RSA® NETWITNESS®
Logs
Implementation Guide

Claroty Platform 2.1

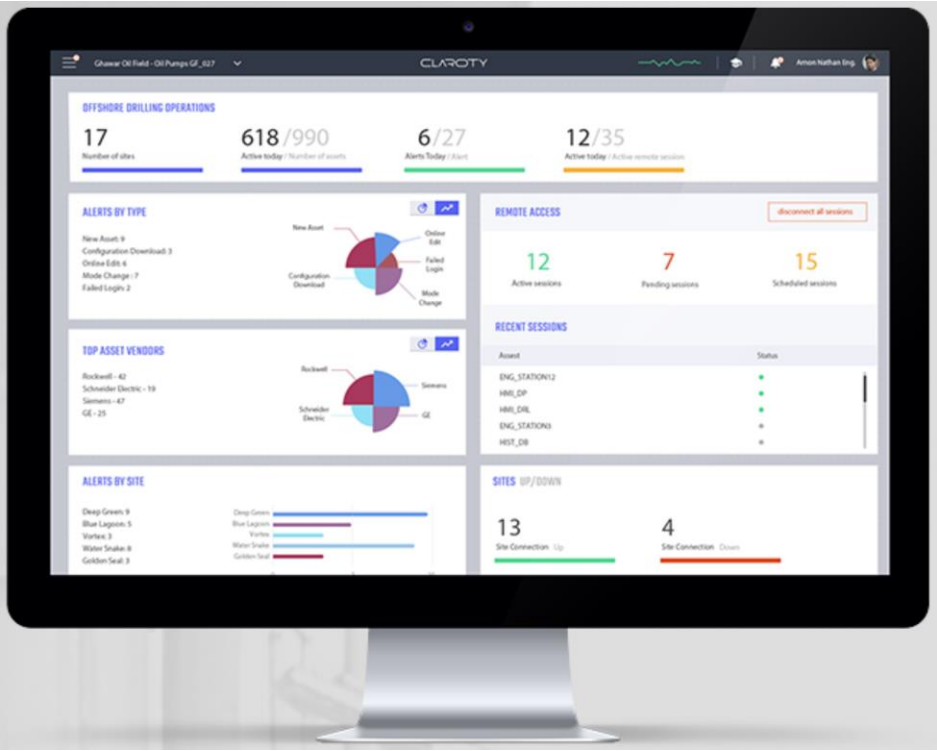
Jeffrey Carlson, RSA Partner Engineering
Last Modified: April 30th, 2018

RSA READY

Solution Summary

Claroty enables customers to secure and optimize the industrial control networks that run the world's most critical infrastructure. The company's enterprise-class OT security platform is designed to address the unique safety and reliability requirements necessary to protect industrial networks– e.g., industrial control systems, SCADA, industrial IOT and others.

RSA NetWitness Features	
Claroty Platform 2.0	
Integration package name	Common Event Format
Event source class	Analysis
Device display name within NetWitness	claroty_ctd
Collection method	Syslog



RSA NetWitness Community

The RSA NetWitness Community is an online forum for customers and partners to exchange technical information and best practices with each other. All NetWitness customers and partners are invited to register and participate in the **RSA NetWitness Community**.

Release Notes

Release Date	What's New In This Release
10/25/2017	Initial support for Claroty Platform 2.0
04/17/2018	Updated support for Claroty Platform 2.1

Important: The RSA NetWitness CEF parser is dependent on the partner adhering to the CEF Rules outlined in the *ArcSight Common Event Format (CEF) Guide*.

Eg. Jan 18 11:07:53 host CEF:Version|Device Vendor|Device Product|Device Version|Signature ID|Name|Severity|[Extension]

Important: The time displayed in the CEF log header is parsed into evt.time.str. For this integration, there is also a custom field, receipt.time, that contains the timestamp listed in the cef key "rt".

Partner Product Configuration

Before You Begin

This section provides instructions for configuring Claroty Platform with RSA NetWitness. This document is not intended to suggest optimum installations or configurations.

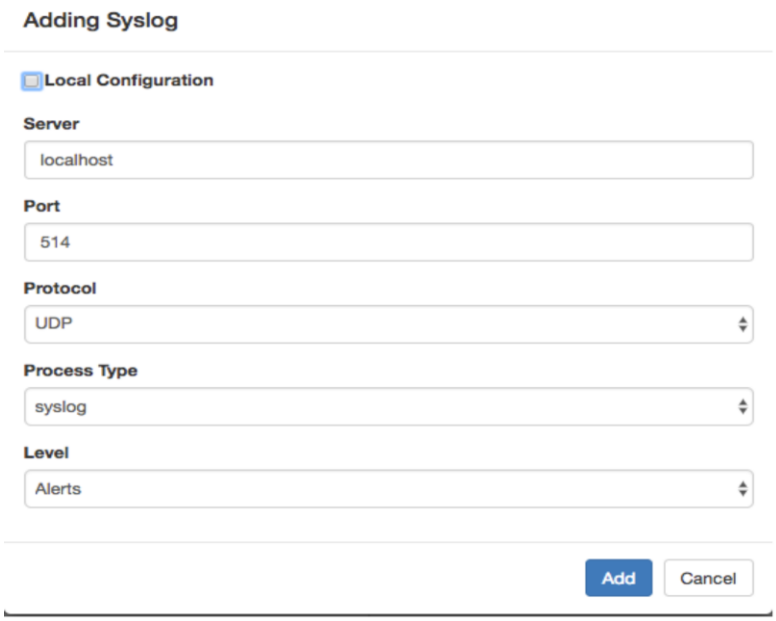
It is assumed that the reader has both working knowledge of all products involved, and the ability to perform the tasks outlined in this section. Administrators should have access to the product documentation for all products in order to install the required components.

All Claroty components must be installed and working prior to the integration. Perform the necessary tests to confirm that this is true before proceeding.

! Important: The configuration shown in this Implementation Guide is for example and testing purposes only. It is not intended to be the optimal setup for the device. It is recommended that customers make sure Claroty Platform is properly configured and secured before deploying to a production environment. For more information, please refer to the Claroty Platform documentation or website.

Claroty Platform Configuration

In order to send events and alerts to RSA NetWitness, the Claroty Platform configuration tool (port 5001) should be used to configure syslog output:



Claroty Platform collects traffic from the network. Each deviation is considered an event. Multiple events are aggregated into a human readable alert. Both events and alerts can be configured to be

outputted by the system in CEF format. An alert may consist of one or multiple events, depending on the type of alert.

