AlarmPoint Systems

AlarmPoint for HP Network Node Manager i-series Software

Validation Date
January 30, 2008
Version 2.0

Contents

1. Introduction

This document describes how to install and configure the AlarmPoint for HP Network Node Manager i-series Software (NNMi) Integration. The intended audience for this guide is experienced HP consultants, system administrators, and other technical readers.

1.1 Summary

This integration supports Event notifications (from NNMi to AlarmPoint) through the configuration of NNMi incident types. It also supports inbound actions (from AlarmPoint to NNMi) to acknowledge the original incident, alter its priority, and add informational annotations..

You will need to modify this configuration to suit your particular business requirements and adjust it to suit your expected loads. The default integration features automatic status annotations on the NNMi incident; in a high-volume production system, this can significantly affect performance. Consider your expected volume of injected events and your server capacity when designing your own integration with AlarmPoint.

1.1.1 Overview

The following is an overview of the integration steps for the AlarmPoint for NNMi integration:

1. Install the AlarmPoint Java Client on the NNMi system.
2. Install the NNMi-specific integration script for the AlarmPoint Java Client.
3. Install the Web Services Library on the AlarmPoint Webservers and Application server.
4. Install the AlarmPoint subscription panel for NNMi on the AlarmPoint Webservers. (Optional)
5. Install the AlarmPoint action scripts for NNMi using the AlarmPoint Developer IDE.
6. Install the integration voice files to the AlarmPoint Application server.
7. Configure an Event Domain (and optionally a Subscription Domain) in AlarmPoint.
8. Configure a Web Services Client within the NNMi system.
9. Configure the NNMi incident types to call the APClient utility, passing the incident parameters to AlarmPoint.
10. Validate that the integration can inject NNMi incident parameters for AlarmPoint notifications, and that AlarmPoint responses properly update the NNMi incident.

1.1.2 Benefits

With the AlarmPoint integration, the appropriate technician can be notified directly via voice, email, pager, or other device. Information about the failure will be presented to the event resolver and decisions can be made in real-time such as acknowledging, ignoring, annotating, or changing the priority of the event.

Once a response is selected on the recipient’s remote device, AlarmPoint will update the NNMi event in real-time. The benefit is that this process is immediate – significantly faster than the time required for Operations staff to notice the failures or malfunctions, determine who is on call, and manually notify the right person. In addition, the ability to take simple actions on the event from any device gives the event resolver a quick way to deal with many issues and communicate to other team members the current status of the event.

During the process, every notification, response, and action is logged in AlarmPoint. In addition, AlarmPoint automatically annotates the original NNMi incident with status information.

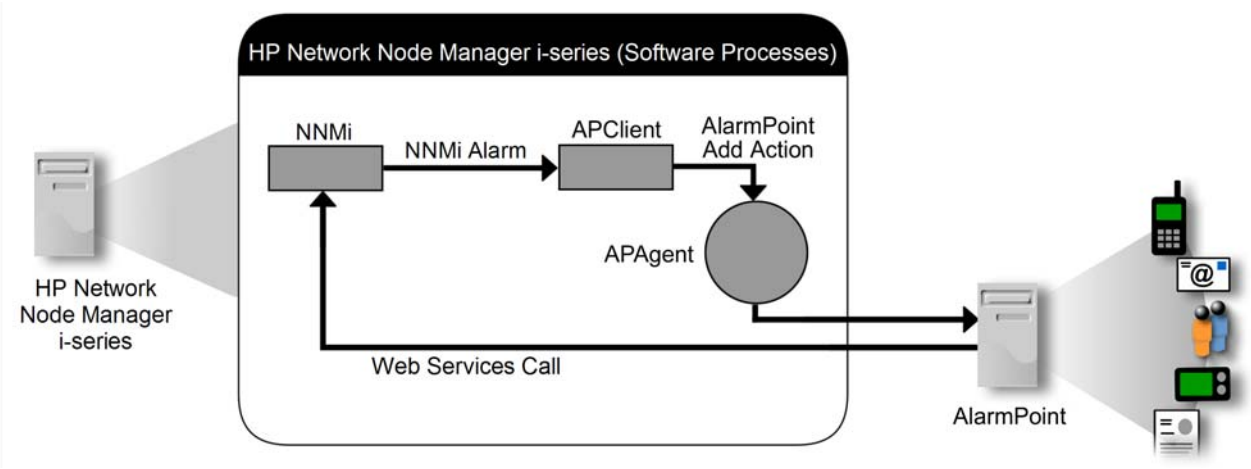
The AlarmPoint product features a self-service web user interface to allow accurate assignment of responsible personnel for each job. AlarmPoint also includes an optional enhanced Subscription panel that allows both managed and self subscription to NNMi events.

1.2 Architecture

The software components in the architecture include:

- NNMi: HP Network Node Manager i-series Software
- APClient: Management System message injection point.
- APAgent: Integration framework for passing messages to and from the AlarmPoint System.
- AlarmPoint Server: Represents the AlarmPoint Application Server Node.

The following figure illustrates the software processes of the integration components that allow the NNMi and AlarmPoint systems to interact by delivering notifications and injecting responses:



Whenever NNMi detects a problem in the network (e.g., “NonSNMPNodeUnresponsive”), it triggers the following steps, as illustrated in the architecture diagram:

1. NNMi calls the AlarmPoint Client (APClient) with the parameters describing the problem (e.g., computer affected, situation).
2. APClient submits the information to the AlarmPoint Agent (APAgent).
3. APAgent ensures delivery of the problem details to AlarmPoint, which in turn notifies the appropriate recipient.
4. The recipient responds to the notification and the acknowledgement, annotation or priority change updates NNMi through a web services call.

1.3 System Requirements

The following software components are required for the NNMi integration:

- AlarmPoint Java Client version 3.2.1 or later.
- AlarmPoint version 3.2.1 or later.
- HP Network Node Manager i-series version 8.01 or higher, with an NNM SDK license to enable web services.

1.3.1 Supported Operating Systems

The following operating systems can be used for integration between AlarmPoint and NNMi:

- Solaris
- Microsoft Windows
- HP-UX

1.4 Conventions & Terminology

This section describes how styles are used in the document, and provides a list of definitions.

1.4.1 Conventions

Some instructions appear in the following format: **MENU > OPTION**; for example, **File > Open** means click the **File** menu, and then click the **Open** menu option.

Words in **bold** typically reference text that appears on the screen.

Words in `monospace` font represent the following:

- text that must be typed into the computer
- directory and file names

1.4.2 Terminology

With respect to this integration, the following definitions apply:

Term	Meaning
Incident	Item of interest detected by HP Network Node Manager i-series.
Event	Information about the incident that was sent to AlarmPoint.
Incident ID	Unique identifier associated with an incident (also referred to as incident_id).

With respect to the AlarmPoint System, the following definitions apply:

Term	Meaning
AlarmPoint Admin	Administrative tool to control AlarmPoint Agent.
AlarmPoint Agent	Communication layer between third-party applications (e.g., a Management System) and AlarmPoint.
AlarmPoint Client	The Management System uses this to communicate with the AlarmPoint Agent.
AlarmPoint Application Server Node	The core AlarmPoint application, consisting of various components that process events and perform notifications.
AlarmPoint Java Client	Umbrella term for the AlarmPoint Admin, AlarmPoint Agent, and the AlarmPoint Client (both Java and native versions).
AlarmPoint Notification Server Node	Delivers notifications to a person in a variety of ways (pager, phone, e-mail, etc.).

Term	Meaning
AlarmPoint System	Umbrella term for all AlarmPoint software components.
AlarmPoint Web User Interface	Browser-accessible interface for controlling AlarmPoint components and information.
Management System	A synonym for HP Network Node Manager i-series.
User Guides	The AlarmPoint documentation suite, which includes the *AlarmPoint Installation and Administration Guide*, the *AlarmPoint Developer's Guide & Scripting Reference*, the *AlarmPoint User Guide*, and the *AlarmPoint Java Client User Guide*.

2. Installation

2.1 AlarmPoint System

This section covers installation of the core components required for integrating the AlarmPoint System with HP NNMi. Components may be installed in any order.

2.1.1 AlarmPoint

Consult the AlarmPoint User Manual for installation details.

2.1.2 AlarmPoint Java Client

Install the AlarmPoint Java Client on the NNMi server. During installation, set the address of the AlarmPoint server.

For complete installation instructions, refer to the *AlarmPoint Java Client Guide*.

It is recommended that you install the AlarmPoint Java Client into the default directory:

Windows:

```
C:\APAgent
```

Unix:

```
/opt/alarmpointsystems/APAgent
```

If installed in a different directory, the paths specified in the Incident Configuration need to be updated.

2.2 HP Network Node Manager i-series

Refer to the appropriate NNMi guides for installation details.

Note: *To allow for Web Service calls to NNMi, the web services license key must be applied to your NNMi installation. To enable annotations, NNMi must be at least version 8.01.*

2.3 Integration

This section covers the installation steps required for the integration files.

2.3.1 Integration Extraction

Extract the `AP-HP-NNMi` archive file to access the integration components. The following lists the notable files and folders in the archive:

```
|-- components
| |-- alarmpoint
| | |-- libs
| | | '-- alarmpoint-nnmi.jar
| | |-- scripts
| | | '-- AP-HP-NNMi.aps
| | |-- sub_panel
| | | '-- NNMiSubscriptionForm.jsp
| | '-- vox
```

```
| |   '-- english <English voice files>
| '-- alarmpoint-java-client
|   '-- hp_nnmi.xml
|-- documentation
| '-- AP-HP-NNMi.pdf
'-- release-notes.txt
```

2.3.1.1 Component Description

The following table describes some of the notable integration components and folders:

Component Name	Description
hp_nnmi.xml	Contains the parameter mapping for messages injected from the NNMi system to AlarmPoint.
alarmpoint-nnmi.jar	Contains the Web Services Library, which is used in the Action Scripts to inject responses back to NNMi, and used by the Subscription Panel to retrieve the available Source Node Names
NNMiSubscriptionForm.jsp	Custom Subscription JSP that allows users to subscribe to Events associated with specific criteria (CATEGORY, SEVERITY, etc.).
AP-HP-NNMi.aps	Contains the AlarmPoint Action Scripts required for the integration.

2.3.2 Installing hp_nnmi.xml

The `hp_nnmi.xml` file provided in the extracted integration archive must be installed into the default APAgent directory.

Windows:

- Source File: `\AP-HP-NNMi\components\alarmpoint-java-client\hp_nnmi.xml`
- Destination Directory: `C:\APAgent\etc\integrations`

Unix:

- Source File: `/AP-HP-NNMi/components/alarmpoint-java-client/hp_nnmi.xml`
- Destination Directory: `/opt/alarmpointsystems/APAgent/etc/integrations`

2.3.3 Installing the Web Services Library

To enable Web Service calls between the AlarmPoint and NNMi servers, you must copy the JAR file into the AlarmPoint Node, and the AlarmPoint Web Server library folders.

