RSA enVision EventSource Integrator 1.2.1 Overview Guide

About RSA enVision EventSource Integrator

RSA enVision EventSource Integrator is a graphical tool that enables you to integrate event sources with the RSA enVision platform. Using EventSource Integrator, you can define how enVision reads and monitors the events from event sources. These definitions are stored as an XML file, called an event source XML file, which is deployed on the enVision platform.

Using EventSource Integrator, you can create a new event source XML file or edit an existing event source XML file.

You can create a new event source XML file for an event source that is not currently supported by enVision. After you deploy the event source XML file, enVision will be able to interpret the events and monitor the event source.

You can edit an existing event source XML file to add or edit definitions for events, or to correct errors. You may need to edit an event source XML file in one of the following situations:

- You upgrade to a new version of an event source that contains new, updated, or deprecated event messages.

  For example, if you upgrade from Firewall 1.0, which generates *n* events, to Firewall 2.0, which generates *n*+20 events, you can edit the existing event source XML file to add definitions for the additional events.

- You want to include additional definitions for existing events.

  For example, you have upgraded to Firewall 2.0 and added definitions for 10 of the 20 additional events. You can add definitions for the remaining 10 events by editing the event source XML file.

- You want to update the definition for an existing event in an event source XML file.
- You want to correct errors in an event source XML file.

Obtaining a Log File

To create the event source XML file that enVision will use to monitor the events from an event source, you must obtain a log file from the event source that you want to integrate with enVision. After you obtain the log file, you can use EventSource Integrator to create an event source XML file.

Before getting started with EventSource Integrator, you must know the log collection protocol that was configured when the event source was deployed with enVision.

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

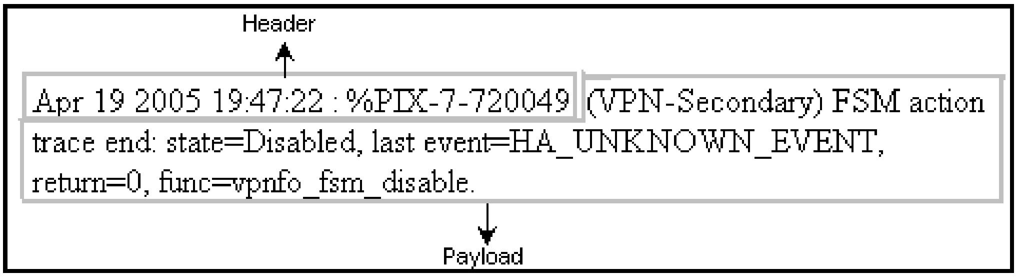
If you configured any other log collection protocol, you must use the LSData utility to obtain a log file in syslog format. For more information, see the EventSource Integrator Help topic "Use LSData to Obtain a Log File in Syslog Format."

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 enVision. 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 2 KB.
- The maximum size of the log file is 10 MB.
- The log file contains one or two instances of each unique event.

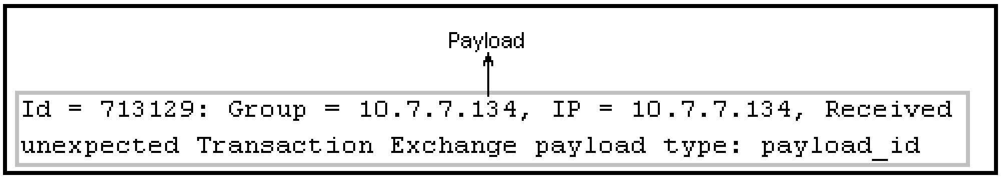
Understanding Events

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.



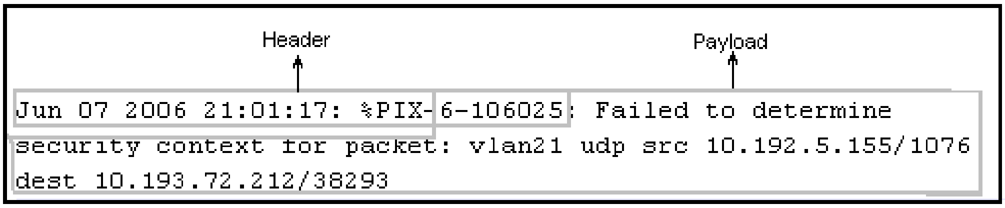
```
Apr 19 2005 19:47:22 : %PIX-7-720049 (VPN-Secondary) FSM action
trace end: state=Disabled, last event=HA_UNKNOWN_EVENT,
return=0, func=vpnfo_fsm_disable.
```

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.



