NetWitness® Platform XDR

Version 12.1.0.0

Log Parser Customization Guide

NETWITNESS
Platform XDR

Contact Information

NetWitness Community at https://community.netwitness.com contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

RSA and other trademarks are trademarks of RSA Security LLC or its affiliates ("RSA"). For a list of RSA trademarks, go to https://www.rsa.com/en-us/company/rsa-trademarks. Other trademarks are trademarks of their respective owners.

License Agreement

This software and the associated documentation are proprietary and confidential to RSA Security LLC or its affiliates are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by RSA.

Third-Party Licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed on the product documentation page on NetWitness Community. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any RSA Security LLC or its affiliates ("RSA") software described in this publication requires an applicable software license.

RSA believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." RSA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

October, 2022

Contents

Log Parser Customization

Note: The JSON Mapping information in this guide applies to NetWitness Version 11.5 and later.

You use the Log Parser Rules view (available from the (Configure) view) to customize rules and to map meta for your log parsers.

The *default log parser* parses logs that do not match any installed log parsers. The information contained in such a log is processed against the default log parser's rules, and metadata is then extracted by those rules and is available for Enrichment, Investigation, Reporting, and Alerting. This provides immediate visibility into logs from custom or unsupported sources.

You can also add or extend a log parser. For example, you may need to parse certain fields differently than in the manner provided by the log parser for a particular event source. You can add rules that change the way meta information is extracted from the logs for the event source. And you can then map the extracted information to NetWitness meta keys.

Finally, you can view and test sample log messages and rules for your log parsers, including the default log parser.

To access this tab, go to **(Configure) > Log Parser Rules**. For more details, see Parser Rules Tab.

Dynamic Rules

The **Dynamic Rules** entry for a log parser displays the following information:

- The default log parser that parses logs that are not associated with a particular log parser
- Native XML-defined device parsers that have been extended with dynamic log parser rules, and
- User-created custom device parsers used to parse unsupported custom event sources

The dynamic parse rules are used to parse arbitrary values triggered by a literal token on the left hand side of the value in unstructured logs. Currently, this is used to parse name-value pairs.

Following are some examples (the token is in bold red):

```
src: 1.2.3.4
src = 1.2.3.4
src=1.2.3.4
Source address is 1.2.3.4
```

JSON Mapping

This allows you to parse JSON nodes from Logstash as well as Log Collection Plugins by mapping JSON nodes to an appropriate meta on which the value should be saved.

The **JSON Mappings** for a log parser displays the following information:

- Sample JSON Message - This allows you paste the log messages.
- The list of JSON Mappings: these are the names that represent the meta information.
- Details of each mapping: for each mapping, the display name, path, NetWitness meta key, and a text description.

The JSON mapping functionality is for strictly paring structured JSON logs, and mapping values from the log to meta or fine parsing. The parsing is not applied to arbitrary logs; only logs where we know the exact structure of the data.

For example:

```
{ "event": { "source": { "address": "1.2.3.4", "port":8080 } } }
```

NetWitness knows the structure of logs when it knows the event source type, or when you add specific JSON mappings.

Add or Delete a Log Parser

Note: The information in this topic applies to NetWitness Version 11.2 and later.

For version 11.2, NetWitness has added the ability to add log parsers through the UI. You can also delete log parsers, as long as they have never been deployed to a Decoder. You can create a new log parser definition from scratch, or extend an existing one.

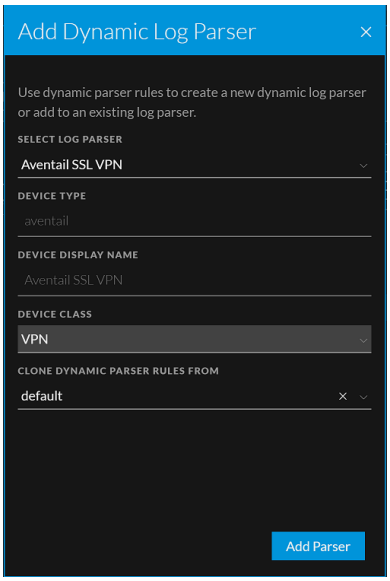
You can add a log parser to extend the functionality for an existing parser. For example, if you have some unknown messages for the Cisco Pix parser, you could add rules to match your unknowns.

IMPORTANT: If you are adding a new log parser, for example when onboarding an event source, you must map the event source IP to the new log parser in order for messages to be parsed. For details, see "Acknowledging and Mapping Event Sources" in the *Event Source Management User Guide*.

Add a Log Parser

1. In the NetWitness UI, navigate to **(Configure) > Log Parser Rules**.
2. From the **Log Parsers** pane, click **Add Parser**.

   The Add Dynamic Log Parser dialog box is displayed.



3. Fill in details for this dialog box. For details, see Add Dynamic Log Parser Parameters below.
4. Click **Save** to save the new log parser.

   This updates the definition file in the file system. It *does not* deploy the changes.
5. To deploy your changes to all of your Decoders, click **Deploy**.

Delete a Log Parser

You can use the UI to delete a log parser.

To delete a log parser:

1. In the NetWitness UI, navigate to **(Configure) > Log Parser Rules**.
2. From the **Log Parsers** pane, select a log parser, then click **Delete**.

   Delete Parser dialog box is displayed.
3. Click **Delete Parser** to remove the log parser from the system.

Note: If you have encounter any issues when you attempt to delete a parser, see the Troubleshooting section, Delete a Log Parser Manually.

Add Dynamic Log Parser Parameters

When you are adding a log parser, the following parameters are available.

Field	Details
SELECT LOG PARSER	Select NEW, or choose an existing log parser. By choosing an existing log parser, you can add rules to that parser, essentially extending its parsing capabilities. **Note:** If you select an existing log parser, the remaining fields are auto-filled based on the values for selected log parser.
DEVICE TYPE	Enter a string to define the device type. The name must be between 3 and 30 alphanumeric characters (including underscores), and must not match the name of any existing log parsers.
DEVICE DISPLAY NAME	Enter the display name for the log parser. **Note:** The display name must be 64 characters or fewer, and must not match the name of any other device display name.
DEVICE CLASS	Select a device class.
CLONE DYNAMIC PARSER RULES FROM	Leave blank to start with no rules, or select one of the existing log parsers to clone its rules.

Import or Export a Log Parser

To import the log parser:

1. In the NetWitness UI, navigate to **(Configure) > Log Parser Rules**.
2. From the **Log Parsers** panel, click **Import**.

3. Select a log parser to import.
   The log parser is imported successfully. This will import the selected log parser rules along with the JSON mapping (if any).

To export the log parser:

1. In the NetWitness UI, navigate to  **(Configure) > Log Parser Rules**.
2. Select the log parser to export.
3. In the **Log Parsers** panel, click **Export**.
   The selected parser is exported successfully in a .json format. This will export the selected log parser rules along with the JSON mapping (if any).

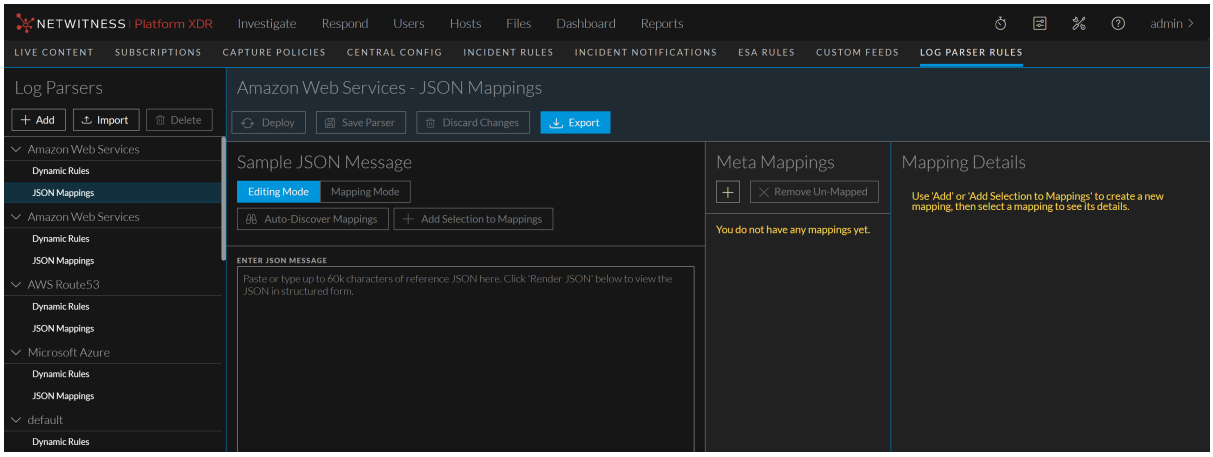
JSON Mappings

JSON mappings is applicable from version NetWitness Platform 11.5 and Later.

View JSON Mappings

1. In the NetWitness UI, go to **(Configure) > Log Parser Rules**.
2. From the Log Parsers pane, select a parser, then click JSON Mappings.

   The JSON Mappings and Mapping Details are shown for the parser you selected.



The Mapping Details pane displays the following information.

Field	Details
DISPLAY NAME	This name corresponds to the name displayed in the JSON Mappings panel.
PATH	The path to where the values for this portion of the log are stored.
DESCRIPTION	Optionally, you can enter a text description for this mapping.
META	Select a meta key to which this value from the log is mapped. Select a value from the drop-down menu. Optional if you choose a Value Format.

Field	Details
VALUE FORMAT	Choose a value format parser onto which to pass this JSON value. You can either select an existing meta or **Custom Regex Type**. If you select custom regex type, you must define the regex and capture to *fine* parse the value in the meta. Optional if you choose a Meta.
CUSTOM REGEX TYPE	Select Custom Regex Type from the Value Format drop-down, which allows you to add new custom regex type.
REGEX PATTERN	Specify a regex to identify different pieces of data contained within a JSON node value.
FIRST CAPTURE	Select a meta key that should be captured first based on the value defined in the Regex pattern.
ADD A CAPTURE	New capture field is added. By default, it is loaded with meta keys in the drop-down. You can add maximum of 20 captures and this option will be disabled once it reaches maximum.

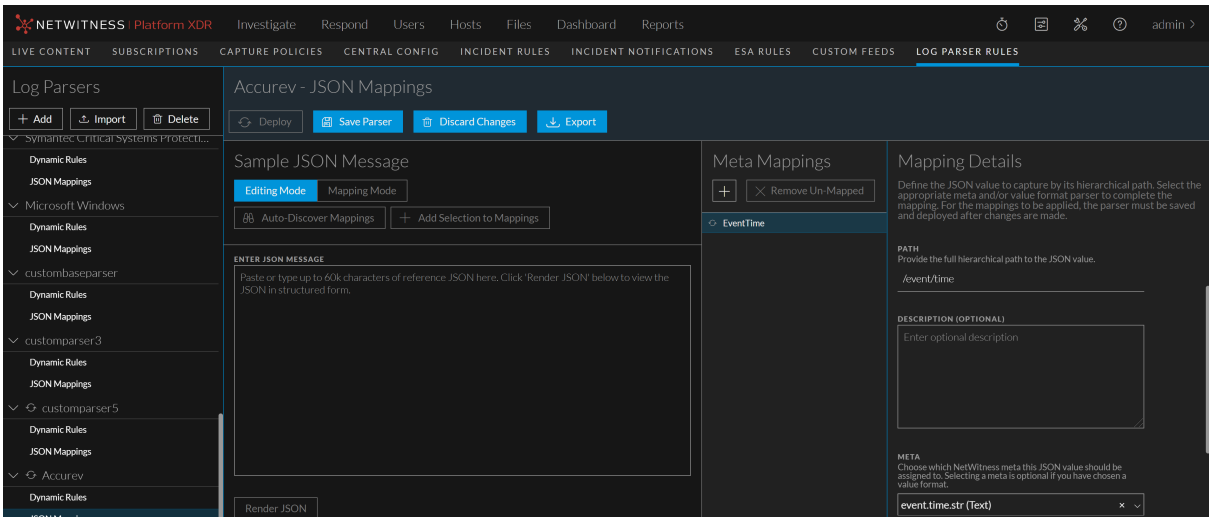
Note: You need to select a meta or enter a Value Format, but you do not need to fill in values for both settings.