Windows:

- **Source File**: `AP-HP-NNMi\components\alarmpoint\libs\alarmpoint-nnmi.jar`
- **Node Destination Directory**: `C:\Program Files\AlarmPointSystems\AlarmPoint\node\lib`
- **Web Server Destination Directory**: `C:\Program Files\AlarmPointSystems\AlarmPoint\webserver\webapps\cocoon\WEB-INF\lib`

Unix:

- **Source File**: `AP-HP-NNMi/components/alarmpoint/libs/alarmpoint-nnmi.jar`

- **Node Destination Directory**: `/opt/alarmpointsystems/alarmpoint/node/lib`
- **Web Server Destination Directory**: `/opt/alarmpointsystems/alarmpoint/webserver/webapps/cocoon/WEB-INF/lib`

Note: *If you have installed more than one web server, install the JAR file into the library folder for each one.*

2.3.4 Installing the Subscription File

To use the optional Subscription Panel, you must copy the JSP file into the AlarmPoint installation folder. If you have installed more than one web server, repeat the following steps for each one.

To install the JSP file:

1. On the AlarmPoint server, locate the web server installation folder:

   `\webserver\webapps\cocoon\alarmpoint\jsp\subscription`

2. Create a subfolder named `nnmi`.
3. Copy `NNMiSubscriptionForm.jsp` from `AP-HP-NNMi/components/alarmpoint/sub_panel` in the extracted integration archive into the new `nnmi` directory.
4. Restart the AlarmPoint Webserver.

3. Configuration

Before using the integration, you must configure AlarmPoint, the AlarmPoint Java Client, and HP Network Node Manager i-series. This section explains the configuration processes required for each product.

3.1 AlarmPoint

Configuring AlarmPoint requires the following steps:

- Import the AlarmPoint Script Package
- Configure the Web Services Connection
- Modify the default callout scripts
- Install voice files
- Define the Event Domain
- Set up a two-way Device
- Configure the Web Service Library
- Configure the Subscription Panel (optional)

Note: *The AlarmPoint Webserver must be running for the following steps.*

3.1.1 Importing the AlarmPoint Script Package

This step requires the AlarmPoint Developer IDE. For installation instructions, refer to the *AlarmPoint Developer's Guide & Scripting Reference*.

To import the AlarmPoint Script Package:

1. Launch the IDE, and configure the database connection (refer to the AlarmPoint Developer IDE Help or the *AlarmPoint Developer's Guide & Scripting Reference* for details)
2. Click **Workspace > Import**.
3. Select the `AP-HP-NNMi.aps` file extracted from the integration archive into the following folder:
   - `AP-HP-NNMi\components\alarmpoint\scripts\`
4. In the Import dialog box, click **Open**, and then click **OK**.
5. Right-click the **HP Network Node Manager i-series (Business)** folder.
6. In the Validation dialog box, select **Validate**, and then click **Close**.
7. Right-click the **HP Network Node Manager i-series (Business)** folder, and then click **Check In**.
8. In the Check In dialog box, click **Create**.
9. In the Script Package dialog box, click **Remove**, and then click **Close**.
10. Close the IDE.

3.1.2 Configuring the Web Services Connection

AlarmPoint for HP NNMi uses the following default settings to connect to NNMi via Web Services:

NNMi Web Services NodeBean (initial PROCESS script):

```
http://localhost:8004/IncidentBeanService/IncidentBean
```

NNMi Web Services NodeBean (Subscription Panel):

```
http://localhost:8004/NodeBeanService/NodeBean
```

Web Services User:

```
webservices
```

Web Services Password:

```
nnm
```

If your NNMi deployment is running on a port other than 8004 (or if you require a different URL or User/Password combination), you must change the settings in the custom Subscription panel and in the initial PROCESS script.

Identifying your NNMi port

You can determine whether the default port setting of 8004 is correct for your NNMi installation by checking the port information contained in the NNMi port configuration file, located in the following folder:

```
${NNM_DATA_DIR}\"shared\nnm\conf\nnm.ports.properties"
```

Note: *The default NNM_DATA_DIR folder (for Windows installations) is C:\Documents and Settings\All Users\Application Data\HP\HP BTO Software.*

If your installation is running on a port other than 8004, use the following steps to change the port number.

To change the Web Services port in the initial PROCESS script:

1. Launch the AlarmPoint Developer IDE.
2. Check out the **HP Network Node Manager i-series (Business)** script package.
3. In the initial PROCESS script, locate the following line:

   ```
   $main.nnmi_incident_url = "http://localhost:8004/IncidentBeanService/IncidentBean"
   ```

4. Edit the port setting to the correct value and save your changes.
5. Check in the script package and close the Developer IDE.

Note: *For instructions on how to configure the Subscription panel's Web Services connection, see "Configuring the Subscription JSP" on page 16.*

3.1.3 Modify the default callout scripts

The default callout scripts are not configured to use web services to annotate the ticket, and must be updated to inject messages back to the NNMi Integration Action Scripts.

To annotate an NNMi incident from the Callout scripts:

1. Launch the AlarmPoint Developer IDE.
2. Check out the Production Callout Script Package.
3. Open both the callout CONTACT script and the authenticate INTERACTION scripts.
4. Add the following lines at the end of both scripts:

```
SendAnnotationMessage:
  IF( $initiatingEvent.agent_client_id == "hp_nnmi" )
    @session::respondToNotification( $notId, "callout_annotate " & $message_note )
  ENDIF
RETURN
```

Note: *If the SendAnnotationMessage is already in the callout scripts due to another integration, update it with logic to handle when the agent_client_id equals "hp_nnmi".*

5. Locate a section of code that sends an ExternalServiceMessage.
6. Add the following lines after the message is added to the ServiceMessage object, and after the send method is executed on the object:

```
$message_note = $connectionEventMessage.message
GOSUB SendAnnotationMessage
```

7. Repeat the previous two steps for each section that sends an ExternalServiceMessage.

Note: *There should be a total of 10 code segments that need to be updated; eight in the callout script, and two in the authenticate script.*

Example:

The following is an example of an enhanced segment of code:

```
$connectionEventMessage.message = "Callout to " & $targetName & " successful (" &
$result & ")."
@connectionEventMessage::send()
$message_note = $connectionEventMessage.message
GOSUB SendAnnotationMessage
```

3.1.4 Installing voice files

These files must be installed into an AlarmPoint deployment running a Voice Device Engine. For more information, refer to the *AlarmPoint Installation and Administration Guide*.

To install the voice files:

1. Copy all of the files in the `AP-HP-NNMi\components\alarmpoint\vox\english` folder from the extracted integration archive to the following node installs folder:

```
node\phone-engine\datastore\domains\common\recordings\english\phrases
```

Note: *This integration provides a complete set of English voice files.*

3.1.5 Defining an Event Domain

The AlarmPoint webserver must be running to perform this portion of the integration.

To define an Event Domain:

1. Login to AlarmPoint as a Company Administrator, and click the **Developer** tab.
2. On the Event Domains page, click **Add New**.
3. Enter the following information into the form:
   - **Name**: hp_nnmi
   - **Description**: HP Network Node Manager i-series Integration
   - **Script Package**: HP Network Node Manager i-series
4. Click **Save**.

Note: *It is strongly recommended that you use the Event Domain Name specified above. For the integration to be successful, the Event Domain name must match the Client ID of the AlarmPoint Java Client.*

3.1.6 Setting up a two-way Device

You can use any User or Group as a target for notifications, such as the default demonstration User named “Bob Smith”. Follow the steps below to ensure that this User exists and has a virtual text phone Device:

To set up a two-way Device:

1. In AlarmPoint, click the **Users** tab.
2. On the Find Users page, click **S**.
3. In the list of returned Users, click **Smith, Bob**.
4. On the Details for Bob Smith page, in the Common Tasks pane, click **User Devices**.
5. Verify that a virtual text phone device exists.
6. Click **Reorder**, and set the virtual text phone to be the first Device in the list.
7. Click **Save**.

Note: *If this user is missing, create a user with a User ID of bsmith and add a virtual text phone Device for the User. For more information and instructions on how to perform these tasks, refer to the* AlarmPoint User Guide.

3.1.7 Initializing the Web Service Library

The AlarmPoint Integration uses Web Services to post responses to NNMi and to populate the Subscriptions Source Node Name list. The AlarmPoint Webserver automatically picks up the installed library if its installed to the correct folder (for instructions, see “Installing the Web Services Library” on page 6).

To enable Web Services for the AlarmPoint node, you must modify the node configuration script so responses may be posted to the NNMi server.

To configure the node to initialize the Web Service Library on Windows:

1. Open the `node-start.conf` script found in `C:\Program Files\AlarmPointSystems\AlarmPoint\common\`.
2. Find the following classpath configuration section:

```
# Class path info
-classpath
```

3. Add the following to the end of the classpath within the quotations:

```
;C:\Program Files\AlarmPointSystems\AlarmPoint\node\lib\alarmpoint-nnmi.jar
```

4. Restart the AlarmPoint Node Service.

To configure the node to initialize the Web Service Library on Unix:

1. Open the `node.sh` script found in `/opt/alarmpointsystems/alarmpoint`.
2. Find the declaration of the CLASSPATH variable.
3. Add the following to the end of the classpath, within the quotations:

```
:/opt/alarmpointsystems/alarmpoint/node/lib/alarmpoint-nnmi.jar
```

4. Restart the AlarmPoint Node Daemon.

Your notification responses will now be posted to the NNMi server using Web Services.

Note: *For AlarmPoint responses to update the NNMi Incident using Web Services, the nnmi_incident_url must be a valid URL for the NNMi server's IncidentBean; for more information, see "Global Configuration Variables" on page 39.*

3.1.8 Configuring the Subscription Panel

To allow Users to subscribe to specific criteria on injected Events, you must configure the Subscription panel. Configuring the Subscription Panel requires the following steps:

- Define the Event Domain predicates
- Define a Subscription Domain
- Create a Subscription
- Create a fail-safe Group

Note: *Before you can configure the custom Subscription Panel, the* `NNMiSubscriptionForm.jsp` *file must be installed as described in "Installing the Subscription File" on page 7.*

3.1.8.1 Defining Event Domain predicates

For the default Subscription panel provided with the integration, the following Event Domain predicates must be defined (case sensitive):

- SEVERITY
- CATEGORY

- NAME
- SOURCENODENAME
- SOURCEOBJECTNAME
- PRIORITY
- FAMILY
- NATURE

Note: *You can also use the following steps to add other predicates that you consider important and which you plan to add to the integration as explained in "Adding data elements" on page 32.*

To define the Event Domain predicates:

1. In AlarmPoint, click the **Developer** tab.
2. On the Event Domains page, click **hp_nnmi**.
3. On the Event Domain Details page, click **Add New**.
4. Add the following predicates to the Event Domain:

Predicate	Type	Important	Values	Description
SEVERITY	List	Yes	Manually entered	Severity is a list predicate containing some or all of the following values (case sensitive): • Critical • Major • Minor • Normal • Warning. The items listed for Severity should be specifically chosen to match the severity of the Events forwarded from NNMi. Exclude any severities that will not be submitted for notification. This predicate corresponds to the $severity variable in NNMi.

