Syslogs and Traps

SL1 version 8.14.0

Table of Contents

Introduction to Syslogs and Traps 3
Appliances that Process Syslog and SNMP Trap Messages 4
Multi-byte Character Support 4
SNMP Traps 5
What Happens When a Message Collector Receives an SNMP Trap 6
Traps That Do Not Match Event Policies 7
Traps From Unknown Devices 7
Filtering Traps 8
Global Settings that Affect SNMP Trap Processing 10
System Settings that Affect SNMP Trap Processing 11
Manually Updating Varbind OIDs 12
Configuring SNMPv3 Traps 12
Syslog Messages 15
Syslogs That Do Not Match Event Policies 16
Syslogs From Unknown Devices 16
IP Address Conflicts 18
IP Addresses Associated with Devices 19
IP Conflict Events 19
Resolving IP Conflicts 20
Event Policies for Syslogs and Traps 23
Creating a Trap Event Policy 24
Defining Pattern Matching and Advanced Behavior 27
Example Trap Event Policy 29
Creating a Syslog Event Policy 33
Defining Pattern Matching and Advanced Behavior 35
Example Syslog Event Policy 37

Chapter 1

Introduction to Syslogs and Traps

Overview

This manual describes how Syslog and SNMP Trap messages are processed by SL1 appliances that perform Message Collection.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ( ).
- To view a page containing all of the menu options, click the Advanced menu icon ( ).

This chapter includes the following topics:

Appliances that Process Syslog and SNMP Trap Messages 4
Multi-byte Character Support 4

Appliances that Process Syslog and SNMP Trap Messages

In SL1, three types of Appliances can process Syslog and SNMP Trap messages from monitored devices. The following appliances can perform the Message Collection function:

- All-In-One Appliances
- Message Collectors
- Data Collectors

> **NOTE:** A Data Collector can perform Message Collection only if that Data Collector is in a Collector Group that contains no other Data Collectors

For more information about SL1 appliances functions and architecture, see the ***Architecture*** manual.

For information on how to create a collector group, see the ***System Administration*** manual.

Multi-byte Character Support

SL1 supports inbound syslog and SNMP trap messages that include multi-byte characters. Multi-byte characters can be displayed in the following pages:

- The **Event Console** page (**[Events]** tab) can display multi-byte characters in syslog and SNMP trap event messages.
- The **Device Logs** page (**[Logs]** tab under the Device Administration panel and the Device Reports panel) can display multi-byte characters in syslog and SNMP trap log messages.
- The ***Ticket Description*** and ***Ticket Notes*** fields in the **Ticket Editor** page can display BMP characters populated from an event message by an automation action. SMP characters are not supported in these fields.

Multi-byte characters can be included in the following fields and functions:

- Outbound SNMP Trap messages generated by the automation engine can now include an event message that contains multi-byte characters.
- Multi-byte characters can be included in the ***Event Message***, ***First Match***, ***Second Match***, and ***Identifier Pattern*** fields in the **Event Policy Editor** page.
- Multi-byte characters can be included in the ***Varbind OID Pattern*** field in an SNMP Trap Filter (Registry > Events > SNMP Trap Filters).
- Multi-byte characters can be included in the ***Expression Match*** field in a Redirect Policy (**[Redirects]** tab under the Device Administration panel).

Chapter 2

SNMP Traps

Overview

This chapter describes how SL1 handles SNMP traps.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ( ).
- To view a page containing all of the menu options, click the Advanced menu icon ( ).

This chapter includes the following topics:

What Happens When a Message Collector Receives an SNMP Trap 6
Traps That Do Not Match Event Policies 7
Traps From Unknown Devices 7
Filtering Traps 8
Global Settings that Affect SNMP Trap Processing 10
System Settings that Affect SNMP Trap Processing 11
Manually Updating Varbind OIDs 12
Configuring SNMPv3 Traps 12

What Happens When a Message Collector Receives an SNMP Trap

When an appliance that performs Message Collection receives an SNMP Trap, it performs the following:

1. If the trap matches a defined filter, the trap is discarded. See ***Filtering Traps***.
2. Matches the IP address of the sender to an IP address of a device monitored by a collector group that includes the Appliance.
    - If the IP address of the sender does not match an IP address of a device monitored by a collector group that includes the Appliance, the message is discarded and a log message is generated. See ***Traps From Unknown Devices***.
3. Using the MIBs compiled on the SL1 system, translates varbind OIDs to symbolic values.

> **NOTE**: By default, Message Collectors and Data Collectors are not populated with information about all varbind OIDs. The first time a Message Collector or Data Collector attempts to translate a specific varbind OID, that varbind OID will not be translated, but information about that varbind OID will be added to theMessage Collector or Data Collector. All instances of a varbind OID after the first will then be translated correctly. To make SL1 translate the first occurrence of a varbind OID correctly, you can manually run a process that pre-populates Message Collectors and Data Collectors with information about all varbind OIDs. For steps on how to run this process, see the ***Manually Updating Varbind OIDs***.

4. Compares the trap to the defined trap event policies:
    - If the trap does not match an event policy, the trap is logged in the Device Logs for the device that sent the trap. See ***Traps That Do Not Match Event Policies***.
    - If the trap does match an event policy, the Source Host Varbind value for the event policy is evaluated. If the Source Host Varbind value matches a varbind OID in the trap, and the value of the varbind matches an IP address or hostname of a device monitored by a collector group that includes the Message Collector, the event is generated and aligned with the device with that IP address or hostname.
    - If the trap does match an event policy and is not realigned using the Source Host Varbind value, the event is generated and aligned with the device the trap was matched with in step two.

> **NOTE:** By default, the event policy "Trap: Unknown trap received" is enabled. This event policy matches all traps that do not match other event policies.

For more information on Trap events, see the ***Events*** manual.

Traps That Do Not Match Event Policies

If an Appliance that performs Message Collection receives a trap that:

- Is from a device that is monitored by a collector group that includes the Message Collector.
- Does not generate an event.

SL1 will log the receipt of the trap in the device logs for the device. If SL1 includes a compiled MIB that contains OIDs used in the received trap, SL1 will include the symbolic translation of those OIDs in the log message. The Device Log will have the following format:

```
Trap Received | Trap Detail: varbind OID or symbolic translation: varbind data type:
varbind data; (Trap OID: trap OID)
```

NOTE: Device Logs that are not associated with an Event are retrieved from Collection Units at five-minute intervals. It may take up to five minutes for traps that do not match event policies to appear in the Device Logs.

Traps From Unknown Devices

If an Appliance that performs Message Collection receives a trap from an unknown device, a "From unknown device: <*ip-address-of-unknown-device*>, received the following Trap message:" event will be generated. An unknown device is defined as either:

- A device monitored by the SL1 system, but by a collector group that does not include the Appliance.
- A device not monitored by the SL1 system.

The "From unknown device: <*ip-address-of-unknown-device*>, received the following Trap message:" event will appear in the Event Console page associated with the System organization.

For the first trap received from an unknown device, the event will have a Severity value of "Notice". If multiple traps are received from the same unknown device, additional events will be generated at the following thresholds:

- ***10, 25 Traps Received***. Severity value of "Minor".
- ***100 Traps Received, and every 100 traps up to and including 900 Traps Received***. Severity value of "Minor".
- ***1,000 Traps Received, and every 1,000 traps up to and including 9,000 Traps Received***. Severity value of "Minor".
- ***10,000 Traps Received, and every 10,000 traps received thereafter***. Severity value of "Major".

NOTE: The counters for the number of traps received from unknown devices will be reset to zero if the Event Engine on the Appliance that performs Message Collection is restarted, or the Appliance is restarted.

NOTE: The default threshold for incoming traps is set to 25 messages per second to prevent degraded performance.

Filtering Traps

In some situations, you might want to filter or limit the traps that are processed by SL1. SNMP Trap Filters allow you to define policies that filter incoming traps to an Appliance that performs Message Collection. When a trap is filtered, the Appliance that performs Message Collection receives the trap, but does not store the trap, does not act on the trap, and does not pass the trap on to be examined by the ScienceLogic event engine.

You can filter incoming SNMP traps using one, multiple, or all of the following parameters:

- IP or hostname of the host that sent the trap. You can also specify "all hosts"
- Trap OID
- Varbind OID
- Varbind content

So you can:

- Filter all incoming traps from a specific host.
- Filter incoming traps with a specific trap OID from all hosts.
- Filter incoming traps with a specific trap OID and from a specific host.
- Filter traps with a specific trap OID and specific varbind OID from all hosts.
- Filter traps with a specific trap OID and specific varbind OID from a specific host.

To create an SNMP Trap Filter, perform the following steps:

1. Go to Registry > Events > SNMP Trap Filters. The **SNMP Trap Filters** page is displayed.
2. In the **SNMP Trap Filters** page, select the **[Create]** button. The **SNMP Trap Filter** modal page is displayed.
3. In the **SNMP Trap Filter** modal page, supply a value in the following fields:
    - ***Filter State***. Specifies whether the SNMP Trap Filter is currently active. When the SNMP Trap Filter is active, all incoming traps that match the criteria in the filter are dropped, and the Appliance does not act upon them. Choices are "Enabled" or "Disabled".
    - ***Host Filter***. Specifies hosts to filter-on. All incoming traps sent from the specified host(s) that match the other parameters will be dropped by the Message Collector.

- If you select the checkbox next to the field name, you can enter a host name or an IP address. All incoming traps from the specified host that also match the other parameters will be dropped by the Appliance.
- If you do not select the checkbox next to the field name, this field will contain the value *All*. In this case, incoming traps from all hosts that also match the other parameters will be dropped by the Appliance.

- ***Trap OID Filter***. Specifies the trap OID to filter on. All incoming traps that are named with the specified OID(s) and match the other parameters will be dropped by SL1.
  - If you select the checkbox next to the field name, you can enter an OID value in standard dotted-decimal notation in this field. All incoming traps that are named with the specified OID that also match the other parameters will be dropped by the Appliance.
  - If you do not select the checkbox next to the field name, this field will contain the value *All*. In this case, all incoming traps named with all OIDs that also match the other parameters will be dropped by the Appliance.

- ***Varbind OID Filter***. A varbind consists of an object, specified by an OID, and its value. In this field, you specify the varbind OID to filter on. All incoming traps that contain the specified varbind OID and also match the other parameters will be dropped by the Appliance.
  - If you select the checkbox next to the field name, you can enter an OID value in standard dotted-decimal notation in this field. All incoming traps that contain that varbind OID and match the other parameters will be dropped by the Appliance.
  - If you do not select the checkbox next to the field name, this field will contain the value *All*. In this case, all incoming traps that contain all OIDs will be dropped by the Appliance.