Add a JSON Mapping

After you add a parser, as described in Add a Log Parser, you can then add JSON mappings.

1. Follow the procedure to add a parser.
2. Select the **JSON Mappings** entry for the newly-added parser.

   The following screen shows an example where an Accurev is added:



3. Click **Add New** to begin adding a mapping.
4. Enter values for display name, path, meta or value format (or both), and (optionally) a description.
5. Click **Save Parser** to save your new mapping.

Auto Discover JSON Mappings

Beginning with NetWitness version 11.5.1, you can automatically create the mappings without the need to manually enter the name and path of the mapping.

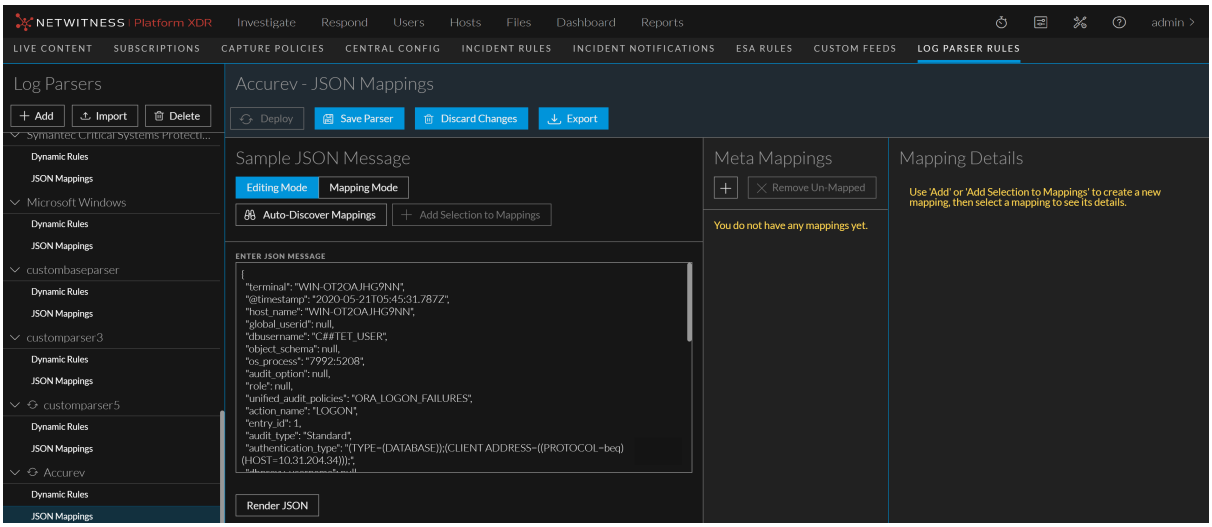
As an example, we are using the following JSON log:

```
{"terminal":"WIN-OT2OAJHG9NN","@timestamp":"2020-05-21T05:45:31.787Z","host_
name":"WIN-OT2OAJHG9NN","global_userid":null,"dbusername":"C##TET_
USER","object_schema":null,"os_process":"7992:5208","audit_
option":null,"role":null,"unified_audit_policies":"ORA_LOGON_
FAILURES","action_name":"LOGON","entry_id":1,"audit_
type":"Standard","authentication_type":"(TYPE=(DATABASE));(CLIENT ADDRESS=
((PROTOCOL=beq)(HOST=10.31.204.34)));","dbproxy_username":null,"external_
userid":null,"@version":"1","new_schema":null,"new_name":null,"statement_
id":1,"proxy_sessionid":0,"os_username":"WIN-
OT2OAJHG9NN\\Administrator","system_privilege":null,"sql_
binds":null,"timestamp":"2020-05-21 10:22:12","client_program_
name":"sqlplus.exe","sessionid":4125005309,"userhost":"WORKGROUP\\WIN-
OT2OAJHG9NN","rman_device_type":null,"object_name":null,"event_timestamp_
utc":"2020-05-20T23:22:12.452Z","system_privilege_used":null,"return_
code":1017,"version":"19.0.0.0.0","instance_name":"orcl","sql_
text":null,"target_user":null,"fga_policy_name":null,"rman_object_
type":null,"dbid":1566661212,"rman_operation":null}
```

To auto-discover JSON mappings:

1. Select the **JSON Mappings** entry for the appropriate parser.
2. Paste JSON formatted log message in to the **Sample JSON message** text box, and click **Render JSON**.

The screen looks like this:



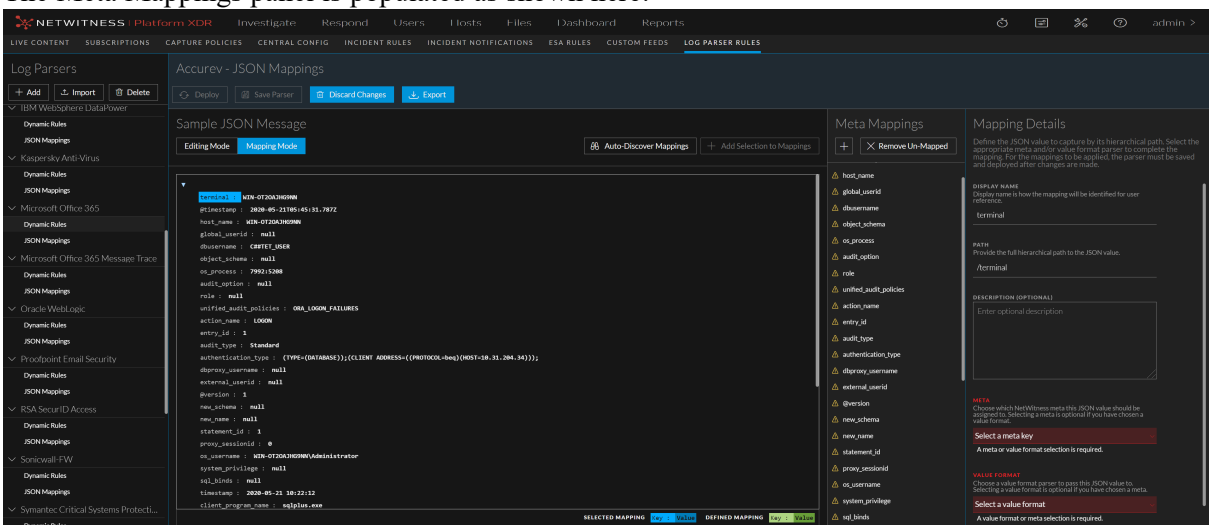
Rendering JSON in Editing Mode allows you to view and edit (if needed) the logs in a pretty format. Additionally, if your text is not valid JSON, the text field is displayed in red.

3. Click **Mapping Mode**, to view the JSON in a collapsable tree format which also highlights the mapping.

Note: In **Mapping mode**, you cannot edit the sample Log.

4. Click **Auto-Discover Mappings** to discover the JSON nodes and create mappings.

The Meta Mappings panel is populated as shown here:



5. After you auto-discover, note that all the mappings are invalid (preceded by this icon: ⚠). You cannot save your changes until all the mappings are valid (mapping is preceded by this icon: 🔄) or removed.

Note:
The meta value is highlighted in blue if it matches the selected mapping including the regex (if used).
The meta value is highlighted in green if it matches any other mappings that are not selected.

6. To make a mapping valid, you need to select a **Meta Key** or **Value Format** for all the mappings that you want to parse and save.

Note: In the **Value Format** you can either select an existing meta or **Custom Regex Type**.

7. If you want to *fine* parse the value in the meta in the **Value Format** drop-down, select **Custom Regex Type** and enter the regex in the **Regex Pattern** field. For example,`\s*(\b(?:[0-9]{1,3}\.){3}[0-9]{1,3}\b):?(\d*).`

Note: The custom regex will be added to the database, when you **Save Parser**.

8. For a value in the Regex, you can select a meta key and store the capture as below:
   - **First capture**: The first portion of the string, up to the period character is stored to the meta key. For example, `ip.src`
   - To add more capture, click **Add Capture** and select the meta key.
     For more information on Regex, see Regex Values.
9. If there are mappings that you do not want to save, you can select the mapping and click **Delete**. Alternatively, after you complete all of the mappings that you want to keep, you can click **Remove Unmapped** to remove all mappings that you have not yet validated.
10. After you have either completed or removed all of your mappings, click **Save Parser** to save your new mappings. Note that the icon preceding each mapping is removed.

Deploy JSON Parser

You need to deploy a JSON parser so that logs coming in to any decoder are parsed appropriately and meta is generated and stored correctly.

To deploy a parser, select it from the list and click **Deploy**. The parser, its dynamic rules, and its mappings are sent to all Log Decoders.

Note: A JSON parser must have at least one rule or mapping to enable deployment.

Add or Delete a Log Parser Rule

Note: The information in this topic applies to NetWitness Version 11.2 and later.

For version 11.2, NetWitness has added the ability to create custom rules for log parsers. You can create rules to change how meta values are parsed for a particular log parser. Prior to version 11.2, you could only view the out-of-the-box log parser rules.

About Log Parser Rules

Parsers are described within their XML files. Each log parser has an XML file that contains rules on how to parse messages for that parser. The out-of-the-box rules are contained within these XML files. For details, see the Log Parser Customization topic in the Link space for NetWitness Content.

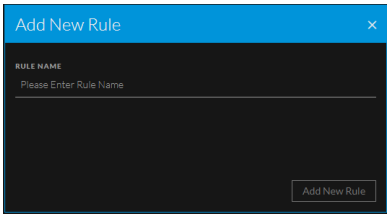
Custom Log Parser Rules

When you create a new log parser rule, it is saved to another XML definition file for the parser. These files are known as token files. This is important, since the out-of-the-box rules are overwritten if you update the parser through Live, but any custom log parser rules are not overwritten, since Live does not update the token files for log parsers.

To create a custom log parser rule:

1. In the NetWitness UI, navigate to (Configure) > **Log Parser Rules**.
2. From the **Log Parsers** pane, select a log parser, then **Dynamic Rules**.
3. From the **Rules** pane, click **Add Rule**.

   The Add Rules dialog box is displayed.



IMPORTANT: If you click outside of the Add Rule dialog box before you save your rule, your changes will be lost.

4. Enter a name for the new rule, and click **Add New Rule**.
5. Add at least one meta key and a value to match, in order to create a valid rule.
6. Click **Save** to save your new rule.

   This updates the definition file in the file system. It *does not* deploy the changes.
7. To deploy your changes to all of your Decoders, click **Deploy**.

Guidelines for Custom Rules

When you are creating a custom rule, keep in mind the following:

- For the list of tokens that match strings from the log file, very short tokens are not useful. For example, a one- or two-character string can match more items than desired.
- Remember to add the delimiter (especially if it is a space) as part of the token. For example "domain=" or "email ".
- When constructing regular expressions, the more complexity you add, the more performance overhead added to the system to compare against the rule.
- To see examples of good tokens and regular expressions, examine the rules that are provided for the default log parser.

Default Log Parser and Log Parser Rules

Note: The information in this topic applies to NetWitness Version 11.1 and later.

This tab displays information about pattern matching and rules for the parsers in your system. The features on this tab apply to all log parsers, including the Default Log Parser.

Default Log Parser

The NetWitness default log parser is used to parse logs coming from the Log Decoder that do not match any of the configured log parsers. This default parser parses these logs by using a default set of rules and tokens.

You can view the default log parser and its details by going to **Admin > Event Sources > Log Parser Rules** and selecting **default** from the Log Parsers panel.