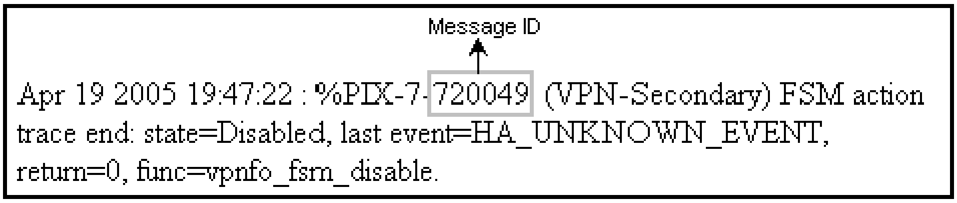
```
Jun 07 2006 21:01:17: %PIX-6-106025: Failed to determine
security context for packet: vlan21 udp src 10.192.5.155/1076
dest 10.193.72.212/38293
```

Note: Events may contain non-English characters. For information about support for non-English characters, see the EventSource Integrator Help topic “Events with Non-English Characters.”

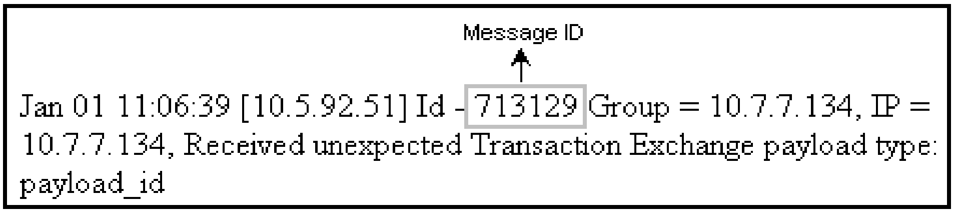
Header

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.

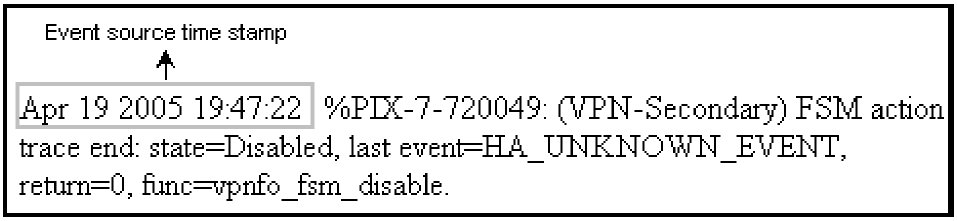


```
Apr 19 2005 19:47:22 : %PIX-7-720049 (VPN-Secondary) FSM action
trace end: state=Disabled, last event=HA_UNKNOWN_EVENT,
return=0, func=vpnfo_fsm_disable.
```



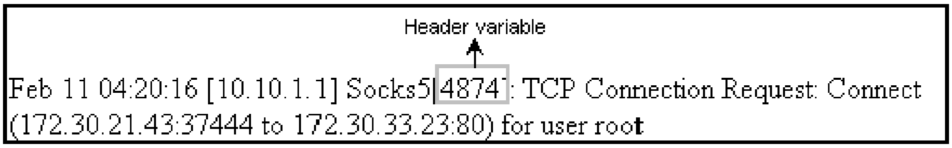
```
Jan 01 11:06:39 [10.5.92.51] Id - 713129 Group = 10.7.7.134, IP =
10.7.7.134, Received unexpected Transaction Exchange payload type:
payload_id
```

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.

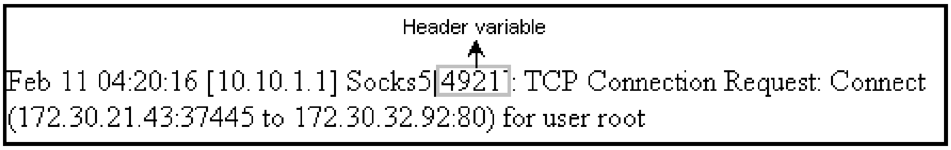

Event source time stamp

```
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
```