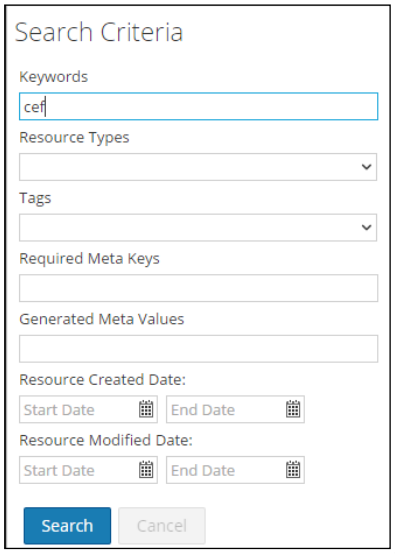
Every new alert (or the resolution of an alert) and the events associated with it, will be sent through to RSA NetWitness to have a unified view integrated in the full context of the organization's security monitoring.

RSA NetWitness Configuration

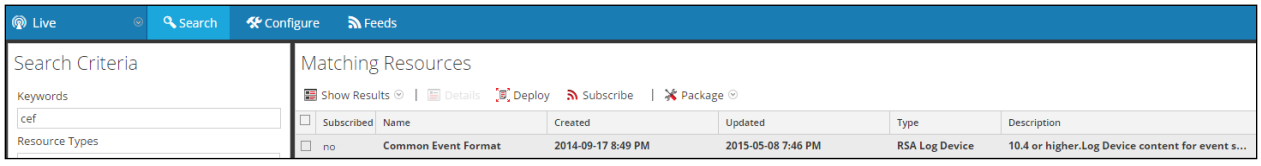
Deploy the Common Event Format (CEF) Parser

In order to ingest events from Claroty Platform, you will need to deploy the *Common Event Format parser* from the **NetWitness Live** module. Log into NetWitness and perform the following actions:

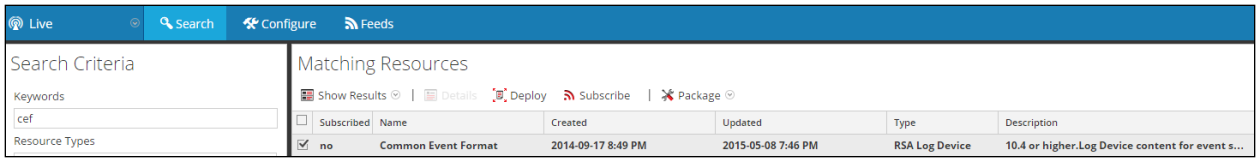
1. From the NetWitness menu, select **Live > Search**.
1. In the keywords field, enter: **CEF**



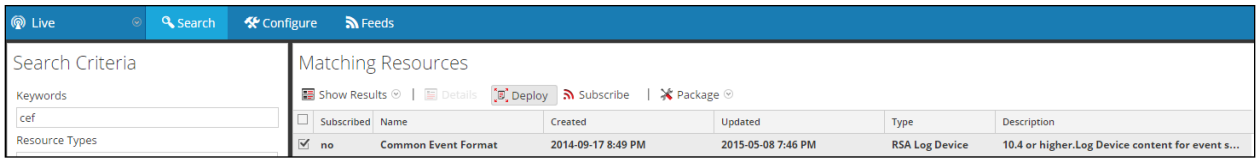
2. RSA NetWitness will display the **Common Event Format** in Matching Resources.



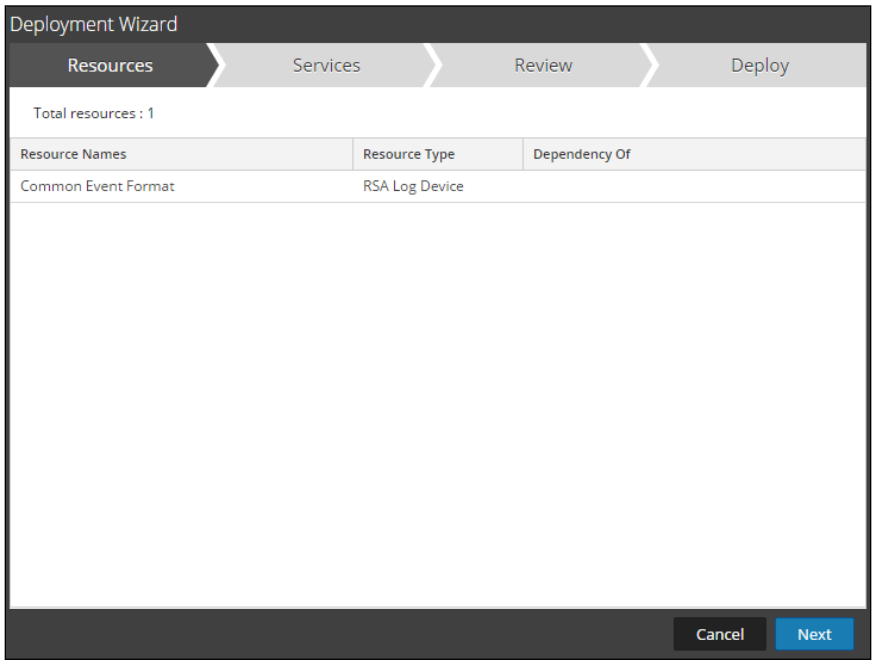
3. Select the checkbox next to **Common Event Format**.