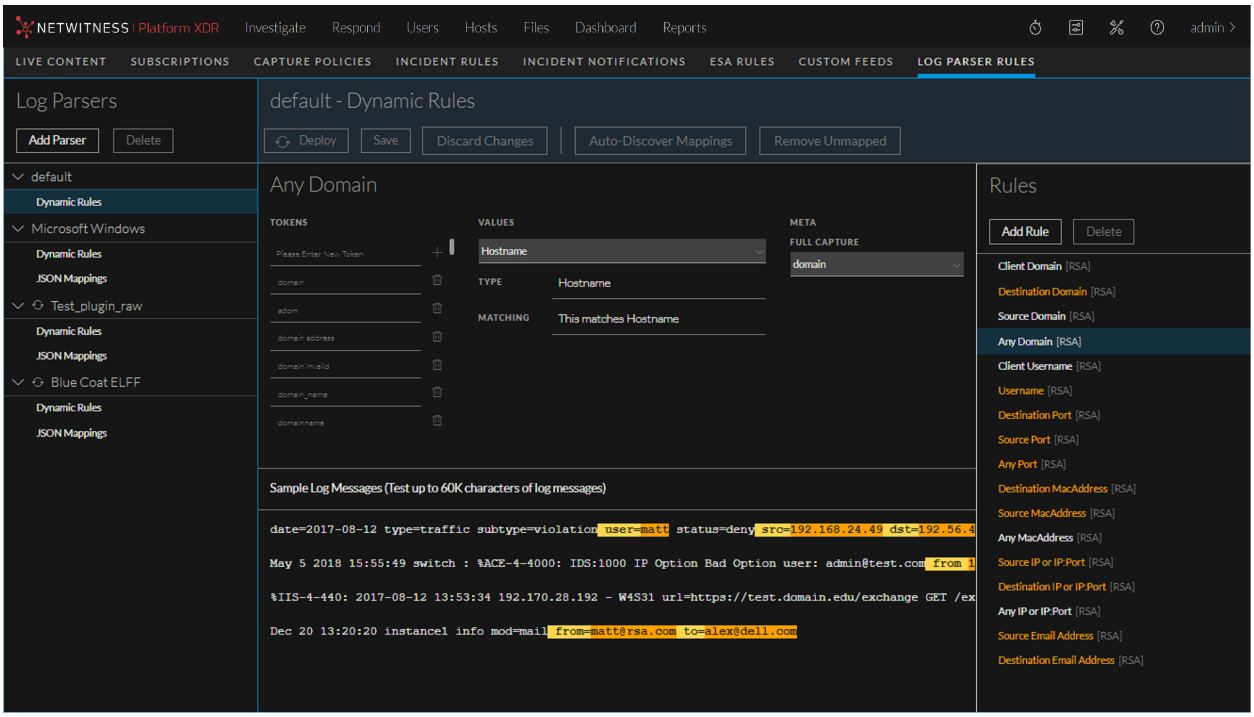
Note: If you do not see the default log parser and its rules, you might need to go to Live and deploy the Content to your log decoders. Additionally, you must have at least one Log Decoder at version 11.2 to view the default log parser.

You can view the default log parser and its details, depending on your version:

- For NetWitness version 11.1, go to **Admin > Event Sources > Log Parser Rules**, then select **default** from the Log Parsers panel.
- For NetWitness version 11.2 and later, go to **Configure > Log Parser Rules**, then select **default** from the Log Parsers panel.

Note: The list of log parsers is based on the first Log Decoder that is installed or registered by the Orchestration Server. If you have more than one Log Decoder, this tab only lists log parsers that have been configured on the first one.

This is a view of the Log Parser Rules tab, showing the **Default Log Parser** and **Any Domain** rule selected:



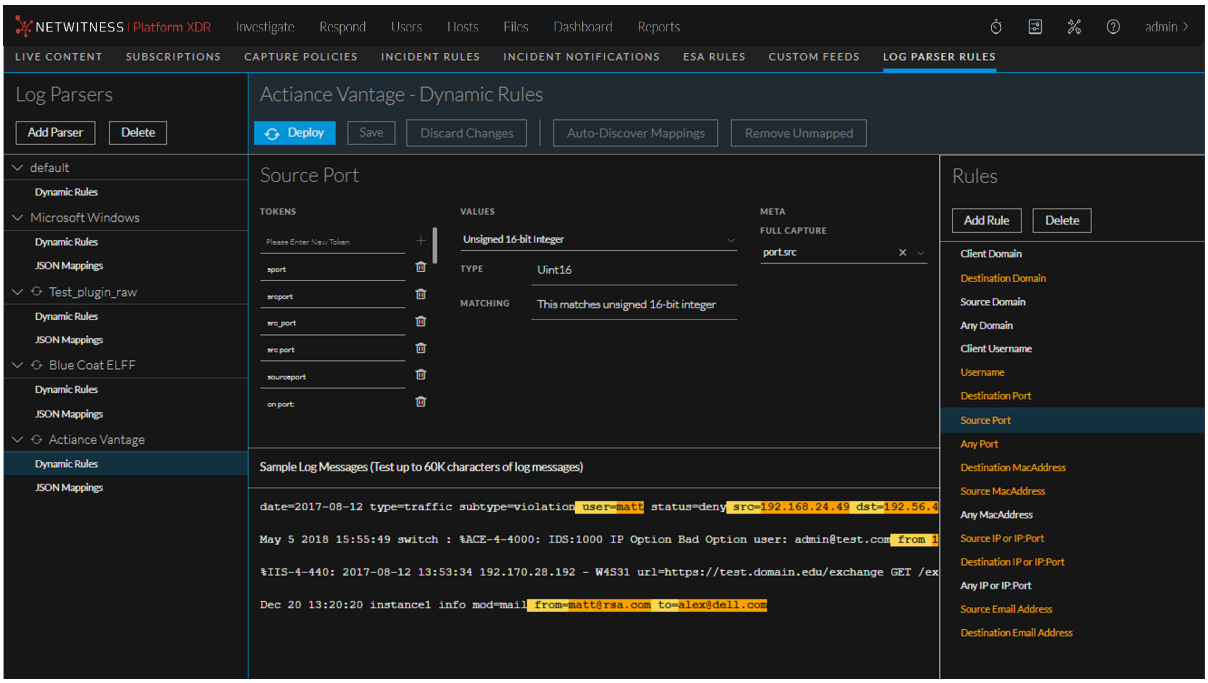
The Parser Rules Tab topic describes the items available for the Log Parsers tab.

Highlight Matching Patterns

You can paste logs into the Log Messages text box, and the system highlights the matching literals and patterns for the rules for the selected event source type. Use this feature to confirm that the parser is behaving as expected.

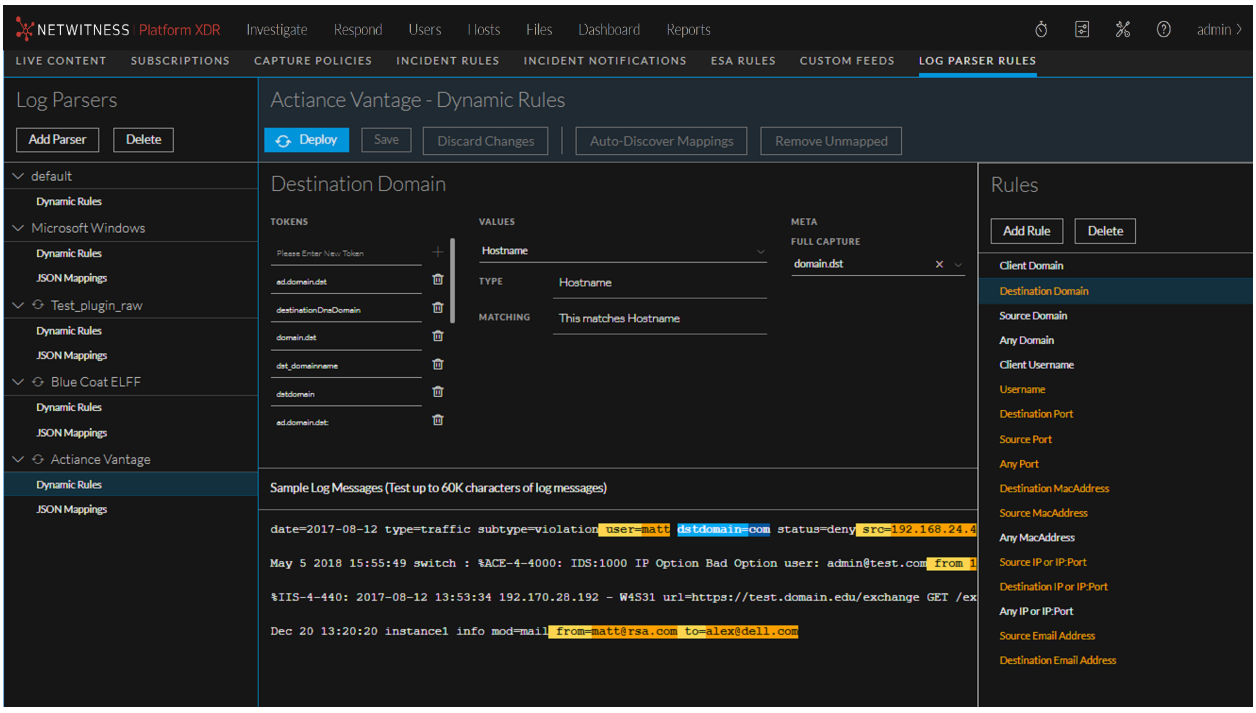
1. In the NetWitness UI, navigate as follows, depending on your version:
   - For NetWitness version 11.1, go to **Admin > Event Sources > Log Parser Rules**.
   - For NetWitness version 11.2 and later, go to  **(Configure) > Log Parser Rules**.
2. From the **Log Parsers** pane, select the **Dynamic Rules** entry for a log parser.
3. From the **Rules** pane, select a rule.

   For example, this screen shows the **Source Port** rule for the **Actiance Vantage** log parser:



4. Add text or paste in a sample log message.

Strings that match tokens for the selected rule are highlighted in blue. Strings that match other rules for the parser (and the rules themselves) are highlighted in orange.



For example, in the previous screen, note:

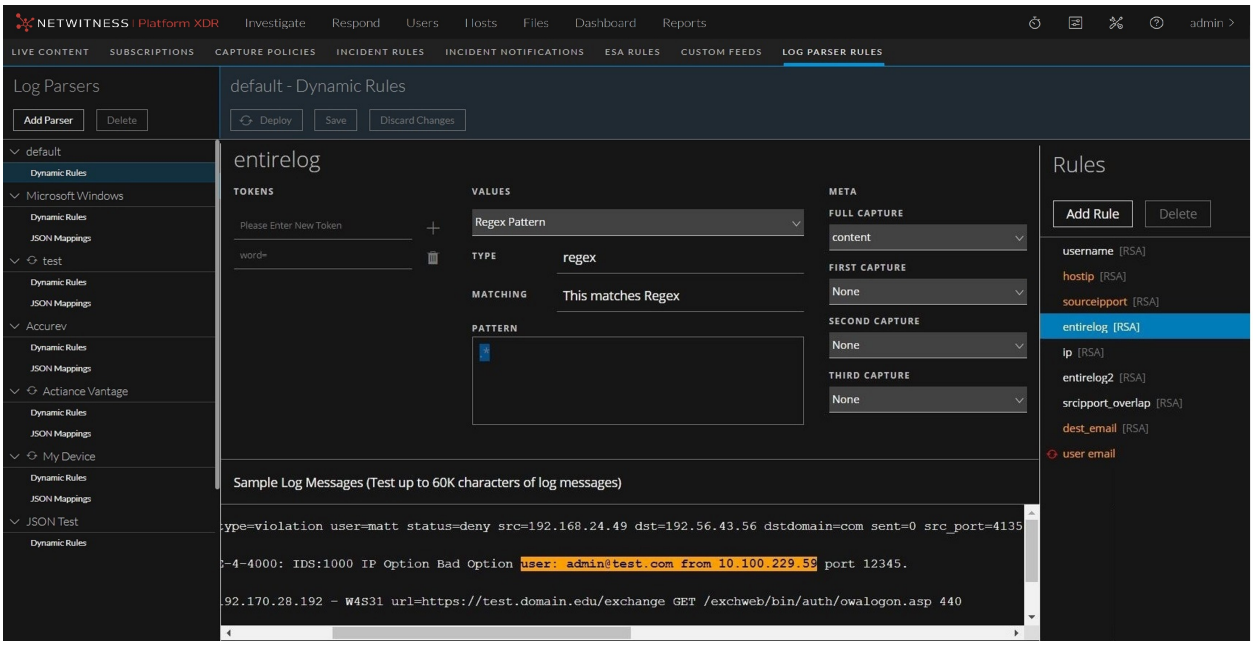
- The destination domain address, matching the **dstdomain** token, is highlighted in blue. The token is in dark blue, and the matching string is highlighted in light blue. This is because the **Destination Domain** is the currently selected Rule.
- The strings highlighted in orange match tokens for rules for **Username**, **Source IP or IP:Port**, **Destination IP or IP:Port**, **Source Port**, **Source Email Address**, and **Destination Email Address**. This is because they are in rules for the default parser that are not currently selected.

