Security Analytics Event Source Integrator Overview Guide

About Security Analytics Event Source Integrator

Security Analytics Event Source Integrator (ESI) is a graphical tool that enables you to create and edit event source definitions or parsers for the Security Analytics Log Decoder. Using the Event Source Integrator, you can define how a Log Decoder identifies, parses, and extracts information from the events of a specific event source. These parser definitions are stored as an XML file, called an event source XML file, which is deployed on the Security Analytics platform.

You can create a new event source parser for an event source that is not currently supported by Security Analytics.You can also edit an existing event source parser to add or edit definitions for events, or to correct errors. You may need to edit an event source parser in one of the following situations:

- You upgrade to a new version of an event source that contains new, updated, or deprecated event messages.
- You want to include additional definitions for existing events.
- You want to update the definition for an existing event in an event source parser.
- You want to correct errors in an event source parser.

Parser Structure

The Event Source Integrator uses the device name of the event source parser to create the structure for the parser. ESI also appends the device name to the directory that you specify for your parser.

When you create an event source parser, you select a device name for it. The device name must start with a letter. The RSA naming convention is to make the device name all lowercase and remove the spaces. For example, Cisco ASA would have the device name **ciscoasa** and Actiance Vantage would be **actiancevantage**. It is not necessary to follow the RSA naming convention to use this tool.

In your event source parser directory, the Event Source Integrator creates two files with the correct name for Security Analytics:

- **INI file.** This is the parser configuration file. (Example: ciscoasa.ini)
- **XML file.** This is the event source XML file that contains the parser definitions. (Example: v20_ciscoasamsg.xml) The device name is prepended with **v20_** and appended with **msg**.

Both of these files are required to deploy your parser in Security Analytics.

When you finish creating or updating your parser, you have the option of retrieving the completed parser in two formats:

- **XML and INI** (In the main menu, select **File > Save** or **Save As**) This option creates a device name folder containing an XML file and a configuration INI file.
- **.envision** (In the menu, select **File > Export Parser**) This option creates an event source package that consists of the event source XML and configuration INI file.

Obtaining a Log File

To create an event source parser that Log Decoder can use to identify, parse, and extract information from a specific event source, you must obtain a log file from the event source that you want to integrate with Security Analytics. After you obtain the log file, you can use Event Source Integrator to create an event source parser.

Before getting started with Event Source Integrator, you must know the log collection protocol that was configured when the event source was deployed with Security Analytics.

If the log collection protocol that you configured when you set up the event source in Security Analytics is Syslog, you can use a log file generated by the event source to create or edit an event source parser.

If you configured any other log collection protocol, you must export a log file from Security Analytics in text format.

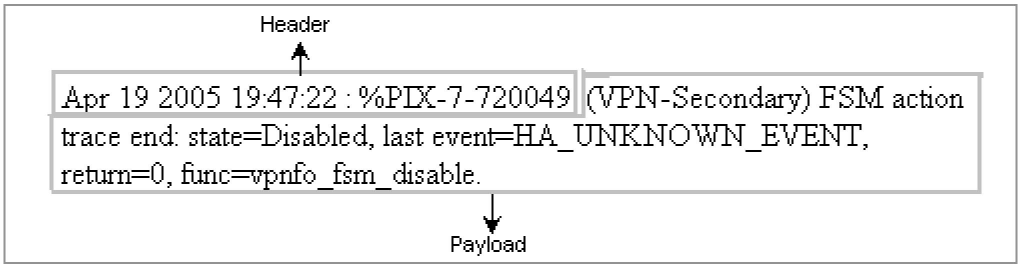
RSA recommends that you compile a log file that contains all the unique events generated by the event source that you want to integrate with Security Analytics. While compiling the log file, ensure that:

- All the events are from a single event source.
- Each event is listed in a single line, without any line breaks.
- The maximum size of any event is 64 KB.
- The recommended maximum size of the log file is 10 MB.
- The log file contains one or two instances of each unique event.

For events transmitted by Syslog, you can put the raw logs directly in the ESI tool.For all other event formats, you need to get the log data from Security Analytics. To get log data from Security Analytics, see "Get a Log File from Security Analytics" on page 6.

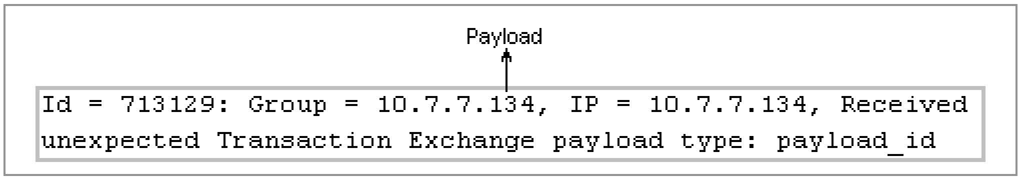
Understanding Events

Typically an event consists of two main elements, a header and a payload. The following figure shows an example of an event with a header and a payload.



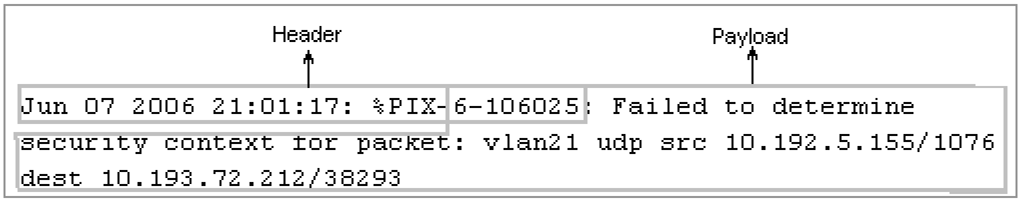
```
Apr 19 2005 19:47:22 : %PIX-7-720049 (VPN-Secondary) FSM action
trace end: state=Disabled, last event=HA_UNKNOWN_EVENT,
return=0, func=vpnfo_fsm_disable.
```

In some events, you may define the entire event as payload as shown in the following figure.



```
Id = 713129: Group = 10.7.7.134, IP = 10.7.7.134, Received
unexpected Transaction Exchange payload type: payload_id
```

In some events, you may define the payload to begin from the header, and the header and payload may overlap.

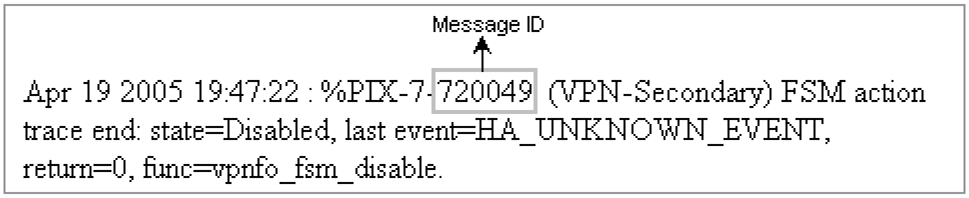


```
Jun 07 2006 21:01:17: %PIX-6-106025: Failed to determine
security context for packet: vlan21 udp src 10.192.5.155/1076
dest 10.193.72.212/38293
```

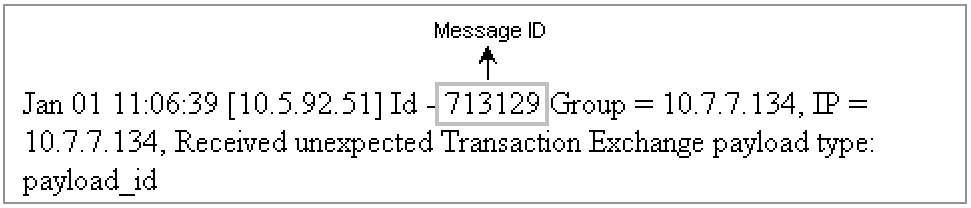
Header

The header consists of the following elements, which are common across multiple events:

Message ID. Indicates a unique identifier for the message in the event. In the examples in the following figures, the message ID is unique to the event.