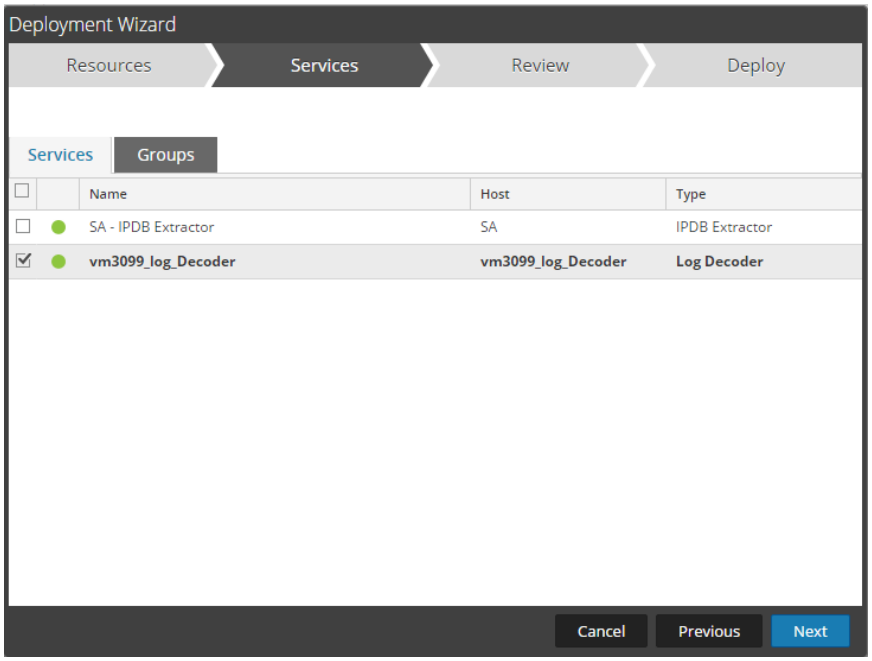
4. Click **Deploy** in the menu bar.



5. Select **Next**.

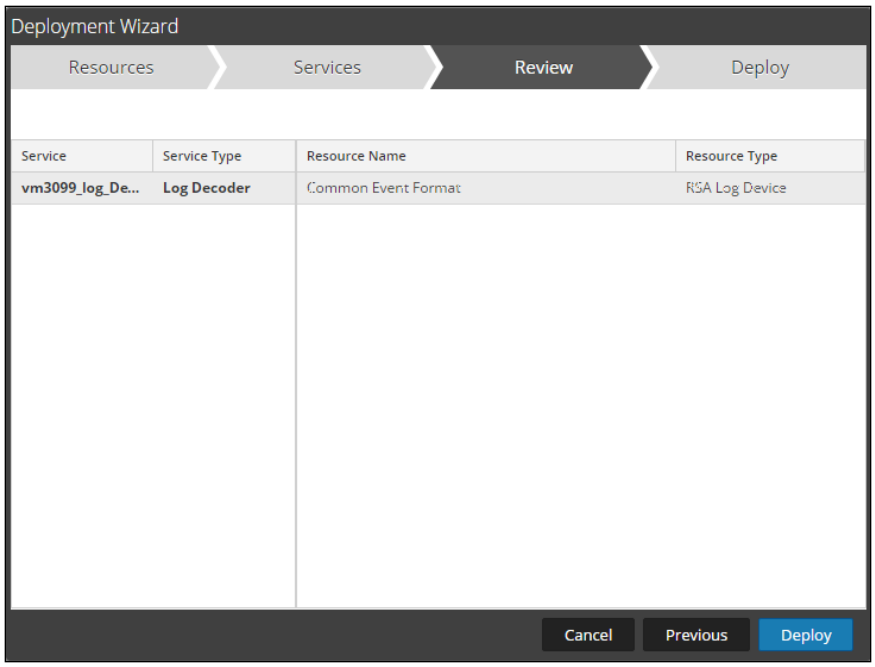


6. Select the **Log Decoder** and Select **Next**.

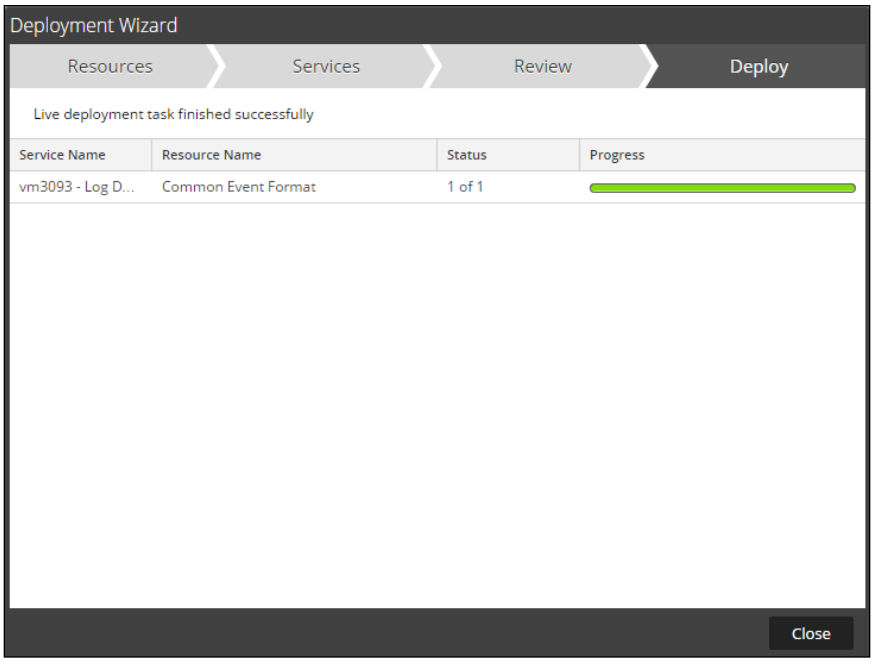


Important: In an environment with multiple Log Decoders, deploy the Common Event Format parser to each Log Decoder in your network.

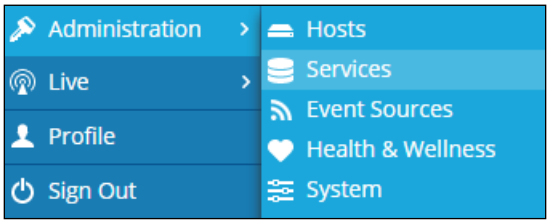
7. Select **Deploy**.



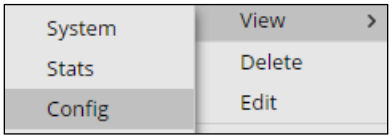
8. Select **Close**, to complete the deployment of the Common Event Format parser.



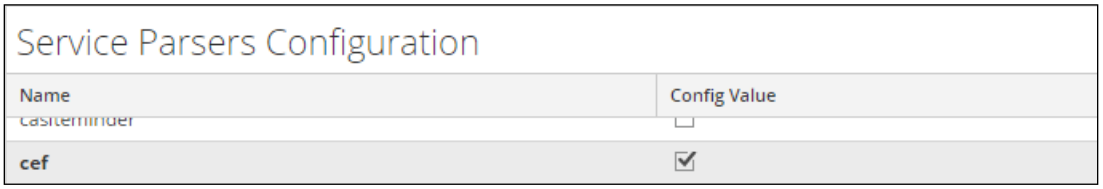
9. Ensure that the CEF parser is enabled on the Log Decoder(s) by selecting **Administration, Services** from the NetWitness Dashboard.