Highlight Overlapping Patterns

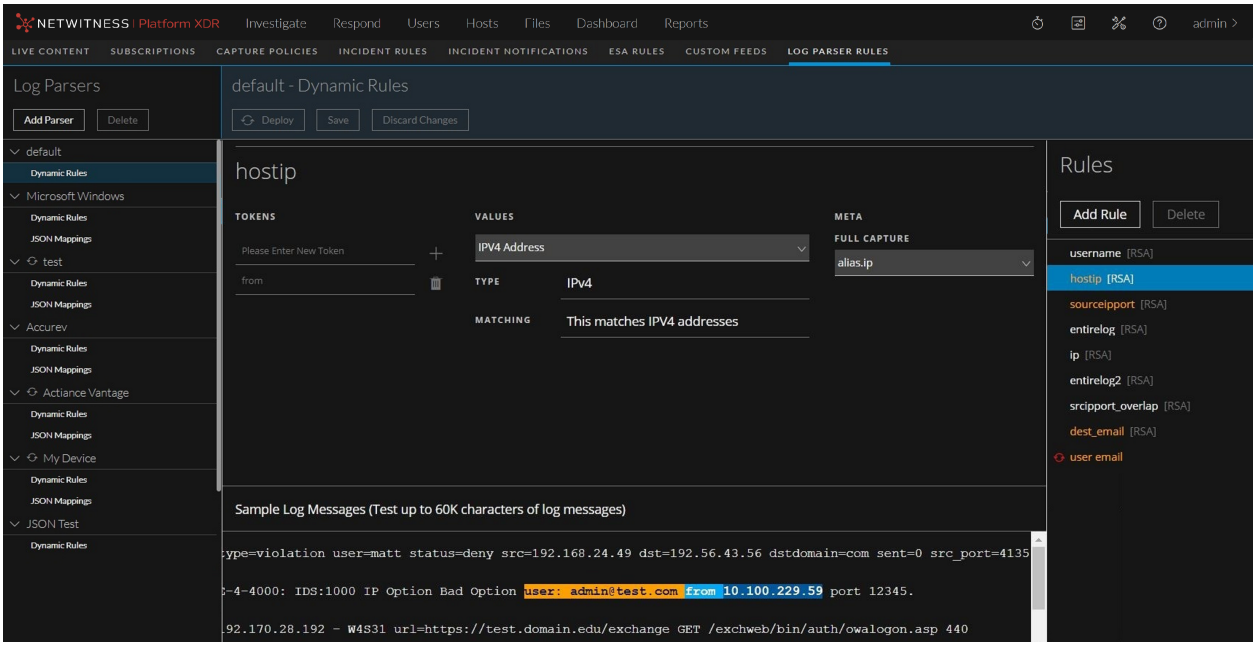
When you have patterns that overlap rules (that is, one pattern matches more than one rule), the behavior is as follows:

- The pattern is displayed in a single color (yellow)
- When you select one of the matching rules, the exactly-matched pattern is displayed in light and dark blue

For example, the pattern `user: admin@test.com from 10.100.229.59` matches several rules.



When you select the **hostip** rule, the highlighting that matches only this rule is shown in dark and light blue.



Use Cases

This topic describes the procedures you use to either on board a new event source, or to extend the parsing capabilities for an existing log parser.

Use Case 1: On Board a New Event Source

In this case, a customer has an event source and wants to add it into the NetWitness. Perform the following tasks:

I. For your event source, get examples of the logs.

II. In the **CONFIGURE > Log Parser Rules** view, add the Log Parser.

III. From your sample logs, paste applicable sections into the Sample Log Messages section of the **Log Parser Rules** screen.

IV. Use the sample area to understand which items are being parsed by the current parser, and note the items that are not being parsed.

V. For anything that is not currently being parsed, add rules.

- If the new rules apply to all parsers, you can add them to the Default parser.
- If not, add them only to the new log parser you are creating.

VI. Save the new rules, and deploy them to all Log Decoders.

VII. Map the IP address for the newly added event source to the newly-created log parser. For details, see "Acknowledging and Mapping Event Sources" in the *Event Source Management User Guide*.

Use Case 2: Modify an Existing Parser

In this case, a customer wants to parse some items from the logs that are not currently being parsed by the existing log parser. Perform the following tasks:

I. For your event source, get examples of the logs.

II. In the **CONFIGURE > Log Parser Rules** view, add the Log Parser.

III. From your sample logs, paste applicable sections into the Sample Log Messages section of the **Log Parser Rules** screen.

IV. Use the sample area to understand which items are being parsed by the current parser, and note the items that are not being parsed.

V. For anything that is not currently being parsed, add rules.

VI. Save the new rules, and deploy them to all Log Decoders.

For a detailed walk through of some of the steps in these use cases, see Extend an Existing Log Parser Example.

Extend an Existing Log Parser Example

Note: The information in this topic applies to NetWitness Version 11.2 and later.

Parse Rules can be used to parse unknown logs from existing devices. If a log is identified as a particular type (**device.type** is populated), and is not already being parsed (**msg.id** is not populated), then Parse Rules can be added to pull out relevant data from these logs.

If the neither **device.type** nor **msg.id** are populated for the logs from an existing device, then you need to map the device before Parse Rules can be processed against these logs.

Note: If a log message is already being parsed (**msg.id** is populated) then Parse Rules will not be processed against that log.

Task Overview

In this example, a customer wants to parse some items from the logs that are not currently being parsed by the existing log parser. Perform the following tasks:

I. For your event source, get examples of the logs.

II. In the CONFIGURE > Log Parser Rules view, Add the Log Parser

III. From your sample logs, paste applicable sections into the Sample Log Messages section of the **Log Parser Rules** screen.

IV. Use the sample area to understand which items are being parsed by the current parser, and note the items that are not being parsed.

V. For anything that is not currently being parsed, add rules as described in Add Rules and Deploy.

VI. Save the new rules, and deploy them to all Log Decoders.

Notes

Note: All the procedures in the topic use the CONFIGURE > Log Parser Rules view.

In the Log Parser Rules tab, you may see the Refresh icon () next to an item. This indicates that the item has undeployed changes.

Add the Log Parser

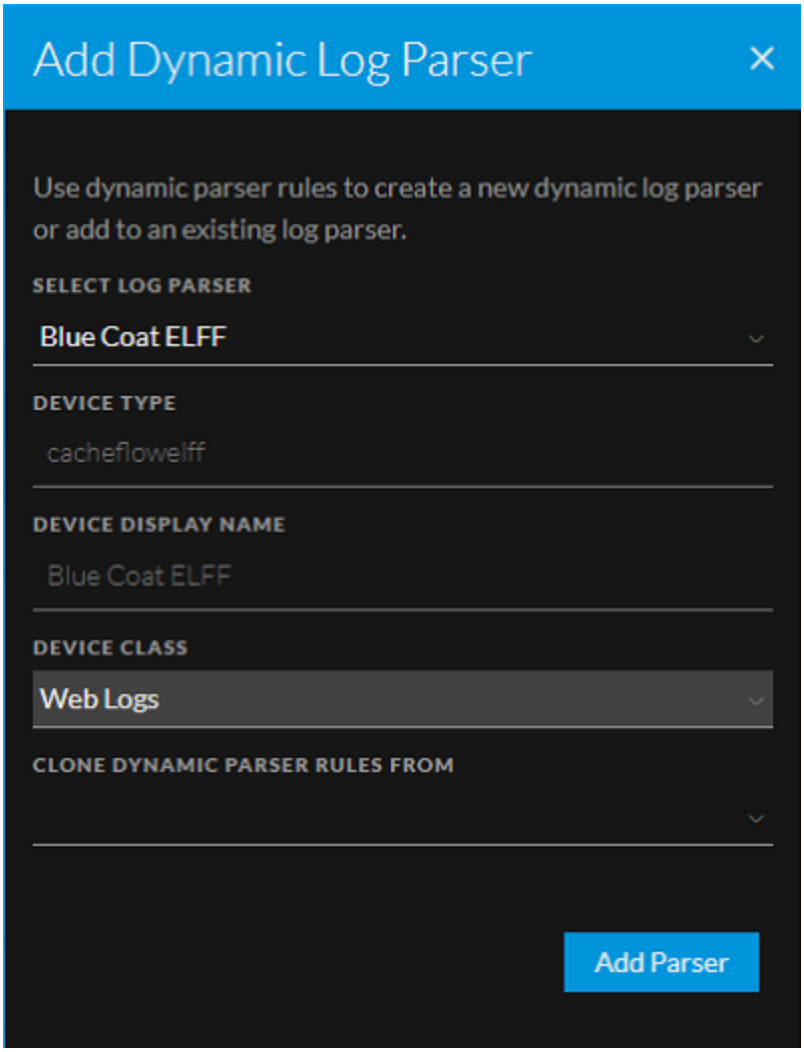
The first step in the process is to add a log parser, based on an existing log parser that you want to customize.

To add a log parser

1. In the NetWitness menu, navigate to **(Configure) > Log Parser Rules**.
2. In the Log Parsers panel, click **Add Parser**.

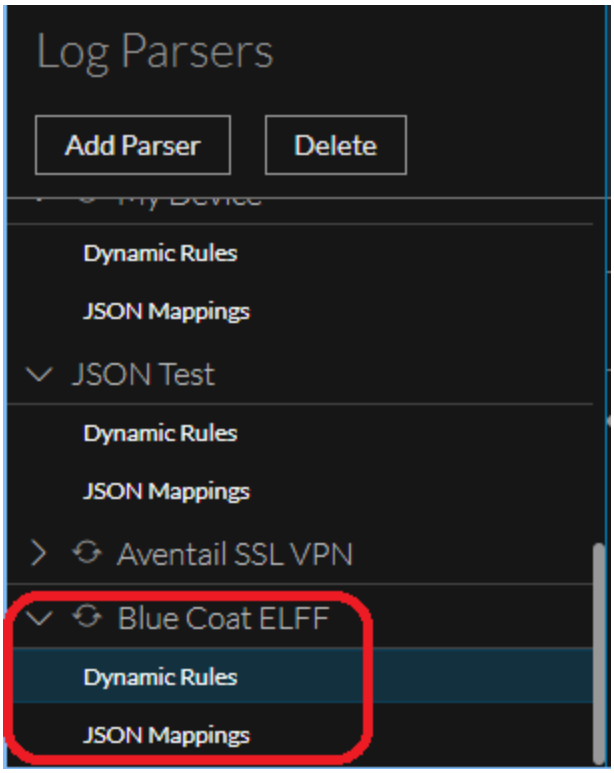
   The Add Dynamic Log Parser dialog box is displayed.
3. In the **SELECT LOG PARSER** field, select the existing parser to extend. In this example, we use Blue coat ELFF.