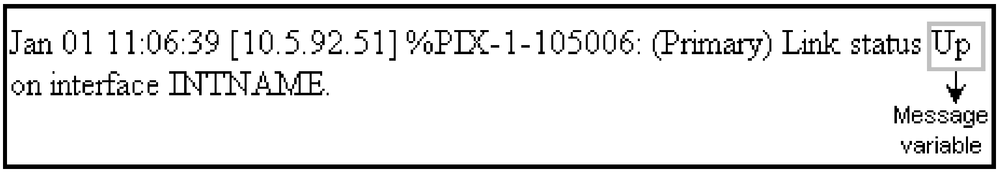

Header variable

```
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
```

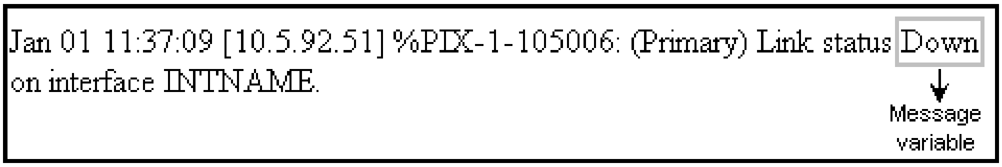
Payload

The payload contains detailed information about the event. The payload is the message in the event. RSA enVision 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.

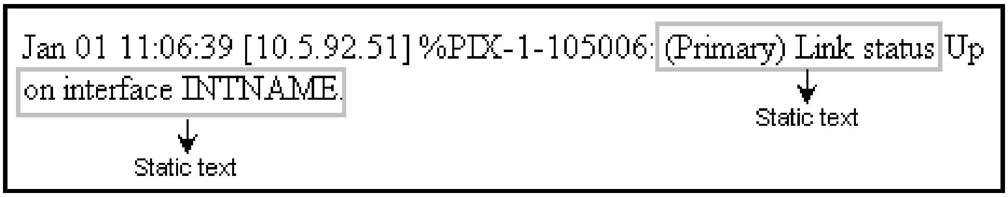


```
Jan 01 11:06:39 [10.5.92.51] %PIX-1-105006: (Primary) Link status Up
on interface INTNAME.
```