Predicate	Type	Important	Values	Description
CATEGORY	List	No	Manually entered	Category describes the type of incident; allowed values are: com.hp.nms.incident.category.Fault com.hp.nms.incident.category.Status com.hp.nms.incident.category.Config com.hp.nms.incident.category.Accounting com.hp.nms.incident.category.Performance com.hp.nms.incident.category.Security com.hp.nms.incident.category.Alert NNMi will generate only the values listed above. Exclude from your list any categories that will not be submitted for notification. This predicate corresponds to the $category variable in NNMi.
NAME	List	Yes	Manually entered	Name is the incident type of the event. You can have any number of incident names listed here, but they should be only those incident types you have configured to inject messages into AlarmPoint. Some example incident types are: • AddressNotResponding • InterfaceDisabled • InterfaceDown • NodeDown • NonSNMPNodeUnresponsive • WANEdgeRouterUnresponsive This predicate corresponds to the $name variable in NNMi.
SOURCENODENAME	List	Yes	Automatically generated	Source Node Name is the name of the Node that is the source of the incident. Leave the values of this list blank as it will be populated through a Web Services Call to NNMi. This predicate corresponds to the $sourceNodeName variable in NNMi. By default the SOURCENODENAME predicate value list is populated through a Web Service Call to NNMi, it may be populated with pre-defined values as well, for more information see “Configuration Variable Reference” on page 37.

Predicate	Type	Important	Values	Description
SOURCEOBJECTNAME	Text	Yes	Manually entered	Source Object Name is the name of the Object that generated the incident. The Object can be determined through a combination of the Object Name and Family. As this is a text field, you can use any number of filters on the results.
PRIORITY	List	No	Manually entered	Priority is how important fixing the incident is to Users. This is in contrast to Severity, the level of which is automatically determined by NNMi. Note that AlarmPoint allows the priority to be altered by notificaiton recipients. Valid values for Priority are: • com.hp.nms.incident.priority.None • com.hp.nms.incident.priority.Low • com.hp.nms.incident.priority.Medium • com.hp.nms.incident.priority.High • com.hp.nms.incident.priority.Top. This predicate corresponds to the $priority variable in NNMi.
FAMILY	List	Yes	Manually entered	Family is the type of object that generated the incident. Valid values for Family are: • com.hp.nms.incident.family.Address • com.hp.nms.incident.family.Interface • com.hp.nms.incident.family.Node • com.hp.nms.incident.family.OSPF, • com.hp.nms.incident.family.HSRP • com.hp.nms.incident.family.AggregatePort • com.hp.nms.incident.family.Board • com.hp.nms.incident.family.Connection • com.hp.nms.incident.family.Correlation This predicate corresponds to the $family variable in NNMi.
NATURE	List	No	Manually entered	Nature is a list predicate that describes how NNMi views the incident. Valid values for Nature are: • ROOTCAUSE • SECONDARYROOTCAUSE • SYMPTOM • USERROOTCAUSE. This predicate corresponds to the $nature variable in NNMi.

Note: *For more information about the predicates and their corresponding NNMi variables, see "Configuring NNMi Incident Types for Automatic AlarmPoint Notifications" on page 21.*

3.1.8.2 Defining a Subscription Domain

The Subscription Domain is the reference point to the optional Subscription panel and provides a means to control certain aspects of it. You must create a Subscription Domain before you can create Subscriptions with the new panel.

To create a Subscription Domain:

1. On the Developer tab, in the Developer menu, click **Add Subscription Domain**.
2. In the **Event Domain** drop-down list, select **hp_nnmi**, and then click **Continue**.
3. On the Subscription Domain details page, in the **Name** field, type `NNMi`.
4. In the **Custom Page URL** field, enter the following path:

   ```
   jsp/subscription/nnmi/NNMiSubscriptionForm.jsp
   ```

5. Click **Continue**.
6. On the Select Appropriate Response Choices page, specify the available responses for this Subscription, and then click **Continue**.
   - By default, the scripts support the following response choices: "Acknowledge", "Set Priority <level>" (where <level> equals "Top", "High", "Medium", or "Low"), and "Ignore". To enable two-way communications for Subscriptions, define all six response choices on the Select Appropriate Response Choices page. If you require only one-way, informational notifications, do not specify any response choices.

Note: *By default, Subscriptions are FYI (informational-only notifications). To enable two-way subscription notifications, set the $subscription_fyi variable to false in the configuration block of the initial PROCESS script.*

7. On the Select Appropriate Predicates page, add all of the predicates to the Applied Predicates list, and then click **Continue**.
8. On the Select Roles page, specify the Roles you want to be able to create Subscriptions on the Domain, and then click **Save**.

Note: *For more information about working with Event and Subscription Domains, see the AlarmPoint Installation and Administration Guide.*

3.1.8.3 Configuring the Subscription JSP

This integration is packaged with an optional Subscription panel which reads the Source Node Name list values from NNMi through Web Services. This feature allows Administrators to change the source of the content supplied for these lists from Web Service Calls to predefined predicate value lists. To configure the Subscription panel in a demo mode, using pre-defined predicate list values, you must modify the Subscription JSP.

To manually populate the predicate list values:

1. Open the `NNMiSubscriptionForm.jsp` found in the `\webservers\webapps\cocoon\alarmpoint\jsp\subscription\nnmi` folder on the AlarmPoint Webserver install.
2. Set the Boolean variable `QUERY_SOURCENODE_PREDICATE_VALUES` found on line 28 to `false`.

3. Save and close the `NNMiSubscriptionForm.jsp` file.
4. In AlarmPoint, click the **Developer** tab.
5. On the Event Domains page, click **hp_nnmi**.
6. On the Event Domain Details page, click **SOURCENODENAME** in the Predicates list.
7. Add to the predicate list values.

The SOURCENODENAME list on the Subscription will now be populated with the predefined list values instead of the Web Service Call results.

Note: *Changing Subscriptions by adding or removing Event Domain predicates may cause existing Subscriptions to fail. For more information about working with Event and Subscription Domains, see the* AlarmPoint Installation and Administration Guide.

If you want to populate the predicate values lists from NNMi through Web Service Calls rather than the pre-defined predicate list values, you must configure the connection properties within the Subscription JSP.

To configure the Subscription JSP to connect to NNMi through Web Services:

1. Open the `NNMiSubscriptionForm.jsp` found in the `\webservers\webapps\cocoon\alarmpoint\jsp\subscription\nnmi` folder on the AlarmPoint Webserver install.
2. Within the Subscription JSP, find the following section:

```
final String NNM_NODE_SERVICE_WS_URL = "http://localhost:8080/NodeBeanService/
NodeBean";
final String NNM_WS_USER = "webservices";
final String NNM_WS_PASSWORD = "nnm";
```

3. Replace the value within quotes for each parameter as described in the following table:

Parameter	Value
NNM_NODE_SERVICE_WS_URL	The URL for the NNMi Web Services NodeBean.
NNM_WS_USER	User name of the NNMi Web Service Client.
NNM_WS_PASSWORD	Password of the NNMi Web Service Client.

4. Save and Close the JSP.

Note: *The NNM_WS_USER and NNM_WS_PASSWORD must match the User configured in "Create a Web Services Client" on page 20.*

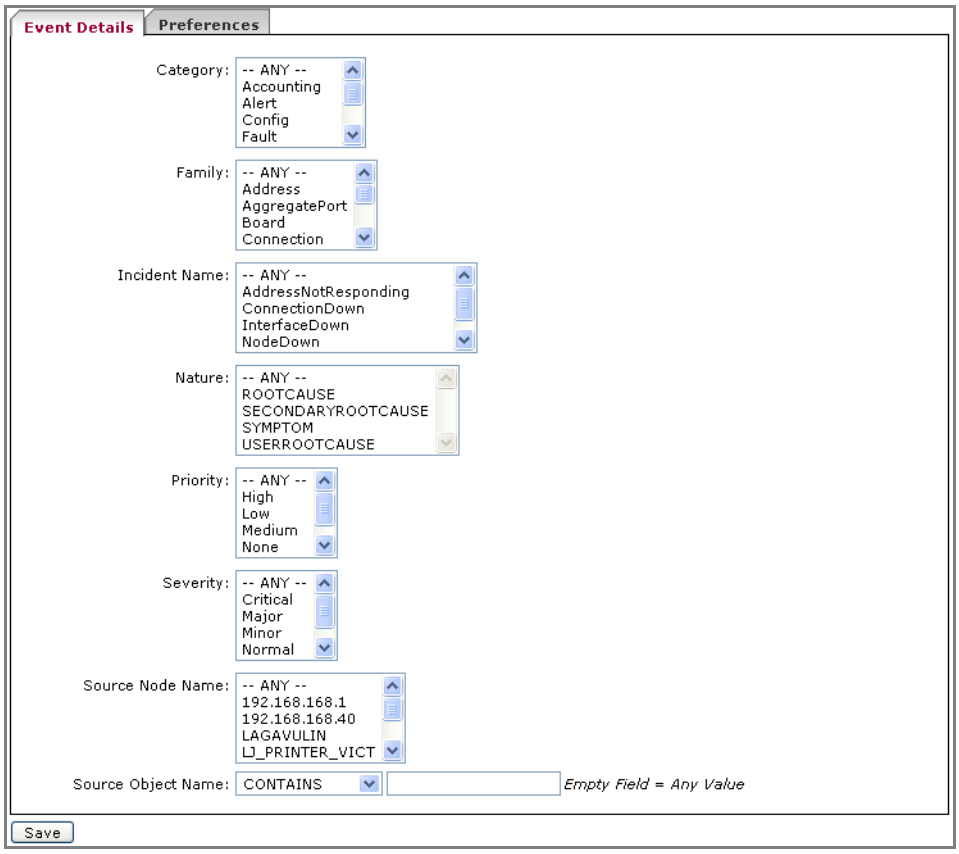
3.1.8.4 Creating a Subscription

You can now use the Subscription panel to subscribe to NNMi Events of specific criteria, such as those of "Critical" Severity.

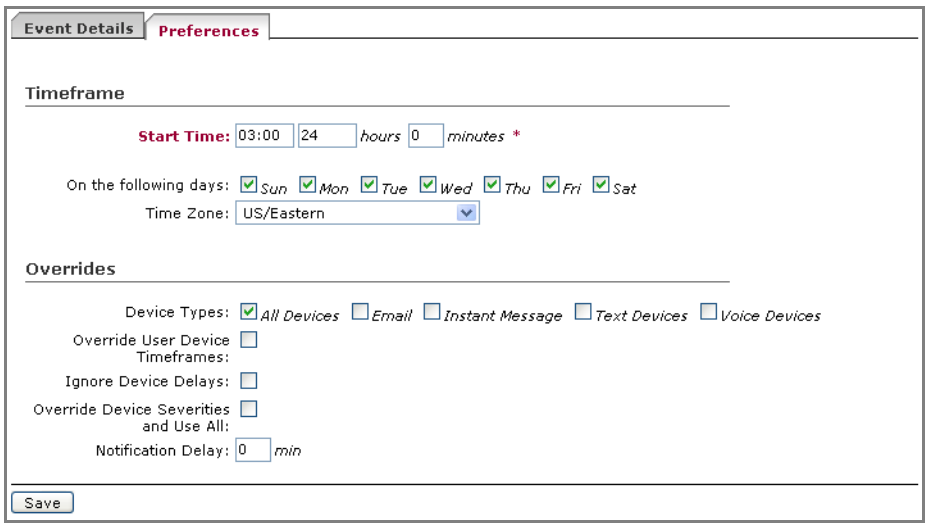
To create a Subscription:

1. Click the **Alerts** tab, and then click **My Subscribed Alerts**.
2. On the My Subscribed Alerts page, in the **Subscription Domain** drop-down list, select **NNMi**.