   Note: Values for Device Type, Device Display Name, and Device Class are filled in automatically, based on the log parser you selected.



   You can clone the rules from any of your existing parsers, including the **default** parser. For simplicity, in this example we leave this field blank: thus, only the rules we create are added to the new parser.
4. Click **Add Parser** to create the new parser.

The new parser is listed in the Log Parsers panel. Note the symbol next to the new parser—this indicates that your changes have not yet been saved.



About Custom Rules

When you create a new log parser rule, it is saved to an XML definition file for the parser. These files are known as token files. This is important, because the built-in rules are overwritten if you update the parser through Live, but any custom log parser rules are not overwritten, since Live does not update the token files for log parsers.

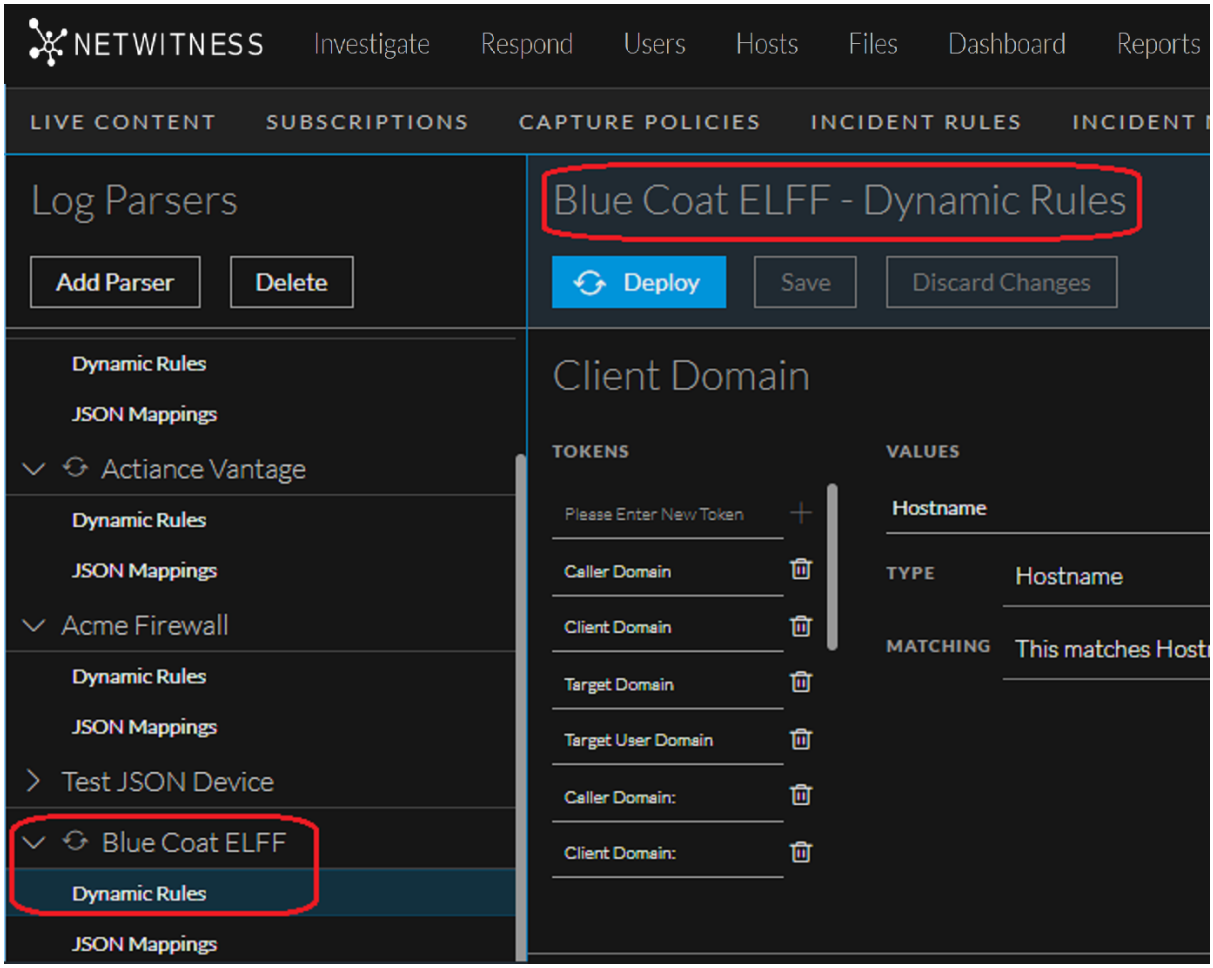
Add Rules and Deploy

After you have added the parser, the next step is to add one or more rules.

For example, you know that your log messages have some email addresses that follow a "source_mail" string. You could add the following rule to parse these strings:

IMPORTANT: If you click on another parser in the **Log Parsers** panel, before you save your rule, your changes are lost.

1. Make sure the Dynamic Rules entry for the Blue Coat ELFF parser is selected.

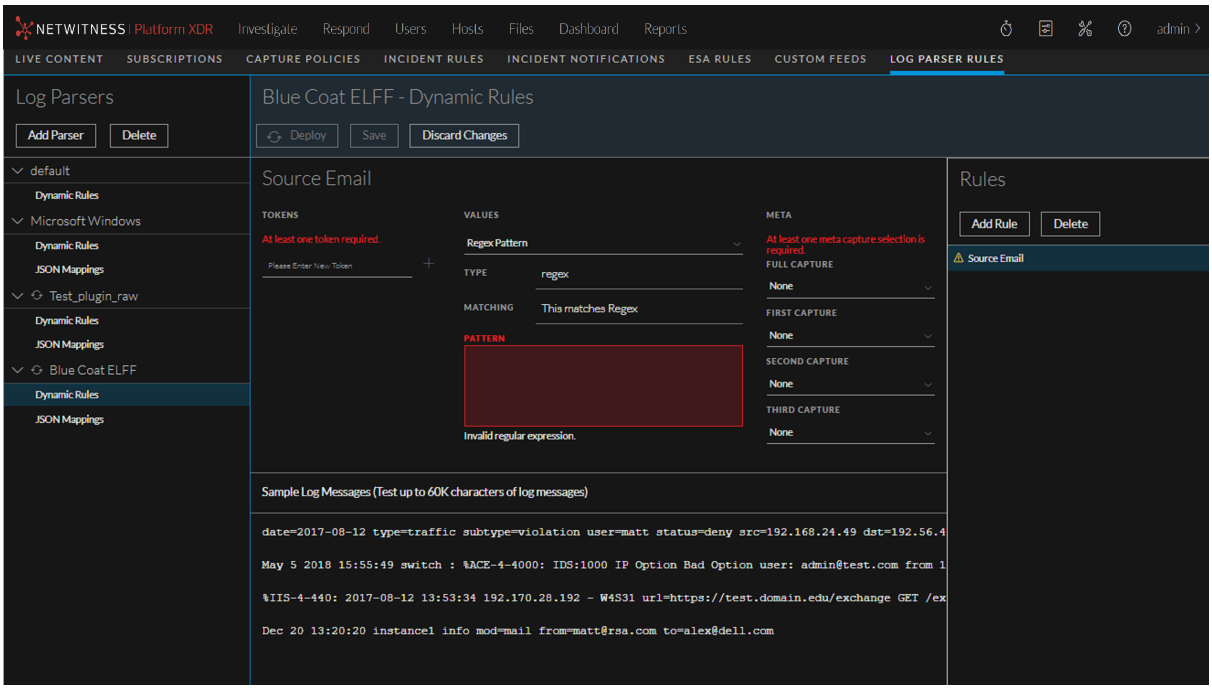


2. In the Rules panel, click **Add Rule**.

   The Add New Rule dialog box is displayed.

3. Enter a name for the rule, and click **Add New Rule**.

   The center panel is updated to reflect that you are working on a new rule.



In the TOKENS section, enter a string for the token that you want to match, then click +.

In this example, we entered **email** .

Note: Make sure to include a delimiter for your token. For example, in this case, the token consists of 6 characters: the string "email," and then a space. Some tokens might use a colon, semicolon, or some other character as the delimiter, but it can be easy to forget to add the space character when that is the delimiter.

4. You can enter more tokens, or continue to add values.
5. In the VALUES section, choose the value for the rule. If you choose to match a Regex Pattern, you need to enter the pattern in the PATTERN field. Other values do not require any options.

   In this example, we selected **Email Address**.

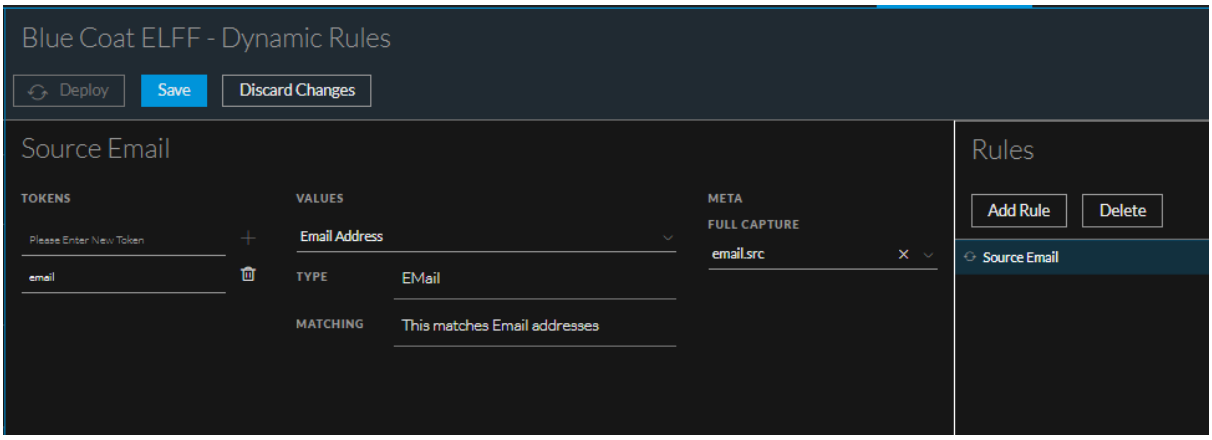
6. In the META section, click  to select a meta key to which the rule stores its information. Some notes:

   - Enter characters to filter the list of available meta keys.
   - For Regex values, you can select "pieces" of the value, and store each piece to its own meta key.

Note: If any new meta keys are added to the Log Decoder, they do not appear in the list of Meta immediately. They appear automatically after 24 hours, or you can restart the **content server** service to view them.

In this example, we selected the **email.src** meta key.

The following image shows an example rule:



7. Click **Save** to save the rule. Repeat this procedure to continue adding rules.
8. After you have added all of your rules, click **Deploy** to deploy the new parser to your Log Decoders. Some notes about deploying rules:
   - You deploy an entire set of rules for a parser. That is, you can continue adding rules for a specific parser until you have all of your rules, and then you can deploy them all at once.
   - After you deploy a custom parser, you can no longer delete it. You can only delete parsers that you have not yet deployed.

Note: In this example, we extended an existing log parser. However, if you are creating a new log parser for a new event source, make sure to map the new log parser to the IP address of the event source, as described in "Acknowledging and Mapping Event Sources" in the *Event Source Management User Guide*.

Regex Values

Custom Log Parser Rules can match regular expression patterns. If you select a Regex pattern for your Value, you can capture the entire matched token, or sections of it:

- Full capture: the entire matched string is stored to your selected meta key.
- First capture: the first portion of the string, up to the period character, is stored to the meta key.
- Second capture: the second portion of the string, starting after the first period character, is stored to the meta key.
- Third capture: the third portion of the string, starting after the second period character, is stored to the meta key.

You can choose any or all four of these captures, depending on the token you are matching.