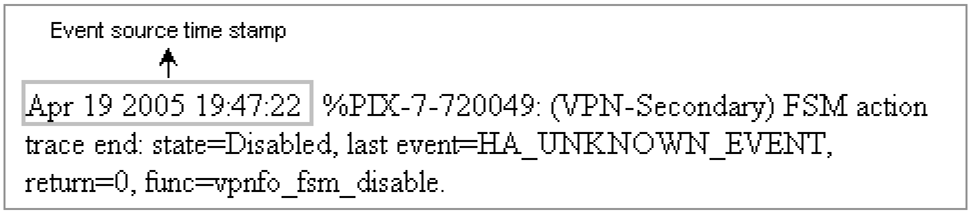


```
Apr 19 2005 19:47:22 : %PIX-7-720049 (VPN-Secondary) FSM action
trace end: state=Disabled, last event=HA_UNKNOWN_EVENT,
return=0, func=vpnfo_fsm_disable.
```



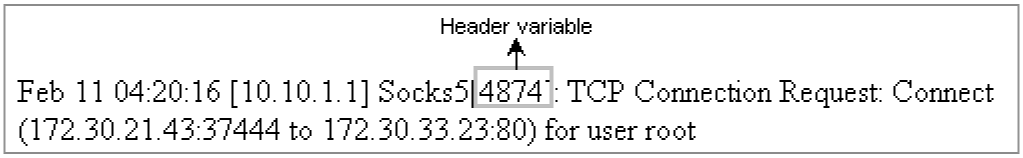
```
Jan 01 11:06:39 [10.5.92.51] Id - 713129 Group = 10.7.7.134, IP =
10.7.7.134, Received unexpected Transaction Exchange payload type:
payload_id
```

Event source time stamp. (Optional) Consists of the date and time when the event was generated by the event source. Some events may not contain an event source time stamp.

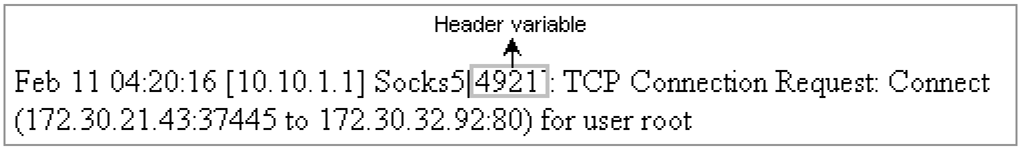

Event source time stamp

```
Apr 19 2005 19:47:22 %PIX-7-720049: (VPN-Secondary) FSM action
trace end: state=Disabled, last event=HA_UNKNOWN_EVENT,
return=0, func=vpnfo_fsm_disable.
```

Header Variable. (Optional) Contains a value in the event header that varies across similar types of events. In the examples in the following figures, 4874 and 4921 are header variables that indicate the session ID in the events.


Header variable

```
Feb 11 04:20:16 [10.10.1.1] Socks5[4874]: TCP Connection Request: Connect
(172.30.21.43:37444 to 172.30.33.23:80) for user root
```


Header variable

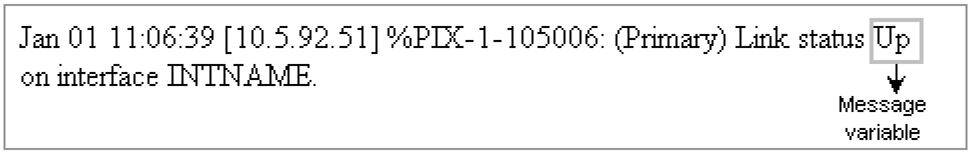
```
Feb 11 04:20:16 [10.10.1.1] Socks5[4921]: TCP Connection Request: Connect
(172.30.21.43:37445 to 172.30.32.92:80) for user root
```

Caution: If you create a header that too widely matches event content, it could match logs that are currently parsed through other parsers.

Payload

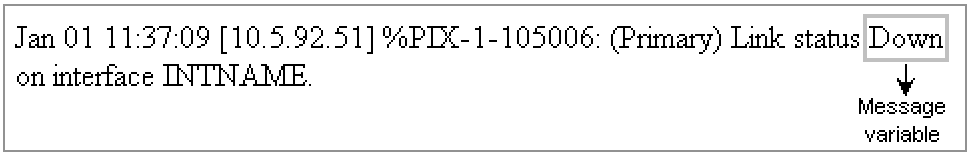
The payload is everything in the event that is not the header. It contains detailed information about the event. The payload is the message in the event. Security Analytics uses this information for analysis and reporting. The payload consists of message variables and static text.

A message variable is a value in the payload that varies across similar types of events. In the examples in the following figures, Up and Down are message variables that indicate the link status of the INTNAME interface.



```
Jan 01 11:06:39 [10.5.92.51] %PIX-1-105006: (Primary) Link status Up
on interface INTNAME.
```

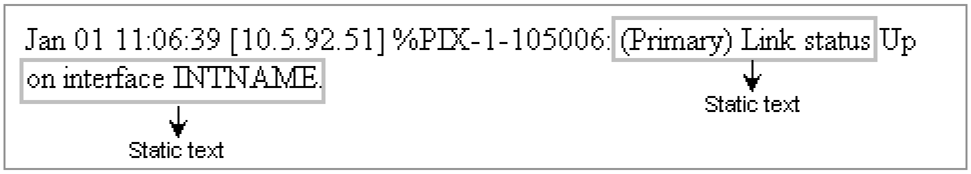
Message
variable



```
Jan 01 11:37:09 [10.5.92.51] %PIX-1-105006: (Primary) Link status Down
on interface INTNAME.
```