- ***Varbind OID Pattern***. A varbind consists of an object, specified by an OID, and its value. In this field, you specify a pattern to search for in the varbind value. All incoming traps that contain a varbind value with this pattern and also match the other parameters will be dropped by the Appliance.
  - If you select the checkbox next to the field name, you can enter an alpha-numeric pattern, including multi-byte characters, to search for. All incoming traps that contain a varbind with that value and also match the other parameters will be dropped by the Appliance.
  - If you do not select the checkbox next to the field name, this field will contain the value *All*. In this case, all incoming traps that contain all varbind values that also match the other parameters will be dropped by the Appliance.

4. Select the **[Save]** button to save the new SNMP Trap Filter.
5. The new SNMP Trap Filter should now appear in the **SNMP Trap Filters** page. If the filter is enabled, SL1 will not store or process traps that meet the filter criteria.

To edit an SNMP Trap Filter, perform the following steps:

1. Go to Registry > Events > SNMP Trap Filters. The **SNMP Trap Filters** page is displayed.
2. In the **SNMP Trap Filters** page, find the filter you want to edit. Select its wrench icon (). The **SNMP Trap Filter** modal page is displayed.

3. In the **SNMP Trap Filter** modal page, change the values in one or more fields.
4. Select the **[Save]** button to save your changes to the SNMP Trap Filter.

To delete an SNMP Trap Filter, perform the following steps:

1. Go to Registry > Events > SNMP Trap Filters. The **SNMP Trap Filters** page is displayed.
2. In the **SNMP Trap Filters** page, find the filter you want to delete. Select its checkbox (☑). To select all checkboxes for all filters, select the big checkbox icon (☑) at the top of the page.
3. In the ***Select Action*** drop-down list, select *Delete filter definitions*. Select the **[Go]** button.
4. The selected SNMP Trap Filters will be deleted. SL1 will stop filtering the incoming SNMP traps that were previously filtered with the deleted SNMP Trap Filters.

Global Settings that Affect SNMP Trap Processing

The following global setting affects how SL1 processes SNMP traps:

- ***use_v1trap_envelope_addr***. In environments where Network Address Translation is performed on SNMP v1 trap messages sent to SL1, you can configure SL1 to read the envelope address (the address of the host sending the trap) instead of the agent address (the IP address variable sent as part of the trap). To use the envelope address instead of the agent address for SNMP v1 trap messages, the *use_v1trap_envelope_addr=1* configuration option can be added to the [LOCAL] section of silo.conf on Message Collectors, Data Collectors that perform message collection, and All-In-One Appliances. If *use_v1trap_envelope_addr* is not defined in silo.conf or *use_v1trap_envelope_addr=0* is defined, SL1 will use the agent address for SNMP v1 trap messages.

To add a settings to the silo.conf file on an appliance:

1. Either go to the console of the SL1 appliance or use SSH to access the server.
2. Login as user ***em7admin*** with the password you configured during setup.
3. At the shell prompt, enter the following:

```
sudo vi /etc/siteconfig/siloconf.siteconfig
```

4. On a line of its own, add the new entry.
5. Save your changes and exit the file (:wq).
6. At the command line, enter the following command to rebuild the configuration file:

```
sudo /opt/em7/share/scripts/generate-silo-conf.py > /etc/silo.conf
```

System Settings that Affect SNMP Trap Processing

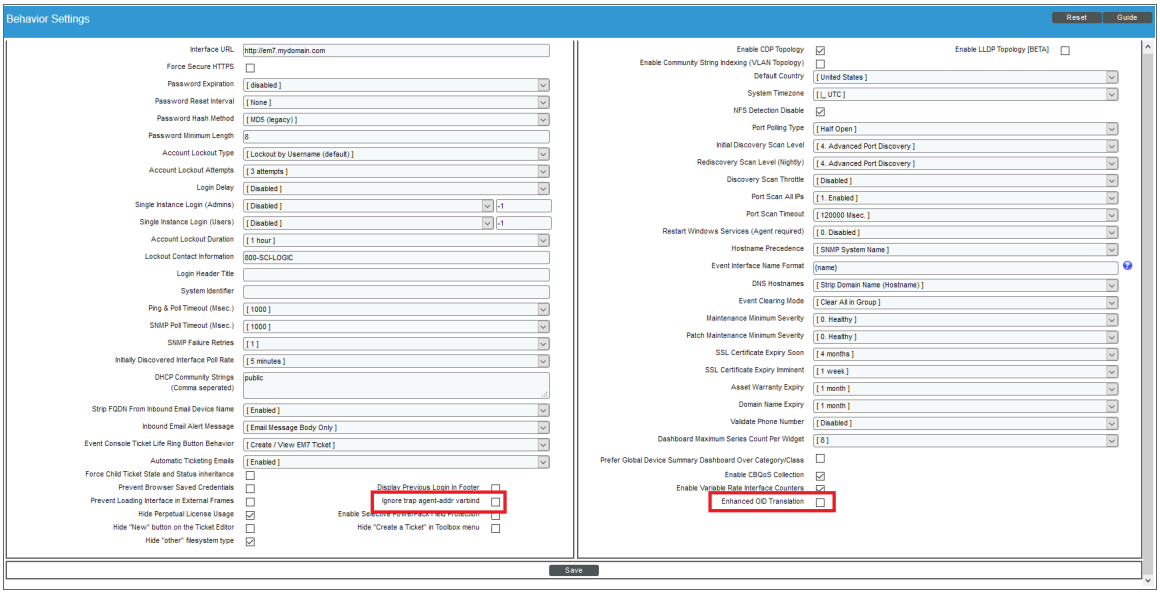
The following system setting affects how SL1 processes SNMP traps:

- ***Ignore trap agent-addr varbind***. If you select this checkbox, SL1 will align the SNMP trap with the forwarder (last hop) instead of searching for the IP address of the originator of the trap.
- ***Enhanced OID Translation***. If selected, ensures that varbind OIDs that use multi-dimensional indexes are translated correctly. The symbolic translation of the known portion of the OID is included in the log message associated with the trap.

NOTE: Enabling the ***Enhanced OID Translation*** option might affect performance on large environments with a large number of traps.

To enable these settings:

1. Go to the **Behavior Settings** page (System > Settings > Behavior).
2. Select the checkbox next to the setting or settings you want to enable:



3. Click **[Save]** to save the settings.

Manually Updating Varbind OIDs

By default, Message Collectors and Data Collectors are not populated with information about all varbind OIDs. The first time a Message Collector or Data Collector attempts to translate a specific varbind OID, that varbind OID will not be translated, but information about that varbind OID will be added to the Message Collector or Data Collector. All instances of a varbind OID after the first will then be translated correctly.

To make SL1 translate the first occurrence of a varbind OID correctly, you can manually run a process that pre-populates Message Collectors and Data Collectors with information about all varbind OIDs. You should run this process after adding new MIBs to SL1.

To manually populates Message Collectors and Data Collectors with information about all varbind OIDs, perform the following steps:

1. Go to the **OID Browser** page (System > Tools > OID Browser).
2. Select the **[Update]** button.

Configuring SNMPv3 Traps

To configure a Message Collector or Data Collector to accept an SNMPv3 trap or inform, you must edit the trap configuration file on the Message Collector or Data Collector. In the trap configuration file, enter the credentials that allow SL1 to communicate with the device that sends traps to SL1. Enter credentials for each device that sends traps to SL1. This information is configured in a configuration file at the operating-system level.

For SNMPv3 traps, the credential entry must include the engine ID of the device sending the trap. Therefore, there will be an entry in the trap configuration for each device that will send SNMPv3 traps. For SNMPv3 informs, the entry does not need to specify the engine ID. The engine ID of the recipient, the Message Collectoror Data Collector, is used for SNMPv3 informs. Therefore, if all the managed devices use the same credentials to send SNMPv3 informs, you need to add only one entry to the trap configuration file.

NOTE: Existing trap event policies will be triggered by SNMPv3 traps and SNMPv3 informs with no additional configuration.

To add a credential entry to the trap configuration file:

1. Either go to the console of the the Message Collector or Data Collector or use SSH to access the server.
2. Log in as user ***em7admin*** with the appropriate password.
3. Open the /etc/snmp/snmptrapd.conf file in a text editor.
4. At the end of the file, add a new "createUser" line. See the section below for the appropriate syntax.
5. Save the file.
6. Restart the trap engine by executing the following command:

```
sudo service snmptrapd restart
```

The syntax of createUser is different for each security level and whether you are configuring traps or informs:

- Informs, no authentication, no encryption (noAuthNoPriv):

  ```
  createUser <security name>
  ```

  For example:

  ```
  createUser em7defaultv3
  ```

- Informs, authentication, no encryption (authNoPriv):

  ```
  createUser <security name> <auth protocol> <security passphrase>
  ```

  For example:

  ```
  createUser em7defaultv3 SHA em7authpass
  ```

NOTE: For FIPS-compliant systems, authentication with MD5 will fail.

- Informs, authentication and encryption (authPriv):

  ```
  createUser <security name> <auth protocol> <security passphrase> <privacy
  protocol> <privacy pass phrase>
  ```

  For example:

  ```
  createUser em7defaultv3 SHA em7authpass DES em7privpass
  ```

- Traps, no authentication, no encryption (noAuthNoPriv):

  ```
  createUser -e <engine ID> <security name>
  ```

  For example:

  ```
  createUser -e 0x0102030405 em7defaultv3
  ```

- Traps, authentication, no encryption (authNoPriv):

  ```
  createUser -e <engine ID> <security name> <auth protocol> <security passphrase>
  ```

  For example:

  ```
  createUser -e 0x0102030405 em7defaultv3 SHA em7authpass
  ```

- Traps, authentication and encryption (authPriv):

  ```
  createUser -e <engine ID> <security name> <auth protocol> <security passphrase>
  <privacy protocol> <privacy pass phrase>
  ```

For example:

```
createUser -e 0x0102030405 em7defaultv3 SHA em7authpass DES em7privpass
```

Here are some example commands for how to send a test coldStart trap from a SL1 appliance using authPriv and the credential information from the examples above:

```
snmptrap -e 0x0102030405 -v3 -u em7defaultv3 -a SHA -A em7authpass -x DES -X
em7privpass <message collector ip> 0 .1.3.6.1.6.3.1.1.5.1
```

```
snmpinform -v3 -u em7defaultv3 -a SHA -A em7authpass -x DES -X em7privpass <message
collector ip> 0 .1.3.6.1.6.3.1.1.5.1
```

Chapter 3

Syslog Messages

Overview

When an Appliance that performs Message Collection receives a Syslog message, it performs the following:

1. Matches the IP address of the sender to an IP address of a device monitored by a collector group that includes the Appliance.
    - If the IP address of the sender does not match an IP address of a device monitored by a collector group that includes the Appliance, the message is discarded and an event is generated. See *Syslogs From Unknown Devices*.
2. Compares the syslog to the defined syslog event policies:
    - If the syslog does not match an event policy, the syslog is logged in the Device Logs for the device that sent the syslog. See *Syslogs That Do Not Match Event Policies*.
    - If the syslog matches an event policy, the event is generated. The generated event is aligned with the device the syslog was matched with in step 1.