For example, we examine the **Source IP or IP:Port** rule:

- Regex Pattern: \s*(\b(?:[0-9]{1,3}\.){3}[0-9]{1,3}\b):?(\d*)
- Full capture: none

- First capture: **ip.src**
- Second capture: **port.src**
- Third capture: none
- Assume example string of "src=192.168.24.4:8080", where **src** is one of the tokens defined for this rule:
  - **192.168.24.4** is saved to the **ip.src** meta key.
  - **8080** is saved to the **port.src** meta key.

For more details, see any online reference that describes PERL regular expressions. There are many tutorials available online.

IMPORTANT: Be careful when constructing regular expressions in your custom rules. Badly constructed regular expressions could impact your performance.

Appendix A: Select the Reference Log Decoder

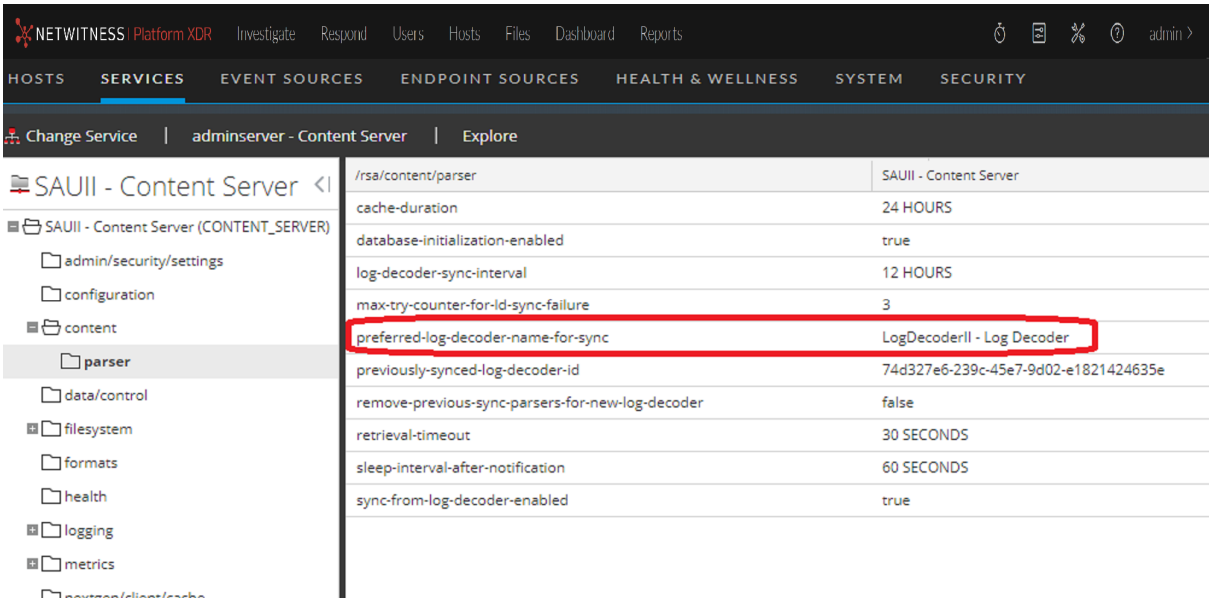
For version 11.2, NetWitness has added the ability to add log parsers and log parsing rules through the UI, using the Log Parsers view. The Log Parsers tab is populated based on your reference Log Decoder. If you have more than one Log Decoder, you can select which acts as the reference one for populating the tab in the UI. This topic describes the procedure to do so.

Note: If you have previously set a reference log decoder, make sure that it is at NetWitness version 11.5 or later to get full functionality.

To change the reference log decoder:

1. In the NetWitness UI, navigate to **(Admin) > Services**.
2. For the **Content Server**, select **View > Explore**.
3. From the left navigation panel, expand **content > parser**.
4. To set the reference log decoder, enter a value for `preferred-log-decoder-name-for-sync`.

   Enter the name listed in the **Name** column on the **ADMIN > Services** screen for your preferred log decoder.



5. The change takes effect during the next system sync, based on the `log-decoder-sync-interval`. To sync sooner, you can do either of the following:

   - To sync immediately, restart the Content Sever: in the **(Admin) > Services** view, from the **Actions** menu for the Content Server, select **> Restart**.
   - Change the `log-decoder-sync-interval` parameter from its default of 12 hours to your preferred interval. Note that the minimum value for this parameter is **1 HOUR**.

Appendix B: Move Log Parsers to Production

You may have a development or test environment where you work on new and updated log parsers and log parser rules. In this case, at some point you need to move your new and updated log parsers into your production environment. This topic describes how to do this.

To move custom log parsers and log parser rules from development to production environment:

1. On the development system, do the following:
   a. SSH to the NetWitness Server
   b. Export the log parser information by running the following command:

      ```
      mongodump --host localhost --port 27017 --db "content-server" --username
      "deploy_admin" --password "netwitness" --authenticationDatabase admin
      ```

   c. Copy the "dump" folder to your production NetWitness Server.
2. On the production system, do the following:
   a. SSH to the NetWitness Server
   b. Drop the content-server table from Mongo by running below commands in the order listed:

      ```
      mongo --username deploy_admin --password netwitness --
      authenticationDatabase admin
      use content-server
      db.logDeviceParser.drop()
      db.patternFormatType.drop()
      exit
      ```

   c. Run the following restore command:

      ```
      mongorestore --host localhost --port 27017 --db "content-server" --
      username "deploy_admin" --password "netwitness" --authenticationDatabase
      admin PATH_TO_DUMP_FOLDER
      ```

      Make sure to replace *PATH_TO_DUMP_FOLDER* with the actual path to the "dump" folder.
   d. Restart the content-server by running the following command:

      ```
      systemctl restart rsa-nw-content-server
      ```

Appendix C: Troubleshooting and Limitations

This section describes some common issues that can occur when you customize log parsers and log parser rules.

Troubleshooting

You do not see any log parsing against a newly created parser.	You may have forgotten to map the new parser. To map a parser, go to **(Admin) > Event Sources > Discovery** tab. See the "Discovery Tab" topic in the *Event Source Management Guide* for details.
Deployment fails	If you click Deploy to deploy a new or updated log parser, and it fails, you should check the log for your reference log decoder. You access this log in the following location on the NetWitness Server: `/var/log/netwitness/content-server/content-server.log`

Delete a Log Parser Manually

If you have any issues when you attempt to remove a Log Parser through the UI, you can manually delete a log parser by using **NwConsole**.

To delete a log parser that has been deployed:

1. Access the NetWitness Console, using the **NwConsole** command. For details, see "Access NwConsole and Help" in the *NwConsole User Guide*.
2. Run the following command:

   ```
   [localhost:50002] /decoder/parsers> send . delete file=filename.xml
   type=device
   ```

   where ***filename*** is the name of the XML file for the log parser. For example, to delete the log parser for Oracle Access Manager, run the following command:

   ```
   [localhost:50002] /decoder/parsers> send . delete file=oracleam.xml
   type=device
   ```

Notes about the log parser filename:

- Log parser files are located on the Log Decoder in the following path:

  `/etc/netwitness/ng/envision/etc/devices`

- Each log parser has its own sub-folder. For example, the Cisco ASA parser files are in the following folder:

  `/etc/netwitness/ng/envision/etc/devices/ciscoasa`

- Some log parser file names begin with **v20_**, while others do not—the only way to tell is by examining the `devices` folders. For Cisco ASA, the log parser file name is **v20_ ciscoasamsg.xml**. However, in the previous command, when you specify the filename, do ***not*** use the **v20_** prefix.

NwLogPlayer

NwLogPlayer is a troubleshooting tool that simulates syslog traffic. `NwLogPlayer.exe` is a command line utility located on the Log Decoder host in `/usr/bin`.

At the command line, type `nwlogplayer.exe -h` to list the available options, as reproduced here:

Option	Description
`--priority arg`	set log priority level
`-h [ --help ]`	show this message
`-f [ --file ] arg (=stdin)`	input message; defaults to **stdin**
`-d [dir ] arg`	input directory
`-s [ --server ] arg (=localhost)`	remote server; defaults to **localhost**
`-p [ --port ] arg (=514)`	remote port; defaults to **514**
`-r [ --raw ] arg (=0)`	Determines raw mode. • 0 = add priority mark (default) • 1= File contents will be copied line by line to the server. • 3 = auto detect • 4 = enVision stream • 5 = binary object
`-m [ --memory ] arg`	Speed test mode. Read up to 1 Megabyte of messages from the file content and replays.
`--rate arg`	Number of events per second. This argument has no effect if **rate** > eps that the program can achieve in continuous mode.
`--maxcnt arg`	maximum number of messages to be sent
`-c [ --multiconn ]`	multiple connection
`-t [ --time ] arg`	simulate time stamp time; format is `yyyy-m-d-hh:mm:ss`
`-v [ --verbose ]`	If **true**, output is verbose
`--ip arg`	simulate an IP tag
`--ssl`	use SSL to connect
`--certdir arg`	OpenSSL certificate authority directory
`--clientcert arg`	use this PEM-encoded SSL client certificate
`--udp`	send in UDP

Limitations

Please note the following limitations when using the Log Parser Rules tab:

- **Log Decoder must be at version 11.2:** For the functionality in the Log Parser Rules tab to work, your installation must have at least one Log Decoder running NetWitness version 11.2 or later.
- **JSON Mapping: Log Decoder must be at version 11.5:** For the JSON Mapping Beta functionality to work, your installation must have at least one Log Decoder running NetWitness version 11.5 or later.
- **Mixed Mode:** If any Log Decoders are at version 11.2 or later, and the NetWitness Server is at version 11.2 or later, the Log Decoders will have parseall rules enabled by default, and thus will begin to parse logs accordingly. However, the 11.2 NetWitness Server does not support Log Decoders with versions less than 11.2, so the Log Parser Rules tab in the UI stays blank.
- **Meta key fields list refresh:** If any new meta keys are added to the Log Decoder, they do not appear in the list of Meta in the Log Parser Rules tab immediately. They appear automatically after 24 hours, or you can restart the **content server** service to view them.
- **Field Restrictions:** Note the following field restrictions:
  - **Rule name** must be 64 characters or fewer.
  - **Parser Name** must be between 3 and 30 alphanumeric characters (including underscores), and must not match the name of any existing log parsers.
  - **Parser Display Name** must be 64 characters or fewer, and cannot match any other parser display name.
  - **Regex Expression** must be 1-255 characters, and a valid regex (closed capture list allowed).
  - **Tags** cannot be duplicates.
- **Deploy only to 11.2 Log Decoders:** The Deploy operation only deploys log parsers to version 11.2 or later Log Decoders.
- **Cannot Remove Deployed Parsers:** Once deployed, you cannot delete a log parser using the UI.
- **See log for errors:** Refer to content-server logs for more details on deploy failure details and log decoder names.

Parser Rules Tab

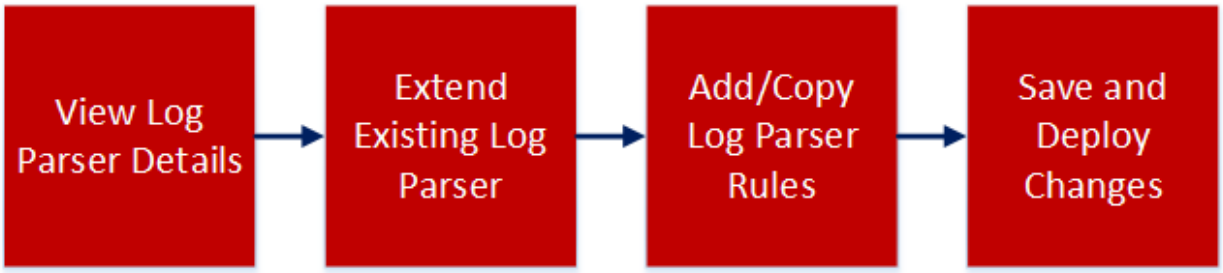
Note: The information in this topic applies to NetWitness Version 11.1 and later.