10. Locate the Log Decoder and click the gear to the right and select **View, Config**.



11. **Check** the box next to the **cef** parser within the Service Parsers Configuration and select **Apply**.



12. Restart the **Log Decoder services**.

Edit the cef.xml File to Collect Claroty Event Times

Important: The cef.xml file is overwritten by NetWitness Live during updates, it is important to maintain backups of the file in the event of a typing error or unforeseen event.

1. Using WinSCP or other application to access the RSA NetWitness Log Decoder, open a connection and locate the **/etc/netwitness/ng/envision/etc/devices/cef** folder. Backup cef.xml and edit the existing cef.xml file.
2. Locate the end of the <MESSAGE section and copy/paste the following lines below into the file after the **/>** of the preceding <MESSAGE and contents;

```
<MESSAGE
	id1="claroty_ctd"
	id2="claroty_ctd"
	eventcategory="1901000000"
	functions="&lt;@event_name:*HDR(event_description)&gt;@event_time_string:
*EVNTTIME($HDR,'%B %F
%Z',param_starttime)&gt;;&lt;@msg:*PARMVAL($MSG)&gt;&lt;@rt:*EVNTTIME($MSG,'%B
%F %W %Z',param_event_time)&gt;"
content="&lt;param_event_time&gt;&lt;msghold&gt;"/>
```

Edit the cef-custom.xml File to Support Custom Fields

Important: The cef-custom.xml file is not overwritten by NetWitness Live during updates, however it is important to maintain backups of the file in the event of a typing error or unforeseen event.

1. Using WinSCP or other application to access the RSA NetWitness Log Decoder, open a connection and locate the **/etc/netwitness/ng/envision/etc/devices/cef** folder. If the cef-custom.xml file does not exist create one. If the file exists create a backup cef-custom.xml and edit the file.
2. If this is a new cef-custom.xml file, copy the following into the file, otherwise copy only the required sections.

```
<!-- ** Please insert your custom keys or modifications below this line ** -->

<VendorProducts>
	<Vendor2Device vendor="Claroty" product="CTD" device="claroty_ctd"
	group="Analysis"/>
</VendorProducts>

<ExtensionKeys>

	<ExtensionKey cefName="rt" metaName="param_event_time">
		<device2meta device="claroty_ctd" metaName="receipt_time"/>
	</ExtensionKey>
	<ExtensionKey cefName="version" metaName="version"/>
	<ExtensionKey cefName="level" metaName="severity"/>
	<ExtensionKey cefName="cs1" metaName="cs_fld" >
		<device2meta device="trendmicrodsa" metaName="context"/>
		<device2meta device="bluecat" metaName="action" label="query"/>
		<device2meta device="websense" metaName="policyname"
		label="Policy"/>
		<device2meta device="mcafeewg" metaName="virusname" label="Virus
		Name"/>
		<device2meta device="bit9" metaName="checksum" label="File Hash"/>
		<device2meta device="mcafeereconnex" metaName="policyname"/>
```

```
        <device2meta device="claroty_ctd" metaName="site"/>
    </ExtensionKey>
    <ExtensionKey cefName="cs1Label" metaName="cs_fld" />

    <ExtensionKey cefName="cs2" metaName="cs_fld">
        <device2meta device="bit9" metaName="v_instafname"
        label="installerFilename"/>
        <device2meta device="claroty_ctd" metaName="Network" />
    </ExtensionKey>
    <ExtensionKey cefName="cs2Label" metaName="cs_fld"/>

    <ExtensionKey cefName="cs3" metaName="cs_fld">
        <device2meta device="websense" metaName="content_type"
        label="ContentType"/>
        <device2meta device="bit9" metaName="policyname"/>
        <device2meta device="mcafeereconnex" metaName="content_type"/>
        <device2meta device="claroty_ctd" metaName="ResolvedAs"/>
    </ExtensionKey>
    <ExtensionKey cefName="cs3Label" metaName="cs_fld"/>

    <ExtensionKey cefName="cs4" metaName="cs_fld">
        <device2meta device="mcafeewg" metaName="info" label="URL
        Categories"/>
        <device2meta device="claroty_ctd" metaName="SiteId"/>
    </ExtensionKey>
    <ExtensionKey cefName="cs4Label" metaName="cs_fld"/>

    <ExtensionKey cefName="smac" metaName="smacaddr"/>
    <ExtensionKey cefName="dmac" metaName="dmacaddr"/>
    <ExtensionKey cefName="externalId" metaName="hardware_id"/>

    </ExtensionKeys>

</DEVICEMESSAGES>
```

Edit the table-map-custom.xml File

Important: The Table-Map-Custom.xml file is not overwritten by NetWitness Live during updates, however it is important to maintain backups of the file in the event of a typing error or unforeseen event.

1. Using WinSCP or other application to access the RSA Netwitness Log Decoder, open a connection and locate the **/etc/netwitness/ng/envision/etc/** folder.
2. If one exists, backup the **table-map-custom.xml** and then edit the existing table-map-custom.xml file.
3. Copy and paste the entire section below into a new file or only the lines between the <mappings>…</mappings> if the table-map-custom.xml file exists;

```
<!-- Custom keys for Claroty -->
<mapping envisionName="receipt_time" nwName="receipt.time" format="Text"
flags="None"/>
<mapping envisionName="Network" nwName="Network" flags="None"/>
<mapping envisionName="ResolvedAs" nwName="ResolvedAs" flags="None"/>
<mapping envisionName="SiteId" nwName="SiteId" flags="None"/>
<mapping envisionName="hardware_id" nwName="hardware.id" flags="None"/>
```

Edit the index-concentrator-custom.xml File

Important: The index-custom-concentrator.xml file is overwritten by NetWitness Live during updates, it is important to maintain backups of the file in the event of a typing error or unforeseen event.