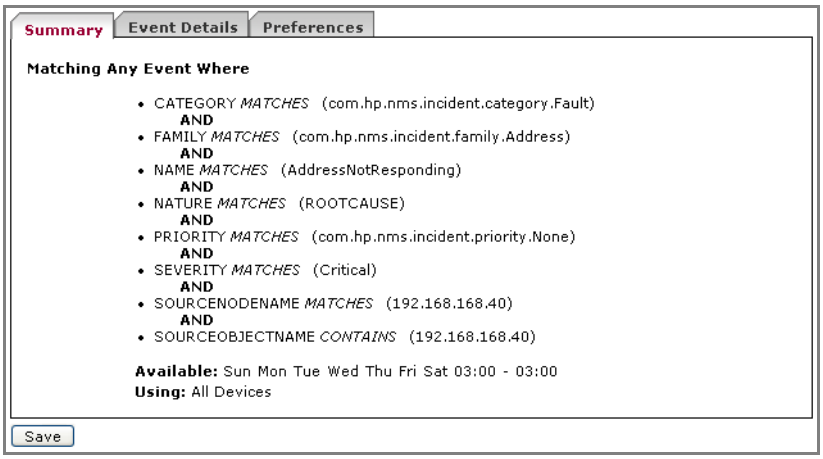
3. Click the **Add New** link above the Self-made Subscriptions table.
4. On the Subscription Details page, enter a name for the Subscription and specify the Subscription criteria using the Event Details and Preferences tabs.
    - The Event Details tab (Ctrl-click to select more than one value):



    - The Preferences tab (defines the Timeframe and Overrides applied to events for Subscription notifications):



5. When you are satisfied with the criteria, click **Save** to create the Subscription.
    - You can review the Subscription details at any time on the Summary tab:



3.1.8.5 Creating a Fail-Safe Group

If a notification is submitted to AlarmPoint when the fail-safe functionality is enabled, and if it matches the necessary circumstances, AlarmPoint sends the notification to the fail-safe recipient. The fail-safe recipient is typically a Group, but can be configured as a User.

To create a fail-safe Group:

1. In AlarmPoint, click the **Groups** tab.
2. Create a new Group named **NNMi FailSafe**, with at least one User as a Team member to receive notifications.

For more information about creating Groups and Teams, see the *AlarmPoint User Guide*.

Note: *If you want to use a pre-existing group or a different group name, modify the value for the* `$fail_safe_group` *variable defined in the initial PROCESS script in the AlarmPoint Action Scripts. You can also eliminate notifying any failsafe group by setting* `$fail_safe` *to* `disabled`.

3.1.8.6 Excluding Subscriptions

If you do not want to use the Subscription functionality, you can remove it from the action scripts by setting the `$enable_subs` variable to `false` in the configuration block of the initial PROCESS script.

3.2 AlarmPoint Java Client

To configure the AlarmPoint Java Client, you must edit the `APAgent.xml` file.

3.2.1 Editing APAgent.xml

You must add the following line to the <alarmpoint-agent-client> section of the `APAgent.xml` file:

```
<alarmpoint-agent-client id="hp_nnmi" filename="integrations/hp_nnmi.xml"/>
```

The `APAgent.xml` file is located at:

Windows:

```
c:\APAgent\etc\APAgent.xml
```

Unix:

```
/opt/alarmpointsystems/APAgent/etc/APAgent.xml
```

Note: *Restart the APAgent after editing the* `APAgent.xml` *file.*

3.3 HP Network Node Manager i-series

Configuring NNMi for the integration requires the following steps:

- Create a Web Services Client
- Configure NNMi Incident Types for automatic AlarmPoint notifications

3.3.1 Create a Web Services Client

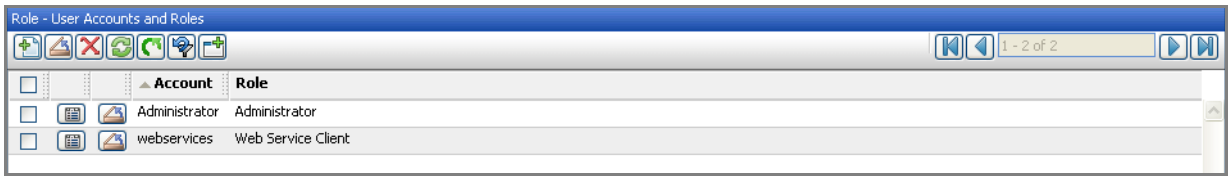
Configuring a Web Services Client allows notification responses to update the NNMi incidents appropriately.

To create a Web Services Client:

1. Launch the NNMi Web Console, and log in as an Administrator.
2. Under the Configuration Workspace, click **User Accounts and Roles**.
3. On the Role – User Accounts and Roles page, click **New**.
4. On the Role page, under Basics, select **New** in the **Account** drop-down list.
5. On the User Account page, specify the **Name** and **Password** for the Web Services Client User:

Note: *By default, the user name and password configured within the AlarmPoint Action Scripts is* `webservices/nnm`*. If you want to use a different name or password, you must update the Configuration Variables in the initial PROCESS Action Script. For more information, see "Configuring the Web Services Connection" on page 9.*

6. Click **Save and Close**.
   - The "webservices" user is now specified in the Account field on the Role page.
7. In the **Role** drop-down list, select Web Service Client.
8. Click **Save and Close**.
   - The Web Service Client will now allow AlarmPoint responses to update NNMi incidents using Web Service Calls. The webservices user is listed on the User Accounts and Roles page:



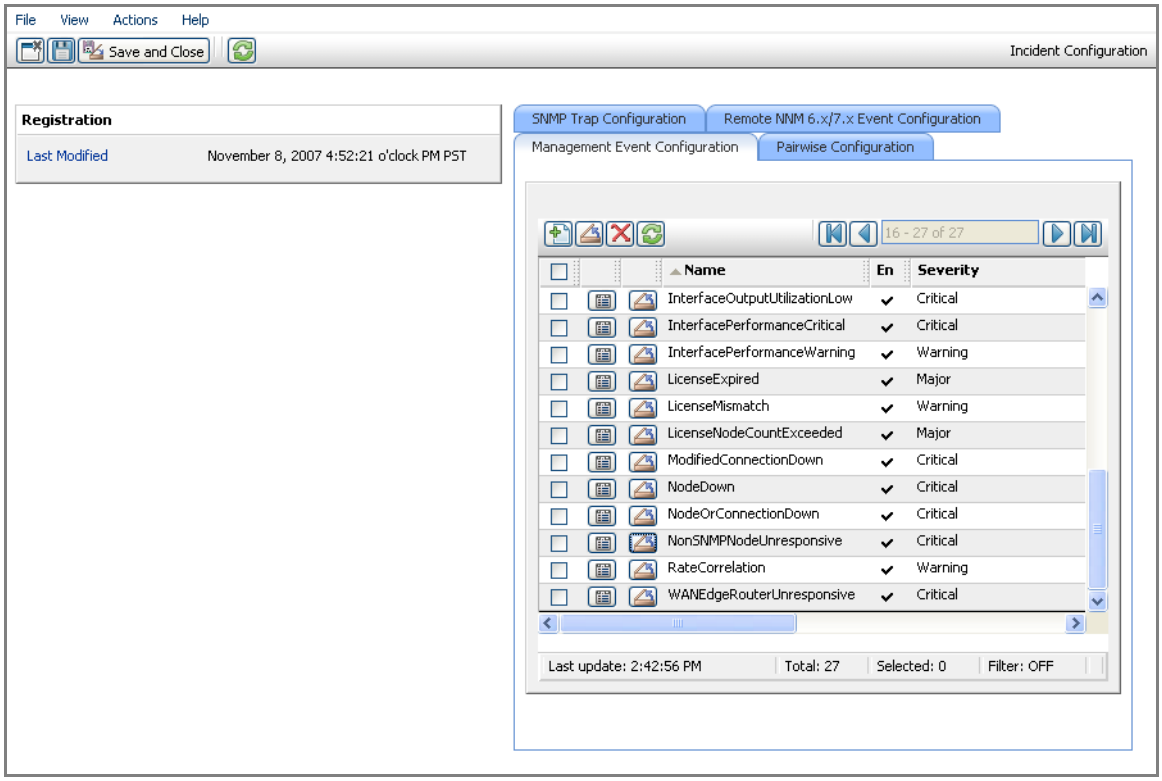
3.3.2 Configuring NNMi Incident Types for Automatic AlarmPoint Notifications

To trigger notifications within AlarmPoint, NNMi must be configured to make a command line call passing the incident parameters into AlarmPoint through the APClient utility. For specific NNMi incident types to trigger Notifications within AlarmPoint, their Action Configuration must be enabled to make a command line call to the APClient utility.

The APClient call must have the desired Incident Parameters passed through the command line for them to be available within AlarmPoint.

To configure an Incident Type for automatic notification:

1. Launch the NNMi Web Console, and log in as an Administratior.
2. Under the Configuration Workspace, select **Incident Configuration**.
   - The Incident Configuration page appears, displaying several types of events:



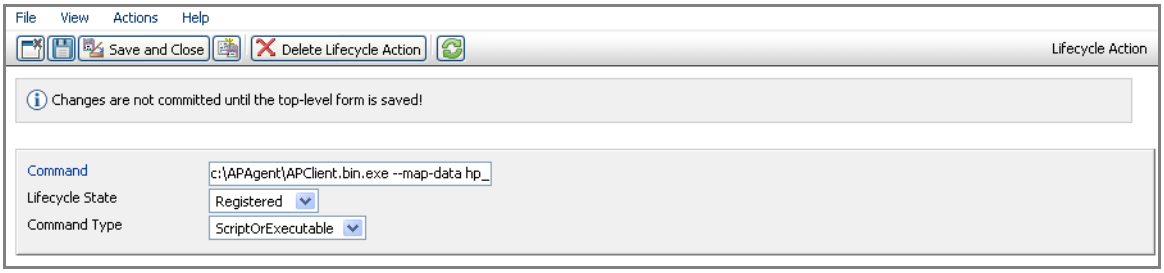
   - Depending on the business behavior desired, use the following steps to configure each type of Event requiring AlarmPoint notifications. It is recommended that you forward only Critical Events to AlarmPoint for notification. The following steps use the **NonSNMPNodeUnresponsive** Management Event as an example of the configuration process.