For more information on syslog events, see the ***Events*** manual.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ( ).
- To view a page containing all of the menu options, click the Advanced menu icon ( ).

This chapter includes the following topics:

Syslogs That Do Not Match Event Policies *16*
Syslogs From Unknown Devices *16*

Syslogs That Do Not Match Event Policies

If an Appliance that performs Message Collection receives a syslog that:

- Is from a device that is monitored by a collection group that includes the Appliance.
- Does not generate an event.

SL1 will log the receipt of the syslog in the device logs for the device. The message field for the Device Log will be the same as the syslog message field.

NOTE: Device Logs that are not associated with an Event are retrieved from Collection Units at five-minute intervals. It may take up to five minutes for syslogs that do not match event policies to appear in the Device Logs.

Syslogs From Unknown Devices

If an Appliance the performs Message Collection receives a syslog from an unknown device, a "From unknown device: *<ip-address-of-unknown-device>*, received the following syslog message:" event will be generated. An unknown device is defined as either:

- A device monitored by the SL1 system, but by a collector group that does not include the Appliance.
- A device not monitored by the SL1 system.

The "From unknown device: *<ip-address-of-unknown-device>*, received the following syslog message:" event will appear in the **Event Console** page associated with the System organization.

For the first syslog received from an unknown device, the message will have a Severity value of "Notice". If multiple syslogs are received from different unknown devices, additional events will be generated at the following thresholds:

- ***10, 25 Syslogs Received***. Severity value of "Minor".
- ***100 Syslogs Received, and every 100 syslogs up to and including 900 Syslogs Received***. Severity value of "Minor".
- ***1,000 Syslogs Received, and every 1,000 syslogs up to and including 9,000 Syslogs Received***. Severity value of "Minor".
- ***10,000 Syslogs Received, and every 10,000 syslogs received thereafter***. Severity value of "Major".

NOTE: Multiple messages received from the same unknown device will not increase the event count of syslog messages received or the event severity.

NOTE: The counters for the number of syslogs received from unknown devices will be reset to zero if the Event Engine on an Appliance that performs Message Collection is restarted, or the Appliance is restarted.

NOTE: The default threshold for incoming syslogs is set to 25 messages per second to prevent degraded performance.

Chapter 4

IP Address Conflicts

Overview

This chapter describes how SL1 handles IP address conflicts.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ( ).
- To view a page containing all of the menu options, click the Advanced menu icon ( ).

This chapter includes the following topics:

IP Addresses Associated with Devices 19
IP Conflict Events 19
Resolving IP Conflicts 20

IP Addresses Associated with Devices

There are three types of IP addresses that can be associated with a device:

- ***Admin Primary***. This is the IP address that SL1 used to discover a device, and is used by Data Collectors to communicate with a device. This IP address is always the ***Admin Primary*** address and cannot be demoted to a secondary address.
- ***Primary***. One or more IP addresses that SL1 uses to match incoming syslog an trap messages with a device.
- ***Secondary***. SL1 gathers information about this IP address, but does not use this IP address to communicate with the device or match incoming syslog or trap messages with a device.

SL1 will allow devices with the same admin primary IP address to be monitored; however, devices with the same admin primary IP address must be in separate collector groups. The admin primary IP address is the IP address SL1 uses to monitor a device, and is listed in the "IP Address" column in the **Device Manager** page (Registry > Devices > Device Manager).

A Message Collector can be aligned with multiple collector groups. Because Message Collectors can be included in multiple collection groups, it is possible for the IP address associated with a syslog or trap to match multiple devices.

This chapter describes how a Message Collector reports IP conflicts in this situation.

NOTE: The information in this chapter does not apply to Data Collectors and All-In-One Appliances because Data Collectors and All-In-One Appliances can be in only one Collector Group.

IP Conflict Events

For each Message Collector, daily maintenance compares the IP addresses for all devices monitored by the collector groups that include the Message Collector. If the daily maintenance task finds duplicate admin primary IP addresses, SL1 generates the following event, with a default severity of major:

```
Primary IP address overlap on devices managed by Message Collector: <appliance-id-of-
message-collection-unit> | Collector Groups: <id-of-collector-groups> | IP Address:
<duplicate-ip-address> | Device IDs: <device-ids-using-ip-address>
```

If the daily maintenance task finds duplicate secondary IP addresses, SL1 generates the following event, with a default severity of minor:

```
Secondary IP address overlap on devices managed by Message Collector: <appliance-id-
of-message-collection-unit> | Collector Groups: <id-of-collector-groups> | IP Address:
<duplicate-ip-address> | Device IDs: <device-ids-using-ip-address>
```

When a Message Collector is:

- Aligned with multiple collector groups
- Receives a syslog or trap from a primary IP address associated with multiple devices
- The IP address is associated with multiple devices, all of which are are monitored by the same collector group that contains the the Message Collector

SL1 generates the following event, with a default severity of minor:

```
Could not match asynchronous message to a device due to a primary IP address ambiguity
address: <duplicate-ip-address>
```

If a received syslog or trap triggers the address ambiguity event, and the Message Collector is discovered on the system, any events or logs generated by the syslog or trap are aligned with the Message Collector. If a received syslog or trap causes the address ambiguity event to be generated, and the Message Collector is not on the system, any events or logs generated by the syslog or trap are aligned with the System organization.

Resolving IP Conflicts

To prevent syslog and trap messages from aligning with the Message Collector or System organization because of an IP conflict, every device monitored by the same Message Collector must use a unique IP address to send syslog and trap messages. Even if these devices that share an IP address are in different collector groups, if the devices share one or more Message Collectors, the devices should use unique IP addresses to send syslog and trap messages.

By default, SL1 uses only the admin primary IP address to align syslog and trap messages to devices. If the admin primary IP address for a device is not unique, you can configure a secondary IP address for use as a primary IP address for message collection.

NOTE: Configuring a secondary IP address as a primary IP address for message collection will not affect any data collection performed byData Collectors. Data Collectors will always use the admin primary IP address when polling devices.

To configure a secondary IP address for a device as a primary IP address for message collection:

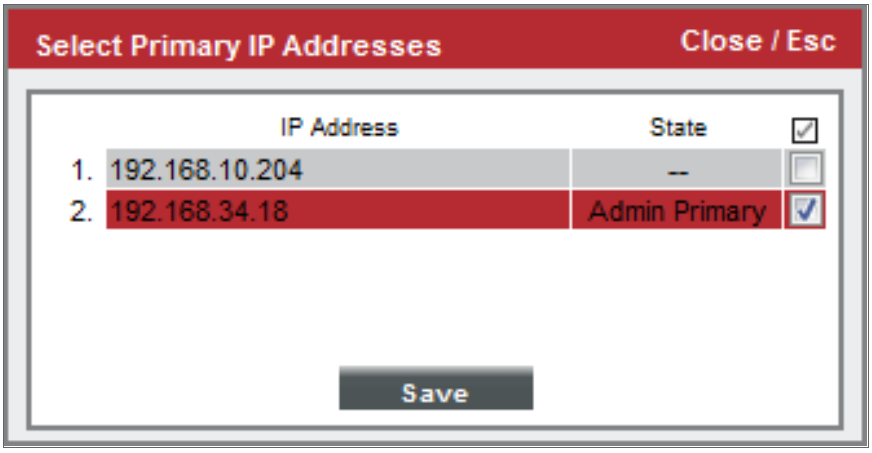
1. Go to the **Device Manager** page (Registry > Devices > Device Manager).
2. Select the wrench icon ( )for the device you want to configure. The **Device Properties** window will be displayed.

3. To check that SL1 has discovered the secondary IP address that you want to configure as the primary IP address for message collection, select the ***IP Address*** drop down list. If the secondary IP address is not displayed in the list of IP addresses, you can add it manually:

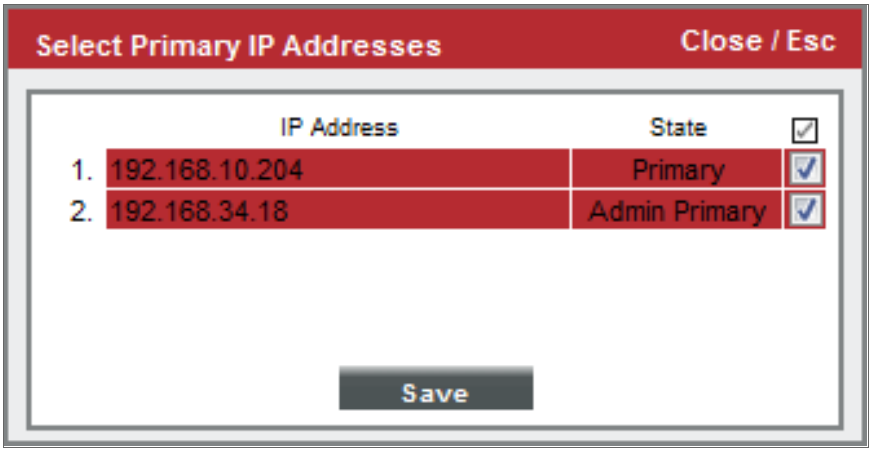
    - Select the plus icon to the right of the ***IP Address*** drop down list. The **Add IP Address** modal window is displayed:

    - Enter the secondary IP address in the ***IP Address*** field.
    - Enter the subnet mask for the secondary IP address in the ***Subnet Mask*** field.
    - Select the **[Save]** button. The **Add IP Address** modal window will close and the message "Unverified IP Added to Device" is displayed.

