CA Spectrum®

AlarmNotifier User Guide

Release 9.4

CA Technologies Product References

This guide references CA Spectrum®.

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: AlarmNotifier Overview 7

About AlarmNotifier 7
Alarm Monitoring Process 8
CA Spectrum Alarm Notification Manager (SANM) 9

Chapter 2: Operating AlarmNotifier 11

Start AlarmNotifier 11
Stop AlarmNotifier 12
Start AlarmNotifier with the Process Daemon 13
    Run Multiple AlarmNotifiers 14
AlarmNotifier Output 15
Script Parameter Definitions 16
Persistent and Stale Alarms 19

Chapter 3: Customizing AlarmNotifier 21

Modifying Scripts 21
    Enable Email Notifications in a Script 21
    Mail Service on Windows Platform 22
Limiting Script Output 23
Passing CA Spectrum Attributes to Scripts 24
    Global Alarm Attributes 25
Sending Data to a Third-Party Application 27
    Arguments—USE_NEW_INTERFACE Set to True 28
    Arguments—USE_NEW_INTERFACE Set to False 28
    Date and Time 30
Customizing the .alarmrc Resource File 30
    .alarmrc Parameters 31
Reinstalling or Upgrading CA Spectrum 32

Index 33

Chapter 1: AlarmNotifier Overview

This section contains the following topics:

About AlarmNotifier (see page 7)
Alarm Monitoring Process (see page 8)
CA Spectrum Alarm Notification Manager (SANM) (see page 9)

About AlarmNotifier

AlarmNotifier is a SpectroSERVER-client application that installs with core CA Spectrum components. The AlarmNotifier application connects to a single SpectroSERVER and invokes scripts that provide notifications about CA Spectrum alarm status.

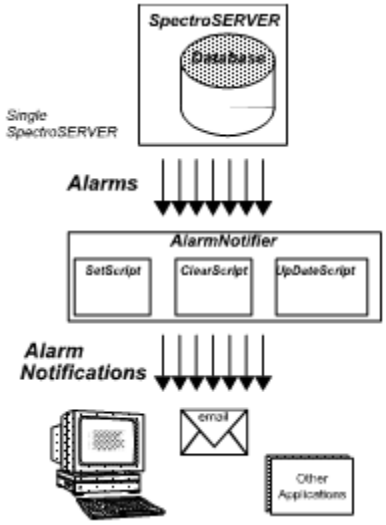
Start AlarmNotifier from a terminal shell command prompt. Once started, it continuously displays output from scripts that are invoked whenever alarms are either set, cleared, or updated. AlarmNotifier provides the following features for CA Spectrum:

- Single SpectroSERVER alarm monitoring.
- Three scripts that generate alarm information: SetScript, ClearScript, and UpdateScript.

  These scripts contain settings that can be customized for your environment.
- Resource file parameters that can be configured to modify AlarmNotifier operational features.

Alarm Monitoring Process

AlarmNotifier supplements CA Spectrum alarm monitoring and notification features. The following diagram illustrates the relationship between AlarmNotifier and CA Spectrum:



CA Spectrum performs some alarm functions, while AlarmNotifier performs others. CA Spectrum polls the modeled network elements and updates the status information about each element that is stored in the SpectroSERVER database.

CA Spectrum generates an alarm when it receives a trap from the network or when it detects a critical status change in a network-element model. In the OneClick Topology view, the condition of the model icon changes from green to another color to indicate alarm severity. CA Spectrum posts information about the alarm in the Alarms tab. Event information for the alarm appears in the Events tab in the OneClick Contents panel.

When AlarmNotifier is started, it registers with CA Spectrum. Then a model named AlarmNotifier of type ClientApp is created. This model is not visible in any of the CA Spectrum Topology views. However, you can see it in the Events tab. The Events tab displays information such as the application start and stop time for this model.

AlarmNotifier queries the SpectroSERVER and requests information about existing alarms. AlarmNotifier runs scripts and generates notifications about existing alarms.

Each time an alarm is set, cleared, or updated, AlarmNotifier receives information from the SpectroSERVER and invokes the relevant script. AlarmNotifier scripts can initiate email notifications of alarms that are sent to network personnel. They can also transmit alarm information to third-party applications.

CA Spectrum Alarm Notification Manager (SANM)

CA Spectrum Alarm Notification Manager (SANM) is an add-on component for CA Spectrum that can enhance AlarmNotifier features. The SANM Policy Administrator lets you create multiple alarm-filtering policies, which you can associate with uniquely named instances of AlarmNotifier applications. Use these policies to instruct AlarmNotifier to generate notifications only for the alarms that you consider relevant.