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

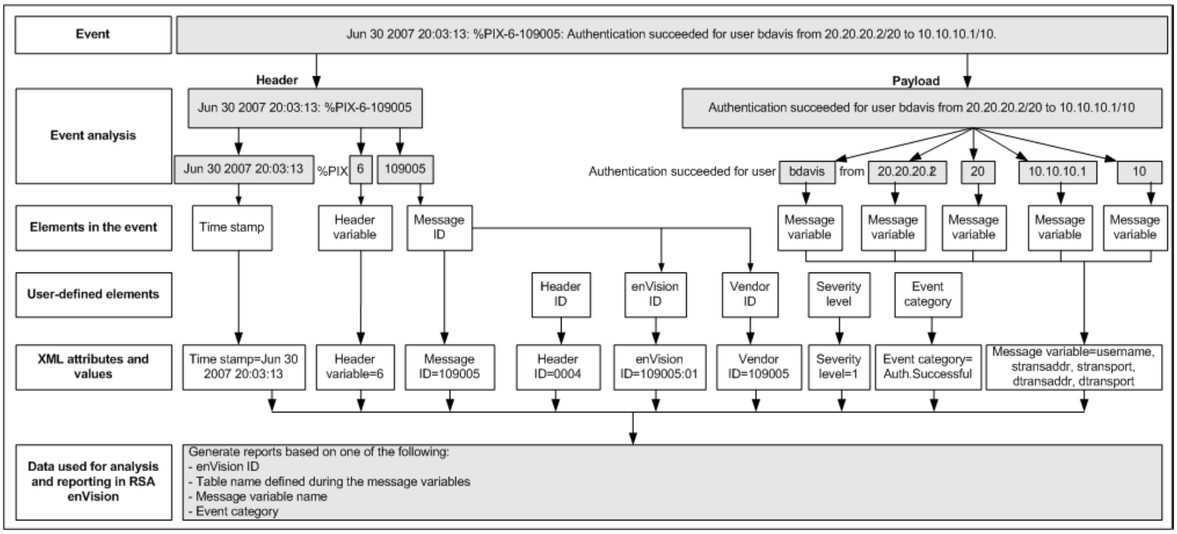
EventSource Integrator classifies all the values in the payload that are not message variables as static text. The following figure shows an example of values that EventSource Integrator classifies as static text.



```
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 an RSA enVision Table

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 RSA enVision. 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.
Header
Payload
Event analysis
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
Jun 30 2007 20:03:13
%PIX
6
109005
Authentication succeeded for user
bdavis
from
20.20.20.2
20
10.10.10.1
10
Elements in the event
Time stamp
Header variable
Message ID
Message variable
Message variable
Message variable
Message variable
Message variable
User-defined elements
Header ID
enVision ID
Vendor ID
Severity level
Event category
XML attributes and values
Time stamp=Jun 30 2007 20:03:13
Header variable=6
Message ID=109005
Header ID=0004
enVision ID=109005:01
Vendor ID=109005
Severity level=1
Event category= Auth.Successful
Message variable=username, stransaddr, stransport, dtransaddr, dtransport
Data used for analysis and reporting in RSA enVision
Generate reports based on one of the following:
- enVision ID
- Table name defined during the message variables
- Message variable name
- Event category

.

Getting Started with RSA enVision EventSource Integrator

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

Goal	Task	Reference
Integrate an event source that is not supported by enVision.	1. Create an XML file that contains definitions for the events generated by the event source.	"Creating an Event Source XML File"
	2. Validate the event source XML file for data pattern warnings.	"Validating an Event Source XML File"
	3. Validate the precedence of XML definitions.	"Validating the Precedence of XML Definitions"
	4. (Optional) View events that are parsed by a header or message definition.	"Viewing Parsed Events and Associated Definitions"
	5. (Optional) View the header and message definition that parse a selected event.	"Viewing Parsed Events and Associated Definitions"
	6. (Optional) Generate reports to analyze the events parsed against XML definitions.	"Generating a Report on Event Parsing"
	7. Create an event source package that consists of the event source XML file and configuration files.	"Creating an Event Source Package"
	8. Deploy the event source package in the enVision platform to integrate the event source.	"Deploying an Event Source Package in RSA enVision"

Goal	Task	Reference
Upgrade an event source that is already supported by enVision.	1. Edit the definitions for events in an event source XML file.	"Editing an Event Source XML File"
	2. Validate the event source XML file for data pattern warnings.	"Validating an Event Source XML File"
	3. Validate the precedence of XML definitions.	"Validating the Precedence of XML Definitions"
	4. (Optional) View events that are parsed by a header or message definition.	"Viewing Parsed Events and Associated Definitions"
	5. (Optional) View the header and message definition that parse a selected event.	"Viewing Parsed Events and Associated Definitions"
	6. (Optional) Generate reports to analyze the events parsed against XML definitions.	"Generating a Report on Event Parsing"
	7. Replace the event source XML file in the enVision platform to upgrade the event source.	"Deploying an Edited Event Source XML 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
2. Defining a Header
3. Defining a Message

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. EventSource 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 define the following elements in the header:

- Message ID
- Event source time stamp
- Header variable
- Payload

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

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
Vendor ID	Indicates the event identifier provided by the event source vendor. The vendor ID is automatically generated from the message ID defined in the header definition.
enVision ID	Indicates the identifier by which enVision identifies the event exclusively. The enVision ID can be defined in one of the following ways: • Same as vendor ID. • A combination of the message ID defined in the header definition and a unique variant. For example, if the message ID is 109801, the enVision ID can be defined as 109801:02. • A brief description of the event to identify the event. For example, in the following event, the message ID is 187698, and the enVision 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 enVision taxonomy. An event category is required for reporting in enVision.
Static values	(Optional) Assigns user-defined values to variables from an enVision table. For more information on enVision tables, see the EventSource Integrator Help topic “RSA enVision Tables.”

Message Element	Description
Functions	(Optional) Defines actions to be performed on variables in an event to generate user-defined values. EventSource Integrator supports the following functions: • PARMVAL assigns the value of a message variable to another variable. • SYSVAL populates the value of a message variable. • HDR assigns the value of a header variable to a variable from an enVision table. • CALC performs a calculation on values and variables in the event. • SUM, MIN, and MAX calculate the total, minimum, and maximum, respectively, of values and variables in the event. • EVNTTIME assigns the date and time information in the event to a time stamp variable. For more information, see the Help topic “Functions.”

Validating an Event Source XML File

After defining the event source XML, you should validate it for data pattern warnings. Validate the event source XML file against the corresponding log file. The results display the data pattern warnings that occur in the message variables. For better analysis and reporting, RSA recommends that you resolve all the warnings.You can then parse the log file to verify that the events are parsed by the header and message definitions in the event source XML file.

Validating the Precedence of XML Definitions

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, enVision parses an A B C event from the event source using the Message 10123 definition. RSA enVision 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.

