JUNIPER NETWORKS | Engineering Simplicity

Juniper Secure Analytics Log Event Extended Format Guide

Published
2020-10-08

Release
7.4.1

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Secure Analytics Log Event Extended Format Guide
7.4.1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation

IN THIS SECTION

The Log Event Extended Format (LEEF) is a customized event format for JSA. Use this guide to understand how to generate, integrate, identify, and process LEEF events.

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at https://www.juniper.net/documentation/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at https://www.juniper.net/books.

Documentation Conventions

Table 1 on page v defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page v defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		

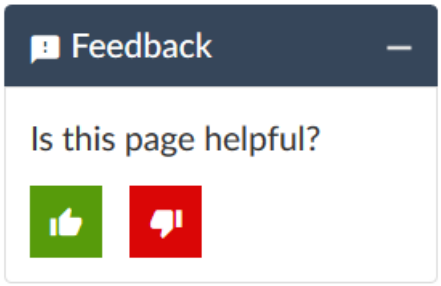
Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active Juniper Care or Partner Support Services support contract, or are

covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://entitlementsearch.juniper.net/entitlementsearch/

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see https://support.juniper.net/support/requesting-support/.

CHAPTER 1

Log Event Extended Format (LEEF)

Log Event Extended Format (LEEF)

The Log Event Extended Format (LEEF) is a customized event format for JSA.

Any vendor can use this documentation to generate LEEF events.

JSA can integrate, identify, and process LEEF events. LEEF events must use UTF-8 character encoding.

You can send events in LEEF output to by using the following protocols:

- Syslog
- File import with the Log File Protocol

NOTE: Before JSA can use LEEF events, you must complete Universal LEEF configuration tasks. For more information about configuring the log file protocol to collect Universal LEEF events, see the *Configuring DSMs Guide*.

The method that you select to provide LEEF events determines whether the events can be automatically discovered in JSA. When events are automatically discovered the level of manual configuration that is needed in JSA is reduced.

As LEEF events are received, JSA analyzes the event traffic in an attempt to identify the device or appliance. This process is referred to as traffic analysis. It typically takes at least 25 LEEF events to identify and create a new log source in JSA. Until traffic analysis identifies the event source, the initial 25 events are categorized as SIM Generic Log DSM events and the event name is set as Unknown Log Event. After the event traffic is identified, JSA creates a log source to properly categorize and label any events that are forwarded from your appliance or software. Events that are sent from your device are viewable in JSA on the **Log Activity** tab.

NOTE: When a log source cannot be identified after 1,000 events, JSA creates a system notification and removes the log source from the traffic analysis queue. JSA is still capable of collecting the events, but a user must intervene and create a log source manually to identify the event type.

RELATED DOCUMENTATION

LEEF Event Components | 11

LEEF Event Components

The Log Event Extended Format (LEEF) is a customized event format for JSA that contains readable and easily processed events for JSA. The LEEF format consists of a Syslog header, a LEEF header, and event attributes.

Syslog Header

The syslog header contains the timestamp and IPv4 address or host name of the system that is providing the event. The syslog header is an optional component of the LEEF format. If you include a syslog header, you must separate the syslog header from the LEEF header with a space. The syslog header must conform to the formats specified in RFC 3164 or RFC 5424.

RFC 3164 header format:

NOTE: The priority tag is optional for JSA.

<priority tag><timestamp><IP address or hostname>

The priority tag, if present, must be 1 - 3 digits and must be enclosed in angle brackets. For example <13>.

Examples of RFC 3164 header:

- <13>Jan 18 11:07:53 192.168.1.1
- Jan 18 11:07:53 myhostname

RFC 5424 header format:

NOTE: The priority tag is required.

<priority tag>1<timestamp><IP address or hostname>

The priority tag must be 1 - 3 digits and must be enclosed in angle brackets. For example, <13>. The timestamp must be in the format: yyyy-MM-ddTHH:mm:ss.SSSZ.