1. Using WinSCP or other application to access the RSA NetWitness Concentrator, open a connection and locate the /etc/netwitness/ng folder.
2. If one exists, backup the index-concentrator-custom.xml and then edit the index-concentrator-custom.xml file.
3. Add custom keys as needed to the file, for example:

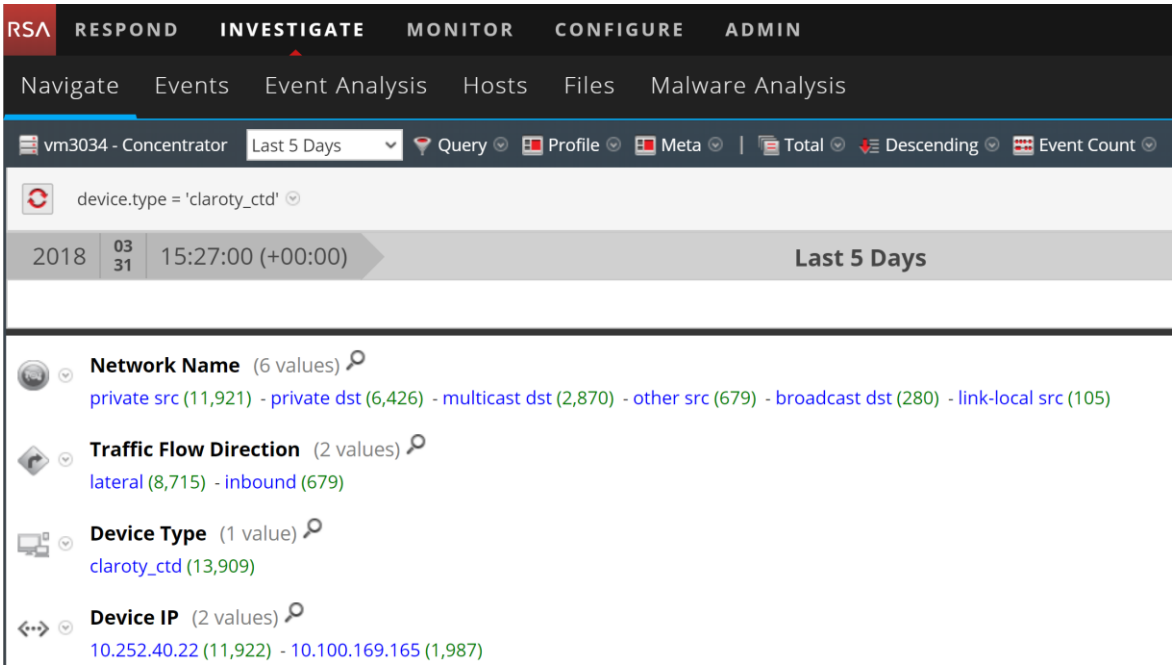
```
<!-- Add your custom index keys below this line -->

      <key description="Site" level="IndexValues" name="Site" format="Text"
valueMax="100000"/>
      <key description="Network" level="IndexValues" name="Network"
format="Text" valueMax="100000"/>
      <key description="ResolvedAs" level="IndexValues" name="ResolvedAs"
format="Text" valueMax="100000"/>
      <key description="SiteId" level="IndexValues" name="SiteId" format="Text"
valueMax="100000"/>

<!-- Add your custom index keys above this line -->
```

Claroty Collection Example within RSA NetWitness Investigator

Once the above changes have been made, events and alerts sent from Claroty Platform will show within the NetWitness Investigator:



In addition to alerts and events, Claroty Platform can also provide a custom feed with additional device information for further enrichment and visibility.

RSA NetWitness Custom Feed Configuration

Exporting the Claroty Assets Report

The Claroty Platform feed data is provided via a .csv file. That is exported as an **Assets Report** within the Claroty Platform UI. To do this, perform the following steps:

1. In the **Assets View** page, click the Export icon:
2. Specify a custom report name in the **name** field.
3. Select the report format as **CSV**.
4. Click **Download**.

Note that if the report contains a header line, for example:

ICS Ranger Assets Report, Produced by ICS Ranger on Monday, Sep 25, 2017, 19:32 UTC+03:00

Remove this line before importing into RSA NetWitness.

RSA NetWitness Custom Feed Configuration

Depending on your deployment and if you have elected to add an RSA SA Log Decoder and/or Packet Decoder, follow the steps below for your integration. The column headers of the .csv file need to be mapped to existing RSA NetWitness keys, or where existing keys are not available, you can create custom keys using the instructions found here:

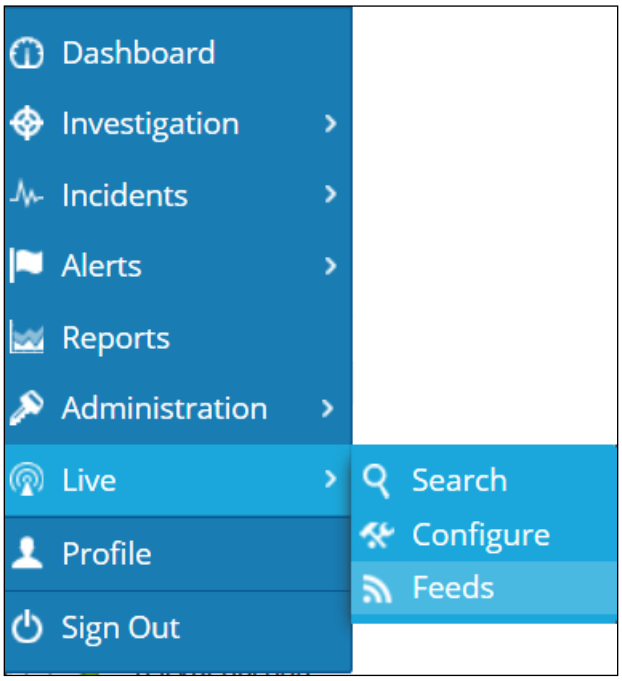
https://community.rsa.com/docs/DOC-78049

Ensure that any custom keys have been added, and any relevant services have been restarted, before configuring the custom feed as described below.

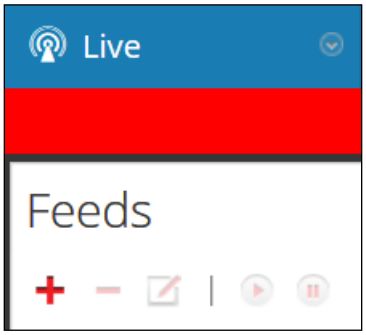
Log Decoder Configuration

RSA NetWitness Feed Configuration

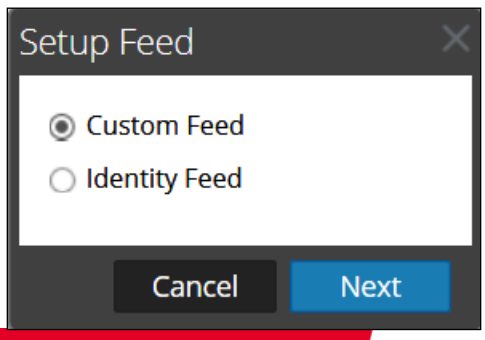
1. From the RSA SA Dashboard Select **Live**, **Feeds**.