4. From the **[Actions]** menu, select ***Select Primary IP Addresses***. The **Select Primary IP Addresses** modal window is displayed:



5. Select the checkbox for the secondary IP address you want to configure as a primary IP address. Select the **[Save]** button. The *State* of the selected IP address is now "Primary":



NOTE: You cannot change the state of the admin primary address. If a listed IP address is already in use as an admin primary or primary IP address for another device in the same collector group, you cannot set it as a primary IP address and the checkbox will not be displayed. You can select multiple secondary IP addresses to set as primary addresses.

Chapter 5

Event Policies for Syslogs and Traps

Overview

This chapter describes how to set up Event Policies for events with a source of Syslog and Trap messages.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon ( ).
- To view a page containing all of the menu options, click the Advanced menu icon ( ).

This chapter includes the following topics:

Creating a Trap Event Policy *24*
Example Trap Event Policy *29*
Creating a Syslog Event Policy *33*
Example Syslog Event Policy *37*

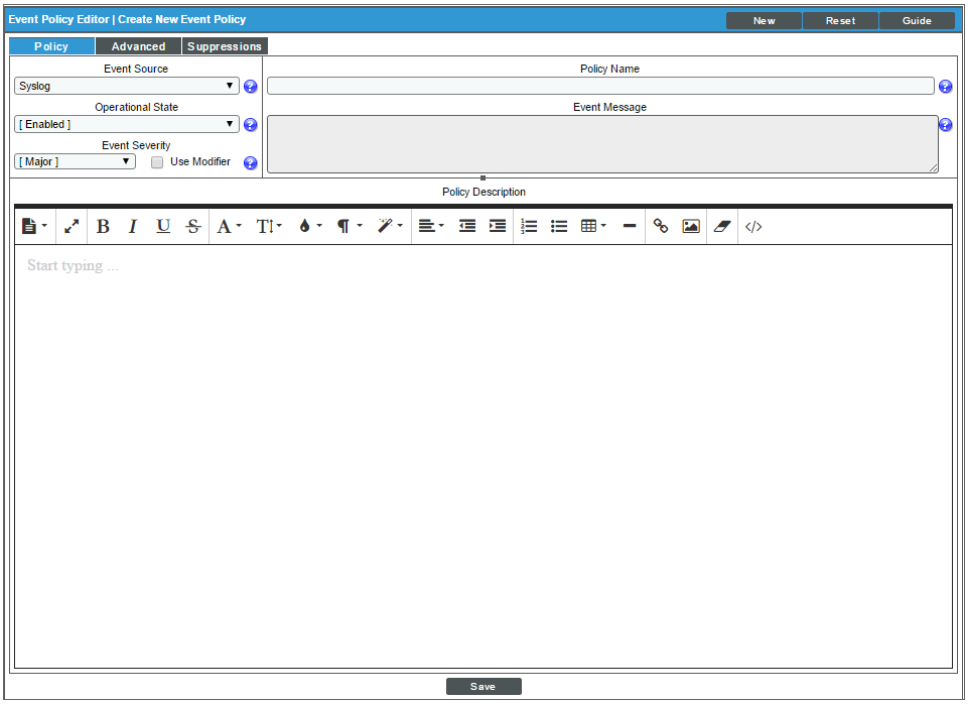
Creating a Trap Event Policy

SL1 includes pre-defined events for the most commonly encountered conditions on the most common platforms. However, if the pre-defined events do not meet the needs of your organization, you can define new events that better suit your needs.

From the **Event Policies** page (or the **Event Policy Manager** page in the classic SL1 user interface), you can define a new event. You can define custom events to meet your business requirements. You can also define events to be triggered by any custom Dynamic Application alerts you have created.

To create an event definition:

1. Go to **Event Policy Manager** page (Registry > Events > Event Manager).
2. In the **Event Policy Manager** page, click the **[Create]** button. The **Event Policy Editor** page appears:



3. In the **Event Policy Editor** page and set of tabs, you can define a new event. The **Event Policy Editor** page contains three tabs:

   - ***Policy***. Allows you to define basic parameters for the event. This tab is described in the following section.
   - ***Advanced***. Allows you to define pattern-matching for the event and also define event roll-ups and suppressions.

- ***Suppressions***. Allows you to suppress the event on selected devices. When you suppress an event, you are specifying that, in the future, if this event occurs again on a specific device, the event will not appear in the **Event Console** page or the **Viewing Events** page for the device.

4. Supply values in the following fields:

   - ***Event Source***. Select *Trap*.
   - ***Policy Name***. The name of the event. Can be any combination of alphanumeric characters, up to 48 characters in length.
   - ***Operational State***. Specifies whether event is to be operational or not. Choices are *Enabled* or *Disabled*.
   - ***Event Message***. The message that appears in the **Event Console** page or the **Viewing Events** page when this event occurs. Can be any combination of alphanumeric characters. Variables include the characters "%" (percent) and "|" (bar). You can also use regular expressions and variables that represent text from the original log message to create the ***Event Message***:

     - To include regular expressions in the Event Message:

     Surround the regular expression with %R and %/R. For example:

     %RFilename: .*? %/R

     Would search for the first instance of the string "Filename: " (Filename-colon-space) followed by any number of any characters up to the line break. The %R indicates the beginning of a regular expression. The %/R indicates the end of a regular expression.

     SL1 will use the regular expression to search the log message and use the matching text in the event message.

     For details on the regular expression syntax allowed by SL1, see http://www.python.org/doc/howto/.

     - You can also use the following variables in this field:
       - *%I* ("eye").This variable contains the value that matches the ***Identifier Pattern*** field in the **[Advanced]** tab.
       - *%M*. The full text of the log message that triggered the event will be displayed in ***Event Message*** field.
       - *%V*. Data Value from log file will be displayed in the ***Event Message*** field.
       - *%T*. Threshold value from the log file will be displayed in ***Event Message*** field.

   - ***Event Severity***. Defines the severity of the event. Choices are:

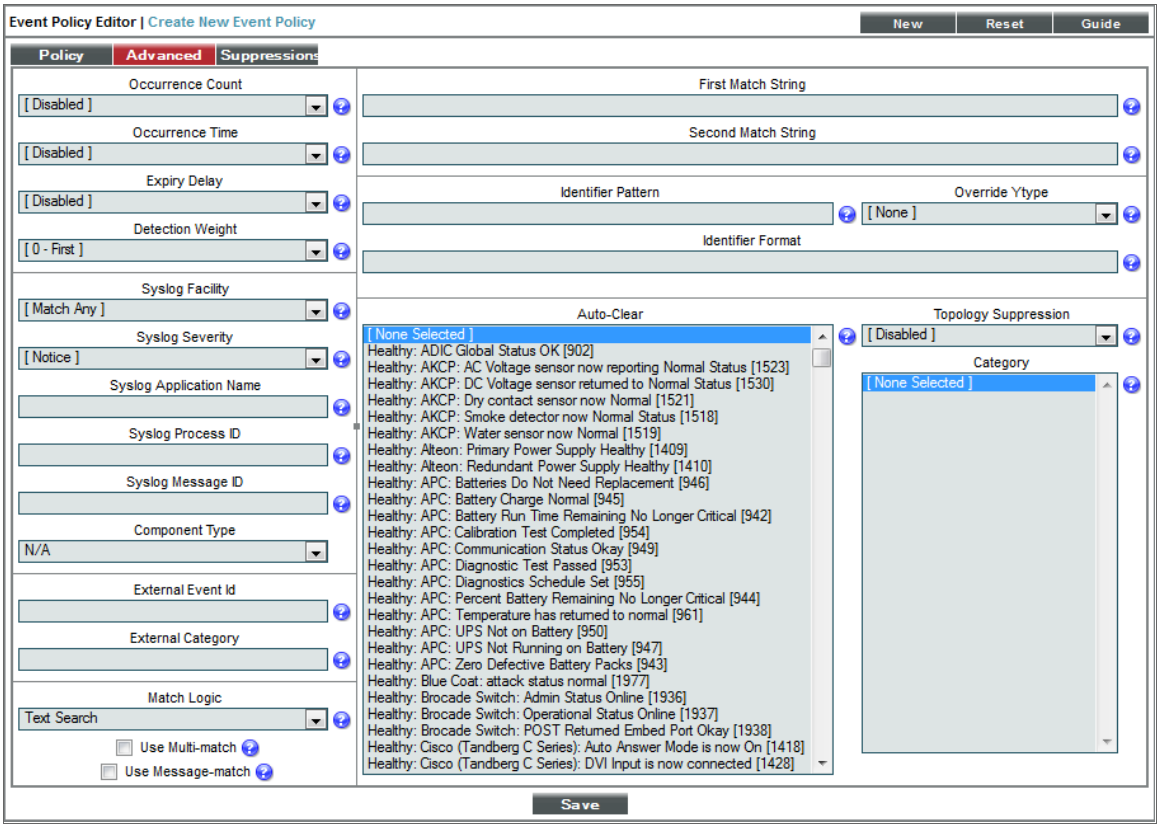
     - *Healthy*. Healthy Events indicate that a device or condition has returned to a healthy state. Frequently, a healthy event is generated after a problem has been fixed.
     - *Notice*. Notice Events indicate a condition that does not affect service but about which users should be aware.

- *Minor*. Minor Events indicate a condition that does not currently impair service, but the condition needs to be corrected before it becomes more severe.
- *Major*. Major Events indicate a condition that is service impacting and requires immediate investigation.
- *Critical*. Critical Events indicate a condition that can seriously impair or curtail service and require immediate attention (i.e. service or system outages).

- ***Use Modifier***. If selected, when the event is triggered, SL1 will check to see if the interface associated with this event has a custom severity modifier. If so, the event will appear in the **Event Console** with that custom severity modifier applied to the severity in the ***Event Severity*** field. For example, if an interface with an ***Event Severity Adjust*** setting of *Sev -1* triggers an event with an ***Event Severity*** of *Major* and that event has the ***Use Modifier*** checkbox selected, the event will appear in the **Event Console** with a severity of *Minor*.
- ***Policy Description***. Text that explains what the event means and what possible causes are.