EventSource Integrator displays the errors that occur in the header and message definition order. You can rearrange the header and message definitions and revalidate the precedence. 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 parsed events in the log file for a header or message definition in the event source XML file. 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.

Generating a Report on Event Parsing

After creating an event source XML file, you can generate a report to analyze how the events in the log file are parsed by the header and message definitions. A report provides a complete breakdown of the header and message definitions that shows the parsed values and associated variables for events in the log file.

You can generate reports for:

- A single event in the log file
- Selected events in the log file
- All the events in the log file

Creating an Event Source Package

You must create an event source package to deploy in enVision. EventSource Integrator generates the event source package in .zip format. The following table describes the files that the event source package contains.

File	Description
DeployEventSourceSetup.vbs	Deploys the event source package in enVision: • For a new event source, triggers the **UpdateESType.vbs**, **update_esi.bat**, and **update_content.zip** files. • For an existing event source, updates the event source XML file in the enVision environment.
update_esi.bat	Deploys or updates the event source in a site.
UpdateESType.vbs	Generates a unique event source type ID and assigns it to the event source in the enVision environment.
update_content.zip	Contains information about the event source that is added or updated in a single or multiple appliance site.

Note: During the event source packaging process, the “msg” suffix is appended to the event source XML filename, but not the event source package name. For example, CiscoPIX.xml is updated to CiscoPIXmsg.xml, and the event source package name remains CiscoPIX.zip. However, if the XML filename already ends with “msg,” no suffix is added to the XML filename, and “msg” is not included in the event source package name. For example, CiscoPIXmsg.xml remains CiscoPIXmsg.xml, and the event source package name is CiscoPIX.zip.

Deploying an Event Source Package in RSA enVision

You must deploy the event source package in your enVision site so that enVision can support the event source. To deploy the package, you must copy the .zip file to an enVision appliance, extract the contents, and run the **DeployEventSourceSetup.vbs** script. After you run the script, you must restart the NIC Service Manager Service.

You only need to deploy the event source XML package on one appliance:

- In a single appliance site, you deploy the event source XML package on the appliance.
- In a multiple appliance site, with or without Enhanced Availability, you deploy the event source XML package on any Application Server, and the package is automatically replicated to all other components in the site.
- In a multiple site deployment, with or without Enhanced Availability, you deploy the event source XML package on any Application Server, and the package is automatically replicated to all other components in all the sites.

Editing an Event Source XML File

Before you edit an event source XML file, you must ensure that you have identified and analyzed the events in the corresponding log file. You can edit an event source XML file that was created by EventSource Integrator or any other source.

Using the EventSource Integrator, you can perform two types of edit of an event source XML file:

Quick Edit. You can edit the elements in header or message definitions that do not have an impact on the parsing flow of the definition.

In a header, you can edit:

- Header ID
- Header variables
- Comment

In a message, you can edit:

- enVision ID
- Severity level
- Event category
- Message variables
- Static values
- Comment

Complete Edit. You can edit all the elements in header and message definitions.

In a header, you can edit:

- Header ID
- Message ID
- Event source time stamp
- Header variables
- Comment
- Payload

In a message, you can edit:

- enVision ID
- Severity level

- Message variables
- Event category
- Static values
- Comment
- Functions

Validating and Parsing an Edited Event Source XML File

After editing an event source XML file, you can perform the following tasks, just as you do when creating an event source XML file:

- Validate the edited event source XML file for data pattern warnings. For more information, see "Validating an Event Source XML File."
- Validate the precedence of header and message definitions in the edited event source XML file. For more information, see "Validating the Precedence of XML Definitions."
- View the parsed events in the log file for header and message definitions in the edited event source XML file or view an associated XML definition for a selected event in the log file. For more information, see "Viewing Parsed Events and Associated Definitions."
- Generate a report on event parsing for the edited event source XML file. For more information, see "Generating a Report on Event Parsing."

Deploying an Edited Event Source XML File

If you have previously generated an event source package for the event source and deployed the package in enVision, you must copy the .zip file to an enVision appliance, extract the contents, and run the **DeployEventSourceSetup.vbs** script. You must not change the name of the event source XML file. When you replace the event source XML file, the edited definitions are then available for parsing in enVision. After you run the script, you must restart the NIC Service Manager Service.

If you have not yet deployed an event source package for the event source, follow the instructions in "Creating an Event Source Package" and "Deploying an Event Source Package in RSA enVision."

You only need to deploy the event source XML package on one appliance:

- In a single appliance site, you deploy the event source XML package on the appliance.
- In a multiple appliance site, with or without Enhanced Availability, you deploy the event source XML package on any Application Server, and the package is automatically replicated to all other components in the site.
- In a multiple site deployment, with or without Enhanced Availability, you deploy the event source XML package on any Application Server, and the package is automatically replicated to all other components in all the sites.

February 2012

Trademarks