The SANM Policy Administrator lets you associate policies with AlarmNotifier applications as required. You can also automate this association process using the Scheduler utility.

With SANM installed, AlarmNotifier offers the following capabilities:

- Distributed SpectroSERVER alarm monitoring
- Additional script parameters that provide more alarm information
- Commands for acknowledging and clearing alarms
- Additional startup options that let you log AlarmNotifier activities and concurrently run multiple instances of AlarmNotifier
- SANM alarm-filtering tools

Note: For more information about SANM, see the *Alarm Notification Manager User Guide*.

Chapter 2: Operating AlarmNotifier

This section contains the following topics:

Start AlarmNotifier (see page 11)
Stop AlarmNotifier (see page 12)
Start AlarmNotifier with the Process Daemon (see page 13)
AlarmNotifier Output (see page 15)
Script Parameter Definitions (see page 16)
Persistent and Stale Alarms (see page 19)

Start AlarmNotifier

AlarmNotifier is installed in the following directory:

`<$SPECROOT>/Notifier`

By default, this directory contains the following files:

- .alarmrc
- AlarmNotifier
- ClearScript
- README
- SetScript
- UpdateScript

Additional files are present in this directory if SANM is installed. For more information, see the *Alarm Notification Manager User Guide*.

Note: In a Windows environment, configure the mail service before starting AlarmNotifier for the first time.

Follow these steps:

1. Verify that the LANDSCAPE parameter in the <*$SPECROOT*>/Notifier/.alarmrc resource file specifies the landscape to which you intend to connect.
2. Verify that this landscape is up and running.
3. Launch AlarmNotifier with either the default resource file (.alarmrc) or a modified version of the resource file as follows:
   - To start AlarmNotifier using the default resource file:
     `<$SPECROOT>/Notifier/AlarmNotifier`
   - To start AlarmNotifier using a different resource file:
     `<$SPECROOT>/Notifier/AlarmNotifier -r <resource file>`

   Note: Additional options are available for the AlarmNotifier command when you are using SANM. For more information, see the *Alarm Notification Manager User Guide*.

More information:

Mail Service on Windows Platform (see page 22)

Stop AlarmNotifier

Once started, AlarmNotifier runs continuously. To stop AlarmNotifier, use one of the following commands:

Stop AlarmNotifier Running in Foreground

Follow these steps:

Windows/Solaris

Enter:

`Control-C`

Stop AlarmNotifier Running in Background

Follow these steps:

Windows

To end the AlarmNotifier process, use $SPECROOT/lib/SDPM/kill –TERM <pid>.

Solaris

Enter:

```
kill <option> <AlarmNotifier PID>
```

Start AlarmNotifier with the Process Daemon

You can automate the startup of AlarmNotifier processes using processd, the CA Spectrum process management daemon.

The processd automatically starts during CA Spectrum installation and whenever the system restarts. Once processd is started, it automatically starts and manages other processes.

To enable processd to launch and track the AlarmNotifier application, create an install ticket file.

Note: AlarmNotifier.idb is shipped by default. Therefore, you do not need to create the file from scratch. You can change AUTOBOOTSTART to Y.

Follow these steps:

1. Using a text editor, create a file with the following content:

   ```
   # Processd Install Ticket for Alarm Notifier
   PARTNAME;ALARMNOTIFIER;
   APPNAME;Alarm Notifier;
   WORKPATH;$SPECROOT/Notifier;
   LOGNAMEPATH;$WORKPATH/ALARMNOTIFY.OUT;
   ADMINPRIVS;y;
   AUTORESTART;y;
   AUTOBOOTSTART;y;
   #STATEBASED;N;
   NUMPROCS;1;
   RETRYTIMEOUT;6000;
   TICKETUSER;<USERNAME>;
   RETRYMAX;20;
   STARTPRIORITY;30;
   #ENV;<var>=<value>;
   ARGV;$WORKPATH/AlarmNotifier<CSEXE>; //
   ```

2. Name this file AlarmNotifier.idb and save it in the following directory:

   ```
   <$SPECROOT>/lib/SDPM/partslist
   ```

   The LOGNAMEPATH parameter specifies the name and path of the log file for the AlarmNotifier application.