3. Open the Event Type you want to configure.
   - To open the NonSNMPNodeUnresponsive Event Type, click the **Management Event Configuration** tab, and then click the **Open** icon beside the NonSNMPNodeUnresponsive Event Type.
4. On the Events Configuration page, ensure the **Enable** check box is selected, and then click the **Action Configuration** tab.
5. On the Action Configuration tab, select the **Enable** check box.
6. Within the Lifecycle Actions table, click the **New** button to add a new Lifecycle Action.
7. On the LifeCycle Action page, enter the following information into the fields:

Field	Detail

Command	**Windows:**
	`c:\APAgent\APClient.bin.exe --map-data hp_nnmi bsmith $category $severity $family $lifecycleState $name $nature $priority $sourceNodeName $sourceObjectName $uuid no`
	Unix:
	`/opt/alarmpointsystems/APAgent/APClient.bin --map-data hp_nnmi bsmith $category $severity $family $lifecycleState $name $nature $priority $sourceNodeName $sourceObjectName $uuid no`

Lifecycle State	Registered
Command Type	ScriptOrExecutable

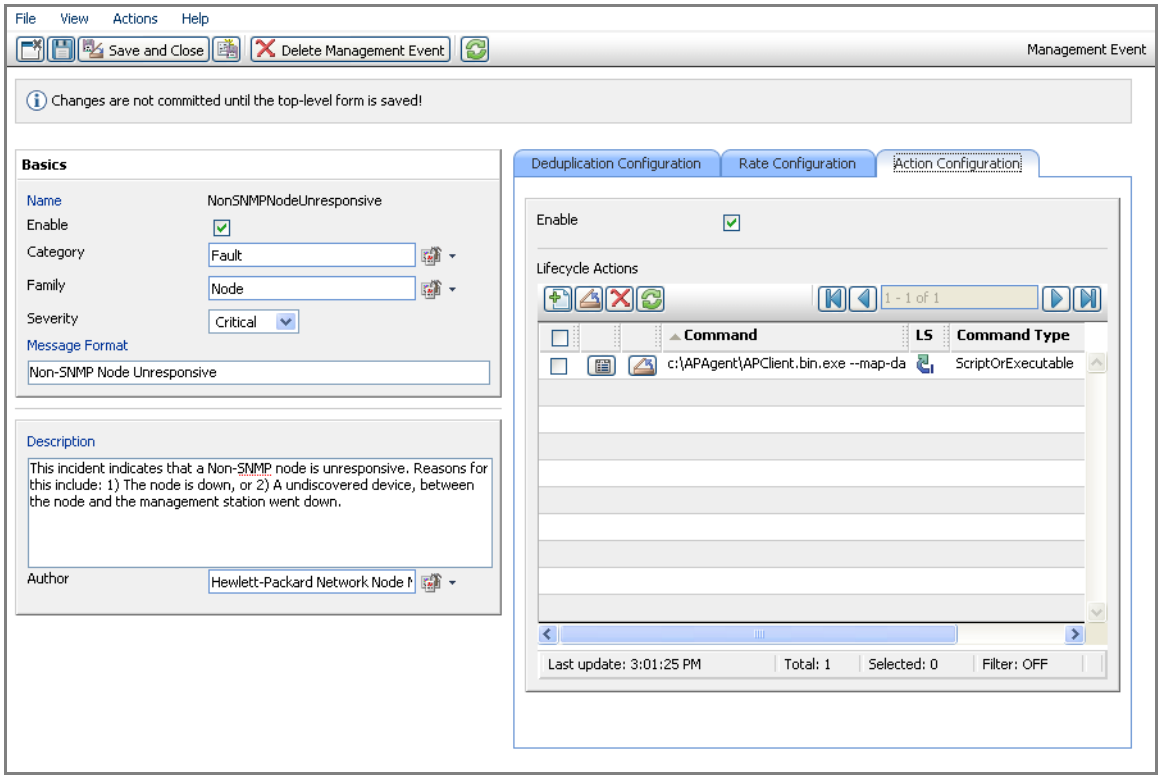


- You can customize the command to inject different event parameters. The following table identifies the default values suggested above:

Variable	Description
$category	Describes the type of incident
$severity	Importance of the incident as specified by NNMi
$family	Type of object that created the incident
$lifecycleState	Current Lifecycle State of the incident
$name	Name of the incident
$nature	Describes how NNMi views the incident
$priority	Importance of fixing the incident as specified by users

Variable	Description
$sourceNodeName	Name of the node where the incident originated
$sourceObjectName	Name of the object that generated the incident
$uuid	Unique identifier of the incident, used by AlarmPoint as the Incident ID
no	Indicates if the event should be informational-only, or allow for response updates (`yes` = FYI and `no` = two-way)

8. Click **Save and Close**.
   - NNMi displays the Action Configuration for the Management Event:



9. Click **Save and Close** on the selected Event Type page, and then click **Save and Close** again on the Incident Configuration page to commit your changes.

The selected Event Type will now trigger incidents within NNMi and will inject a message through the command line for automatic AlarmPoint notification for each incident triggered. The following is an example of the injected command:

```
APClient --map-data hp_nnmi bsmith com.nms.incident.category.Fault Critical
com.hp.nms.incident.family.Address Registered NonSNMPNodeUnresponsive ROOTCAUSE
com.hp.nms.incident.priority.None 192.168.168.40 192.168.168.40 a1810618-8308-
42e9-8f2d-6758d739c2fd no
```

Note that the configuration of the alerts in NNMi is the responsibility of the system integrator. The recommended implementation pattern is to send in alerts only on items that are marked as “Critical”, and to target a group that is responsible for addressing the problem. Membership of the group can then be maintained in AlarmPoint. Individuals that want to receive information about the alert can create a Subscription.

Note: *The recipient ID specified in the "Command" (bsmith) is not automatically mapped within AlarmPoint. Ensure that the recipient target specified in NNMi matches a Group or User ID within AlarmPoint.*

3.4 Software Component Validation

It is recommended that the applications be run in the following order:

- HP Network Node Manager i-series
- AlarmPoint Application and Notification Server Nodes
- AlarmPoint Agent

Consult the respective user manuals for details on starting these applications.

3.4.1 AlarmPoint Agent

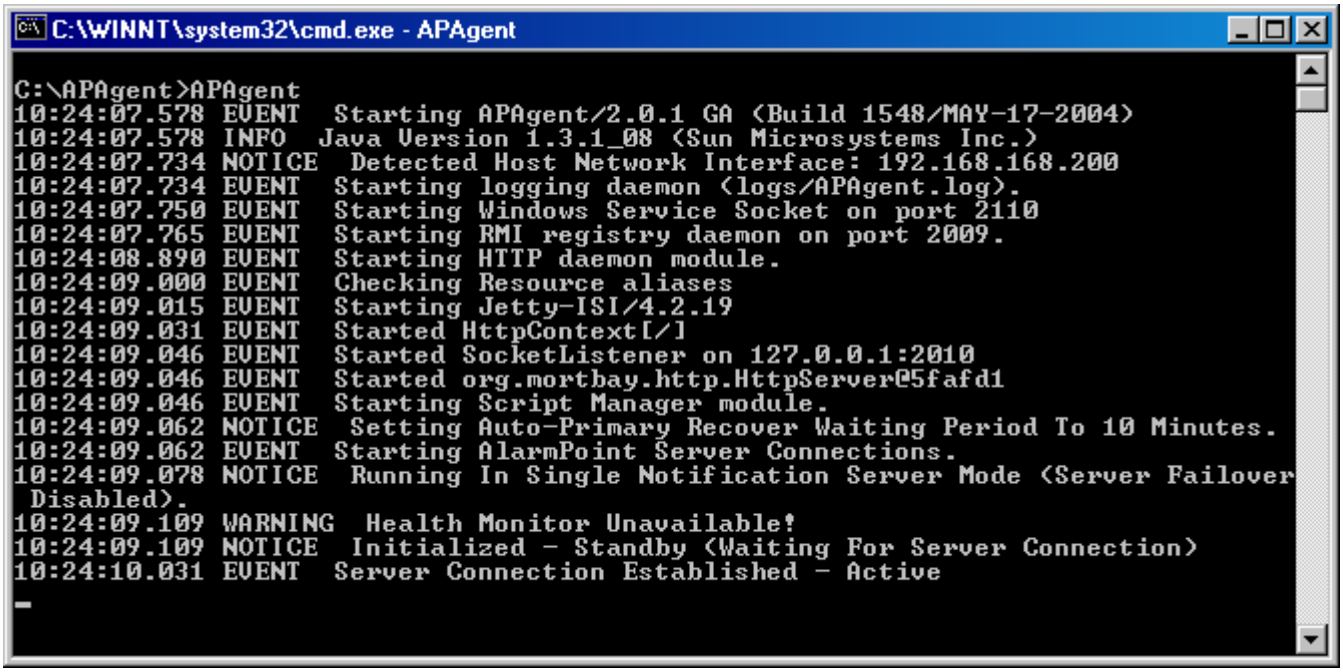
This section shows how to verify that the AlarmPoint Agent can communicate with AlarmPoint. For both Windows and Unix, the last line on the console will indicate a successful connection. If the connection cannot be established, consult the user manual for more information and troubleshooting tips.

3.4.1.1 Testing on Windows

Provided that the AlarmPoint Java Client was not installed as a Windows Service, after installation of NNMi is complete, open a command line window, navigate to the AlarmPoint Agent home directory, and run the following command::

```
APAgent
```

This starts the AlarmPoint Agent. The output should resemble the following figure:



If critical problems arise (file errors, communication difficulties, insufficient memory, etc.), the Health Monitor may be used to send an e-mail. However, the Health Monitor is optional.

Note: *In a production environment, the AlarmPoint Java Client should be installed as a Windows Service. This helps ensure that if the system is restarted, the AlarmPoint Agent will run automatically.*

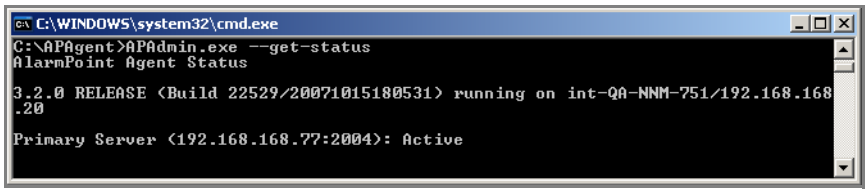
If the AlarmPoint Agent is running as a Windows Service, you can verify the installation using the following steps.