NOTE:

- The 'T' must be a literal T character.
- The 'Z' can be a literal Z or it can be a timezone value in the following format: -04:00

Examples of RFC 5424 header:

- <13>1 2019-01-18T11:07:53.520Z 192.168.1.1
- <133>1 2019-01-18T11:07:53.520+07:00 myhostname

LEEF Header

The LEEF header is a required field for LEEF events. The LEEF header is a pipe delimited (|) set of values that identifies your software or appliance to JSA.

The following list shows:

- **LEEF:Version|Vendor|Product|Version|EventID|**
- **LEEF:1.0|Microsoft|MSExchange|2013 SP1|15345|**
- **LEEF:2.0|Lancope|StealthWatch|6.5|41|^|**

Event Attributes

Event attributes identify the payload information of the event that is produced by your appliance or software. Every event attribute is a key-value pair with a tab that separates individual payload events. The LEEF format contains a number of predefined event attributes, that JSA uses to categorize and display the event.

The following list shows:

- key=value<tab>key=value<tab>key=value<tab>key=value<tab>
- src=192.0.2.0 dst=172.50.123.1 sev=5 cat=anomaly srcPort=81 dstPort=21 usrName=joe.black

Use the **DelimiterCharacter** in the LEEF 2.0 header to specify an alternative delimiter to the attributes. You can use a single character or the hex value for that character. The hex value can be represented by the prefix 0x or x, followed by a series of 1-4 characters (0-9A-Fa-f).

Table 3: Attribute Delimiter Character Examples for LEEF 2.0

Delimiter	Header
Caret (^)	**LEEF:2.0|Vendor|Product|Version|EventID|^|**
Caret (hex value)	**LEEF:2.0|Vendor|Product|Version|EventID|x5E|**
Broken vertical bar (¦)	**LEEF:2.0|Vendor|Product|Version|EventID|xa6|**

The following table describes LEEF formats.

Table 4: LEEF Format Descriptions

Type	Entry	Delimiter	Description
Syslog Header	IP address	Space	The IP address or the host name of the software or appliance that provides the event to JSA. The IP address in the syslog header is used by JSA to route the event to the correct log source in the event pipeline. Don't use an IPv6 address in your syslog header. JSA cannot route an IPv6 address in the syslog header to the event pipeline. Also, an IPv6 address might not display properly in the **Log Source Identifier** field in JSA. When JSA can't understand an IP address in the syslog header, the system defaults to the packet address to properly route the event.
LEEF Header	LEEF:version	Pipe	The LEEF version information is an integer value that identifies the major and minor version of the LEEF format that is used for the event, for example, **LEEF:1.0|Vendor|Product|Version|EventID|**

Table 4: LEEF Format Descriptions *(continued)*

Type	Entry	Delimiter	Description
LEEF Header	Vendor or manufacturer name	Pipe	Vendor is a text string that identifies the vendor or manufacturer of the device that sends the syslog events in LEEF format, for example, **LEEF:1.0|Microsoft|Product|Version|EventID|** The Vendor and Product fields must contain unique values when specified in the LEEF header.
LEEF Header	Product name	Pipe	The product field is a text string that identifies the product that sends the event log to JSA, for example, **LEEF:1.0|Microsoft|MSExchange|Version|EventID|** The Vendor and Product fields must contain unique values when specified in the LEEF header.
LEEF Header	Product version	Pipe	Version is a string that identifies the version of the software or appliance that sends the event log, for example, **LEEF:1.0|Microsoft|MSExchange|2013 SP1|EventID|**
LEEF Header	EventID	Pipe	EventID is a unique identifier for an event. The purpose of the EventID is to provide a fine grain, unique identifier for an event without the need to examine the payload information. An EventID can contain either a numeric identified or a text description, for example, • **LEEF:1.0|Microsoft|MSExchange|2013|7732|** • **LEEF:1.0|Microsoft|MSExchange|2013|Logon Failure|** **NOTE:** The value of the event ID must be a consistent and static across products that support multiple languages. If your product supports multi-language events, you can use a numeric or textual value in the EventID field, but it must not be translated when the language of your appliance or application is altered. The EventID field cannot exceed 255 characters.
LEEF Header	Delimiter Character	Pipe	Use the **DelimiterCharacter** in the LEEF 2.0 header to specify an alternate delimiter to the attributes. You can use a single character or the hex value for that character. The hex value can be represented by the prefix 0x or x, followed by a series of 1-4 characters (**0-9A-Fa-f**).