Message
variable

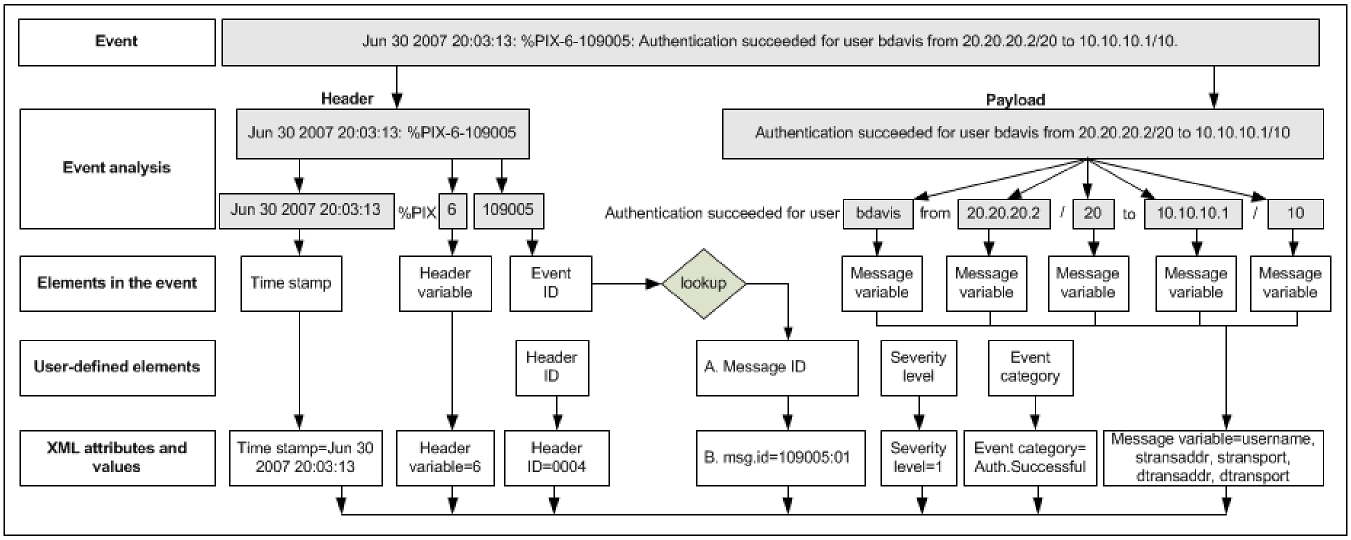
Event Source Integrator classifies all the values in the payload that are not message variables as static text. The following figure shows an example of values that Event Source Integrator classifies as static text.



```
Jan 01 11:06:39 [10.5.92.51] %PIX-1-105006: (Primary) Link status Up
on interface INTNAME.
```

How Log Data is Mapped to Security Analytics Metadata

The following figure shows an example of how you can create an XML definition that makes the event data available for analysis and reporting in Security Analytics. It also shows the various elements in an XML definition.



Event
Jun 30 2007 20:03:13: %PIX-6-109005: Authentication succeeded for user bdavis from 20.20.20.2/20 to 10.10.10.1/10.
Event analysis
Header
Jun 30 2007 20:03:13: %PIX-6-109005
Payload
Authentication succeeded for user bdavis from 20.20.20.2/20 to 10.10.10.1/10
Jun 30 2007 20:03:13
%PIX
6
109005
Authentication succeeded for user
bdavis
from
20.20.20.2
/
20
to
10.10.10.1
/
10
Elements in the event
Time stamp
Header variable
Event ID
lookup
Message variable
Message variable
Message variable
Message variable
Message variable
User-defined elements
Header ID
A. Message ID
Severity level
Event category
XML attributes and values
Time stamp=Jun 30 2007 20:03:13
Header variable=6
Header ID=0004
B. msg.id=109005:01
Severity level=1
Event category= Auth.Successful
Message variable=username, stransaddr, stransport, dtransaddr, dtransport

Getting Started with Security Analytics Event Source Integrator

The following table provides a high-level overview of the tasks that you can perform using Event Source Integrator.

Goal	Task	Reference
Integrate an event source that is not supported by Security Analytics.	1. Create an XML file that contains definitions for the events generated by the event source.	"Creating an Event Source XML File" on page 7
	2. (Optional) View events that are parsed by a header or message definition.	"Viewing Parsed Events and Associated Definitions" on page 10
	3. (Optional) View the header and message definition that parse a selected event.	"Viewing Parsed Events and Associated Definitions" on page 10

Goal	Task	Reference
	4. Create an event source package for deployment to a Log Decoder.	"Parser Structure" on page 1
	5. In Security Analytics, deploy the event source package.	See the "Add or Update Supported Event Source Log Parsers" topic in the *RSA Content and Resources* documentation. The "Download Log Parsers from Live and Deploy from Local Network" section provides information on how to upload the event source log parsers from your local network to the Log Decoder.
Upgrade an event source that is already supported by Security Analytics.	1. Edit the existing event source XML file.	"Editing an Event Source XML File" on page 9
	2. (Optional) View events that are parsed by a header or message definition.	"Viewing Parsed Events and Associated Definitions" on page 10
	3. (Optional) View the header and message definition that parse a selected event.	"Viewing Parsed Events and Associated Definitions" on page 10
	4. In Security Analytics, deploy the event source package.	See the "Add or Update Supported Event Source Log Parsers" topic in the *RSA Content and Resources* documentation. The "Download Log Parsers from Live and Deploy from Local Network" section provides information on how to upload the event source log parsers from your local network to the Log Decoder.

Get a Log File from Security Analytics

1. In the **Security Analytics** menu, select **Investigation > Events**.
2. In the **Investigate** dialog, select a Log Decoder, Archiver, Concentrator, or Broker service and click **Events**.
3. In the **Events** view, select the events and in the Actions menu, select **Export > Export All Logs**.
4. In the **Enter file name for extraction** dialog, enter a name for your log file and click **OK**.
5. In the **Export Log Format** dialog, select Text and click **Export**.
   You will receive a Scheduled Job notice.

6. Check the Job Notifications tray to view the status of the log file. Click the **View** link to go the Jobs panel in the Profile view to download the log file.

Creating an Event Source XML File

Creating an event source XML file involves creating a definition for each type of event in the log file generated by an event source. Creating an event definition involves the following tasks:

1. "Selecting an Event from the Log File" below
2. "Defining a Header" below
3. "Defining a Message" on the next page

Selecting an Event from the Log File

Select an event from the log file to define the various elements of the header and message in the event.

Defining a Header

Define the header by assigning the values in the event to header elements. The purpose of defining a header is to identify the event source from which the event is generated. When you define a header with all its elements, the definition can parse similar types of events in the log file.

RSA recommends that you define a generic header definition that will parse multiple events that follow similar formats. Event Source Integrator generates a unique identifier, the header ID, for each header definition to identify the header definitions available in the event source XML file. However, you can change the generated identifier to provide a unique header ID of your choice.

You can include the following elements when defining how to locate the message ID in the header:

- Message ID, used in the Event ID lookup, which enables you to specify one of the following options:
  - Message ID variable
  - Variable suffix
  - Concatenation
- Header variables
- Payload

For more information on these elements, see "Understanding Events" on page 3.

Defining a Message

Define the message by assigning the values in the payload to message variables and defining message elements. A single message definition may parse one or more similar events in your log file.

The following table lists the message elements that you must define.

Message Element	Description
Message ID	Indicates the identifier by which Security Analytics identifies the event uniquely. The message ID can be defined in one of the following ways: • Same as the event ID. • A combination of the event ID defined in the header definition and a unique variant. For example, if the event ID is 109801, the message ID can be defined as 109801:02. • A brief description that identifies the event. For example, in the following event, the event ID is 187698, and the message ID can be defined as CableFailover. `Jan 01 11:06:39 [10.5.92.51] %PIX-1-101001: (PRIORITY) Error reading failover cable status.`
Severity level	Indicates the severity level that you assign to the event. The severity level ranges from 0 to 7, where 0 indicates emergency and 7 indicates debugging information.
Event category	Indicates the category to which the event belongs, based on the Security Analytics taxonomy.
Functions	(Optional) Define actions to be performed on variables in an event to generate user-defined values. Event Source Integrator supports the following functions: • **Assign Constant** assigns user-defined values to variables. • **Assign Header Variable** assigns the value of a header variable to a message variable. • **Assign Message Variable** assigns the value of a message variable to another variable. • **Assign System Variable** populates the value of a message variable with a system value • **Assign Context** assigns the payload from the beginning of the payload to the first instance of a specified character within the payload. • **Calculation** performs a calculation on values and variables in the event. • **Convert IP** converts IP addresses to domain names.