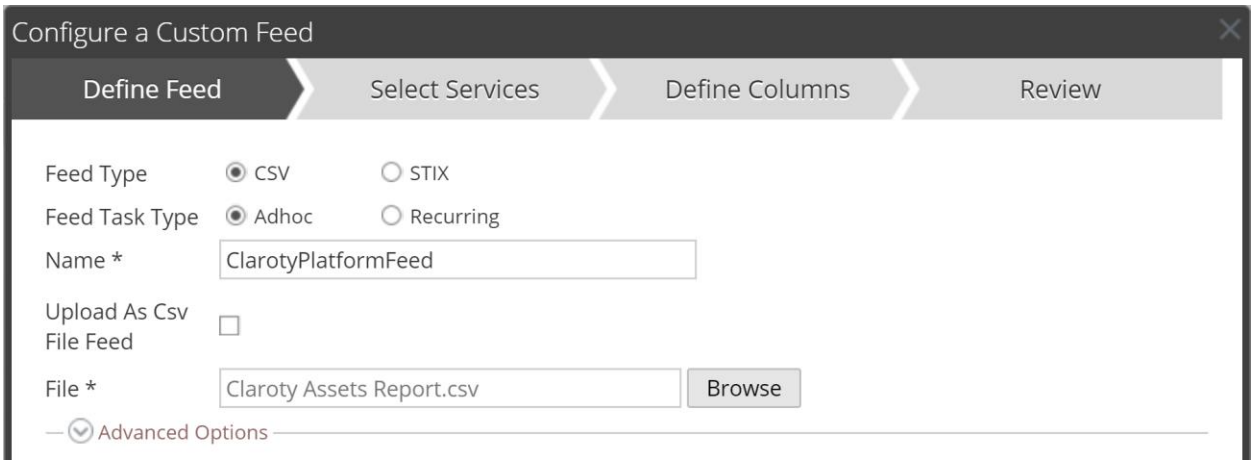
2. Select the + in the Live Feeds Window to setup the feed.



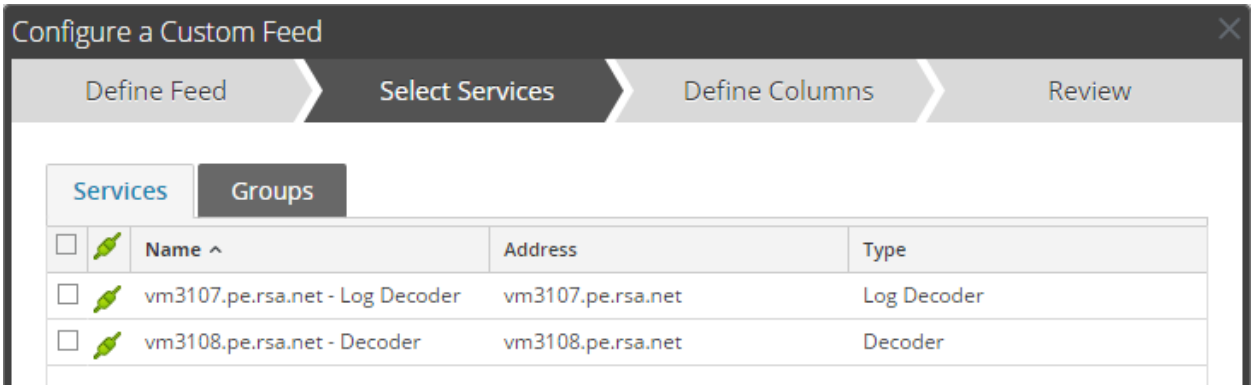
3. Select the **Custom Feed** radio button within the Setup Feed pop-up window and select **Next**.



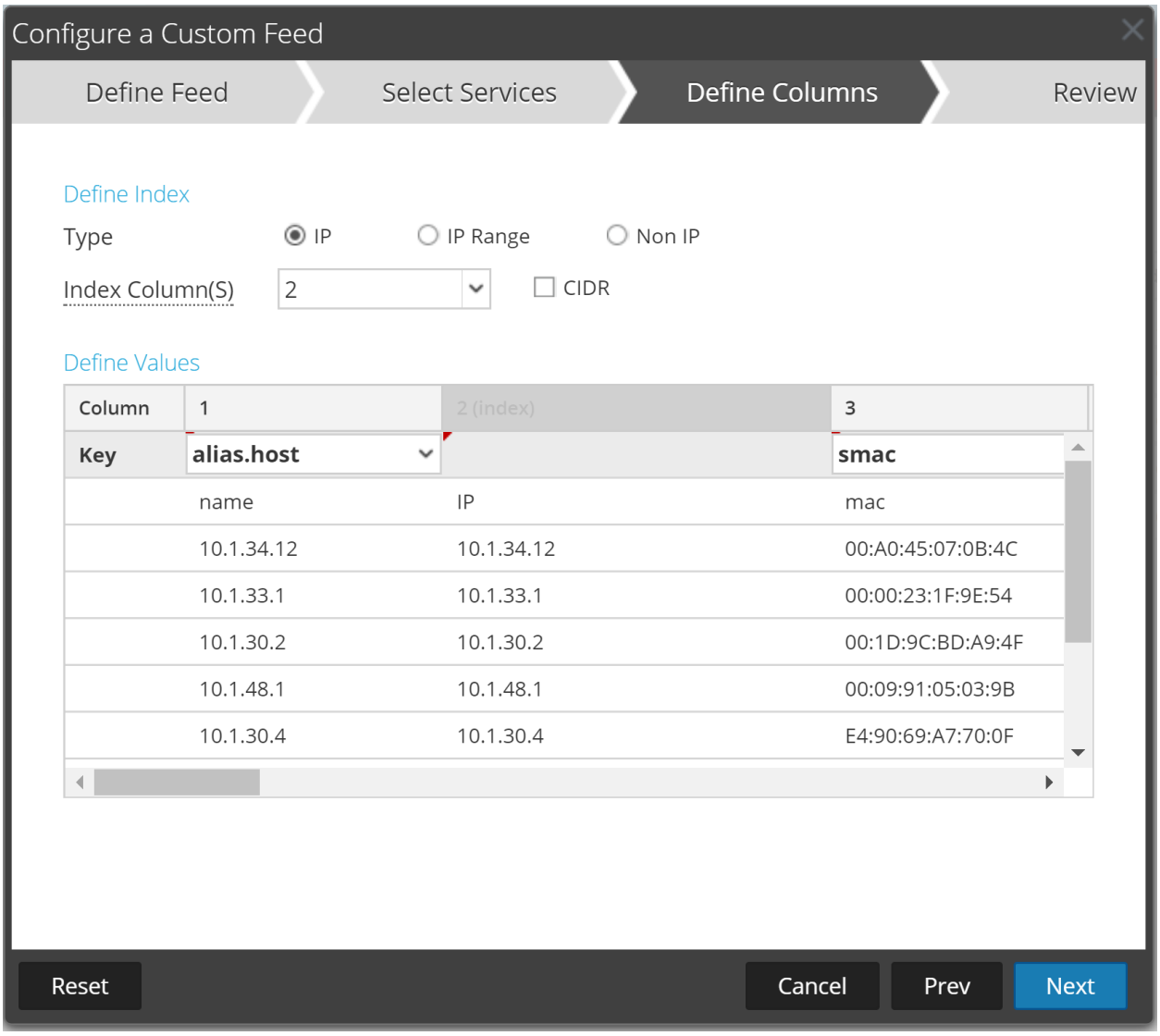
4. Select **Adhoc** if you are uploading the file once or the **Recurring** radio button if you plan to automate the feed.