Each time AlarmNotifier starts, a new log file is generated, and a backup of the previous log file is created. However, the SpectroSERVER only stores the two most recent AlarmNotifier log files. To keep all log file output for AlarmNotifier, create a script that saves the log file output in a separate file.

The value of the TICKETUSER parameter, <USERNAME>, must be the username of a valid CA Spectrum User Model.

The SpectroSERVER must be running before AlarmNotifier starts. Therefore, the STARTPRIORITY parameter can be set to 30, indicating that AlarmNotifier is dependent on the SpectroSERVER. For more information, see the *Distributed SpectroSERVER Administrator Guide*.

Run Multiple AlarmNotifiers

Multiple AlarmNotifiers with different notification policies can be started when CA Spectrum processd starts. You can enable this setup by creating requisite install ticket files for such AlarmNotifiers. Perform the following tasks to run multiple AlarmNotifiers:

- Create requisite install ticket files for multiple AlarmNotifiers or make copies of the default ALARMNOTIFIER.idb file and rename the files accordingly. For more information, see Start AlarmNotifier with the Process Daemon (see page 13).
- After the install ticket files are created, you can change the values of the install ticket file to start an AlarmNotifier with a preferred notification policy. You can provide the following values:

  APPNAME

  Specifies the name of the application that is created to implement the notification policy. For more information, see the *Alarm Notification Manager User Guide*.

  AUTOBOOTSTART

  Notifies the process daemon to start a process when processd starts. Set the value to Y.

  WORKPATH

  Specifies the working path where the application can be found.

  LOGNAMEPATH

  Specifies the name and path of the log file for the AlarmNotifier application.

  Note: LOGNAMEPATH must be unique for each instance.

ARGV

Specify the following value:

```
$SPECROOT/Notifier/AlarmNotifier<CSEXE> -r alarmrc -n <name of the
application that you specified for APPNAME>
```

Note: If we copy the AlarmNotifier executable file to a different directory to run as separate instance, customize the WORKPATH, LOGNAMEPATH, and ARGV accordingly.

AlarmNotifier Output

AlarmNotifier invokes the SetScript, ClearScript, or UpdateScript whenever AlarmNotifier detects an alarm that is set, cleared, or updated in CA Spectrum. Each script generates a notification containing information about alarm status (set, cleared, or updated) and displays it. Each notification contains the parameters that are defined in Script Parameter Definitions (see page 16).

SetScript

Invoked for an alarm in the following situations:

- AlarmNotifier is started and detects an existing alarm. AlarmNotifier invokes SetScript unless the value of GET_EXISTING_ALARMS is set to 'false' in the .alarmrc resource file.

  Default: true.

- CA Spectrum generates an alarm while AlarmNotifier is running.

ClearScript

Invoked when an alarm is cleared.

UpdateScript

Invoked when an alarm is updated. An alarm is defined as updated in these situations:

- A troubleshooter has been assigned to an alarm, or the troubleshooter name has been changed. The RepairPerson parameter in the scripts represents this troubleshooter name.
- The status of an alarm has changed. Status information for an alarm is entered in the Alarms tab. The AlarmStatus parameter represents status in the scripts.
- An alarm is acknowledged or unacknowledged in CA Spectrum.
- A new event, or a change to an existing event, occurs on a device that is in an alarm state.

Script Parameter Definitions

You can update the parameters of the AlarmNotifier scripts to customize their functionality. The following list describes the parameters that are available for SetScript, ClearScript, and UpdateScript.

Note: Additional script parameters are available when you use SANM with AlarmNotifier. For more information, see the *Alarm Notification Manager User Guide*.

Date

Specifies the date when AlarmNotifier detects that the alarm is set, updated, or cleared.

Format: mm/dd/yyyy

Time

Specifies the time when AlarmNotifier detects that the alarm is set, updated, or cleared.

Format: hh:mm:ss

Mtype

Specifies the type of model for which the alarm is set, updated, or cleared.

ModelName

Specifies the name of the model whose alarm is set, updated, or cleared. If the ModelName contains special characters, pass it to the script as an environment variable to avoid errors. A special character is a character that the command shell interprets as having special meaning such as '$' or '*'.

To pass the model name as an environment variable, add the attribute ID for ModelName (0x1006e) to the values for the EXTRA_ATTRS_AS_ENVVARS parameter in the .alarmrc file. For more information, see Passing CA Spectrum Attributes to Scripts (see page 24) and .alarmrc Parameters (see page 31).

AlarmID

Specifies the numeric identifier that CA Spectrum assigned to the alarm.