Message Element	Description
	- **Convert Domain** converts domain names to IP addresses. - **Direction Check** determines the direction of an IP address (in or out). - **Duration** converts a duration represented as date and time information to seconds assigned to a message variable. - **Event Time** assigns the date and time information in the event to a message variable. - **Remove Quotes** removes quotes from a variable. - **URL Part** extracts parts of a URL string.

Editing an Event Source XML File

Before you edit an event source XML file, identify and analyze the events in the corresponding log file. You can edit an event source XML file that was created by Event Source Integrator or any other source.

Editing an event source XML file involves the same header and message definition tasks as creating an event source XML file. However, you may not need to define the header pattern if the headers are all defined. For example, you may want to edit a parser to add new messages to Windows or other platforms. You may also want to adjust an existing parsed message. For example, you may need to change IP source to IP destination in a particular event.

Validating the Precedence of XML Definitions

It is important to consider the precendence of XML definitions when defining headers and messages with the same event ID.

You must arrange the definitions in the event source XML file in order from specific to generic so that events are parsed against the specific header or message definition.

For example, suppose that an event source XML file contains the following message definitions:

- Message 10123 < A B C >, where A, B, and C are elements in the message
- Message 10124 < A B C D >, where A, B, C, and D are elements in the message

If the order of the message definitions is as shown, Security Analytics parses an A B C event from the event source using the Message 10123 definition. Security Analytics also parses an A B C D event from the event source using the Message 10123 definition. The A B C D event must be parsed against the Message 10124 definition, which is specific. Therefore, you must ensure that the specific definition, Message 10124, appears before the generic definition, Message 10123, in event source XML file, as follows:

- Message 10124 <A B C D>
- Message 10123 < A B C >

After validating the event source XML for data pattern warnings, you must validate the precedence of the header and message definitions in the event source XML file.

Event Source Integrator displays the errors that occur in the header and message definition order. For better analysis and reporting, you must resolve all the precedence errors.

Viewing Parsed Events and Associated Definitions

After defining the event source XML, you can view the highlighted header and message definitions in the parsed logs within the ESI tool. You can also view the header and message definition for a selected event in the log file.

While defining the event source XML, you can view parsed events and the associated header and message definitions. For example, if you have defined two header definitions and one message definition in the event source XML file, you can view parsed events for these definitions, and then continue to define more header and message definitions depending on the parsed events already viewed.

Custom Table Mapping Files

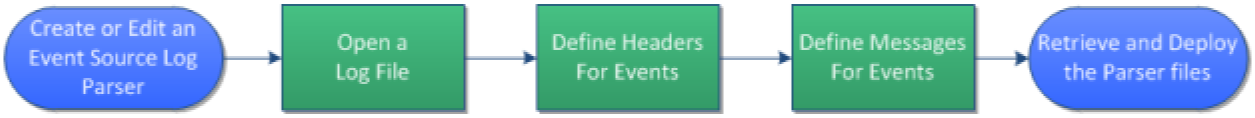
If you changed the table mapping file (table-map.xml)or created a custom table mapping file (table-map-custom.xml), get the changed table mapping files from Security Analytics. You can add these custom table mapping files to the Event Source Integrator. You can find these files in Security Analytics in the Services Config view > Files tab of the Log Decoder service (**Administration > Services > Select Log Decoder > Config > Files tab**).

For Windows, place the table mapping files in the **C:\Users\<username>\AppData\Roaming\NetWitness\envision\etc** directory.

For Macs, place the table mapping files in the **~/.netwitness/envision/etc** directory.

ESI Tool Workflow

The following figure shows the workflow of the Event Source Integrator tool user interface.


Create or Edit an Event Source Log Parser
Open a Log File
Define Headers For Events
Define Messages For Events
Retrieve and Deploy the Parser files

Create a Parser

1. On the opening screen, click **New Parser**.
2. In the **New Parser Details** dialog:
   a. Type a name for the device.
      The device name must start with a letter. The RSA naming convention is to make the device name all lowercase and remove the spaces. For example, Cisco ASA would have the device name **ciscoasa** and Actiance Vantage would be **actiancevantage**. It is not necessary to follow the RSA naming convention to use this tool.
   b. Specify the directory where you want to create the parser and click **OK**.
      In the directory that you specify, the ESI tool creates two files with the correct name for Security Analytics:
      - INI file (Example: ciscoasa.ini)
      - XML file (Example: **v20_**ciscoasa**msg**.xml) The device name is prepended with **v20_** and appended with **msg**.
3. On the main screen **Parser Details** tab, in the **Device Type** field, select the device that you are working on, for example, Firewall. The device type provides additional information about the event.
4. In the **Display Name** field, type a user-friendly name for the parser, for example, Cisco ASA.

Edit a Parser

1. On the opening screen, click **Open**.
2. Select your parser definition XML file and click **Open**.
3. If your parser is not already defined:
   a. On the main screen **Parser Details** tab, in the **Device Type** field, select the device that you are working on, for example, Firewall. The device type narrows down the use of the parser.
   b. In the **Display Name** field, type a user-friendly name for the parser, for example, Cisco ASA.

If you opened your parser previously in the ESI tool, you can click the file link in the **Existing files** section.

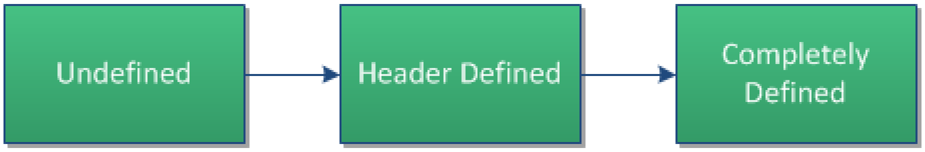
Open a Log File

In the **Select an Event** section, do one of the following to add a log file to parse:

- Click **Open Log File**.
- Drag and drop the log file.
- Drag and drop the text of a plain log.

Understand the Select an Event Section Workflow

The following figure shows the workflow of the Select an Event section.



After you have defined a new parser or opened a parser to edit, work from the Select an Event section to define the headers and messages. Events move from **Undefined** (Header and Message not defined) to **Header Defined** (Message not defined) and then to **Completely Defined** (Header and Message defined).

Determine the Parser Definition Method

There are two main methods to specify a parser definition depending upon the type of information that you need to collect in the logs:

- Identify events with a specific type and extract generic information.
- Extract as much detailed information as possible from an event.

Example: Extract Generic Information

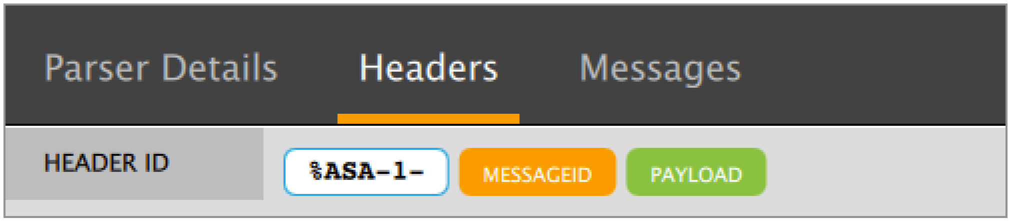
The following example shows how to create a parser that extracts generic information from a Cisco ASA log. It uses the following event from the Cisco ASA log:

%ASA-1-101001: (PRIORITY) Failover cable OK.