Defining Pattern Matching and Advanced Behavior

The **[Advanced]** tab in the **Event Policy Editor** page allows you to define or edit pattern-matching for the trap event and also define event roll-ups and suppressions. In the **[Advanced]** tab, you can define or edit the following fields that pertain to traps:



- ***Link-Trap***. For events with a source of *Trap*, displays a list of trap OIDs that are included in the MIB files that have been compiled in SL1. You can either select one of the listed trap OIDs to associate with the event or manually enter a custom trap OID. You can use an asterisk (*) as a wildcard character at the end of the trap OID. If you add the wildcard character to the end of the trap OID, the event policy will match all trap OIDs that start with the specified OID string. This is useful for creating "catch all" event policies.
- ***Source Host Varbind***. For events with a source of *Trap*, specifies an OID that is included in the trap. This OID will contain the IP address to align with the event.
    - If a value is specified in this field, SL1 examines the OID specified in this field. If the value stored in the OID matches the primary IP address of a device in SL1, the resulting event will be aligned with that device.
    - If a value is specified in this field, SL1 examines the OID specified in this field. If the value stored in the OID does not match a primary IP address of a device in SL1, the resulting event will be aligned with the device that sent the trap.

- If no value is specified in this field, but the trap includes the default snmpTrapAddress OID, SL1 will examine the value stored in the snmpTrapAddress OID. If the value stored in the OID matches the primary IP address of a device in SL1, the resulting event will be aligned with that device.
- If no value is specified in this field and the trap does not include the snmpTrapAddress OID, SL1 will align the resulting event with the device that sent the trap.

- ***First Match String***. A string used to correlate the event with a log message. To match this event policy, the text of a log message or alert must match the value you enter in this field. Can be any combination of alphanumeric characters. SL1's expression matching is case sensitive. This field is required for events generated with a source of Syslog, Security, 3rd Party, and Email.
- ***Second Match String***. A secondary string used to match against the originating log message. To match this event policy, the text of a log message or alert must match the value you enter in this field and the value you entered in the ***First Match String*** field. This field is optional.

NOTE: The ***Match Logic*** field specifies whether SL1 should process ***First Match String*** and ***Second Match String*** as simple text matches or as regular expressions.

NOTE: You can define an event so that it is triggered only when it occurs on a specific interface. You can then include the interface name and SL1's unique interface ID for the interface in the event message. When defining an event, you can use the following three fields below to associate an event with an interface.

- ***Identifier Pattern***. A regular expression used to extract the specific subentity (like the name of a network interface) within the log entry. SL1 will use this value as the yName of the interface. By identifying the subentity, SL1 can create a unique event for each subentity, instead of a single event for the entire device. For example, a log message indicating a link has gone down may include the network interface name. So this field could extract the network interface name from the log message. SL1's expression matching is case sensitive. For details on the regular expression syntax allowed by SL1, see http://www.python.org/doc/howto.
- ***Identifier Format***. If the ***Identifier Pattern*** field returns multiple results, users can specify which results to use and in which order. Each result is represented by a variable. This field is optional.
    - %1. First match with identifier pattern. This is the default behavior if no value is supplied in the ***Identifier Format*** field.
    - %2. Second match with identifier pattern.
    - For example, users could specify "%2:%1" for "Interface %2: Peer %1".

Select the **[Save]** button to save your settings when you have finished editing the fields pertaining to your trap event policy.

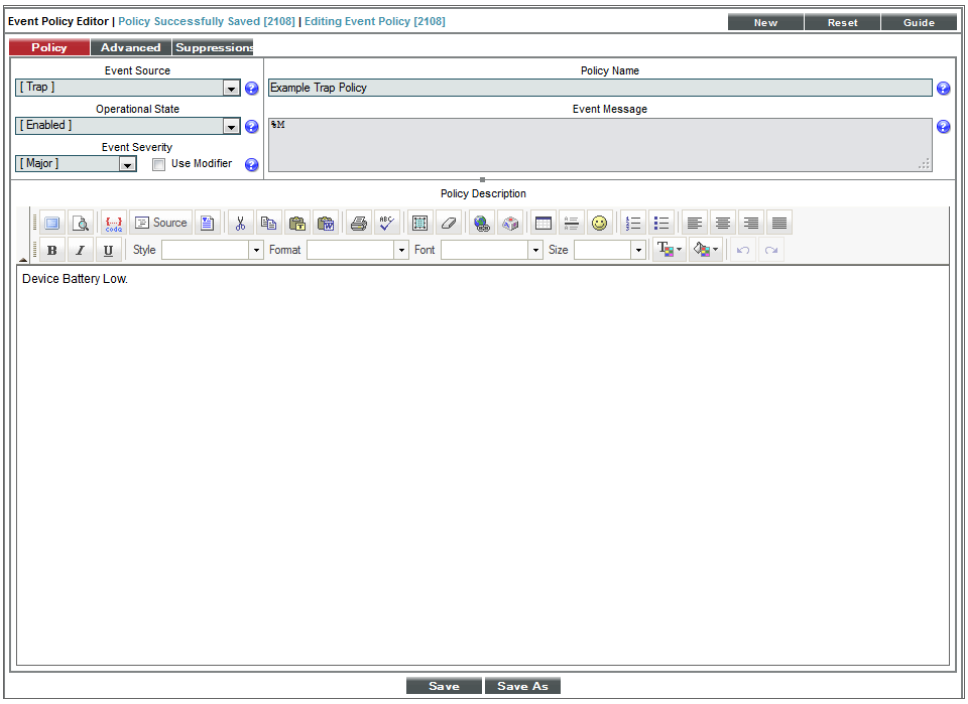
For more information on the remaining fields, as well as the **[Suppressions]** tab, see the ***Events*** manual.

Example Trap Event Policy

Trap messages are sent from devices to SL1 in order to notify the platform of any issues or important events occurring on the device.

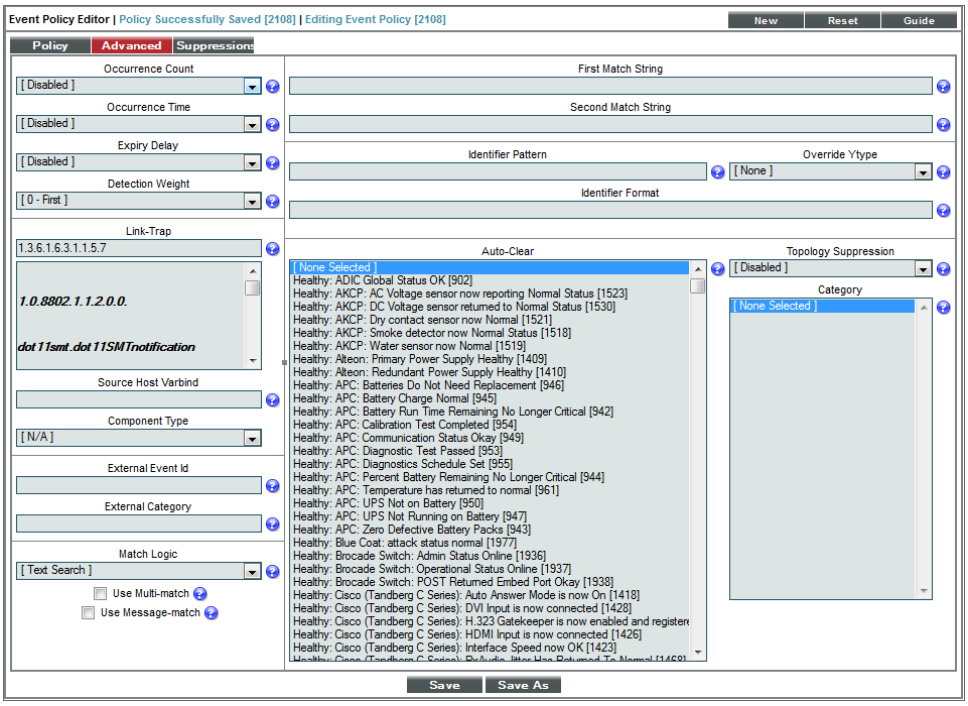
To create a Trap Event Policy:

1. Go to the **Event Policy Manager** page (Registry > Events > Event Manager).
2. Select the **[Create]** button, and the **Event Policy Editor** page will appear.
3. In the **Event Policy Editor** page, enter these values in the following fields:



- ***Event Source***. We selected *Trap*.
- ***Operational State***. We selected *Enabled*.
- ***Event Severity***. We selected *Notice*.
- ***Policy Name***. We entered "Example Trap Policy".
- ***Event Message***. We entered "%M".
- ***Policy Description***. We entered "Device Battery Low."

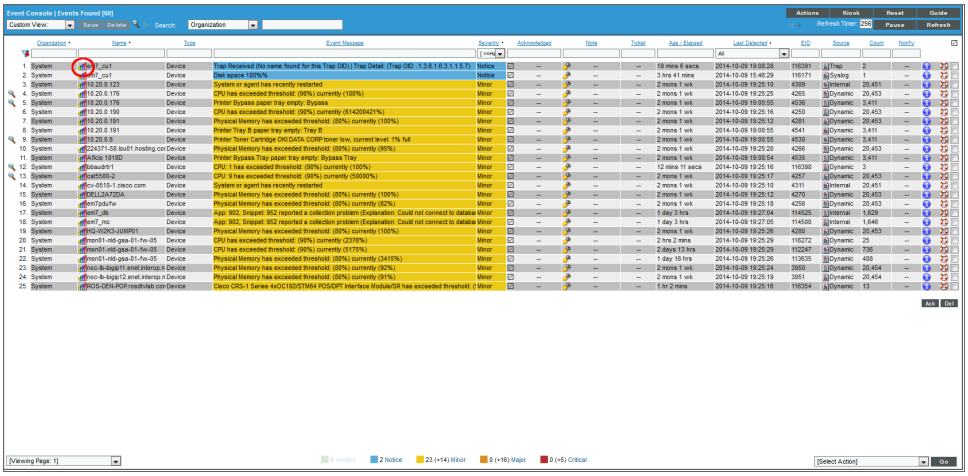
4. Select the **[Save]** button.
5. After saving those settings, select the **[Advanced]** tab. We entered the following values in the following fields:



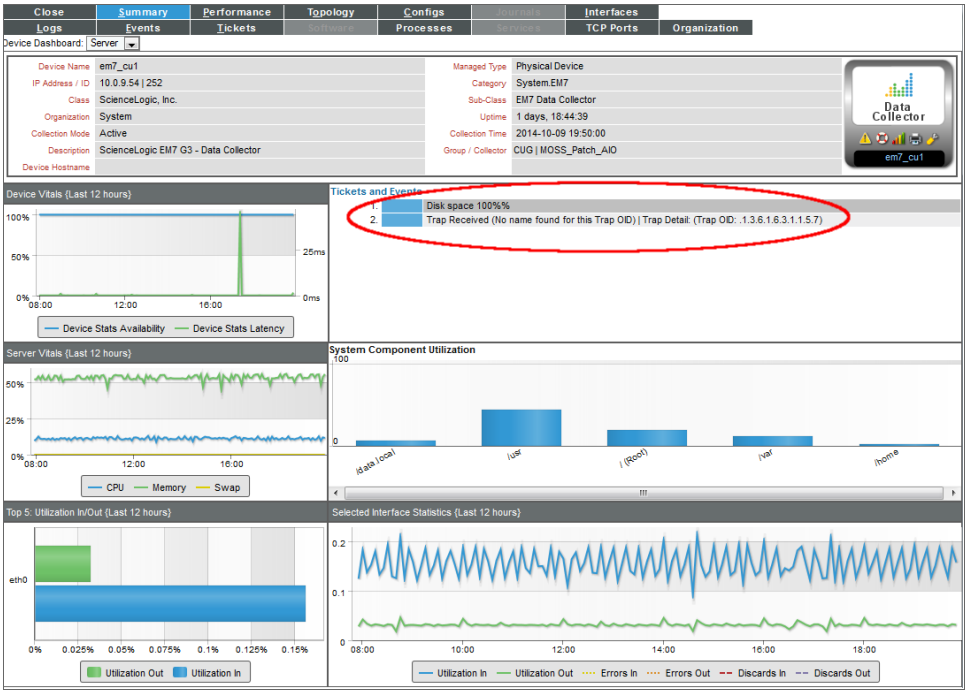
- ***Link-Trap***. We entered the device's trap oid of "1.3.6.1.6.3.1.1.5.7".