Table 4: LEEF Format Descriptions *(continued)*

Type	Entry	Delimiter	Description
Event Attributes	Predefined Key Entries	Tab Delimiter Character	Event attribute is set of key value pairs that provide detailed information about the security event. Each event attribute must be separated by tab or the delimiter character, but the order of attributes is not enforced, for example, **src=172.16.77.100**

RELATED DOCUMENTATION

Predefined LEEF Event Attributes | **15**
Custom Event Keys | **25**
Custom Event Date Format | **27**

Predefined LEEF Event Attributes

The Log Event Extended Format (LEEF) supports a number of predefined event attributes for the event payload.

LEEF uses a specific list of name-value pairs that are predefined LEEF event attributes. These keys outline fields that are identifiable to JSA. Use these keys on your appliance when possible, but your event payloads are not limited by this list. LEEF is extensible and you can add more keys to the event payload for your appliance or application.

The following table describes the predefined event attributes.

Table 5: Pre-defined Event Attributes

Key	Value type	Normalized event field? Yes or No	Description
cat	String	Yes	An abbreviation for event category is used to extend the EventID field with more specific information about the LEEF event that is forwarded to JSA. Cat and the EventID field in the LEEF header help map your appliance event to a JSA Identifier (QID) map entry. The EventID represents the first column and the category represents the second column of the QID map. **NOTE:** The value of the event category must be consistent and static across products that support multiple languages. If your product supports multi-language events, you can use a numeric or textual value in the cat field. The value in the cat field must not be translated when the language of your appliance or application is altered.
cat (continued)	String	Yes	Example 1: Use the **cat** key to extend the EventID with additional information to describe the event. If the EventID is defined as a User Login event, use the category to further categorize the event, such as a success or failed login. You can define your EventIDs further with the **cat** key, and the extra detail from the event can be used to distinguish between events when the same EventID is used for similar event types, for example, **LEEF:1.0|Microsoft|Exchange|2013|Login Event|cat=Failed** **LEEF:1.0|Microsoft|Exchange|2013|Login Event|cat=Success** Example 2: Use the **cat** key to define a high-level event category and use the EventID to define the low-level. This situation can be important when the EventID doesn't match any value in the QID map. When the EventID doesn't match any value in the QID map, JSA can use the category and other keys to further determine the general nature of the event. This "fallback" prevents events from being identified as unknown and JSA can categorize the events based on the known information from the key attribute fields of the event payload, for example, **LEEF:1.0|Microsoft|Endpoint|2015|** **Conficker_worm|cat=Detected**

Table 5: Pre-defined Event Attributes *(continued)*

Key	Value type	Normalized event field? Yes or No	Description
devTime	Date	Yes	The raw event date and time that is generated by your appliance or application that provides the LEEF event. JSA uses the **devTime** key, along with **devTimeFormat** to identify and properly format the event time from your appliance or application. If the **devTime** value is an epoch value of 10 or 13 digits, a **devTimeFormat** string is not required. Otherwise, the **devTime** and **devTimeFormat** keys must be used together to ensure that the time of the event is accurately parsed by JSA. When present in the event payload, **devTime** is used to identify the event time, even when the syslog header contains a date and time stamp. The syslog header date and time stamp is a fallback identifier, but **devTime** is the preferred method for event time identification.
devTimeFormat	String	No	Applies formatting to the raw date and time of the **devTime key**. The **devTimeFormat** key is required if your event log contains **devTime**. For more information, see "Custom Event Date Format" on page 27.
proto	Integer or Keyword	Yes	Identifies the transport protocol of the event. For a list of keywords or integer values, see the Internet Assigned Numbers Authority website, http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml
sev	Integer	Yes	Indicates the severity of the event. 1 is the lowest event severity. 10 is the highest event severity. Attribute Limits: 1-10.
src	IPv4 or IPv6 Address	Yes	The IP address of the event source.