This tab contains details about the rules for the default log parser, as well as any other custom rules and log parsers that have been defined.

To access this tab, go to  **(Configure) > Log Parser Rules**.

Workflow

This workflow shows processes available from the Log Parser Rules view.



What do you want to do?

Role	I want to...	Documentation
Administrator	*View log parser rules.	Default Log Parser and Log Parser Rules
Administrator	*Add, edit or delete a log parser rule (version 11.2 and later)	Add or Delete a Log Parser Rule
Administrator	*Add or remove a log parser (version 11.2 and later)	Add or Delete a Log Parser

*You can perform this task here.

Related Topics

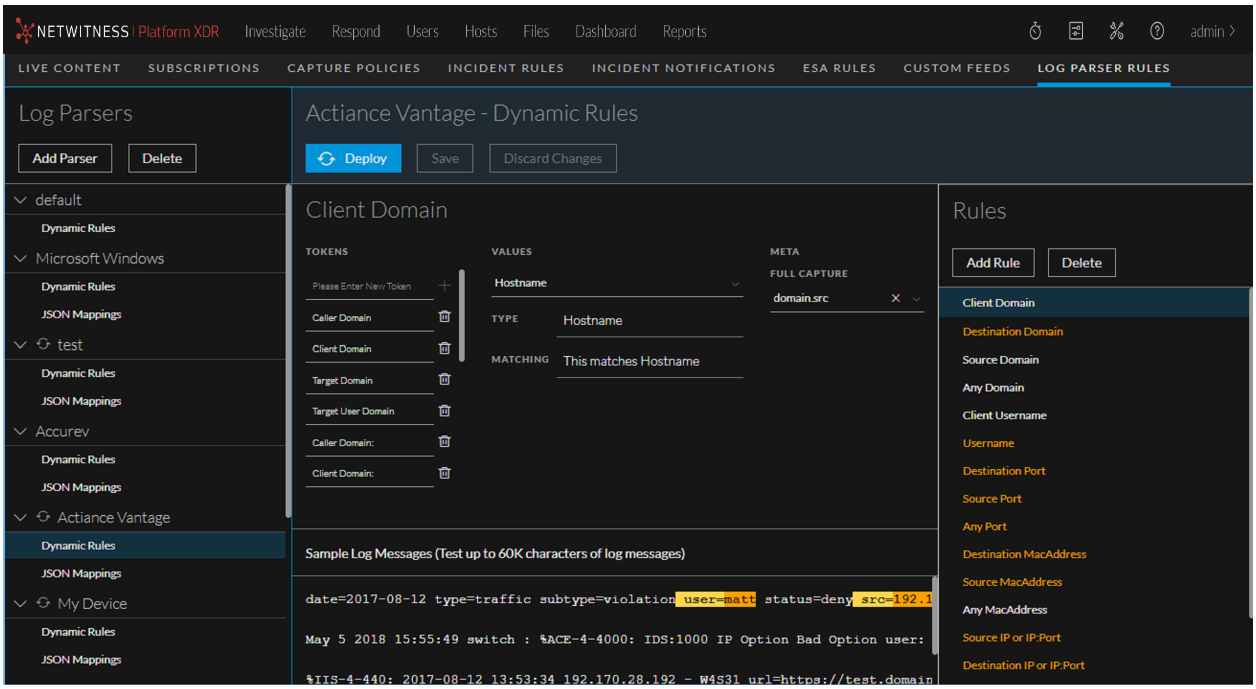
Default Log Parser and Log Parser Rules

Quick Look

Note: The list of log parsers is based on the first Log Decoder that is installed or registered by the Orchestration Server. If you have more than one Log Decoder, this tab only lists log parsers that have been configured on the first one.

The Log Parser Rules tab organizes and displays information about the configured log parsers in your system.

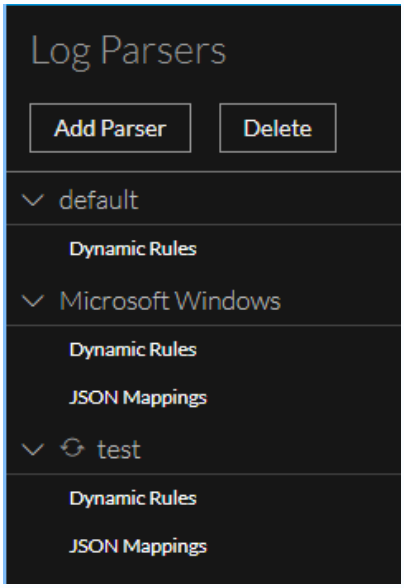
If you select the Dynamic Rules entry for a parser, the screen displays the dynamic rules:



Note: The icon indicates that there is some uncompleted or unsaved work. For example, next to a parser name, it means that the parser has not yet been deployed.

If you select JSON Mappings, the screen displays JSON mappings for the parser.

Log Parsers Panel

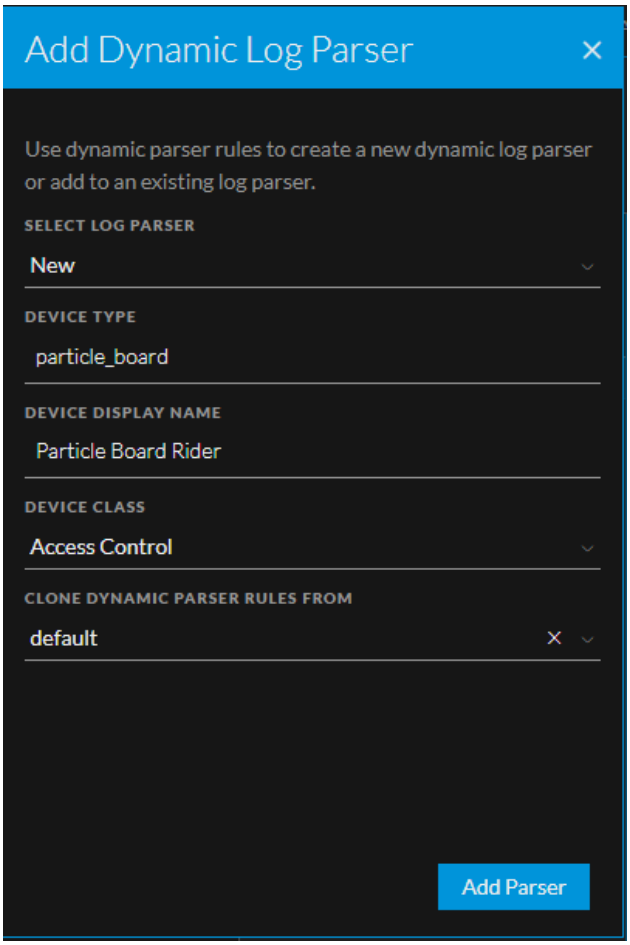


The Log Parsers Panel lists the configured log parsers.

- Until you add rules to existing XML parsers on your reference Log Decoder, (or add a new, custom log parser) only the **default** parser is listed here.
- Select a specific log parser to view its details in the Details and Rules panels.
- Click **Add Parser** to open the Add Dynamic Log Parser dialog box.
- Click **Delete** to delete a log parser.

IMPORTANT: Once you deploy a log parser, you can no longer delete it through this interface. The **Delete** button is not available for deployed parsers. To manually delete a log parser, see Add or Delete a Log Parser.

The Add Dynamic Log Parser dialog box allows you to add a custom log parser.



When you are adding a log parser, the following parameters are available.

Field	Details
SELECT LOG PARSER	Select NEW, or choose an existing log parser. By choosing an existing log parser, you can add rules to that parser, essentially extending its parsing capabilities. **Note:** If you select an existing log parser, the remaining fields are auto-filled based on the values for selected log parser.
DEVICE TYPE	Enter a string to define the device type. The name must be between 3 and 30 alphanumeric characters (including underscores), and must not match the name of any existing log parsers.
DEVICE DISPLAY NAME	Enter the display name for the log parser. **Note:** The display name must be 64 characters or fewer, and must not match the name of any other device display name.
DEVICE CLASS	Select a device class.
CLONE DYNAMIC PARSER RULES FROM	Leave blank to start with no rules, or select one of the existing log parsers to clone its rules.

Dynamic Rules

If you select the Dynamic Rules entry for a parser, you see the following panes:

- Details
- Rules

Details Pane

The details pane shows the three pieces for the selected rule:

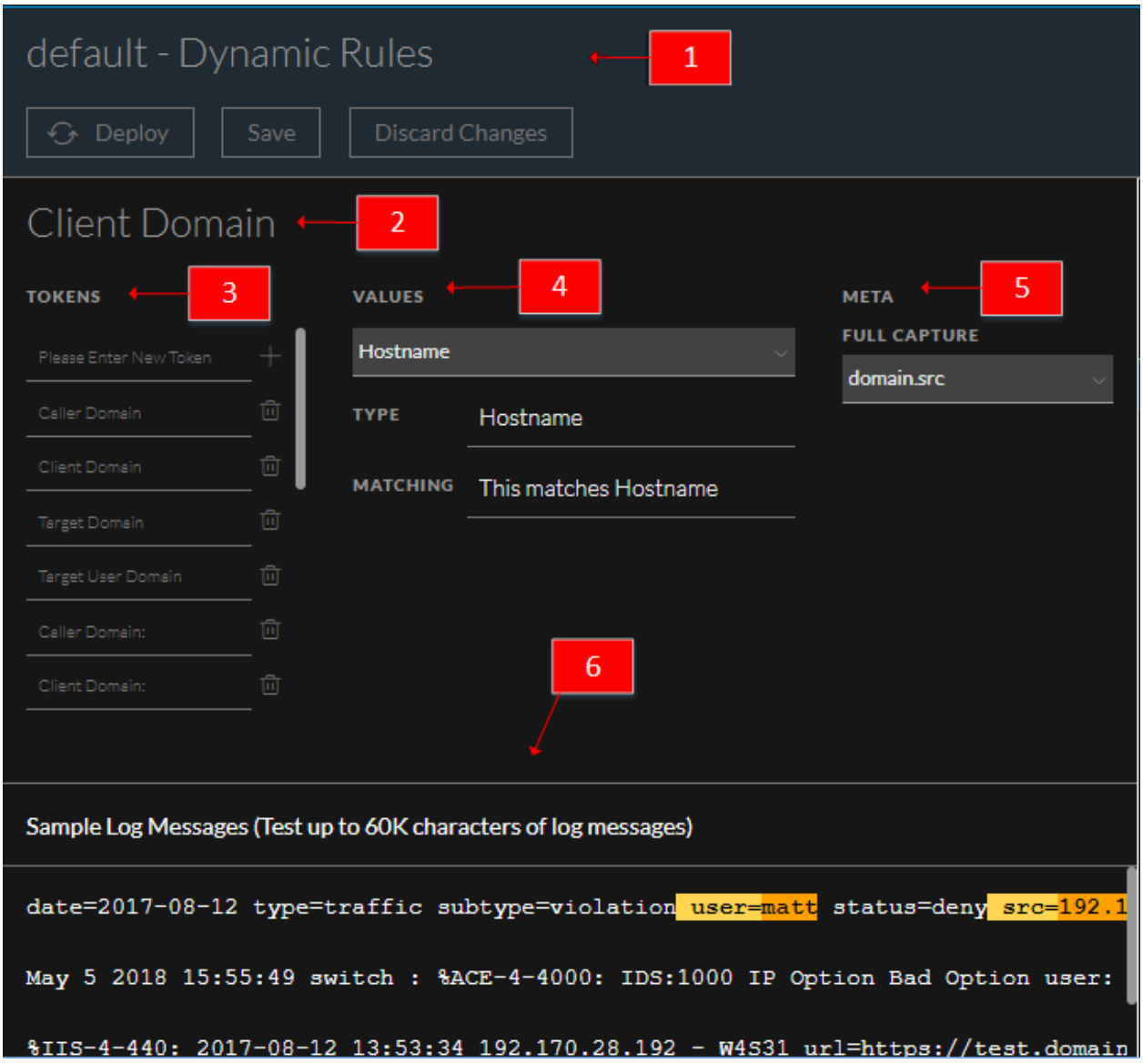
- **Tokens:** one or more tokens to match in the message. For example, the Any Port rule looks for the following strings to match against: **port** , **port:**, **port=**, and others.
- **Values:** the value that follows the token. This is a string that is captured as meta. For example, assume a log contains the following string:

  ```
  port 12345
  ```

  The Any Port rule has a token that matches "port ". When it encounters that string, it assigns the token value, "12345" to a meta key.
- **Meta:** the meta keys to which the value is mapped. For example, the Any Port rule maps the port value to the **port** meta key.