6. We left the rest of the fields at their default settings, and then selected the **[Save]** button.

7. When the device's battery is low, it will send the trap message and trigger an event, which appears in the **Event Console**. Clicking on the graph icon ( ) will bring up the **Device Summary** page for the device for which the event occurred. Clicking on the life ring icon ( ) will create a ticket for the event.

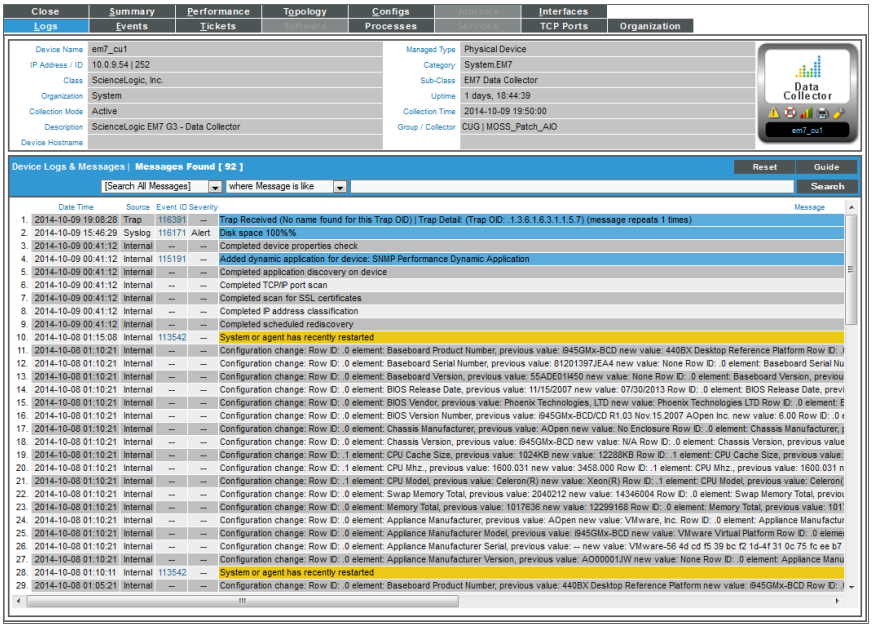


8. Clicking on the graph icon ( ) will bring up the **Device Summary** page. You will see the event listed in the **Device Summary** page, and you can click on the event to view the **Event Summary** modal page.

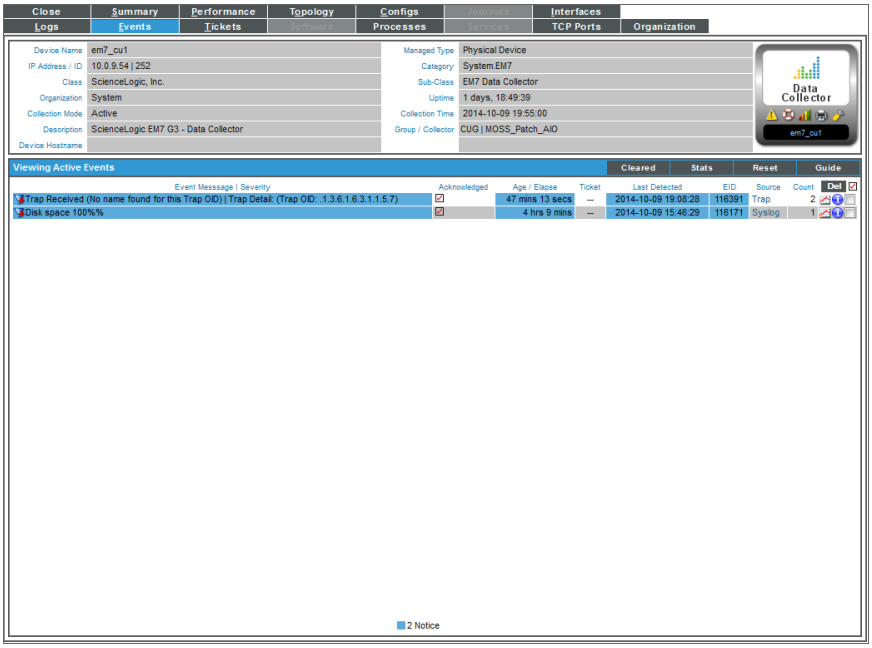


9. You can also select the **[Logs]** tab from the **Device Summary** page to view the **Device Logs & Messages** page. The trap message will appear in the device logs, and you can select the View Events icon ( ) which

will take you to the **Viewing Active Events** page for that device.



10. From the **Viewing Active Events** page, you can select the information icon ( ) to view the **Event Information** modal page, filter the device's events based on event type, or view graphical reports about that device's events based on type.



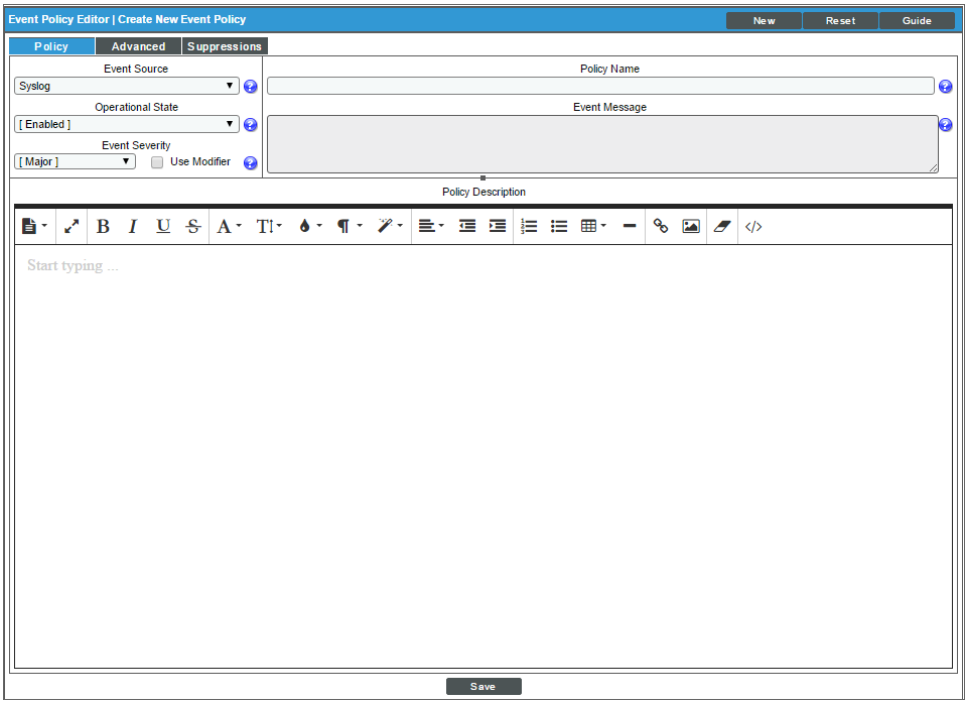
Creating a Syslog Event Policy

SL1 includes pre-defined events for the most commonly encountered conditions on the most common platforms. However, if the pre-defined events do not meet the needs of your organization, you can define new events that better suit your needs.

From the **Event Policies** page (or the **Event Policy Manager** page in the classic SL1 user interface), you can define a new event. You can define custom events to meet your business requirements. You can also define events to be triggered by any custom Dynamic Application alerts you have created.

To create an event definition:

1. Go to **Event Policy Manager** page (Registry > Events > Event Manager).
2. In the **Event Policy Manager** page, click the **[Create]** button. The **Event Policy Editor** page appears:



3. In the **Event Policy Editor** page and set of tabs, you can define a new event. The **Event Policy Editor** page contains three tabs:

   - ***Policy***. Allows you to define basic parameters for the event. This tab is described in the following section.
   - ***Advanced***. Allows you to define pattern-matching for the event and also define event roll-ups and suppressions.

- ***Suppressions***. Allows you to suppress the event on selected devices. When you suppress an event, you are specifying that, in the future, if this event occurs again on a specific device, the event will not appear in the **Event Console** page or the **Viewing Events** page for the device.

4. Supply values in the following fields:
   - ***Event Source***. Select *Syslog*.
   - ***Policy Name***. The name of the event. Can be any combination of alphanumeric characters, up to 48 characters in length.
   - ***Operational State***. Specifies whether event is to be operational or not. Choices are *Enabled* or *Disabled*.
   - ***Event Message***. The message that appears in the **Event Console** page or the **Viewing Events** page when this event occurs. Can be any combination of alphanumeric characters. Variables include the characters "%" (percent) and "|" (bar). You can also use regular expressions and variables that represent text from the original log message to create the ***Event Message***:
     - To include regular expressions in the Event Message:

       Surround the regular expression with %R and %/R. For example:

       %RFilename: .*? %/R

       Would search for the first instance of the string "Filename: " (Filename-colon-space) followed by any number of any characters up to the line break. The %R indicates the beginning of a regular expression. The %/R indicates the end of a regular expression.

       SL1 will use the regular expression to search the log message and use the matching text in the event message.

       For details on the regular expression syntax allowed by SL1, see http://www.python.org/doc/howto/.

