRIPTION "emcGW_start"
    CONDITION
    $e ".1.3.6.1.4.1.1139.3"
    $G 0
    SET
      SEVERITY Normal
      NODE OTHER "EMC_View"
```

```
      OBJECT "<$A>"
      TEXT "EMC Gateway started on <$A>"
      HELPTEXT "Trap sent when EMC Gateway is started."
      HELP "9cec43f2-5131-71d5-03de-ac1793a10000"
  DESCRIPTION "emcGW_status"
    CONDITION
    $e ".1.3.6.1.4.1.1139.3"
    $G 6
    $S 1
    SET
      SEVERITY Minor
      NODE OTHER "EMC_View"
      OBJECT "<$A>"
      TEXT "EMC Status trap: Status: <$1> State: <$2> MO:
<$3>"
      HELPTEXT "Trap sent by EMC Gateway when object status
changes."
      HELP "9cf1c9c6-5131-71d5-03de-ac1793a10000"
  DESCRIPTION "emcGW_delete"
    CONDITION
    $e ".1.3.6.1.4.1.1139.3"
    $G 6
    $S 3
    SET
      SEVERITY Minor
      NODE OTHER "EMC_View"
      OBJECT "<$A>"
```

```
      TEXT "EMC Delete trap: <$1>"
      HELPTEXT "Trap sent by EMC Gateway when object is
deleted."
      HELP "9cf3328e-5131-71d5-03de-ac1793a10000"
  DESCRIPTION "emcGW_event"
    CONDITION
    $e ".1.3.6.1.4.1.1139.3"
    $G 6
    $S 4
    SET
    SEVERITY Warning
    NODE OTHER "EMC_View"
    OBJECT "<$A>"
    TEXT "EMC Event trap: #<$1> Sev:<$5> Msg:<$4> MO:<$6>"
    HELPTEXT "Trap sent by EMC Gateway for general object
events."
    HELP "9cf488d2-5131-71d5-03de-ac1793a10000"
```

Integration Packages creates OVO templates in the Toplevel section named EMC_Traps. The templates need to be assigned to a system (such as the Management Server) running the trap interceptor.

Selecting EMC_Traps and Conditions, rename the default emcGW_event and create copies for the severities you want to appear in OVO.

OVO expects each trap to be of certain severity. As defined in the FCMGMT.MIB, the traps do not contain a severity variable, but the Integration Gateway Agent adds the severity as variable $5. You can define the values of this variable in a template field.

The same procedure can be done with other severities for minor, major, warning, etc The result is a much easier to read Message Browser.

HP NNM

Network Node Manager is an active integration and is supported on Windows, HP-UX, and Solaris. You can modify the display of the messages in the Event Configuration window.

By default NNM expects each trap to be of a specific severity. Because the Integration Gateway sends many different alerts with the same trap a possible severity mapping from $5 would have to be done via actions tab. The trap variables could be passed into a script which makes the mapping and call ovEvent to return it back to NNM.

CA Unicenter

Unicenter runs on Windows. The modifications shown replace the message record definition and run an external batch program to filter events and replace some text strings.

```
Create trap filter and apply to \tng\bin

Modify operations rules and apply to \tng\bin

cagui msgrecord

Remove old message record entry for "* * * * 1139"

Create modified entry

cautil -f tngecc.opr

Reload new message format and activate it

oprcmd opreload



define msgrec
      msgid="* * * * 1139"
       type="MSG"
       msgnode="*"
       desc="EMC Enterprise traps"
       cont='N'
       msgact='Y'
       wcsingle='?'
       wcmany='*'
       case="y"
       regexp="n"
define msgact
       name=(*,10)
```

```
        action="COMMAND"
        attrib="DEFAULT"
        color="DEFAULT"
        condop="  "
        evaluate='Y'
        quiet='Y'
        status="ACTIVE"
        sim='N'
        text="ecc3pi -trap &TEXT"
define msgact
        name=(*,15)
        action="DISCARD"
        attrib="DEFAULT"
        color="DEFAULT"
        condop="  "
        evaluate='Y'
        quiet='N'
        status="ACTIVE"
        sim='N'
define msgact
        name=(*,20)
        action="COMMAND"
        attrib="DEFAULT"
        color="GREEN"
        condop="  "
        evaluate='Y'
        quiet='Y'
```

```
        status="ACTIVE"
        sim='N'
        text="ecctrapfilter.bat &text"

@echo off
rem ecctrapfilter.bat
rem
rem This procedure translates the Gateway traps on TNG
rem to improve readability
rem
rem EMC PS Germany
:noprint
set msg=[
if "%1" == "Object:" goto createmsg
if "%1" == "Description:" goto createmsg
shift
if not "%1" == "" goto noprint
goto end
rem create the new message  out of connUnitEventDescr
:createmsg
if "%1" == "OID:" goto severity
if "%1" == "Alert" goto noprint
set msg=%msg%%1
shift
if not "%1" == "" goto createmsg
:severity
shift
```

```
shift
shift
shift
if "%msg%" == "[" goto end
if "%1" == "8" cawto EMC ControlCenter - INFO/SOLVED - %msg%
if "%1" == "7" cawto EMC ControlCenter - MINOR - %msg%
if "%1" == "6" cawto EMC ControlCenter - WARNING - %msg%
if "%1" == "4" cawto EMC ControlCenter - CRITICAL - %msg%
if "%1" == "2" cawto EMC ControlCenter - FATAL - %msg%
:end
exit
```

Tivoli NetView

NetView is supported as active integration on Windows. The settings of the trap configuration format are similar to NNM. If NetView is run on AIX, you must manually transfer and load the MIB and modify the trap database.

BMC PATROL Enterprise Manager (PATROL EM)

After the ControlCenter supplied software is installed, launch NetCmmnd and select ALFE.

You can duplicate the EMC_Event similar to the HP OVO section and use the emcsev token to map the severity accordingly.

INDEX