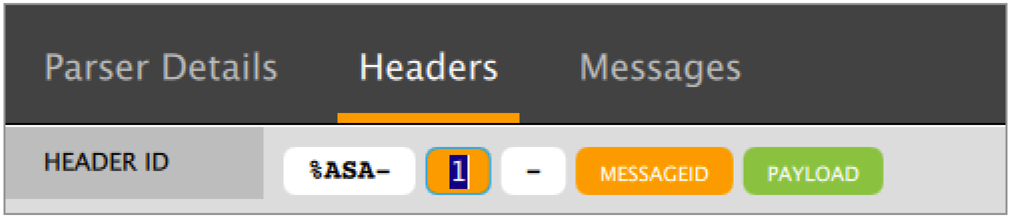
Define the Header Pattern (Generic)

1. Select an undefined event and do one of the following:
   - Select the event and click **Create**.
   - (Optional Shortcut) Highlight the Message ID to select it and click **Create**.
2. In the Headers section, remove any unnecessary static text.
   For example, right-click the text, select **Edit**, remove all of the text to the right of **%ASA-1-** and

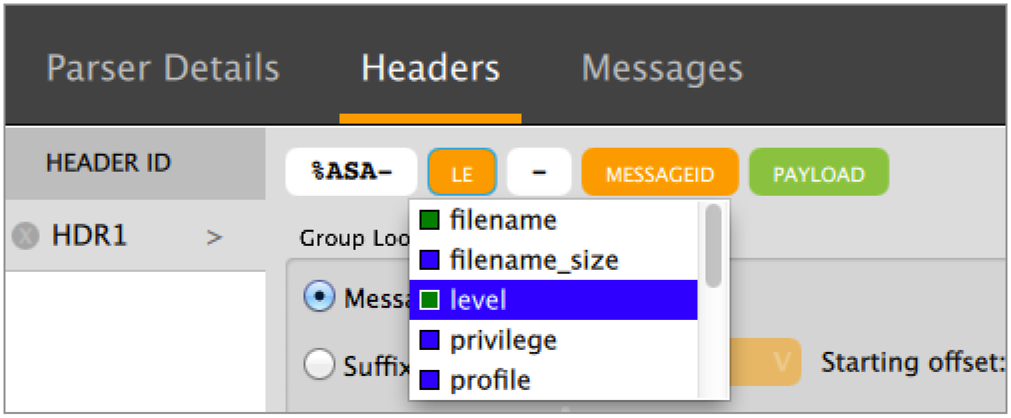
press **ENTER**.



3. Define a variable for anything that can change. To define a variable:
   a. Right-click the text and select **Edit**.
   b. Highlight the text that you want to change to a variable and press **CTRL+K** (**COMMAND+K** for Mac). The background changes to orange, which indicates a variable. This example shows changing 1 to a variable

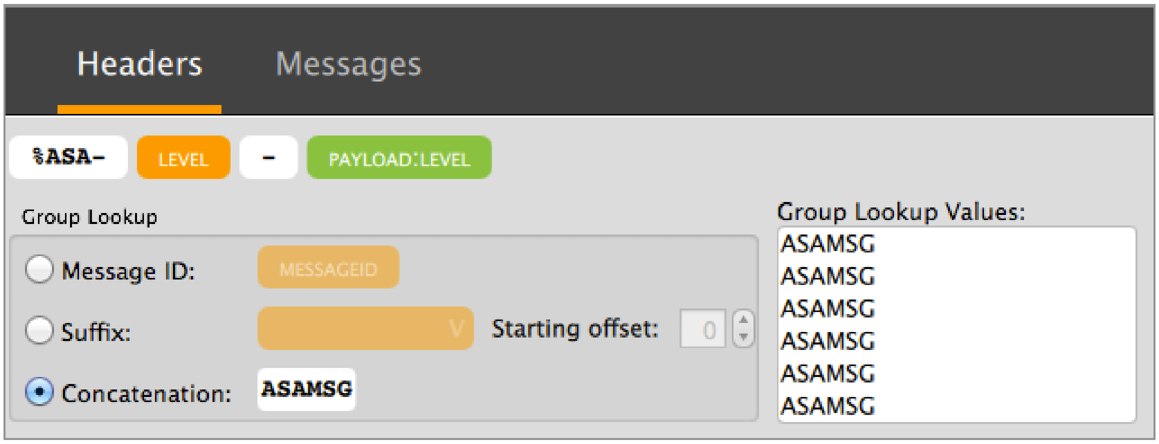


   c. Start typing the name of the variable in the variable field, use the down and up arrow keys to select the variable, and press **ENTER**. In this example, the variable is level.



   As you define the header pattern, the **Group Lookup Values** show the matched values. The number of headers defined in the **Select an Event** section increases. The green bar shows where the payload starts.

4. To change where the payload starts, right-click a variable and select **Set As Payload Start**. For example, right-click the level variable and select **Set As Payload Start**.
5. To change the message ID to a generic value:
   a. Right-click the **messageid** variable next to payload and select **Delete**.
   b. Select **Concatenation**, type a generic text string in the text field, and press **ENTER**. For example, type ASAMSG.



%ASA- LEVEL - PAYLOAD:LEVEL

Group Lookup

○ Message ID: MESSAGEID

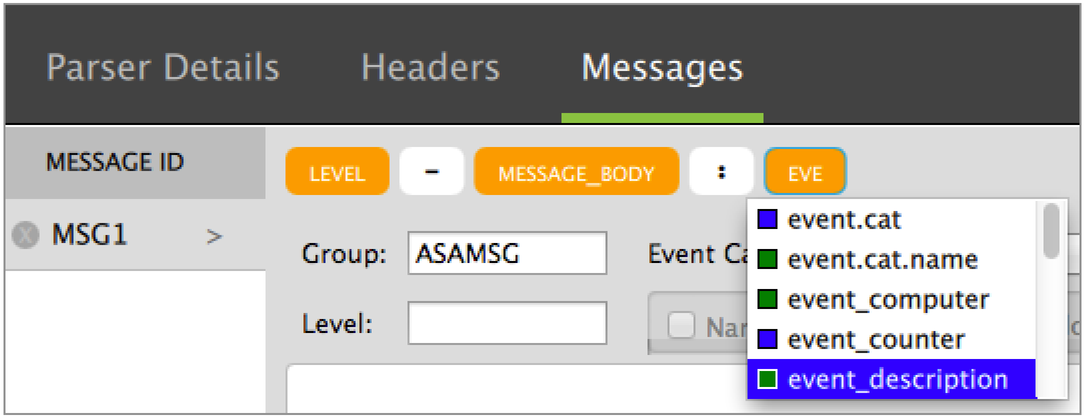
○ Suffix: Starting offset: 0

● Concatenation: ASAMSG

Group Lookup Values:
ASAMSG
ASAMSG
ASAMSG
ASAMSG
ASAMSG
ASAMSG

Define the Message Pattern (Generic)

1. Click the **Messages** tab, select an event with the header defined, and click **Create**.
2. In the message pattern, define variables for the values that you want to extract as meta. To define a variable:
   a. Right-click the pattern text and select **Edit**.
   b. Highlight the text that you want to change to a variable and press **CTRL+K** (**COMMAND+K** for Mac). The background changes to orange, which indicates a variable.
   c. Start typing the name of variable in the variable field, use the down and up arrow keys to select the variable, and press **ENTER**. In this example, the variables are level, message_body, and event_description.