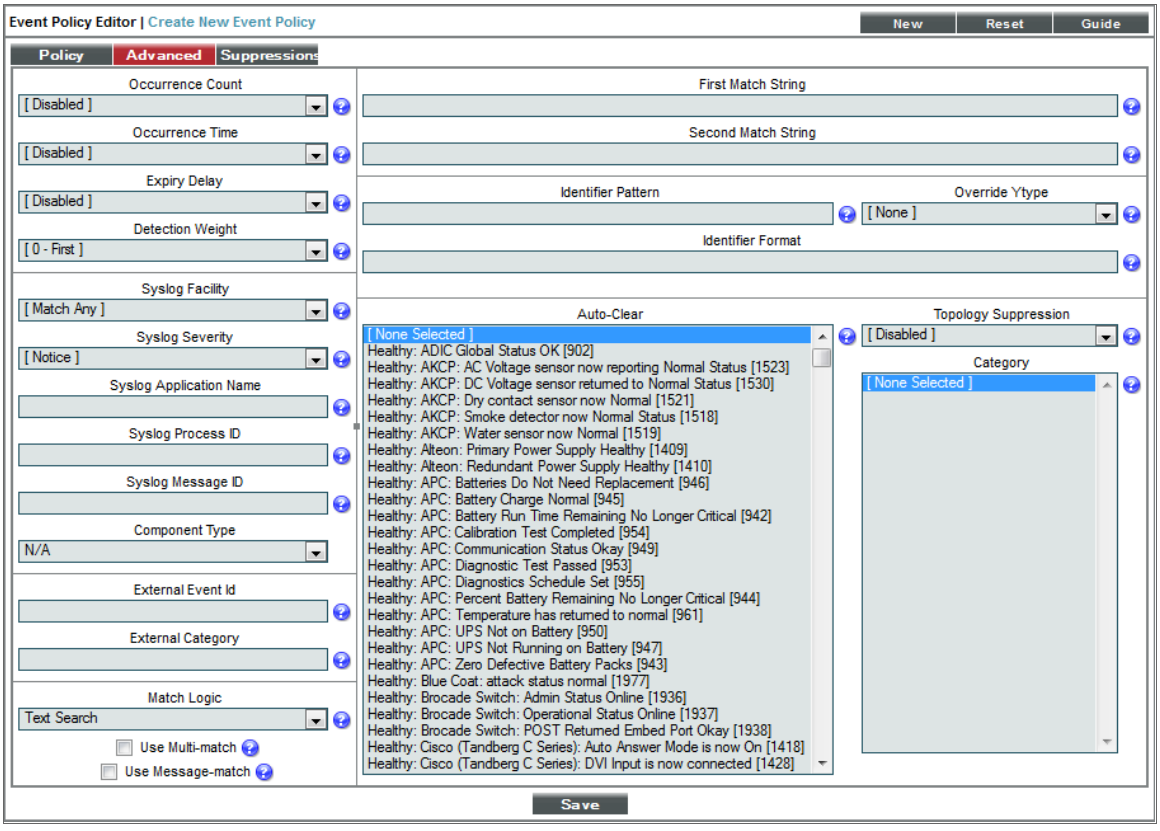
     - You can also use the following variables in this field:
       - *%I* ("eye").This variable contains the value that matches the ***Identifier Pattern*** field in the **[Advanced]** tab.
       - *%M*. The full text of the log message that triggered the event will be displayed in ***Event Message*** field.
       - *%V*. Data Value from log file will be displayed in the ***Event Message*** field.
       - *%T*. Threshold value from the log file will be displayed in ***Event Message*** field.
   - ***Event Severity***. Defines the severity of the event. Choices are:
     - *Healthy*. Healthy Events indicate that a device or condition has returned to a healthy state. Frequently, a healthy event is generated after a problem has been fixed.

- *Notice*. Notice Events indicate a condition that does not affect service but about which users should be aware.
- *Minor*. Minor Events indicate a condition that does not currently impair service, but the condition needs to be corrected before it becomes more severe.
- *Major*. Major Events indicate a condition that is service impacting and requires immediate investigation.
- *Critical*. Critical Events indicate a condition that can seriously impair or curtail service and require immediate attention (i.e. service or system outages).

- ***Use Modifier***. If selected, when the event is triggered, SL1 will check to see if the interface associated with this event has a custom severity modifier. If so, the event will appear in the **Event Console** with that custom severity modifier applied to the severity in the ***Event Severity*** field. For example, if an interface with an ***Event Severity Adjust*** setting of *Sev -1* triggers an event with an ***Event Severity*** of *Major* and that event has the ***Use Modifier*** checkbox selected, the event will appear in the **Event Console** with a severity of *Minor*.
- ***Policy Description***. Text that explains what the event means and what possible causes are.

Defining Pattern Matching and Advanced Behavior

The **[Advanced]** tab in the **Event Policy Editor** page allows you to define or edit pattern-matching for the syslog event and also define event roll-ups and suppressions. In the **[Advanced]** tab, you can define or edit the following fields that pertain to syslogs:



- ***Syslog Facility***. Facility information used by syslog to match an event message.
- ***Syslog Severity***. Severity information used by syslog to match an event message.
- ***Syslog Application Name***. Application Name used by syslog to match an event message.
- ***Syslog Process ID***. Process ID used by syslog to match an event message.
- ***Syslog Message ID***. Message ID used by syslog to match an event message.

NOTE: For more information on the syslog fields for events, see http://www.rfc-archive.org/getrfc.php?rfc=5424 .

- ***First Match String***. A string used to correlate the event with a log message. To match this event policy, the text of a log message or alert must match the value you enter in this field. Can be any combination of alphanumeric characters. SL1's expression matching is case sensitive. This field is required for events generated with a source of Syslog, Security, 3rd Party, and Email.
- ***Second Match String***. A secondary string used to match against the originating log message. To match this event policy, the text of a log message or alert must match the value you enter in this field and the value you entered in the ***First Match String*** field. This field is optional.

NOTE: The ***Match Logic*** field specifies whether SL1 should process ***First Match String*** and ***Second Match String*** as simple text matches or as regular expressions.

NOTE: You can define an event so that it is triggered only when it occurs on a specific interface. You can then include the interface name and SL1's unique interface ID for the interface in the event message. When defining an event, you can use the following three fields below to associate an event with an interface.

- ***Identifier Pattern***. A regular expression used to extract the specific subentity (like the name of a network interface) within the log entry. SL1 will use this value as the yName of the interface. By identifying the subentity, SL1 can create a unique event for each subentity, instead of a single event for the entire device. For example, a log message indicating a link has gone down may include the network interface name. So this field could extract the network interface name from the log message. SL1's expression matching is case sensitive. For details on the regular expression syntax allowed by SL1, see http://www.python.org/doc/howto.
- ***Identifier Format***. If the ***Identifier Pattern*** field returns multiple results, users can specify which results to use and in which order. Each result is represented by a variable. This field is optional.
    - %1. First match with identifier pattern. This is the default behavior if no value is supplied in the ***Identifier Format*** field.
    - %2. Second match with identifier pattern.
    - For example, users could specify "%2:%1" for "Interface %2: Peer %1".

Select the **[Save]** button to save your settings when you have finished editing the fields pertaining to your syslog event policy.

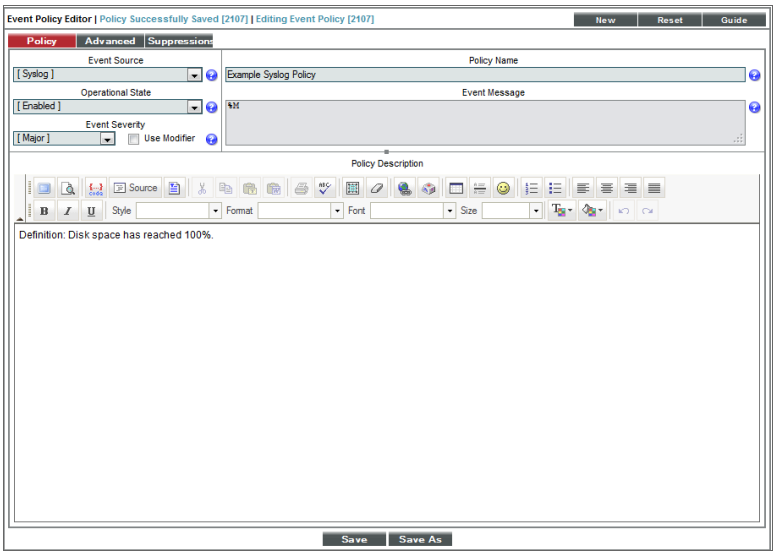
For more information on the remaining fields, as well as the **[Suppressions]** tab, see the ***Events*** manual.

Example Syslog Event Policy

This section will walk through the steps of creating an event policy for syslogs. We will be creating a policy that will send a syslog message when the device's disk space has reached 100% capacity.

To create a Syslog Event Policy:

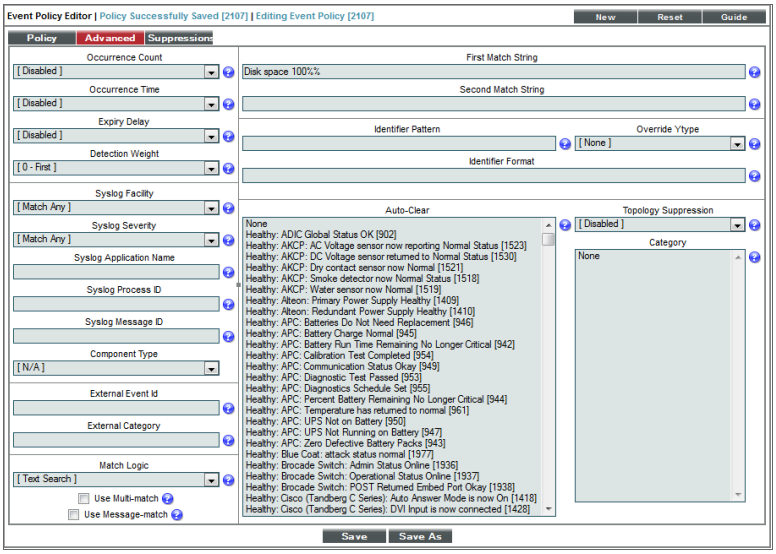
1. Go to the **Event Policy Manager** page (Registry > Events > Event Manager).
2. Select the **[Create]** button, and the **Event Policy Editor** page will appear.
3. In the **Event Policy Editor** page, enter these values in the following fields:



- ***Event Source***. We selected *Syslog*.
- ***Operational State***. We selected *Enabled*.
- ***Event Severity***. We selected *Notice*.
- ***Policy Name***. We entered "Example Syslog Policy".
- ***Event Message***. We entered "%M".
- ***Policy Description***. We entered "Definition: Disk space has reached 100%."