Global AlarmID

Specifies a unique numeric identifier that CA Spectrum assigns to the alarm. Unlike the AlarmID, the global alarm ID is not only unique within the CA Spectrum environment, but also can be passed as a unique identifier to other environments. Use this value to pass a unique identifier to third-party software. By default, Global AlarmID is commented out in each of the AlarmNotifier scripts. Remove the comment mark (#) to pass this parameter.

Severity

Specifies the CA Spectrum severity-level code for the alarm: Critical (Red), Major (Orange), Minor (Yellow), Maintenance (Brown), Suppressed (Gray), or Initial (Blue).

ProbableCauseID

Specifies the hexadecimal identifier that is associated with the probable cause for the alarm.

RepairPerson

Specifies the troubleshooter who is assigned to the alarm in the Alarms tab. AlarmNotifier invokes the UpdateScript whenever a troubleshooter is first assigned and each time thereafter.

The following circumstances determine whether the name of a troubleshooter (a repair person) appears in the notifications that SetScript and ClearScript generate:

- If a troubleshooter is assigned after AlarmNotifier detects that an alarm has been set, SetScript does not display a name. The UpdateScript and the ClearScript do display a name.
- If a troubleshooter is assigned before AlarmNotifier detects the set (for an alarm that exists before AlarmNotifier is started), all three scripts display the name of the troubleshooter.

Note: For more information, see the *Operator Guide*.

AlarmStatus

Indicates the status information for the alarm in OneClick. AlarmNotifier invokes UpdateScript whenever status information is first entered and each time thereafter. Status information typically appears in the notifications that the SetScript and ClearScript generate. The following circumstances are exceptions:

- If status information is entered after AlarmNotifier detects that an alarm has been set, SetScript does not display the information. The UpdateScript and the ClearScript do display the information.
- If status information is entered before AlarmNotifier detects the set (for an alarm that exists before AlarmNotifier is started), all three scripts display the status information.

Note: For more information, see the *Operator Guide*.

SpectroSERVER

Specifies the name of the host for the SpectroSERVER where the alarm has been set, updated, or cleared.

Landscape

Specifies the handle for the landscape from which the alarm has been set, updated, or cleared.

ModelHandle

Specifies the handle of the model for which the alarm has been generated.

ModelTypeHandle

Specifies the handle of the model type for which the alarm has been set, updated, or cleared.

IPAddress

Specifies the IP address of the network element for which the alarm has been set, updated, or cleared.

SecurityString

Specifies the security string of the model for which the alarm has been set, updated, or cleared.

AlarmState

Specifies whether the alarm state is "Existing" or "New."

The state of an alarm is "Existing" if the alarm is set before AlarmNotifier is started. AlarmNotifier invokes SetScript for an existing alarm if the GET_EXISTING_ALARMS parameter in the .alarmrc resource file is set to true.

The state of an alarm is "New" if the alarm is generated after AlarmNotifier is started. The alarm state is also "New" when the SpectroSERVER for the AlarmNotifier restores connections to a previously connected landscape where the alarm occurred.

Acknowledged

Specifies whether the alarm has been acknowledged.

UserClearable

Specifies whether a user can clear the alarm.

DeviceType

Specifies the value of the DeviceType attribute on the model for which the alarm has been set, updated, or cleared. For more information, see the *Certification User Guide*.

Raw Alarm Time

The Alarm date and time is available as $DATE and $TIME. To get the unformatted alarm time, you can reference $RAW_ALARM_TIME.

Note: The unformatted alarm time is the number of seconds that have elapsed since midnight UTC of January 1, 1970.

Persistent and Stale Alarms

When the SpectroSERVER stops and restarts, alarms that were already present continue to exist. These alarms are "persistent" alarms. The persistent alarm feature lets CA Spectrum retain alarm-related information such as troubleshooter assignments and status when the SpectroSERVER shuts down.

In some cases, the underlying cause of an alarm is resolved between the time that the SpectroSERVER shuts down and restarts. The alarm still appears in the Alarms list in OneClick but is considered to be stale. You can clear all stale alarms (which are also known as "residual alarms") manually.

However, stale alarm information is not forwarded to AlarmNotifier by the SpectroSERVER. Instead, a single *new* alarm that indicates that stale or residual alarms exist on the landscape is generated and sent to AlarmNotifier. When you manually clear a stale alarm, that alarm is also cleared in AlarmNotifier. When the final stale alarm is cleared, a "clear" is issued for the stale alarm notification.