To verify the AlarmPoint Agent as a Windows Service:

1. Click **Control Panel > Administrative Tools > Services**.
2. In the Services window, locate the AlarmPoint Java Client entry and ensure that it is started.
3. Open a command line window, navigate to the AlarmPoint Agent home directory, and run the following command:

```
APAdmin --get-status
```

   - If the AlarmPoint Agent is properly configured, the output should resemble the following figure:

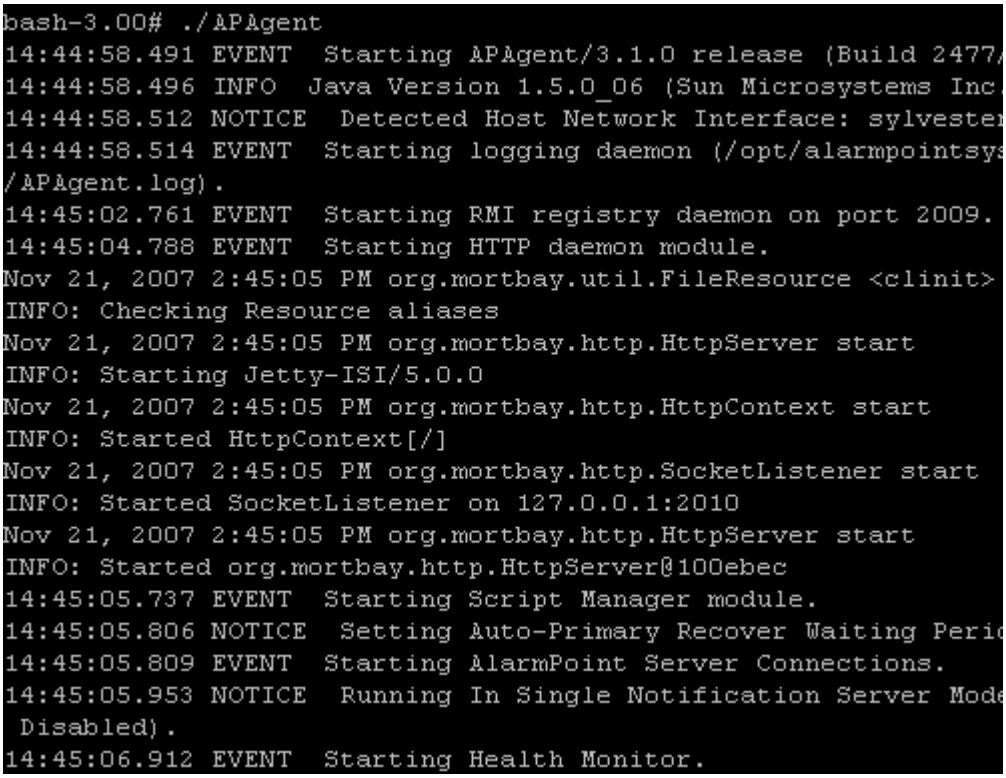


3.4.1.2 Testing on Unix

Open a new shell prompt, navigate to the AlarmPoint Agent home directory, and run the following command to verify connectivity:

```
./APAgent
```

This starts the AlarmPoint Agent. The output should resemble the following figure:


bash-3.00# ./APAgent
14:44:58.491 EVENT Starting APAgent/3.1.0 release (Build 2477
14:44:58.496 INFO Java Version 1.5.0_06 (Sun Microsystems Inc
14:44:58.512 NOTICE Detected Host Network Interface: sylveste
14:44:58.514 EVENT Starting logging daemon (/opt/alarmpointsy
/APAgent.log).
14:45:02.761 EVENT Starting RMI registry daemon on port 2009.
14:45:04.788 EVENT Starting HTTP daemon module.
Nov 21, 2007 2:45:05 PM org.mortbay.util.FileResource <clinit>
INFO: Checking Resource aliases
Nov 21, 2007 2:45:05 PM org.mortbay.http.HttpServer start
INFO: Starting Jetty-ISI/5.0.0
Nov 21, 2007 2:45:05 PM org.mortbay.http.HttpContext start
INFO: Started HttpContext[/]
Nov 21, 2007 2:45:05 PM org.mortbay.http.SocketListener start
INFO: Started SocketListener on 127.0.0.1:2010
Nov 21, 2007 2:45:05 PM org.mortbay.http.HttpServer start
INFO: Started org.mortbay.http.HttpServer@100ebec
14:45:05.737 EVENT Starting Script Manager module.
14:45:05.806 NOTICE Setting Auto-Primary Recover Waiting Peri
14:45:05.809 EVENT Starting AlarmPoint Server Connections.
14:45:05.953 NOTICE Running In Single Notification Server Mod
Disabled).
14:45:06.912 EVENT Starting Health Monitor.

4. Software Component Integration

4.1 Triggering a Notification

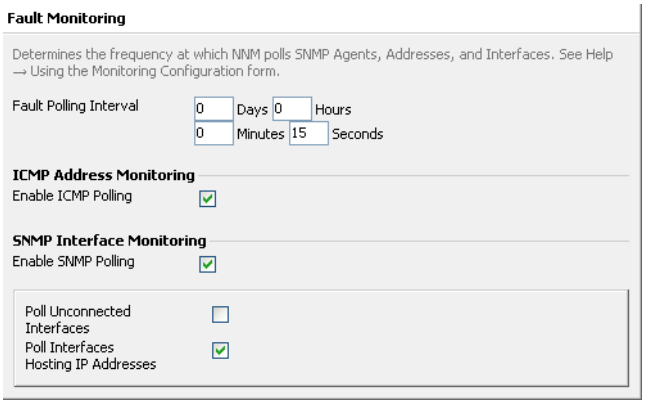
The following example shows resolution of a network outage on a monitored LAN.

4.1.1 Increase the Polling Frequency

The following section describes how the fault polling interval can be decreased to speed up the demonstration.

To adjust the fault polling interval:

1. If it is not already running, launch HP Network Node Manager i-series.
2. Log in to the NNMi Web Console as an Administrator.
3. Select the **Configuration Workspace**.
4. Open **Monitoring Configuration**.
5. In the Fault Monitoring dialog box, set the **Fault Polling Interval** to **15 seconds**:

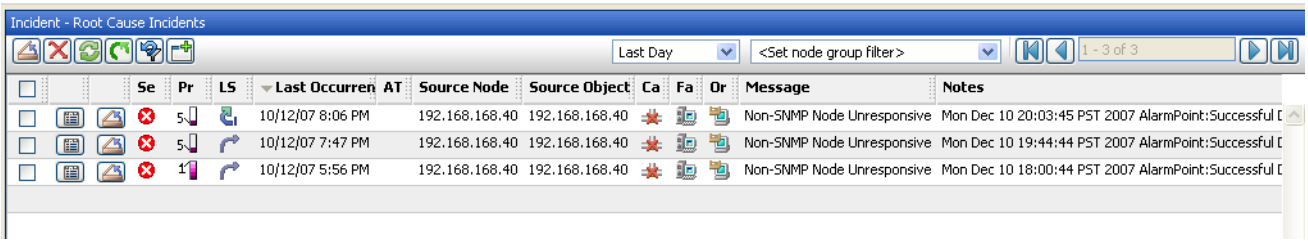


6. Click the **Save and Close** button.

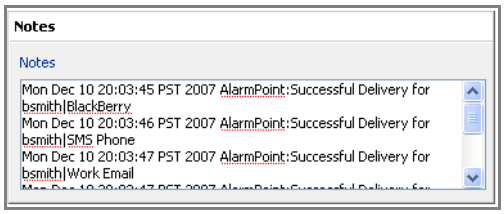
4.1.2 Disconnect a Computer from the LAN

If NNMi is monitoring a LAN, one of the easiest ways to trigger a notification is to interrupt the communication between NNMi and one of the computers on the LAN. The following steps describe how to do this and what to expect.

1. Physically disconnect a computer from the local area network (using a computer other than the AlarmPoint or NNMi servers).
2. When the computer goes offline, an incident will be triggered within NNMi and can be viewed in the Incidents workspace under Root Cause Incidents (or another incident category depending on the trigger).
    - The Notes entry for the open incident indicates that this event has succesfully notified an AlarmPoint User:



3. To display the full Notes for an incident, click the **Open Incident** button to open the incident, and view the Notes area:



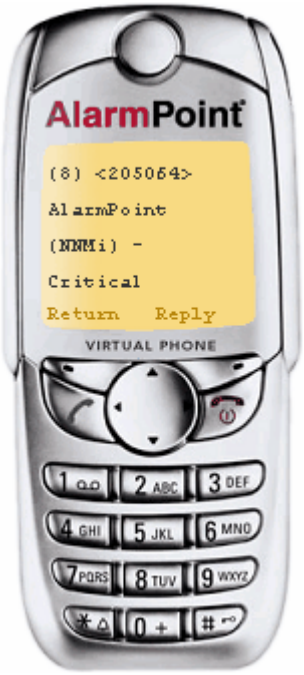
4. The target's specified contact type will receive a message corresponding to the notification, as shown in the following section.

4.2 Responding to a Notification

This section describes how to respond to a notification using the default User's virtual text phone.

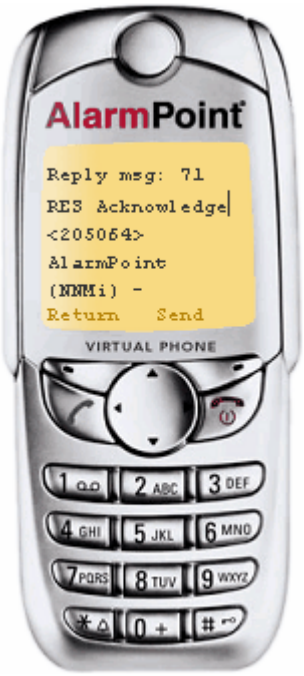
1. When a notification arrives for the default user, the virtual text phone appears and indicates the number of notifications that have been received:

2. To see the first notification, click **Select**:



3. Scroll down using the arrow buttons to view the details and the list of possible responses:

4. Click **Reply**, and then type **RES Acknowledge**:



5. Click **Send**, and AlarmPoint will send the acknowledgement of the Event to NNMi.

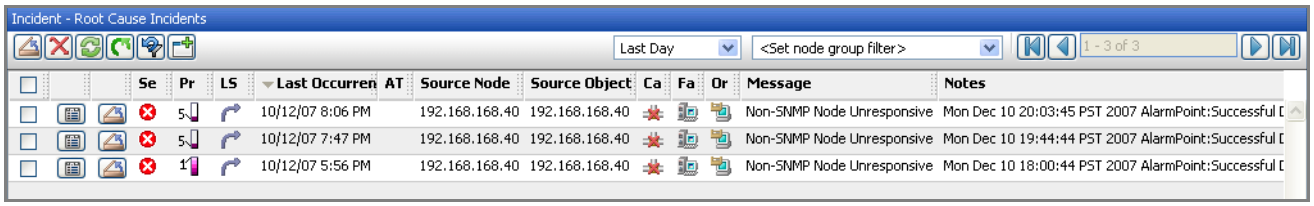
Note: *For more about using or changing the available response choices, see "Response choices" on page 32.*