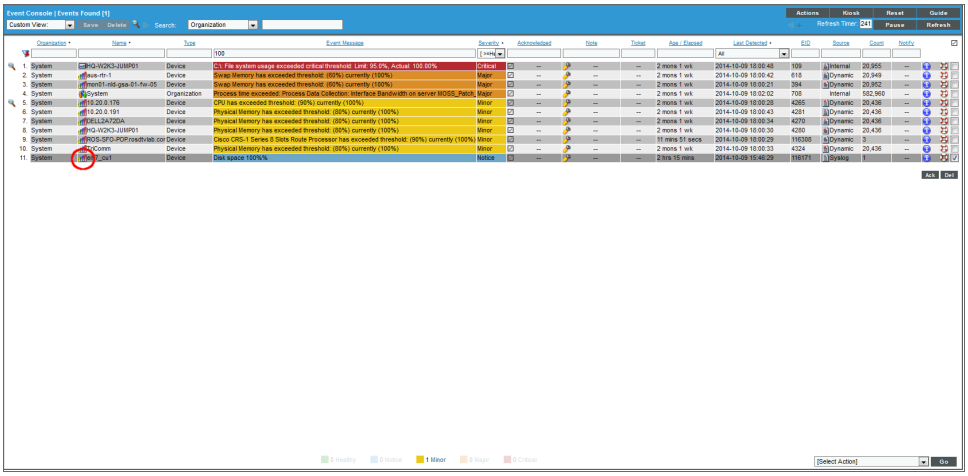
4. Select the **[Save]** button.

5. After saving those settings, select the **[Advanced]** tab. We entered the following values in the following fields:

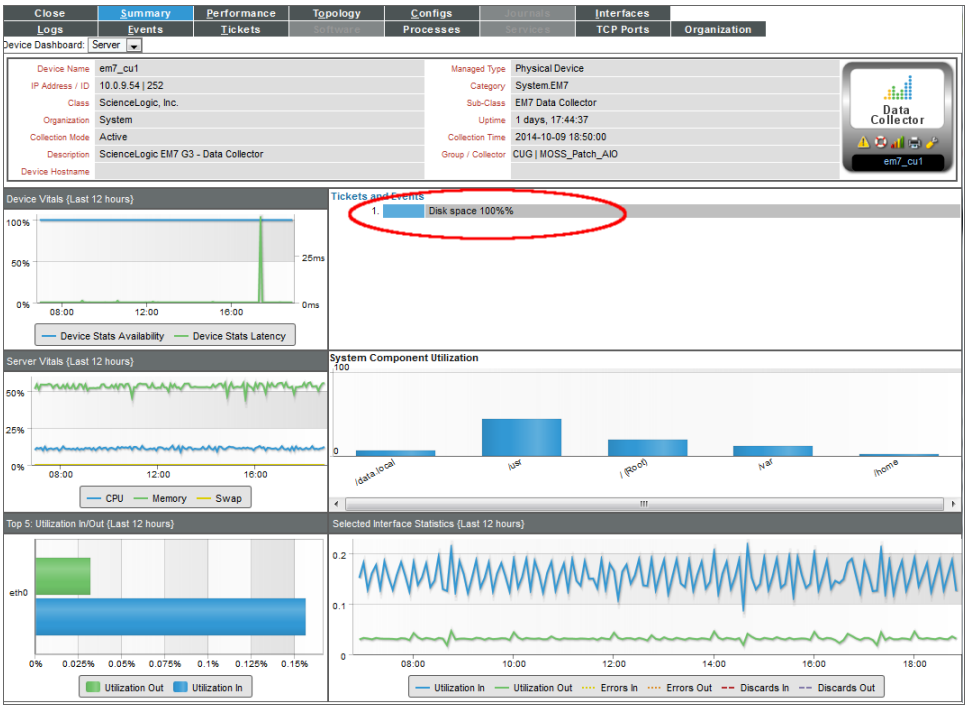


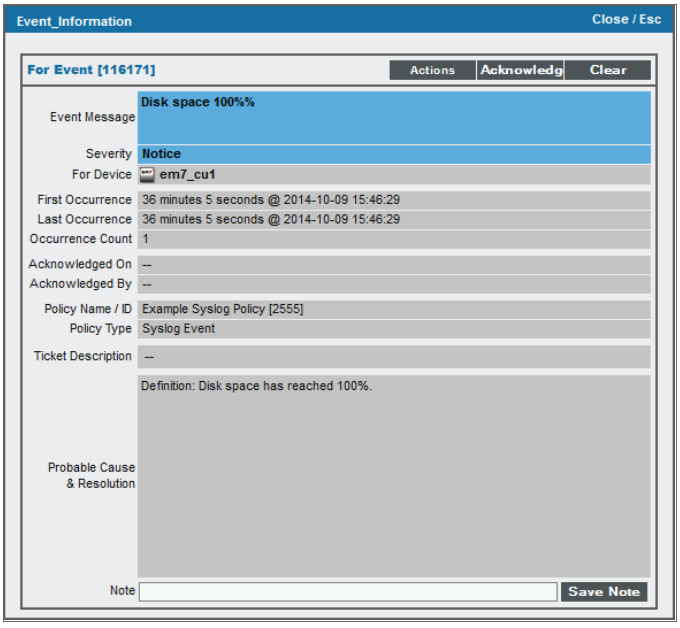
- ***Syslog Facility***. We selected *Match Any*.
- ***Syslog Severity***. We selected *Match Any*.
- ***First Match String***. We entered "Disk space 100%%".

6. We left the rest of the fields at their default settings, and then selected the **[Save]** button.
7. When the device reaches 100% capacity, it will trigger an event, which appears in the **Event Console**. Clicking on the graph icon ( ) will bring up the **Device Summary** page for the device for which the event occurred. Clicking on the life ring icon ( ) will create a ticket for the event.

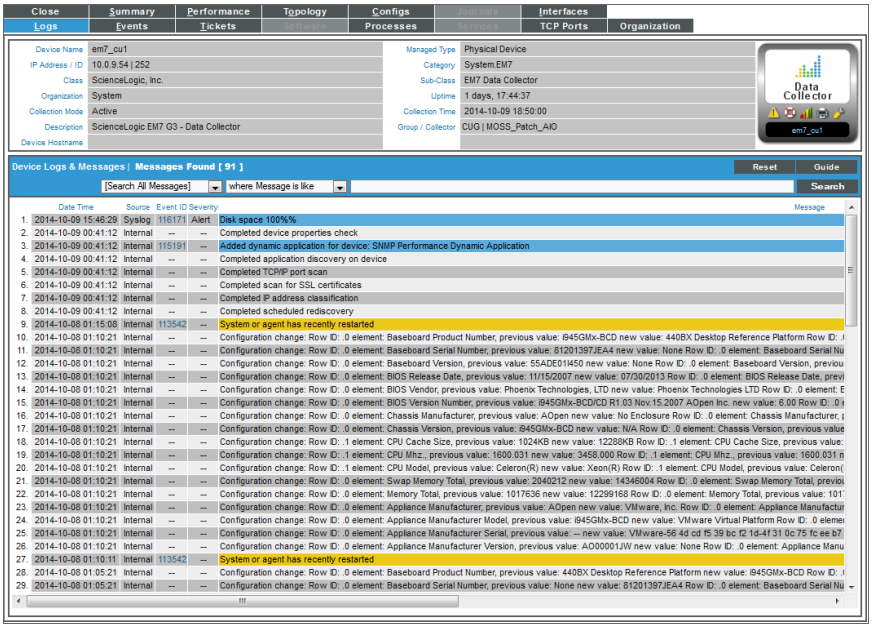


8. Clicking on the graph icon ( ) will bring up the **Device Summary** page. You will see the event listed in the **Device Summary** page, and you can click on the event to view the **Event Summary** modal page.

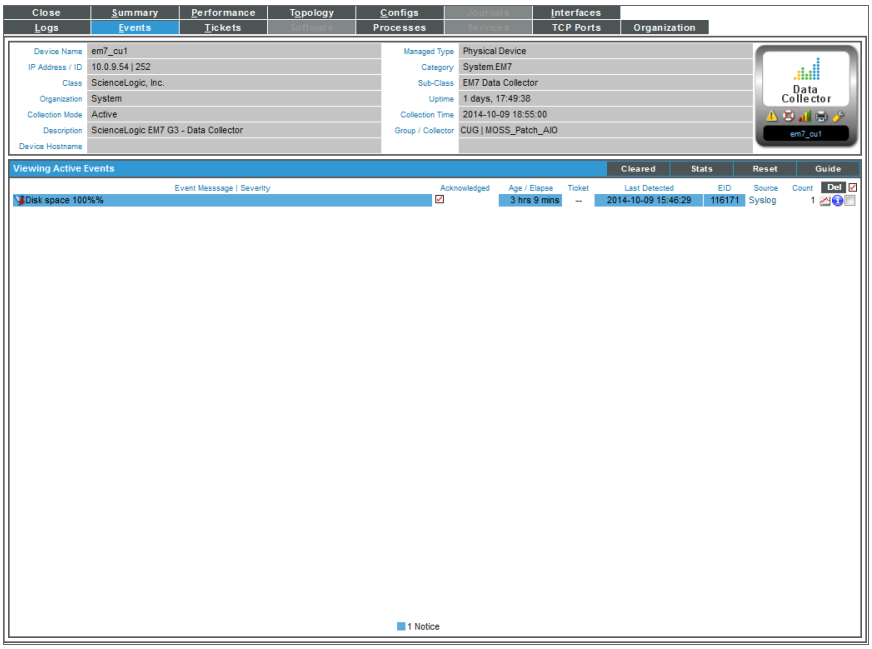




9. You can also select the **[Logs]** tab from the **Device Summary** page to view the **Device Logs & Messages** page. The syslog message will appear in the device logs, and you can select the View Events icon ( ) which will take you to the **Viewing Active Events** page for that device.



10. From the **Viewing Active Events** page, you can select the information icon ( ) to view the **Event Information** modal page, filter the device's events based on event type, or view graphical reports about that device's events based on type.



© 2003 - 2020, ScienceLogic, Inc.

All rights reserved.

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010