Chapter 3: Customizing AlarmNotifier

This section contains the following topics:

Modifying Scripts (see page 21)
Limiting Script Output (see page 23)
Passing CA Spectrum Attributes to Scripts (see page 24)
Sending Data to a Third-Party Application (see page 27)
Customizing the .alarmrc Resource File (see page 30)
Reinstalling or Upgrading CA Spectrum (see page 32)

Modifying Scripts

You can modify AlarmNotifier scripts to customize AlarmNotifier actions and output. You can configure the scripts to initiate email notifications to specified recipients. You can also customize scripts to limit the range of information that alarm transition notifications provide, or to integrate with a third-party application.

Enable Email Notifications in a Script

Each AlarmNotifier script includes two parameters (SENDMAIL and VARFORMAIL) that you can configure to enable email notifications to the troubleshooter for an alarm. You can enable email notifications for one or more scripts.

Note: To preserve default script configuration settings in case of accidental loss, make a backup copy of the default script that you plan to edit.

Follow these steps:

1. Navigate to the *<$SPECROOT>*/Notifier directory, or to the directory where the script is saved.
2. Open the script with a text editor.

   Note: All of the scripts are executed serially. Therefore, you can edit a script without stopping the AlarmNotifier.
3. Set the SENDMAIL parameter in the script to True.
4. Set the VARFORMAIL parameter to RepairPerson.
5. Save and close the script.

   Email is sent to the troubleshooter who is assigned to the alarm in the Alarms tab. This person must be an authorized user of (or user model in) the landscape where the alarm originates.

Note: The value for RepairPerson is established after the alarm has occurred. As a result, mail cannot be sent in response to a set action (using the SetScript). However, if you are also using SANM, you can configure mail to be sent as a result of a set action. For more information, see the *Alarm Notification Manager User Guide*.

Mail Service on Windows Platform

On Windows, use the mail command to enable the mail service so that AlarmNotifier scripts can send email notifications.

Take this step before AlarmNotifier starts. Otherwise, an error message prompts you to configure the mail service and then start AlarmNotifier.

Mail Command Parameters

Several parameters are required when running the mail command from a terminal window. These parameters are as follows:

-m

Is the Return host name. The Return host is the computer where incoming mail is received.

-h

Is the Simple Mail Transfer Protocol (SMTP) host, the computer where outgoing email is sent to be processed.

-u

Is the username.

Note: AlarmNotifier can send notifications to a pager. We recommend first configuring AlarmNotifier to send notifications to a valid local mail account to test this configuration. You can then reconfigure the Mail Service to send the notifications to the pager.

Configure Mail Service

You can set up the mail service on Windows so that AlarmNotifier can send email notifications.

Follow these steps:

1. Consult with your mail server administrator to verify the correct values of the mail command parameters.
2. Open the bash shell.

3. Enter the following command:
   ```
   mail -m your-company.com -h smtp.your-company.com -u username
   ```
   The command usage list appears once the command has completed successfully.
4. To verify that the configuration is complete, view the registry entries for HKEY_LOCAL_USER/Software/SMail.

   The hostname, smtphost, and username keys now contain the information that you included in the command string.

More information:

Mail Command Parameters (see page 22)

Limiting Script Output

You can comment out AlarmNotifier script parameters to reduce the amount of output from a script.

Note: Before you modify a script, review the parameter descriptions in Script Parameter Definitions (see page 16). We recommend a thorough understanding of the information that you are suppressing.

Follow these steps:

1. Close the instance of the AlarmNotifier application that you want to configure.
2. Navigate to the default script directory, <$SPECROOT>/Notifier, or the directory where the script that you want to edit is saved.
3. Open the script with a text editor.
4. Comment out the echo command lines that you want to suppress by typing a pound sign (#) at the beginning of each line. In the following example, the UserClearable parameter is commented out:
   ```
   echo "SecurityString:        " $SECSTR
   echo "AlarmState:    " $ALARMSTATE
   echo "Acknowledged:  " $ACKD
   #echo "UserClearable:        "  $CLEARABLE
   ```
   When this script generates notifications, the information from the parameter that has been commented out does not appear on the screen.
5. Save and close the script.

   Note: Do *not* comment out or modify the assignments of the variables themselves or the shift commands. The script does not display alarm information properly if you change these lines.