4.3 Viewing Notification Results

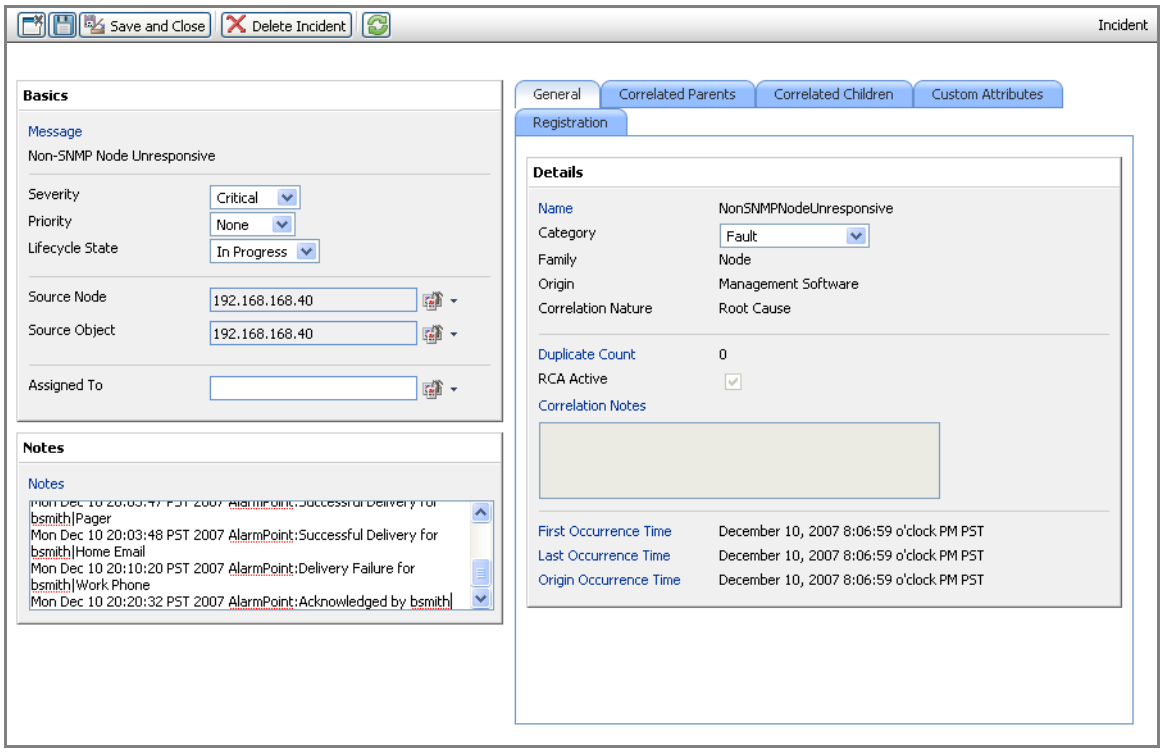
In the Root Cause Incidents table, the In Progress arrow indicates that the incident has been acknowledged, and a message will be logged within the Notes field indicating who took responsibility.

To view the notification results:

1. Open the NNMi Web Console.
2. In the Incident Workspace, under Root Cause Incidents, locate the incident used for testing notifications..
   - The Incidents Life Cycle State has changed to **In Progress**, indicating that the incident was acknowledged from AlarmPoint:



3. To display the acknowledged incident's details, click the **Open** button.
   - The Notes field indicates that the incident was acknowledged by bsmith:


Save and Close
Delete Incident
Incident
Basics
Message
Non-SNMP Node Unresponsive
Severity
Critical
Priority
None
Lifecycle State
In Progress
Source Node
192.168.168.40
Source Object
192.168.168.40
Assigned To
Notes
Notes
bsmith|Pager
Mon Dec 10 20:03:48 PST 2007 AlarmPoint:Successful Delivery for bsmith|Home Email
Mon Dec 10 20:10:20 PST 2007 AlarmPoint:Delivery Failure for bsmith|Work Phone
Mon Dec 10 20:20:32 PST 2007 AlarmPoint:Acknowledged by bsmith
General
Correlated Parents
Correlated Children
Custom Attributes
Registration
Details
Name
NonSNMPNodeUnresponsive
Category
Fault
Family
Node
Origin
Management Software
Correlation Nature
Root Cause
Duplicate Count
0
RCA Active
Correlation Notes
First Occurrence Time
December 10, 2007 8:06:59 o'clock PM PST
Last Occurrence Time
December 10, 2007 8:06:59 o'clock PM PST
Origin Occurrence Time
December 10, 2007 8:06:59 o'clock PM PST

5. Optimizing and Extending the Integration

This section describes some of the available methods you can use to optimize or extend the AlarmPoint for HP Network Node Manager i-series Integration.

5.1 Adding data elements

Additional data elements can be forwarded to AlarmPoint by adding them to the command line call for a particular incident, as described in “Configuring NNMi Incident Types for Automatic AlarmPoint Notifications” on page 21.

Any changes to the parameters passed with the execution of the Command for Automatic Action must be considered in the AlarmPoint Java Client parameter mapping and the Alarmpoint Action Scripts.

For each new parameter to be passed from NNMi to AlarmPoint, add a new line to the `hp_nnmi.xml` mapping file similar to the following:

```
<parameter index="13" type="string">custom_parameter</parameter>
```

The new parameter may then be used within the AlarmPoint Action Scripts. A possible use for the variable would be to incorporate it in the notification content of a Device by adding the $custom_parameter to the presentation script within the Device’s content creation block:

```
$content.message = $content.message & "Custom Field : " & $event.custom_parameter
```

5.2 Response choices

The following response choices are available in this integration:

Table 5-1. Available Response Choices

Response	Description
Acknowledge	User takes ownership of the incident, preventing further notifications to other Users. (The exception is subscription FYI notifications, which are reporting on the service outage. These are not stopped until the problem has actually been solved.)
Ignore	Stops notifying the current User.
Raise Priority	Increases the priority of the incident in NNMi by one level. (**Voice only**)
Lower Priority	Decreases the priority of the incident in NNMi by one level. (**Voice only**)
Set Priority Top	Sets the priority of the incident to Top. (**Email, BES, and browser only**)
Set Priority High	Sets the priority of the incident to High. (**Email, BES, and browser only**)
Set Priority Medium	Sets the priority of the incident to Medium. (**Email, BES, and browser only**)
Set Priority Low	Sets the priority of the incident to Low. (**Email, BES, and browser only**)
Annotate	Allows the User to append a message to the Notes field of the NNMi incident. (**Non-HTML Email only**)

Note: *Users responding with email Devices can add annotations to their responses, as described in "Adding Annotation Messages", below.*

5.2.0.1 Responses for FYI Notifications

FYI notifications do not have any response choices available, except for FYI notifications sent to voice Devices. Voice FYI notifications offer the following response choices so that Users can navigate between multiple notifications. (This navigation is not required on other Devices.)

Table 5-2. Voice-Device Responses for FYI Notifications

Response	Description
Delete	Removes the notification from the User's list. This option is most likely to be selected.
Save	Saves the notification and stops attempting to deliver it to the User's other Devices. Users may select this option to delay listening to the notification when it is delivered, and access the details by calling in, or via the AlarmPoint Web User Interface, at a later time.
Repeat	Repeats the notification.

5.2.0.2 Adding Annotation Messages

Two-way email Device notifications (not FYI) can add extra annotations which will be appended to the NNMi Incidents Notes field. To add an extra annotation, respond to an email notification with the following format in the subject line:

```
RESPONSE <Choice> <Message>
```

<Choice> can be any of the response choices listed in the table above, and <Message> can be any content you want to add as the annotation.

5.2.1 Changing response choices

Changes to the response choices and behavior can be changed in the response business script in the Action Script set. Actions available through Web Services Calls include acknowledging an incident, annotating it, and changing its priority. Any other response functionality for the integration must be configured within the response HANDLER script with NNMi-provided Web Services Calls.

As an example, the following code illustrates adding a response choice of "Be there in 10 minutes" to the integration:

AlarmPoint Action Scripts:

- Presentation script:

```
$content.choices::add( "be there in ten minutes" )
```

- Response script:

```
# Handle responses
$reply = $response.reply
$reply::toLowerCase()
$ten_minutes= $reply::startsWith( "be there in ten minutes" )
...
IF ( $ten_minutes )
   # Perform any changes to the AlarmPoint event and notifications here
   @event::delinkAll() # Consider the incident handled
   $main.continue = true
```

```
   …
# Acknowledge Event on Management System
      GOSUB acknowledgeIncident
...

# Acknowledges the original NNM incident using a web service call, changing the
lifecycle state
acknowledgeIncident:
  @nnmiRequest = new NetworkNodeManagerScriptObject( $main.nnmi_incident_url,
       $main.nnmi_username, $main.nnmi_password )
  IF (! EXISTS($event.nnm_id))
    $event.nnm_id = @nnmiRequest::getNNMIncidentId($event.incident_id)
  ENDIF
  $request_successful = @nnmiRequest::acknowledgeIncident($event.nnm_id)
  IF ($request_successful != true )
    $err_msg = "Failed to acknowledge NNMi incident: " & $event.incident_id
    IF ( $main.debug )
      @script::log( $main.log_prepend & $err_msg )
    ENDIF
    @event::report( $err_msg )
  ENDIF
RETURN
```

Note: *This is only a brief overview of the required components, For more information about AlarmPoint responses and scripting, refer to the AlarmPoint Action Scripts and the* AlarmPoint Developer's Guide & Scripting Reference.

5.3 Filtering and suppression of Event data

If a subset of Event Data from the NNMi incident type needs to be filtered or suppressed, you can construct a rule similar to the following:

"Suppress all messages from Sources that match *DEV* for a period from 17:00 to 08:00 M-F and 24h on Sat & Sun."

For more information about filtering and suppression, see the *AlarmPoint Java Client Guide*.

5.4 Altering the duration of Events

You can modify the amount of time AlarmPoint will send out notifications for a particular event before it times out by changing the `$main.timeout` variable in the initial PROCESS script. This variable stores the number of seconds the notifications will be allowed to continue before timing out.

The default value is 86400, which is the number of seconds in a 24-hour period. You can change the delay to a 2 hour timeout by changing the line to:

```
$main.timeout = 7200
```

5.5 FYI notifications

You can make all notifications informational only, meaning that the user is not offered any response choices. Setting the `$force_fyi` flag to "on" makes all normal and Subscription notifications one-way (FYI).

In the initial PROCESS script, locate the following line:

```
$force_fyi = disable
```

Change the line to:

```
$force_fyi = on
```

5.5.1 Generating FYI notifications for specific incidents

The FYI parameter is an optional parameter added to the end of the command line that can be passed from NNMi into AlarmPoint. If it is set to “yes” in the NNMi Event Configuration, any notifications generated for the event will be informational-only, and have no response choices.

To use this feature, change the Command for Automatic Action field to the following:

Windows:

```
c:\APAgent\APClient.bin.exe --map-data hp_nnmi bsmith $category $severity $family
$lifecycleState $name $nature $priority $sourceNodeName $sourceObjectName $uuid  yes
```

Unix:

```
/opt/alarmpointsystems/alarmpoint/APClient.bin --map-data hp_nnmi bsmith $category
$severity $family $lifecycleState $name $nature $priority $sourceNodeName
$sourceObjectName $uuid  yes
```

Note that within the script, you can choose to ignore the injected FYI variable to make an event informational-only by setting the `$force_fyi` variable to “off” in the configuration section of the initial PROCESS script. For more information, see “Configuration Variable Reference” on page 37.