LEVEL - MESSAGE_BODY : EVE

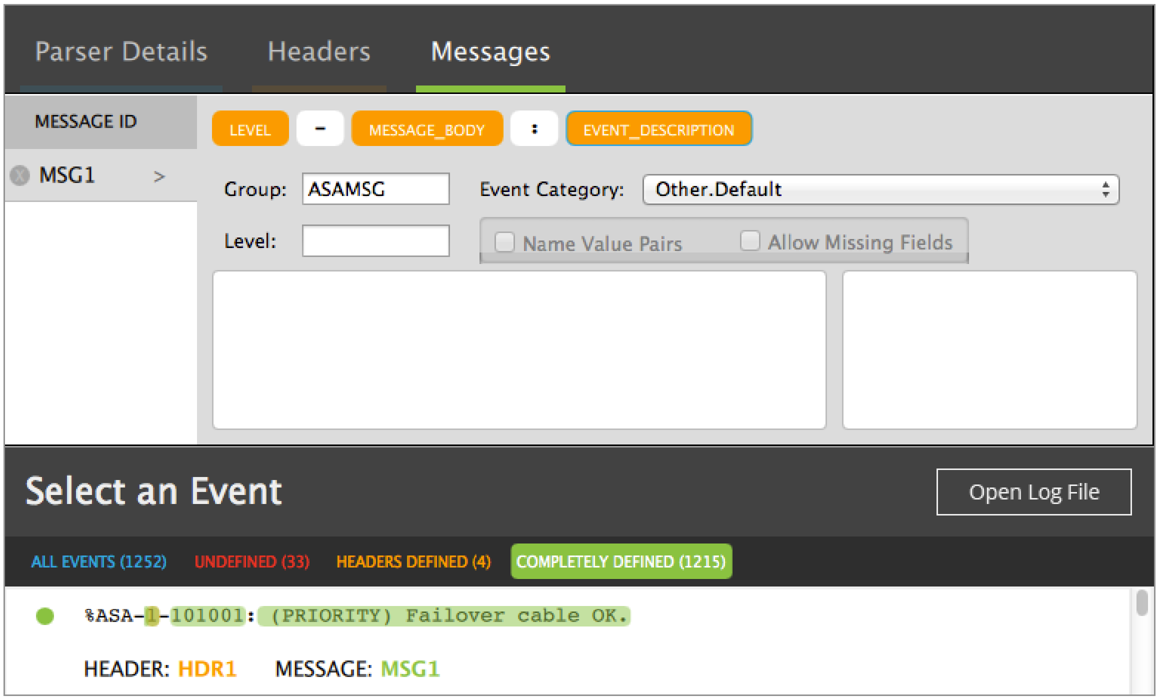
Group: ASAMSG

Level:

event.cat
event.cat.name
event_computer
event_counter
event_description

The message variables that you define create meta in the Log Decoder.

3. In the **Event Category** field, select a generic category for the message. For example, Other.Default.



Group: ASAMSG

Event Category: Other.Default

Level:

□ Name Value Pairs □ Allow Missing Fields

```
%ASA-1-101001: (PRIORITY) Failover cable OK.
```

HEADER: HDR1 MESSAGE: MSG1

The **Group** field populates from the header. The event shows as completely defined in the **Select an Event** section.

4. After you complete and save your changes, retrieve the completed parser files. You have a choice of two formats:
   - **XML / INI** (In the main menu, select **File > Save** or **Save As**)
   - **.envision** (In the menu, select **File > Export Parser**) This option creates an event source package that consists of the event source XML and configuration (INI) file.
5. Deploy the event source package in the Security Analytics platform to integrate the event source. RSA recommends that you first deploy the parser to a test system to verify that it parses log traffic correctly.

Example: Extract Detailed Information

The following example shows how to create a parser that extracts detailed information from a Cisco ASA log. It uses the following event from the Cisco ASA log:

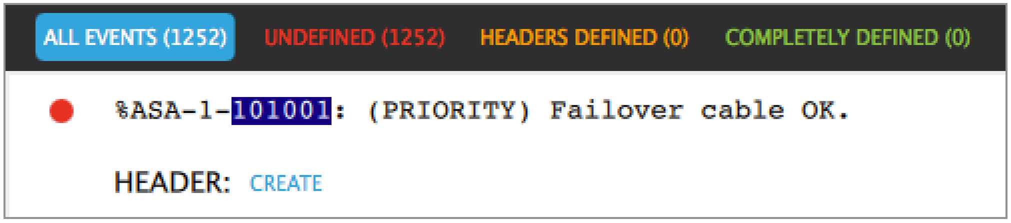
%ASA-1-101001: (PRIORITY) Failover cable OK.

If RSA did not support this type of log, this procedure shows how you would define the header and message patterns for the event.

Define the Header Pattern (Detailed)

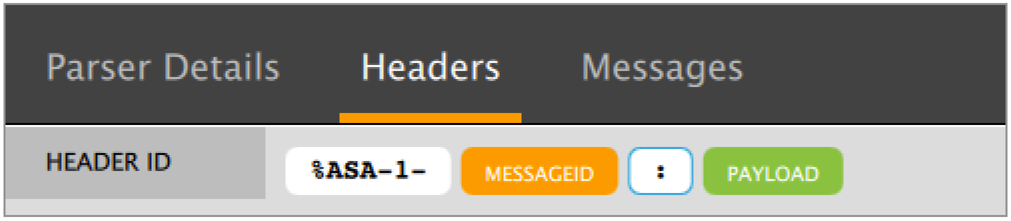
Note: If you are editing an existing parser, you may not need to define the header pattern if the headers are all defined.

1. Select an undefined event and do one of the following:
   - Select the event and click **Create**.
   - (Optional Shortcut) Highlight the Message ID to select it and click **Create**.

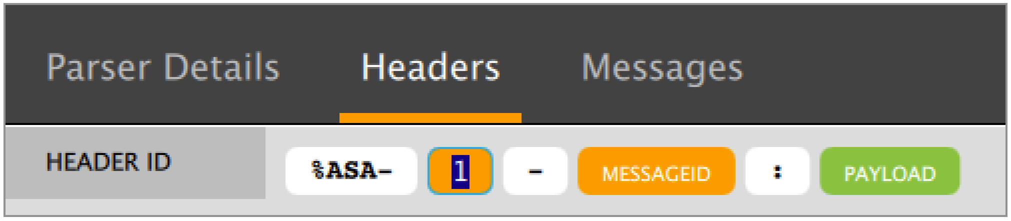


```
%ASA-1-101001: (PRIORITY) Failover cable OK.
```