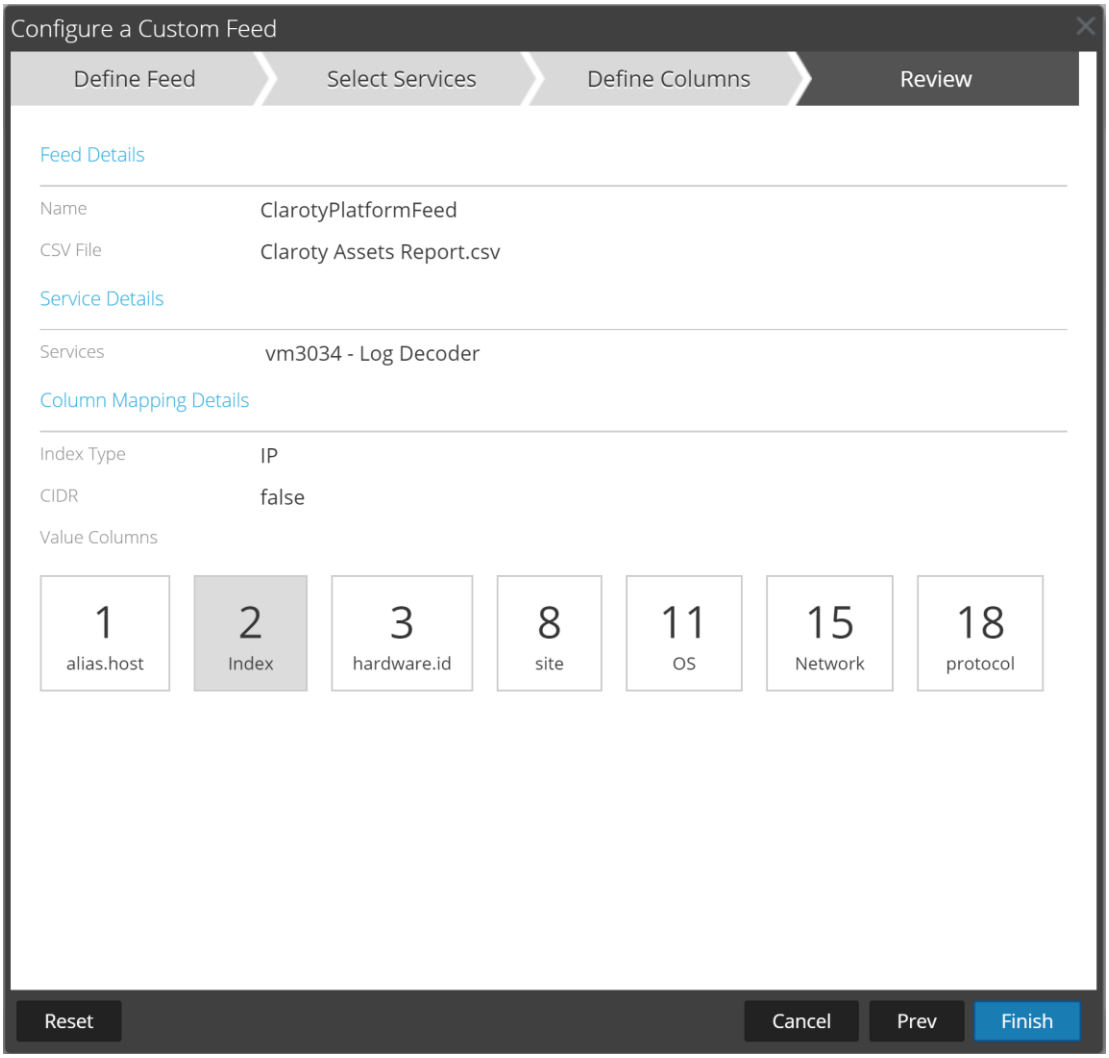
5. Select the RSA Log Decoder Service checkbox and select **Next**.



6. Define the **Type** as **IP** and **Index Column 2** (IP Address Field). Set the header of each column as needed. If the custom keys you have added are not available from the drop-down list, type them in. Select **Next** to continue.


Configure a Custom Feed
Define Feed
Select Services
Define Columns
Review
Define Index
Type
IP
IP Range
Non IP
Index Column(S)
2
CIDR
Define Values
Column
1
2 (index)
3
Key
alias.host
smac
name
IP
mac
10.1.34.12
10.1.34.12
00:A0:45:07:0B:4C
10.1.33.1
10.1.33.1
00:00:23:1F:9E:54
10.1.30.2
10.1.30.2
00:1D:9C:BD:A9:4F
10.1.48.1
10.1.48.1
00:09:91:05:03:9B
10.1.30.4
10.1.30.4
E4:90:69:A7:70:0F
Reset
Cancel
Prev
Next

7. Select **Finish**, to complete the setup of the Feed Integration.



Initially the status will appear as **Waiting** and the Progress will be **yellow** until RSA SA completes the transfer of the Feed. Once completed the Status will display **Completed** and the Progress will be **green**.