Passing CA Spectrum Attributes to Scripts

Attributes of a model with an alarm can be passed to AlarmNotifier. Model attributes can be used as parameters in SetScript, ClearScript, or UpdateScript. Use the .alarmrc parameters EXTRA_ATTRS_AS_ENVVARS or EXTRA_ATTRS_AS_ARGS to pass in attributes. To enable these parameters, set the USE_NEW_INTERFACE .alarmrc parameter to TRUE.

EXTRA_ATTRS_AS_ENVVARS passes attributes to AlarmNotifier as environment variables. EXTRA_ATTRS_AS_ARGS passes attributes as command-line arguments. For most attributes, either of these mechanisms can be used. However, EXTRA_ATTRS_AS_ENVVARS is required in cases where new lines or special characters can cause problems for the script that parses the extra data. When USE_NEW_INTERFACE=TRUE, the environment variable mechanism is used to pass $STATUS, $EVENTMSG, and $PCAUSE to avoid this problem.

In the .alarmrc file, set the appropriate parameter equal to the CA Spectrum attribute IDs that you want to pass. You can reference the attribute ID either in hexadecimal or decimal notation.

If you pass an attribute as an environment variable using EXTRA_ATTRS_AS_ENVVARS, you reference this variable in a script using the following syntax:
`$SANM_<attribute_ID>`

<attribute_ID>

Specifies the attribute ID of the attribute you are referencing. If you have used hexadecimal notation to call this attribute in the .alarmrc file, hexadecimal notation is also required in the script. If you have used decimal notation to call this attribute in the .alarmrc file, use decimal notation in the script.

Windows automatically sets environment variables to uppercase. Therefore, when you reference these variables, use the uppercase format, such as $SANM_0X100C5.

If you use EXTRA_ATTRS_AS_ARGS to pass an attribute as an argument, you can reference this variable in a script by assigning the value to a variable within the script:
`<variable>=$<x>`

<variable>

Specifies the variable that holds the value of the attribute.

<x>

Specifies the appropriate variable number for the order and number of arguments that you have passed.

Example

The following example shows four sample CA Spectrum attributes that are passed to AlarmNotifier in the .alarmrc file. The attributes are then referenced in a script.

.alarmrc File Reference

```
USE_NEW_INTERFACE=TRUE
EXTRA_ATTRS_AS_ENVVARS=0x100c5,0x11f84
EXTRA_ATTRS_AS_ARGS=0x110df,0x117dc
```

Script Reference

```
#These lines read 0x110df and 0x117dc into the variables MAC_ADDRESS
#and FIRMWARE_VERSION respectively.
shift 9
MAC_ADDRESS=$1
FIRMWARE_VERSION=$2

#These lines print out the value of each attribute.(Solaris Platform)
echo "The value of attribute 0x100c5 is: " $SANM_0x100c5
echo "The value of attribute 0x11f84 is: " $SANM_0x11f84
echo "The value of attribute 0x110df is: " $MAC_ADDRESS
echo "The value of attribute 0x117dc is: " $FIRMWARE_VERSION

#These lines print out the value of each attribute.(Windows Platform)
#references to environmental variables are in uppercase
echo "The value of attribute 0x100c5 is: " $SANM_0X100C5
echo "The value of attribute 0x11f84 is: " $SANM_0X11F84
echo "The value of attribute 0x110df is: " $MAC_ADDRESS
echo "The value of attribute 0x117dc is: " $FIRMWARE_VERSION
```

More information:

.alarmrc Parameters (see page 31)

Global Alarm Attributes

This section lists the CA Spectrum Global Alarm attributes and their corresponding attribute IDs. Pass any of these attributes to AlarmNotifier using the method that is described in Passing CA Spectrum Attributes to Scripts (see page 24).

Acknowledged

0x11f4d

Alarm_Source

0x11fc4

Alarm_Status

0x11f4f

Cause_Code

0x11f50

Cleared_By_User_Name

0x11f51

Creation_Date

0x11f4e

ImpactScope

0x1290e

ImpactSeverity

0x1290d

Last_Occurrence_Date

0x1321a

Occurrences

0x11fc5

Originating_Event

0x1296e

Persistent

0x12942

Primary_Alarm

0x11f54

Severity

0x11f55

Trouble_Shooter_Email

0x12a6c

Trouble_Shooter_mh

0x11fc6

Trouble_Ticket_ID

0x12022

TroubleShooter

0x11f57

User_Clearable

0x11f9b

Customer_Impact

0x12bf6

Service_Impact

0x12bf7

Note: Service_Impact does not always reflect the current health of the service. Service_Impact represents the health status of the service at the time when the device alarm was generated. If only the service health changes, Service_Impact is not affected.