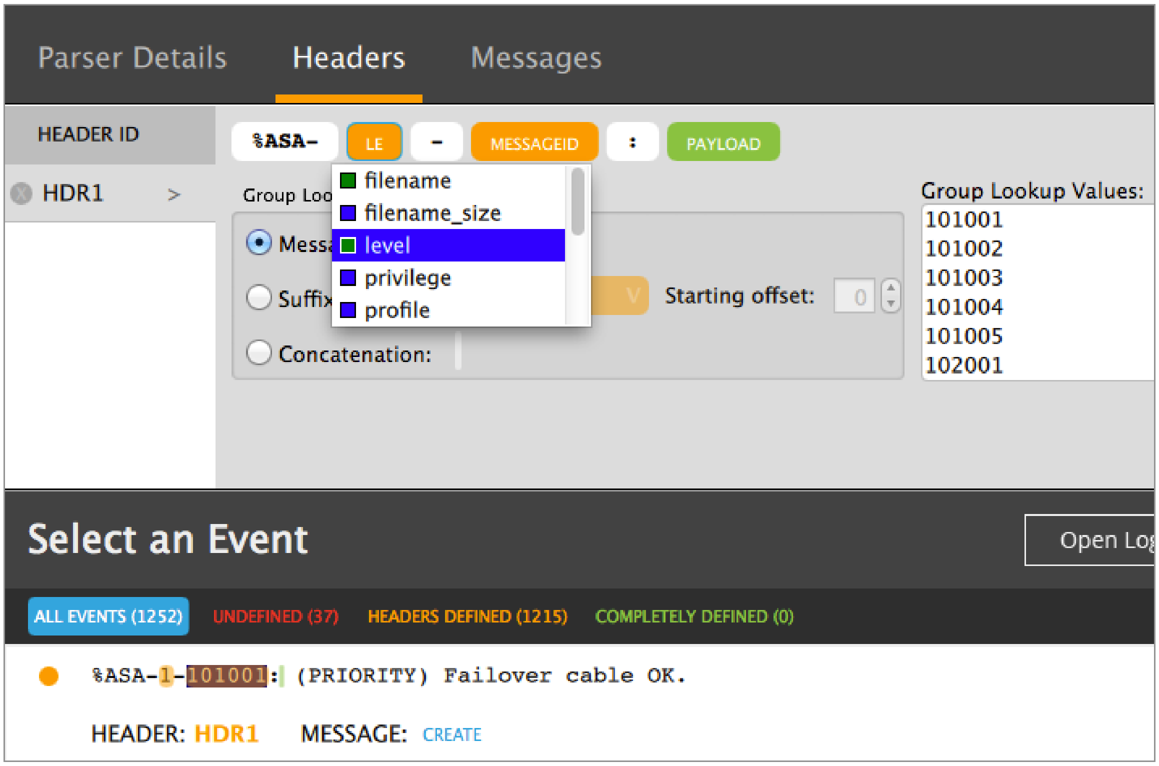
2. In the Headers section, specify the static (anchor) text, which does not change. Any character can be an anchor.
   For example, if a colon (**:**) separates the message ID from the message text, right-click the message text, select **Edit**, remove all of the text to the right of the colon, and press **ENTER**.



3. Define a variable for anything that can change. To define a variable:
   a. Right-click the text and select **Edit**.
   b. Highlight the text that you want to change to a variable and press **CTRL+K** (**COMMAND+K** for Mac). The background changes to orange, which indicates a variable. This example shows changing 1 to a variable.



   c. Start typing the name of variable in the variable field, use the down and up arrow keys to select the variable, and press **ENTER**. In this example, the variable is level.



Group Lookup Values:
101001
101002
101003
101004
101005
102001

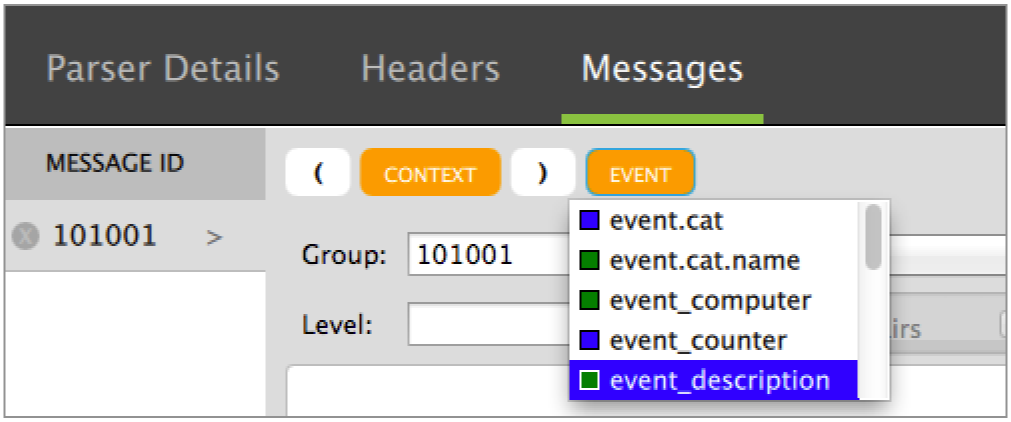
```
%ASA-1-101001: (PRIORITY) Failover cable OK.
```

HEADER: HDR1 MESSAGE: CREATE

As you define the header pattern, the **Group Lookup Values** show the matched values. The number of headers defined in the **Select an Event** section increases. The green bar shows where the payload starts. To change where the payload starts, you can right-click a variable and select **Set As Payload Start**.

Define the Message Pattern (Detailed)

1. Click the **Messages** tab, select an event with the header defined, and click **Create**.
2. In the **Message ID** field, type the actual message ID, and press **ENTER**. For example, change MSG1 to 101001.
3. In the message pattern, define variables for the values that you want to extract as meta or variables used to make the pattern match. To define a variable:
   a. Right-click the pattern text and select **Edit**.
   b. Highlight the text that you want to change to a variable and press **CTRL+K** (**COMMAND+K** for Mac). The background changes to orange, which indicates a variable.
   c. Start typing the name of variable in the variable field, use the down and up arrow keys to select the variable, and press **ENTER**. In this example, the variables are context and event_description.



Group: 101001

Level:

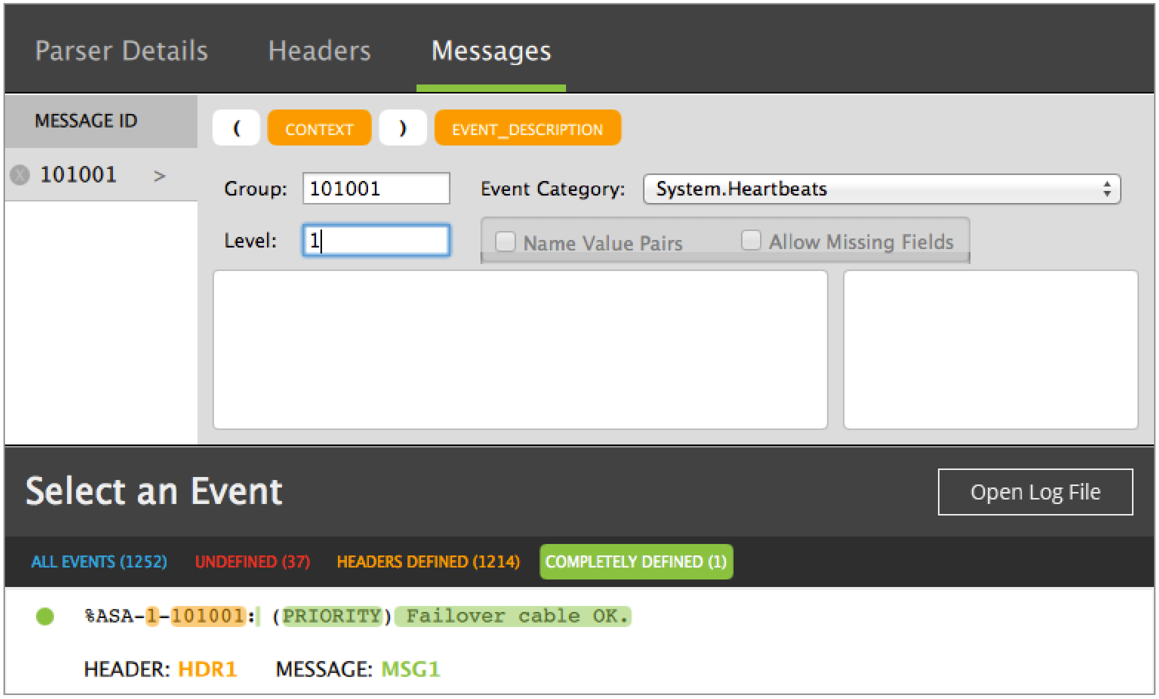
The message variables that you define create meta in the Log Decoder.

d. To add a throw-away field variable for information that you do not care about, see "Define a Throw-away Variable" on the facing page.