8. Once the feed has completed, you should see additional metadata provided by Claroty Platform when performing an investigation if there is a match on an IP address contained in the feed file:

Certification Checklist for RSA NetWitness

Date Tested: April 27th, 2018

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA NetWitness	11.1	Virtual Appliance
Claroty Platform	2.1	

NetWitness Test Case	Result
Device Administration	
Partner's device name appears in Device Parsers Configuration	✓
Device can be enabled from Device Parsers Configuration	✓
Device can be disabled from Device Parsers Configuration	✓
Device can be removed from Device Parsers Configuration	✓
Investigation	
Device name displays properly from Device Type	✓
Displays Meta Data properly within Investigator	✓

✓ = Pass ✗ = Fail N/A = Non-Available Function

Appendix

NetWitness 10.6 Support

This appendix contains information on integrating Claroty 2.0 with NetWitness 10.6, for historical reference only. The necessary edits to key files are listed below.

cef.xml

```
<MESSAGE
       level="4"
       parse="1"
       parsedefvalue="1"
       tableid="74"
       id1="claroty_ranger"
       id2="claroty_ranger"
       eventcategory="1612000000"

       content="&lt;@event_name:*HDR(event_description)&gt;&lt;@msg:*PARMVAL($MSG)&gt;
&lt;@endtime:*EVNTTIME($MSG,'%B %D %W
%Z',param_event_time)&gt;&lt;msghold&gt;&lt;param_event_time&gt;" />
```

cef-custom.xml

```
<VendorProducts>
<Vendor2Device vendor="Claroty" product="Ranger" device="claroty_ranger"
group="Analysis"/>
</VendorProducts>

           <ExtensionKeys>
              <ExtensionKey cefName="Version" metaName="version"/>
              <ExtensionKey cefName="level" metaName="severity"/>

              <ExtensionKey cefName="cs1" metaName="cs_fld" >
                     <device2meta device="trendmicrodsa" metaName="context"/>
                     <device2meta device="bluecat" metaName="action" label="query"/>
                     <device2meta device="websense" metaName="policyname"
label="Policy"/>
                     <device2meta device="mcafeewg" metaName="virusname" label="Virus
Name"/>
                     <device2meta device="bit9" metaName="checksum" label="File Hash"/>
                     <device2meta device="mcafeereconnex" metaName="policyname"/>
                     <device2meta device="claroty_ranger" metaName="Site"/>
              </ExtensionKey>
              <ExtensionKey cefName="cs1Label" metaName="cs_fld" />

              <ExtensionKey cefName="cs2" metaName="cs_fld">
                     <device2meta device="bit9" metaName="v_instafname"
label="installerFilename"/>
                     <device2meta device="claroty_ranger" metaName="Network" />
              </ExtensionKey>
              <ExtensionKey cefName="cs2Label" metaName="cs_fld"/>

              <ExtensionKey cefName="cs3" metaName="cs_fld">
                     <device2meta device="websense" metaName="content_type"
label="ContentType"/>
                     <device2meta device="bit9" metaName="policyname"/>
                     <device2meta device="mcafeereconnex" metaName="content_type"/>
                     <device2meta device="claroty_ranger" metaName="ResolvedAs"/>
              </ExtensionKey>
              <ExtensionKey cefName="cs3Label" metaName="cs_fld"/>

              <ExtensionKey cefName="cs4" metaName="cs_fld">
```

```
                <device2meta device="mcafeewg" metaName="info" label="URL
Categories"/>
                <device2meta device="claroty_ranger" metaName="SiteId"/>
            </ExtensionKey>
            <ExtensionKey cefName="cs4Label" metaName="cs_fld"/>

            <ExtensionKey cefName="smac" metaName="smacaddr"/>

            <ExtensionKey cefName="dmac" metaName="dmacaddr"/>

            <ExtensionKey cefName="externalId" metaName="hardware_id"/>


            </ExtensionKeys>
```

table-map-custom.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
# attributes:
#      envisionName: The name of the column in the universal table
#      nwName:                    The name of the NetWitness meta field
#      format:                    Optional. The language key data type. See
LanguageManager. Defaults to "Text".
#      flags:              Optional. One of None|File|Duration|Transient.
Defaults to "None".
#      failureKey:         Optional. The name of the NW key to write data if
conversion fails. Defaults to system generated "parse.error" meta.
#      nullTokens:         Optional. The list of "null" tokens. Pipe separated.
Default is no null tokens.
-->
<mappings>

      <mapping envisionName="starttime" nwName="start" flags="None"
format="TimeT" envisionDisplayName="StartTime"/>
      <mapping envisionName="endtime" nwName="endtime" flags="None"
format="TimeT" envisionDisplayName="EndTime,rt,end"/>
      <mapping envisionName="version" nwName="version" flags="None"/>
      <mapping envisionName="severity" nwName="severity" flags="None"
envisionDisplayName="Severity|SeverityLevel"/>

      <mapping envisionName="Site" nwName="Site" flags="None"
envisionDisplayName="Site"/>
      <mapping envisionName="msg" nwName="msg" flags="None" format="Text"
envisionDisplayName="Message"/>

      <mapping envisionName="Network" nwName="Network" flags="None"/>
      <mapping envisionName="ResolvedAs" nwName="ResolvedAs" flags="None"/>
      <mapping envisionName="SiteId" nwName="SiteId" flags="None"/>
      <mapping envisionName="hardware_id" nwName="hardware.id" flags="None"/>

      <mapping envisionName="smacaddr" nwName="eth.src" flags="None"
format="MAC" envisionDisplayName="SourceMacAddress"
nullTokens="Unknown|Irresolvable"/>
      <mapping envisionName="dmacaddr" nwName="eth.dst" flags="None"
format="MAC" envisionDisplayName="DestMacAddress|DestinationMacAddress"/>

</mappings>
```

index-concentrator-custom.xml

```
<!-- Add your custom index keys below this line -->

       <key description="Site" level="IndexValues" name="Site" format="Text"
valueMax="100000"/>
       <key description="Network" level="IndexValues" name="Network"
format="Text" valueMax="100000"/>
       <key description="ResolvedAs" level="IndexValues" name="ResolvedAs"
format="Text" valueMax="100000"/>
       <key description="SiteId" level="IndexValues" name="SiteId" format="Text"
valueMax="100000"/>

<!-- Add your custom index keys above this line -->
```