Sending Data to a Third-Party Application

You can customize or replace SetScript, ClearScript, or UpdateScript to create an integration with a third-party application.

You can supplement and extend the functionality of SetScript, ClearScript, or UpdateScript. You can also include CA Spectrum CLI commands in these scripts to retrieve more information from the SpectroSERVER. You can also add code of your own to the script that sends data to a third-party application.

If you do not use the functionality of the existing script, you can direct AlarmNotifier to run your own script or executable. In the AlarmNotifier resource file (.alarmrc), the Set_Script parameter controls the script that runs when an alarm is set. The Clear_Script parameter controls the script that runs when an alarm is cleared, and the Update_Script parameter controls the script that runs when an alarm is updated.

By default, the Set_Script parameter has a value of SetScript, the Clear_Script parameter has a value of ClearScript, and the Update_Script parameter has a value of UpdateScript. You can modify the values of these parameters to launch a different script when an alarm is set, cleared, or updated.

Custom script parameters depend on the data to extract from CA Spectrum and on the third-party application to which the data is sent. To create your own script or executable, first understand the arguments that are passed from CA Spectrum to the receiving script or executable. Your script or executable must receive all of the arguments from CA Spectrum in the correct order.

More information:

AlarmNotifier Output (see page 15)
Customizing the .alarmrc Resource File (see page 30)

Arguments—USE_NEW_INTERFACE Set to True

The following table shows the number, name, and format of each argument that is passed to each script when the USE_NEW_INTERFACE .alarmrc parameter is set to TRUE.

When USE_NEW_INTERFACE is set to TRUE, the Status argument is sent as an environment variable. The order of arguments is therefore affected.

Argument	Name	Format
1	Date	mm/dd/yyyy
2	Time	hh:mm:ss
3	Model Type	Text
4	Model Name	Text
5	Alarm ID	Integer
6	Severity	Text
7	Cause	Text
8	Repair Person	Text
9	Server	Text
10	Landscape	Hexadecimal
11	Model Handle	Hexadecimal
12	Model Type Handle	Hexadecimal
13	IP Address	xxx.xxx.xxx.xxx
14	Security String	Text
15	Alarm State	Text
16	Acknowledged	Text
17	Clearable	Text
18	Device Type	Text

Arguments—USE_NEW_INTERFACE Set to False

The following table shows the number, name, and format of each argument that is passed to each script when the USE_NEW_INTERFACE .alarmrc parameter is set to FALSE.

Note: If you are working with a SANM-enabled AlarmNotifier, additional arguments are passed. For more information, see the *Alarm Notification Manager User Guide*.

Argument	Name	Format
1	Date	mm/dd/yyyy
2	Time	hh:mm:ss
3	Model Type	Text
4	Model Name	Text
5	Alarm ID	Integer
6	Severity	Text
7	Cause	Text
8	Repair Person	Text
9	Status	Text
10	Server	Text
11	Landscape	Hexadecimal
12	Model Handle	Hexadecimal
13	Model Type Handle	Hexadecimal
14	IP Address	xxx.xxx.xxx.xxx
15	Security String	Text
16	Alarm State	Text
17	Acknowledged	Text
18	Clearable	Text
19	Device Type	Text

Date and Time

The following conditions apply to the Date and Time arguments for the Set, Clear, and Update Scripts:

- **For SETs:** Date and Time are derived from the CREATION_DATE (0x11f4e) attribute. SpectroSERVER sets this attribute when the alarm is created.
- **For CLEARs:** Date and Time are derived from the CLEAR_DATE (0x129af) attribute. SpectroSERVER sets this attribute when the alarm is cleared.
- **For UPDATEs:** Date and Time reflect when the AlarmNotifier received notification that the alarm has been updated. SpectroSERVER does not set this value. Do not rely on this value to determine the exact time that the update occurred.

Customizing the .alarmrc Resource File

The .alarmrc resource file, which is saved in the Notifier directory, includes AlarmNotifier operational parameters. You can modify the resource file in the following ways:

- Specify whether AlarmNotifier processes alarms that CA Spectrum generated before AlarmNotifier started.
- Specify optional CA Spectrum attributes to pass to AlarmNotifier.
- Replace SetScript, ClearScript, or UpdateScript with custom scripts.
- Specify the SpectroSERVER to which AlarmNotifier connects.
- Disable the parameters that specify AlarmNotifier actions that you do not plan to deploy, reducing network traffic.