5.5.2 Generating FYI notifications for Subscriptions

When using subscriptions to inform Users about service outages, you may want to remove responses from notifications generated for subscriptions.

To accomplish this, ensure that the configuration section of the initial PROCESS script has the following:

```
$subscription_fyi = true
```

The $enable_subs variable must also be set to true. See the section on configuration variables in the initial PROCESS script for details.

Note: *For more information about the variables in this section, see “Configuration Variable Reference” on page 37.*

5.6 Constructing BES and HTML email notifications

You can configure AlarmPoint to create BES and HTML email notifications.

This feature requires the AlarmPoint Developer IDE. For installation instructions, refer to the *AlarmPoint Developer's Guide & Scripting Reference*.

To enable BES and HTML email, the HP Network Node Manager i-series (Business) script package set must be checked into the Developer IDE Database. If the script package has not been checked in already, see the instructions in “Importing the AlarmPoint Script Package” on page 8.

Note: *Some email clients, such as Microsoft Outlook 2007, may not display HTML elements correctly. It is recommended that you test the HTML compatibility of your email client before implementing the HTML email feature.*

To enable BES and/or HTML email:

1. Launch the Developer IDE.
2. Check out the HP Network Node Manager i-series (Business) Production script package.
3. In the Global Configuration Variables section of the initial PROCESS script, do the following:
   - Set the `$main.enable_HTML_Email` variable to `true`.
   - Set `$main.use_logo` to true or false depending on whether you want your HTML email to show a logo.
   - Set `$AlarmPoint_URL` to the base URL of your AlarmPoint web server. (The default is `localhost`.)
4. Optionally, you can also do any of the following
   - Change `$main.HTML_form_url` to point to a JSP page that you want to process any responses from the HTML email. (The default setting should work out-of-the-box.)
   - Change `$main.logo` to a URL that holds the image you want to display at the top of HTML emails. By default, it points to the AlarmPoint logo.
   - Set `$main.logo_alt_text` to the text you wish to display when the logo can not be fetched. This can be displayed if the email client is configured not to show images, or it could be displayed because the email client cannot access the AlarmPoint web server directly and thus cannot respond by using the links in the HTML.
   - If you are using BES and have access to a BES server, you can set the URL to the BES server in the `$main.bes_pushurl` variable.
5. Save and validate the script, and check in the script package.

Note: *For more information about these and other configuration variables, see “Configuration Variable Reference” on page 37.*

5.7 Known Issues

AlarmPoint Systems plans to resolve these issues in subsequent releases:

- Acknowledgement from within Network Node Manager i-series does not prevent further notification deliveries.
- Subsequent responses using BES or HTML Email on Events previously responded to, but not closed, cause a stack trace. (This issue is addressed in AlarmPoint 3.2.1 patch 001.)

5.7.1 Uninstalling

For instructions on removing an AlarmPoint deployment, refer to the *AlarmPoint Installation and Administration Guide*.

6. Configuration Variable Reference

This section outlines and describes the configuration variables available in the initial PROCESS AlarmPoint Action Script.

6.1 Local Configuration Variables

These variables are available only in this script, and control how the script runs. For more information about the initial PROCESS script, consult the *AlarmPoint Developer's Guide & Scripting Reference*

6.1.1 FYI and Subscription Notification Variables

The following variables configure the behavior of informational-only, or FYI, notifications. The value assigned to each variable is the default value within the script

Note: *For more information on the behavior associated with informational-only notifications, see "FYI notifications" on page 34.*

Variable	Description
$force_fyi = "disable"	Forces notifications to be informational only rather than requiring responses. Possible values are: • **disable**: nothing is forced. • **on**: notifications are forced to be FYI. • **off**: notifications are forced not to be FYI.
$use_email_for_fyi = true **$use_phone_for_fyi = false** **$use_im_for_fyi = true** **$use_text_phone_for_fyi = true** **$use_text_pager_for_fyi = true** **$use_numeric_pager_for_fyi = true** **$use_bes_for_fyi = true** **$use_generic_for_fyi = true**	Configure Device filters for informational-only (FYI) notifications. Setting these flags to `false` prevents that Device type from being notified with informational (FYI) messages.
$enable_subs = true	Enables Subscription functionality. If set to `true`, Users subscribed to criteria matching the event will be notified. If set to `false`, no subscribed Users will be notified even if they match the criteria of the event.

Variable	Description
$subscription_fyi = true	Forces Subscription notifications to be informational only; recipients of a Subscription notification will not be able to respond to the event. **Note**: If the `$use_phone_for_fyi` flag is set to `true`, a User can respond with “delete”, which removes the notification from the phone queue, “save”, which moves to the next notification without deleting, or “repeat”, which replays the notification. The `$force_fyi` flag also forces subscriptions to be informational only. If both the `$force_fyi` flag and the `$subscription_fyi` flag are set to `false`, AlarmPoint will use the FYI flag submitted with the event from the Management System.

6.1.2 Fail-safe Configuration Variables

The following variables configure the fail-safe functionality, and specify when notifications will be sent to the fail-safe recipient. The value assigned to each variable is its default value within the script.

Note: *For instructions on how to set up a fail-safe recipient, see “Creating a Fail-Safe Group” on page 19.*

Variable	Description
$fail_safe = “enabled”	Controls whether the fail-safe recipient is notified, and under which circumstances. Possible values are: • **enabled**: notify the fail-safe Group if no Subscriptions match and there are no notifiable recipients. • **for-subscriptions**: notify if the Subscription functionality is enabled and no Subscriptions match. • **for-recipients**: notify if there are no notifiable recipients. • **disabled**: disable the fail-safe functionality; no notifications will be sent to the fail-safe recipient.
$fail_safe_group = "NNMi FailSafe"	Identifies the fail-safe recipient, which is typically a Group, but may be a User.

6.1.3 Alert Configuration Variables

The following variables configure Alert behavior. The value assigned to each variable is its default value within the script.

Variable	Description
$override_timeframes = false	Overrides any Device Timeframes that have been configured for a User for this notification.
$use_emergency_devices = false	Forces the use of emergency Devices as part of the Device resolution processing.

Variable	Description
$track_delivery = true	Configures the notification to run a response script when the delivery of a notification is successful. As this can limit Node performance, you can set this value to false if the custom behavior for successful delivery events is unnecessary, but you will lose any information about whether a delivery was successful.

6.2 Global Configuration Variables

These variables are available throughout the script package, and are parameters of the "main" object. The value assigned to each variable is its default value within the script.

Variable	Description
$main.timeout = 86400	Amount of time (in seconds) the event is allowed to run before timing out. (86400 seconds = 24 hours.)
$main.debug = false	Indicates whether to log informational messages for debugging purposes. Disabling this variable may improve performance, but will provide less information.
$main.use_logFile = false	Specify whether to use an alternate log file for debugging messages. This variable is ignored unless `$main.debug` is also set to true.
$main.logFile = "../logs/HP_NNMi_Script.log"	Defines the file used to log debugging information (only if `$main.use_logfile` is set to `true`).
$main.maxInvalidResponses = 3	Specifies the maximum number of invalid responses allowed before the notification will no longer be requeued. If a recipient sends an invalid response and this number has not been exceeded, they will be renotified with the same content, prefixed with a message indicating that their response was invalid.
$main.annotate = true	Enables submission of information back to the Management System. Information is logged throughout the script progress; if this variable is set to `true`, these logged messages will be annotated to the originating Event. Setting this variable to `false` may improve performance, but will make debugging difficult as some information may not be annotated to the originating event.
$main.subscription_annotate = false	Enables submission of Subscription information back to the Management System. (As with `$main.annotate`, but specifically for Subscription information.) Most Subscriptions are informational only; this variable can be enabled, for debugging and informational purposes but may reduce performance.
$main.enable_HTML_Email = true	Enables HTML Email functionality for email clients able to support HTML emails. If a client cannot support HTML than the plain text version will be passed.

Variable	Description
$AlarmPoint_URL = "http://localhost:8888"	Identifies the AlarmPoint URL used for the HTML response form and AlarmPoint logo. If the specified URL cannot be reached, the logo will not appear, and the response links will not work.
$main.HTML_form_url = $AlarmPoint_URL & "/jsp/ProcessNotificationResponse.jsp"	Specifies the URL of the AlarmPoint Web Server's Process Notification Response JSP form, used by HTML email and BES to inject responses through the system.
$main.use_logo = true	Specifies whether HMTL email notifications will display the AlarmPoint (or custom) logo.
$main.logo = $AlarmPoint_URL & "/static/images/logos/alarmpoint/UNKNOWN.gif"	Specifies the path to the graphic displayed on HTML (email and BES) notifications.
$main.logo_alt_text = "[If the logo does not appear you may be blocking images or you may be outside a firewall. If the latter, the links will not work for responding and you should respond by replying to this email as described below.]"	The alternate text to display if the HTML email logo is unavailable. **Note**: If the logo does not display, it is unlikely that the HTML_form_url is valid and responses will not be injected from HTML Devices (email and BES).
$main.numeric_pager_number = "555-1212"	The phone number to display for calling in to retrieve event information. This variable has a non-existent number as a default value; a real call-in number must be supplied, or a message indicating that an AlarmPoint event has occurred.
$main.bes_pushurl = "http://localhost:8888/static"	Specifies the URL of the BES server. (Optional.)
$main.nnmi_incident_url = "http://localhost:8004/IncidentBeanService/IncidentBean"	Specifies the URL for the NNMi Incident Web Service Bean. **Note**: The port for UNIX installations of NNMi is `80`, rather than the default `8004`.
$main.nnmi_username = "webservices"	Specifies the user name of the NNMi Web Service Client.
$main.nnmi_password = "nnm"	Specifies the password of the NNMi Web Service Client.

7. Contact Us

You can access the AlarmPoint Systems Web Site at http://www.alarmpoint.com. From this site you can obtain information about the Company, the Products, Support and other helpful information. You may also access the Customer Support Site from the main web page. In this protected site you will find current product releases, helpful hints, patches, release notes, a helpful product knowledge base, trouble ticket submission areas and other helpful tools provided by AlarmPoint Systems, Inc.

AlarmPoint Systems, Inc

4457 Willow Road, Suite 220
Pleasanton, CA 94588

Phone: 925-226-0300
Fax: 925-226-0310

Email: support@alarmpoint.com
Website: http://www.alarmpoint.com

Hewlett-Packard Company

3000 Hanover Street
Palo Alto, CA 94304-1185 USA

Phone: 650-857-1501
Fax: 650-857-5518

Support: http://support.openview.hp.com
Website: http://www.openview.hp.com

8. Copyright

AlarmPoint Systems, Inc and HP produced this integration document to assist customers with joint HP/AlarmPoint Systems implementations. HP and AlarmPoint Systems have made every effort to ensure that the information contained in this document is accurate, but do not guarantee any accuracy now or in the future. AlarmPoint Systems™ and AlarmPoint® are a trademark and registered trademark, respectively, of AlarmPoint Systems, Inc. in the United States, United Kingdom and other jurisdictions. HP Network Node Manager i-series is a registered trademark of HP Software, Inc. All other trademarks are the property of their respective owners.