4. In the **Event Category** field, select a generic category for the message. For example, System.Heartbeats.

   The **Group** field populates from the header.

5. (Optional) In the **Level** field, type a severity level number to match the event. For example, 1.



Group: 101001 Event Category: System.Heartbeats

Level: 1 □ Name Value Pairs □ Allow Missing Fields

● %ASA-1-101001: (PRIORITY) Failover cable OK.

HEADER: HDR1 MESSAGE: MSG1

6. To add functions to your messages, right-click the box below the **Level** field, select **Add Function**, and select the function.
   - To add a Constant, see "Add a Constant Function" on the facing page.
   - To add an Event Time function, see "Add an Event Time Function" on page 20.

7. After you complete and save your changes, retrieve the completed parser files. You have a choice of two formats:

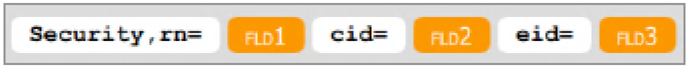
- **XML / INI** (In the main menu, select **File > Save** or **Save As**)
- **.envision** (In the menu, select **File > Export Parser**) This option creates an event source package that consists of the event source XML and configuration (INI) file.

8. Deploy the event source package in the Security Analytics platform to integrate the event source. RSA recommends that you first deploy the parser to a test system to verify that it parses log traffic correctly.

Define a Throw-away Variable

To add a throw-away field variable for information that you do not care about, create a variable and give it a name that is not defined in the variable list, such as fld1, fld2, or fld3.

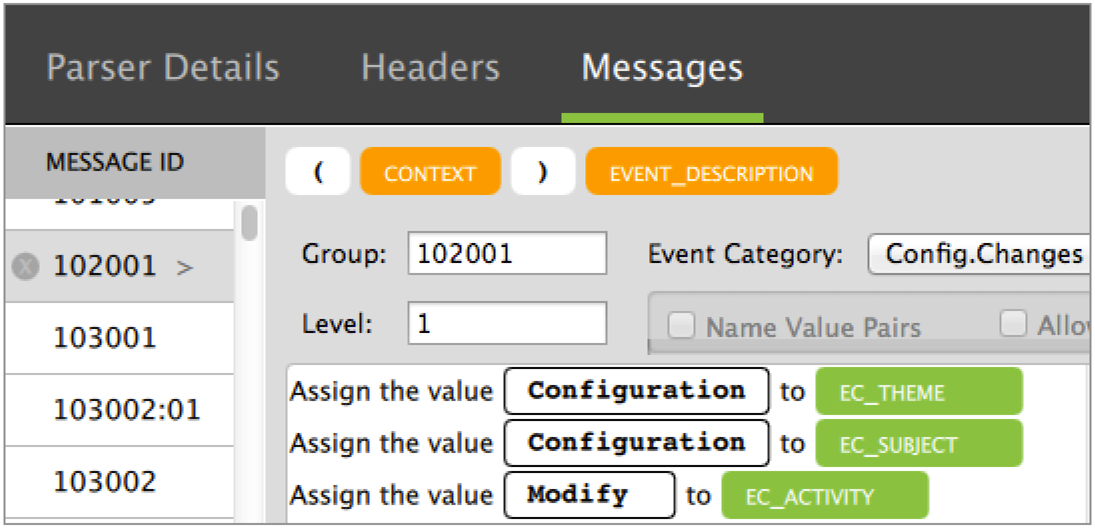
1. Right-click the pattern text and select **Edit**.
2. Highlight the text that you want to change to a variable and press **CTRL+K** (**COMMAND+K** for Mac).
3. Type the name of the throw-away field. For example, fld1
   Since throw-away fields are not in the mapping, the information contained will be removed when parsing.



```
Security,rn= FLD1 cid= FLD2 eid= FLD3
```

Add a Constant Function

1. Right-click the box below the **Level** field and select **Add Function > Assign Constant**.
2. In the **value** field, type the value that you want to assign to the variable.
3. In the **Set Variable** field, type the name of a variable that was not previously selected.



102001 >
103001
103002:01
103002

(CONTEXT) EVENT_DESCRIPTION

Group: 102001

Event Category: Config.Changes

Level: 1

Assign the value Configuration to EC_THEME

Assign the value Configuration to EC_SUBJECT

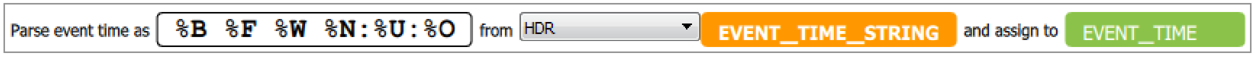
Assign the value Modify to EC_ACTIVITY

Add an Event Time Function

Use the Event Time function to change the format of an event source timestamp.

1. Right-click the box below the **Level** field and select **Add Function > Event Time**.
2. In the **Set Value** field, type the format that you want to assign to the time variable. For example, **%B %F %W %N:%U:%O**, which appears in the format **Jan 27 2015 23:55:29**.
3. In the **from** field, select whether to parse event time in this format from the message (MSG) or from the header (HDR).
4. Right-click the **Select Variable** fields and select a variable from the list. To add additional variables as required, right-click a variable and select **Add**.

Event time example:



Parse event time as %B %F %W %N:%U:%O from HDR EVENT_TIME_STRING and assign to EVENT_TIME

Event Time Function Formatting Characters

The following table shows the format characters that Log Decoder supports for the Event time function.

Format Character	Description
%C	Dates of this format: 04/20/05 14:01:57
%R	Full Month Name, fixed width field: January, February, March, April, May, June, July, August, September, October, November, December
%B	Abbreviated Month Name, fixed width field: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
%M	Numeric Month, fixed width field: 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12
%G	Numeric Month Variable width field: 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12
%D	Numeric Month Day, fixed width field: 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, ... , 23, 24, 25, 26, 27, 28, 29, 31
%F	Numeric Month Day Variable width field: 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, ... , 23, 24, 25, 26, 27, 28, 29, 31
%H	Hour, fixed width field: 00-23
%I	Hour, fixed width field: 00-12
%N	Hour: Variable width field: 00-12 , 00-24
%T	Minute, fixed width field: 00-59

%U	Minute: Variable width field: 00-59
%J	Julian day, fixed width field: 001-365
%P	Alpha, fixed width field: AM or PM
%Q	A.M./P.M.
%S	Seconds, fixed width field: 00-59
%O	Variable width field: Seconds: 00-59
%Y	Year: 00-99
%W	Year, fixed width field: 0000-9999
%Z	Hours:Min:Sec
%A	Days
%X	Unix Time-Stamp (e.g.: 1424849941)

March 2016

Trademarks

RSA, the RSA Logo and EMC are either registered trademarks or trademarks of EMC Corporation in the United States and/or other countries. All other trademarks used herein are the property of their respective owners. For a list of RSA trademarks, go to **www.emc.com/legal/emc-corporation-trademarks.htm**.