Table 5: Pre-defined Event Attributes *(continued)*

Key	Value type	Normalized event field? Yes or No	Description
dst	IPv4 or IPv6 Address	Yes	The IP address of the event destination.
srcPort	Integer	Yes	The source port of the event. Attribute Limits: 0 - 65535
dstPort	Integer	Yes	The destination port of the event. Attribute Limits: 0 - 65535
srcPreNAT	IPv4 or IPv6 Address	Yes	The source IP address of the event message before Network Address Translation (NAT).
dstPreNAT	IPv4 or IPv6 Address	Yes	The destination address for the event message before Network Address Translation (NAT).
srcPostNAT	IPv4 or IPv6 Address	Yes	The source IP address of the message after Network Address Translation (NAT) occurred.
dstPostNAT	IPv4 or IPv6 Address	Yes	The destination IP address of the message after Network Address Translation (NAT) occurred.
usrName	String	Yes	The user name that is associated with the event. Attribute Limits: 255
srcMAC	MAC Address	Yes	The MAC address of the event source in hexadecimal. The MAC address is made up of six groups of two hexadecimal digits, which are colon-separated, for example, **11:2D:1a:2b:3c:4d**

Table 5: Pre-defined Event Attributes *(continued)*

Key	Value type	Normalized event field? Yes or No	Description
dstMAC	MAC Address	Yes	The MAC address of the event destination in hexadecimal. The MAC address is composed of six groups of two hexadecimal digits, which are colon-separated, for example, **11:2D:1a:2b:3c:4d**
srcPreNATPort	Integer	Yes	The port number of the event source before Network Address Translation (NAT). Attribute Limits: 0 - 65535
dstPreNATPort	Integer	Yes	The port number of the event destination before Network Address Translation (NAT). Attribute Limits: 0 - 65535
srcPostNATPort	Integer	Yes	The port number of the event source after Network Address Translation (NAT). Attribute Limits: 0 - 65535
dstPostNATPort	Integer	Yes	The port number of the event destination after Network Address Translation (NAT). Attribute Limits: 0 - 65535

Table 5: Pre-defined Event Attributes *(continued)*

Key	Value type	Normalized event field? Yes or No	Description
identSrc	IPv4 or IPv6 Address	Yes	Identity source represents an extra IPv4 or IPv6 address that can connect an event with a true user identify or true computer identity. Example 1: Connecting a person to a network identity. User X logs in from their notebook and then connects to a shared system on the network. When their activity generates an event, then the **identSrc** in the payload can be used to include more IP address information. JSA uses the **identSrc** information in the event along with the payload information, such as *username*, to identify that user X is bob.smith. The following identity keys depend on **identSrcs** presence in the event payload: **identHostName** **identNetBios** **identGrpName** **identMAC**
identHostName	String	Key	Host name information that is associated with the **identSrc** to further identify the true host name that is tied to an event. The **identHostName** parameter is usable by JSA only when your device provides both the **identSrc** key and **identHostName** together in an event payload. Attribute Limits: 255
identNetBios	String	Yes	NetBIOS name that is associated with the **identSrc** to further identify the identity event with NetBIOS name resolution. The **identNetBios** parameter is usable by JSA only when your device provides both the **identSrc** key and **identNetBios** together in an event payload. Attribute Limits: 255