  Note: Commenting out parameters does not disable them. Instead, their default value is used.

Follow these steps:

1. Navigate to the <*$SPECROOT*>/Notifier directory and make a backup copy of the .alarmrc file.
2. Open the file with your preferred text editor.
3. Edit the file by turning off optional parameters or by entering new parameter values. You can disable a parameter by giving it a value of False or by leaving the value blank.

   Note: Do not disable required parameters or delete parameters.
4. Save and close the file, and then restart AlarmNotifier.

   Your changes go into effect when AlarmNotifier is restarted.

.alarmrc Parameters

The following list describes the resource file parameters that are provided with AlarmNotifier. For more information, see the *Alarm Notification Manager User Guide*.

LANDSCAPE

Identifies the initial SpectroSERVER host to which AlarmNotifier connects. Enter only one name here. If LANDSCAPE is not defined, AlarmNotifier defaults to using the first landscape in the VNM landscape map. An informational window shows the default landscape handle.

VNM_MAIL_TIMEOUT

Specifies the minimum time that the Mail Service waits for a response from the SpectroSERVER before the request is canceled.

Default: 60,000 milliseconds (one minute)

VNM_CONNECT_TIME_LIMIT

Specifies the minimum delay before an initial TCP connect request between AlarmNotifier and a SpectroSERVER times out.

Default: 60,000 milliseconds (one minute)

KEEP_ALIVE_TIMEOUT

Specifies the amount of time before a keep-alive request times out.

Default: 30,000 milliseconds (30 seconds)

KEEP_ALIVE_INTERVAL

Specifies the amount of time between keep-alive requests that are sent to the SpectroSERVER. A keep-alive request checks to see if the SpectroSERVER is still connected to the AlarmNotifier. If AlarmNotifier does not receive a response to the request, it disconnects from the SpectroSERVER. If your SpectroSERVER is slow to respond to these requests, you can increase this value to prevent the AlarmNotifier from disconnecting from the SpectroSERVER.

Default: 60,000 milliseconds (1 minute)

SEND_ALARM_DELAY

Specifies the minimum delay between successive alarm notifications.

Default: 1,000 milliseconds (1 second)

GET_GRAY_INITIAL_ALARMS

Specifies whether you want to receive Gray and Initial alarms. If you do not want Gray or Initial alarms, set this parameter to FALSE. This setting reduces the network traffic that AlarmNotifier generates and improves its performance.

GET_EXISTING_ALARMS

Specifies whether you want to receive reports about the alarms that exist when AlarmNotifier is invoked. Otherwise, you only receive reports of alarms that occur after AlarmNotifier is invoked. The "Waiting for more alarms from the SpectroSERVERs" message appears during any interval between alarm notifications.

Reinstalling or Upgrading CA Spectrum

When you reinstall CA Spectrum or you upgrade the version of CA Spectrum, the install process automatically saves the SetScript, UpdateScript, and ClearScript to a backup directory. Versions of the default scripts that you have saved under another name, for example SetScript_version1 or UpdateScript_modified, are retained in the <*$SPECROOT*>/Notifier directory. That directory also contains the default scripts that are included with the reinstallation or upgrade.

In addition, the install process saves your .alarmrc file to a backup directory. Versions of the .alarmrc resource file that you have saved under another name, .alarmrc1 or .alarmrc2 for example, are retained in the <*$SPECROOT*>/Notifier directory.

The backup scripts and the backup .alarmrc are saved to the following directory:

```
<$SPECROOT>/Install-Tools/SAVES_<date>/<time>/Notifier
```

Index

.

.alarmrc
- customizing • 30
- resource file parameters • 31

A

AlarmNotifier
- limiting output of scripts • 23
- optimizing performance • 30
- starting • 11
- stopping • 12

alarms
- monitoring/notification process • 8, 9

C

contacting technical support • 3
customer support, contacting • 3

D

device type • 28

E

email, enabling in scripts • 21
EXTRA_ATTRS_AS_ARGS • 24
EXTRA_ATTRS_AS_ENVVARS • 24

P

performance, reducing network traffic • 30
persistent alarms • 19

R

residual alarms • 19

S

scripts, overview • 15
SENDMAIL, email script parameter • 21
Spectrum Alarm Notification Manager (SANM) • 9
stale alarms • 19
support, contacting • 3

T

technical support, contacting • 3

V

VARFORMAIL, email script parameter • 21