Table 5: Pre-defined Event Attributes *(continued)*

Key	Value type	Normalized event field? Yes or No	Description
identGrpName	String	Yes	Group name that is associated with the **identSrc** to further identify the identity event with Group name resolution. The **identGrpName** parameter is usable by JSA only when your device provides both the **identSrc** key and **identGrpName** together in an event payload. Attribute Limits: 255
identMAC	MAC Address	Yes	Reserved for future use in the LEEF format.
vSrc	IPv4 or IPv6 Address	No	The IP address of the virtual event source.
vSrcName	String	No	The name of the virtual event source. Attribute Limits: 255
accountName	String	No	The account name that is associated with the event. Attribute Limits: 255
srcBytes	Integer	No	Indicates the byte count from the event source.
dstBytes	Integer	No	Indicates the byte count to the event destination.
srcPackets	Integer	No	Indicates the packet count from the event source.
dstPackets	Integer	No	Indicates the packet count to the event destination.
totalPackets	Integer	No	Indicates the total number of packets that are transmitted between the source and destination.
role	String	No	The type of **role** that is associated with the user account that created the event, for example, Administrator, User, Domain Admin.

Table 5: Pre-defined Event Attributes *(continued)*

Key	Value type	Normalized event field? Yes or No	Description
realm	String	No	The **realm** that is associated with the user account. Depending on your device, can be a general grouping or based on region, for example, accounting, remote offices.
policy	String	No	A **policy** that is associated with the user account. This **policy** is typically the security policy or group policy that is tied to the user account.
resource	String	No	A **resource** that is associated with the user account. This **resource** is typically the computer name.
url	String	No	**URL** information that is included with the event.
groupID	String	No	The **groupID** that is associated with the user account.
domain	String	No	The **domain** that is associated with the user account.
isLoginEvent	Boolean string	No	Identifies if the event is related to a user login, for example, **isLoginEvent=true** **isLoginEvent=false** This key is reserved in the LEEF specification, but not implemented in JSA. Attribute Limits: true or false
isLogoutEvent	Boolean string	No	Identifies if the event is related to a user logout, for example, **isLogoutEvent=true** **isLogoutEvent=false** This key is reserved in the LEEF specification, but not implemented in JSA. Attribute Limits: true or false

Table 5: Pre-defined Event Attributes *(continued)*

Key	Value type	Normalized event field? Yes or No	Description
identSecondIp	IPv4 or IPv6 Address	No	Identity second IP address represents an IPv4 or IPv6 address that is used to associate a device event that includes a secondary IP address. Secondary IP addresses can be in events by routers, switches, or virtual LAN (VLAN) device events. This key is reserved in the LEEF specification, but not implemented in JSA.
calLanguage Attribute Limits: 2	String	No	Identifies the language of the device time (**devTime**) key to allow translation and to ensure that JSA correctly parses the date and time of events that are generated in translated languages. The **calLanaguage** field can include two alphanumeric characters to represent the event language for the device time of your event. All **calLanguage** alphanumeric characters follow the ISO 639-1 format, for example, **calLanguage=fr devTime=avril 09 2014 12:30:55** **calLanguage=de devTime=Di 30 Jun 09 14:56:11** This key is reserved in the LEEF specification, but not implemented currently in JSA. Attribute Limits: 2

Table 5: Pre-defined Event Attributes *(continued)*

Key	Value type	Normalized event field? Yes or No	Description
calCountryOrRegion	String	No	Extends the **calLanguage** key to provide more translation information that can include the country or region for the event device time (**devTime**). The key **calCountryOrRegion** must be used with the **calLanguage** key. The **calCountryOrRegion** field can include two alphanumeric characters to represent the event country or region for the device time of your event. All **calCountryOrRegion** alphanumeric characters follow the ISO 3166 format, for example, **calLanguage=de calCountryOrRegion=DE devTime=Di 09 Jun 2014 12:30:55** **calLanguage=en calCountryOrRegion=US devTime=Tue 30 Jun 09** This key is reserved in the LEEF specification, but not implemented in JSA. Attribute Limits: 2

NOTE: Non-normalized predefined LEEF event attributes are not automatically parsed for all log source types. However, JSA provides custom properties (either built-in or from the Juniper Security App Exchange) for some of these keys. You can configure custom properties for non-normalized keys to parse by using Regex. To configure a key to parse, the input is **key**=([^\t]+).

The following examples show Regex inputs for non-normalized predefined keys, where the delimiter that follows the caret (^) is a horizontal tab in LEEF V1.0:

- The input for **vSrc** is **vSrc**=([^\t]+).
- The input for **vSrcName** is **vSrcName**=([^\t]+).
- The input for **accountName** is **accountName**=([^\t]+).

The following examples show Regex inputs for non-normalized predefined keys, where the delimiter that follows the caret (^) is a customized separator character in LEEF V2.0:

- If you use # as the delimiter, the input for **vSrc** is **vSrc**=([^#]+).
- If you use | as the delimiter, the input for **vSrc** is **vSrc**=([^|]+).

JSA 7.3.2 or later includes property auto-detection for custom properties of both predefined and custom LEEF event attributes. Property auto-detection makes it easier to configure custom properties, without the use of Regex.

RELATED DOCUMENTATION

Custom Event Keys | 25
Custom Event Date Format | 27
LEEF Event Components | 11

Custom Event Keys

IN THIS SECTION

Vendors and partners can define their own custom event keys and include them in the payload of the LEEF format.

Use custom key value-pair attributes in an event payload when there is no default key to represent information about an event for your appliance. Create custom event attributes only when there is no acceptable mapping to a predefined event attribute. For example, if your appliance monitors access, you can require the file name that is accessed by a user where no file name attribute exists in LEEF by default.

NOTE: Event attribute keys and values can appear one time only in each payload. Using a key-value pair twice in the same payload can cause JSA to ignore the value of the duplicate key.

Custom event keys are non-normalized, which means that any specialized key value pairs you include in your LEEF event are not displayed by default on the **Log Activity** tab of JSA. To view custom attributes and non-normalized events on the **Log Activity** tab of JSA, you must create a custom event property. Non-normalized event data is still part of your LEEF event, is searchable in JSA, and is viewable in the event payload. For more information about creating a custom event property, see the *Juniper Secure Analytics Administration Guide*.

Best Practices Guidelines for LEEF Events

LEEF is flexible and can create custom key value pairs for events, but you must follow some best practices to avoid potential parsing issues.

Items that are marked *Allowed* can be included in a key or value, and is not in violation of LEEF but these items are not good practice when you create custom event keys.

The following list contains custom key and value general guidelines:

- Use alphanumeric (A-Z, a-z, and 0-9) characters, but avoid tab, pipe, or caret delimiters in your event payload keys and values (key=value).
  - Correct—usrName=Joe.Smith
  - Incorrect—usrName=Joe<tab>Smith
- Contain a single word for the key attribute (key=value).
  - Correct—file name=pic07720.gif
  - Allowed—file name=pic07720.gif
  - Allowed—file name =pic07720.gif
- A user-defined key cannot use the same name as a LEEF predefined key. For more information, see "Predefined LEEF Event Attributes" on page 15.
- Key values must be human readable, if possible, to help you to investigate event payloads.
  - Correct—deviceProcessHash=value
  - Correct—malwarename=value
  - Allowed—EBFDFBE14D4=value

RELATED DOCUMENTATION

Custom Event Date Format | 27
LEEF Event Components | 11
Predefined LEEF Event Attributes | 15

Custom Event Date Format

To create a customized event format, your device must supply the raw date format by using the **devTime** event attribute in the payload of the event.

Use the **devTimeformat** to format the **devTime** event attribute to display the event in JSA. The suggested **devTimeFormat** patterns are listed in the following table:

Table 6: DevTimeFormat Suggested Patterns

devTimeFormat Pattern	Result
devTimeFormat=MMM dd yyyy HH:mm:ss	Jun 06 2015 16:07:36
devTimeFormat=MMM dd yyyy HH:mm:ss.SSS	Jun 06 2015 16:07:36.300
devTimeFormat=MMM dd yyyy HH:mm:ss.SSS z	Jun 06 2015 02:07:36.300 GMT

For more information about specifying a date format, see the SimpleDateFormat information on the Java Web Page.

RELATED DOCUMENTATION

LEEF Event Components | **11**
Predefined LEEF Event Attributes | **15**
Custom Event Keys | **25**