Essentially, a rule says, "when you are parsing a message, if you match one of my tokens, assign the value that follows the token to the meta key that I want it stored as."

The bottom section of the Details panel contains sample log messages, and how they would be parsed for the selected log parser.



1	Displays the name of the selected log parser, and the buttons for deploying, saving, and discarding changes. This value changes when you select a different parser.
2	Displays the name of the selected rule. This value changes when you select a different rule for this parser.
3	Displays the list of tokens defined for the selected rule.
4	Displays the type and pattern of the value matching for the selected parser. The values here are determined by the type of the selected value. You can also use the Regex option to define a custom regular expression.
5	Displays the NetWitness meta to which the selected rule maps any matched tokens. The values here are determined by the selected Rule.
6	Displays a sample log message, and highlights strings that match tokens in the selected log parser. You can edit this field, and add in your own logs to preview how the selected parser will parse your logs. **Note:** The sample section refreshes whenever a rule is changed or updated, as well as when you

paste in samples from your logs.

For example, consider the following scenario:

- The **default** parser is selected.
- The **Any Domain** rule is selected.
- The Tokens matching list displays all of the tokens that are matched when found in a log message: **Domain**, **Domain Name**, **domain**, **ADMIN_DOMAIN**, and so on.
- The Meta list displays the NetWitness meta to which the value for the token is mapped: **domain**.

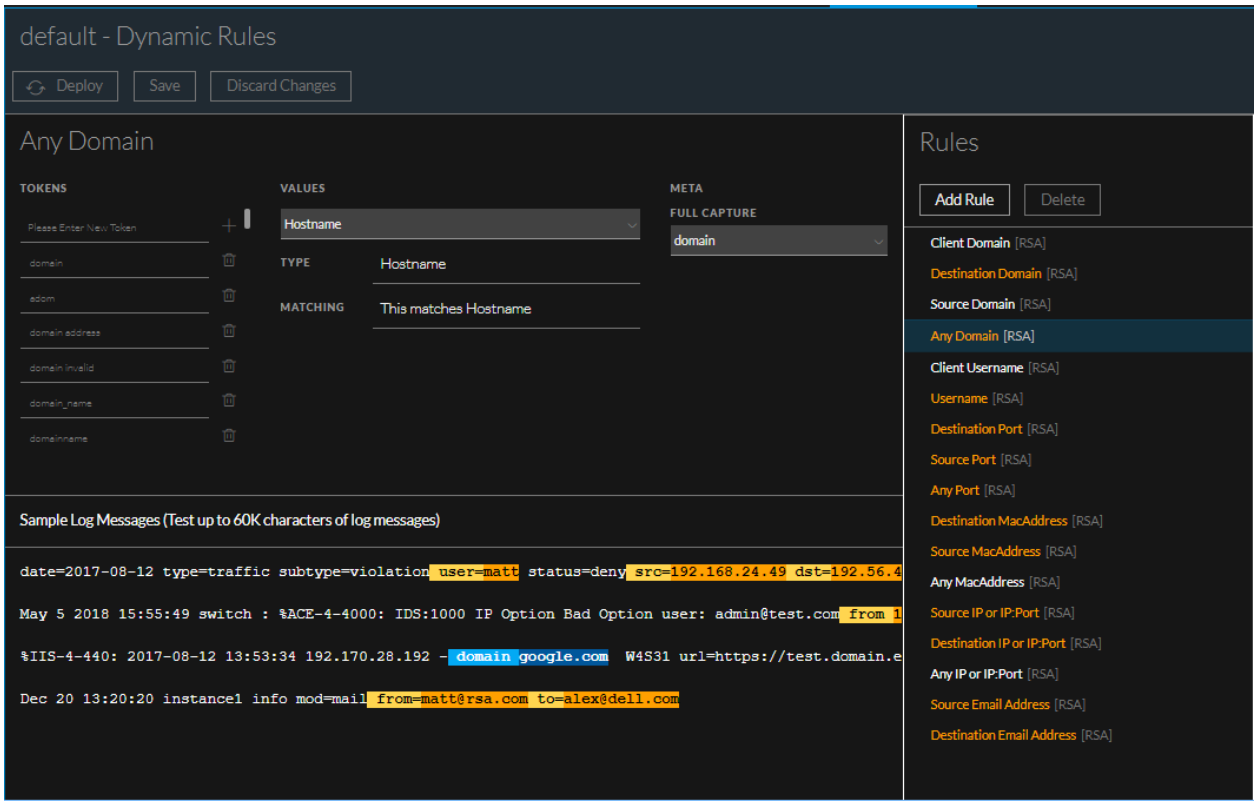
So, let's say the sample log message area has the following text:

```
Below are sample log messages:

May 5 2010 15:55:49 switch : %ACE-4-400000: IDS:1000 IP Option Bad Option
List by user admin@test.com from 10.100.229.59 to 224.0.0.22 on port 12345.

Apr 29 2010 03:15:34 pvg1-ace02: %ACE-3-251008: Health probe failed for
server 218.83.175.75:81, connectivity error: server open timeout (no SYN ACK)
domain google.com with mac 06-00-00-00-00-00.
```

In this case, the Sample Log Message area looks like this:



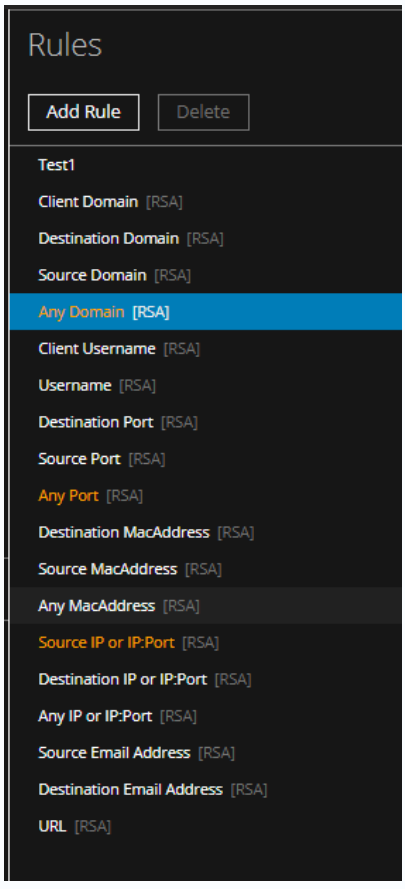
Note that some strings are highlighted, and that there are two "pairs" of highlight colors:

- Dark blue and light blue highlighting is applied to the strings that match the currently selected rule.
  - Dark Blue highlighted strings match a token in the selected rule. In this case, **domain** is the token that is matched for the **Any Domain** rule.
  - Light Blue highlighted strings are the values that correspond to the tokens in dark blue. For example, **google.com** is highlighted in light blue, because it corresponds to the **domain** token.
- Orange and yellow highlighting is applied to the strings that match rules for the current parser that are *not* currently selected.
  - Orange highlighted strings match a token in a rule that is not currently selected.
  - Yellow highlighted strings are the values that correspond to the tokens in orange. For example, the **user** token matches the **Username** rule (which is not currently selected).

In this example, the **domain** meta would be assigned a value of **google.com** for this log message, if it was parsed using the default log parser.

Rules Pane

The Rules pane displays the list of rules used by the selected log parser. When you select a rule, you change the values that are displayed in both the **Tokens** and **Values** areas of the panel.



Note the highlighted rules:

- The currently selected rule is highlighted in blue.
- Other rules that match tokens in the sample log message area are highlighted in orange.

Other notes for the Rules panel:

- RSA rules (the rules provided out-of-the-box for each log parser) are identified by **[RSA]** following the rule name.

  You can copy these rules when adding a new log parser, and then change them as needed.
- The **Delete** button is only available for custom rules; for RSA rules, it is greyed out.
- Use the **Add Rule** button to add a custom rule.

JSON Mappings

If you select the JSON Mappings entry for a parser, you see the following panes:

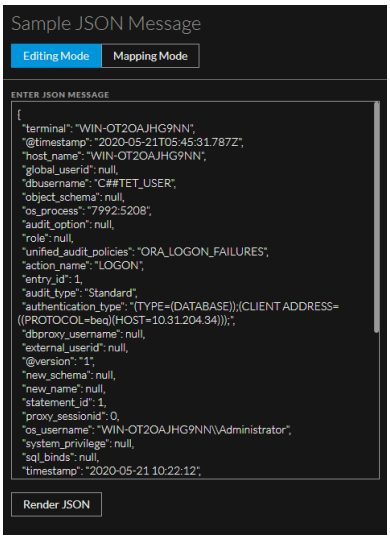
- Sample JSON Message
- Meta Mappings
- Mapping Details

Sample JSON Message

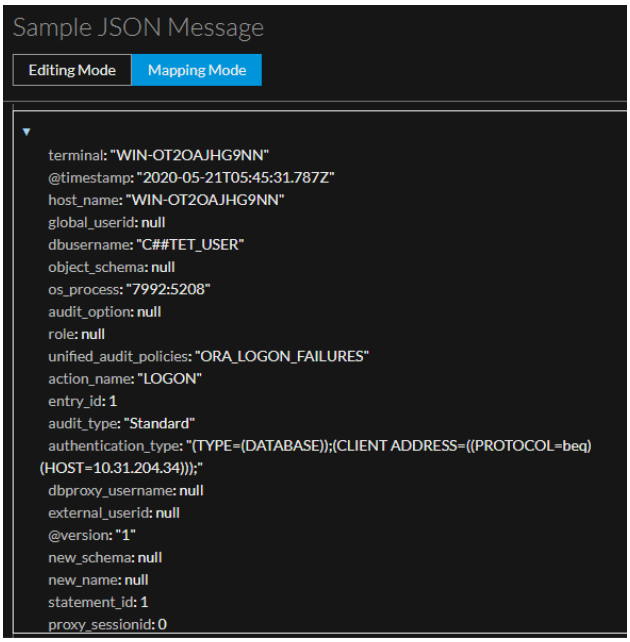
You can enter or paste sample JSON text. Click the **Render JSON** button to automatically format the text into JSON code. If the text is not valid JSON, you receive a message and the text is not formatted.

The following screen shows the Sample JSON Message area with some JSON that has been pasted in:

Since this is valid JSON, clicking **Render JSON** produces the following:

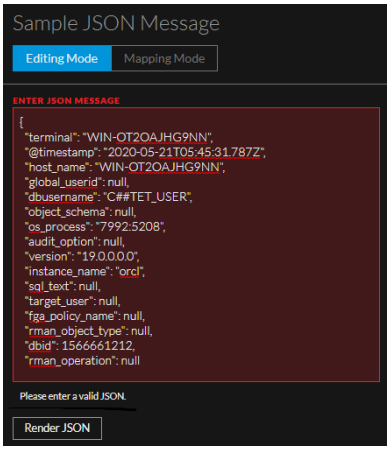


Note that you can see the tree mode of the sample JSON if you click **Mapping Mode**:



Note: You can edit the sample JSON in Editing mode, but not in Mapping mode. Mapping mode is read only.

On the other hand, if you enter text that is not valid JSON, the screen looks as follows:



Auto Discover JSON Mappings

Beginning with NetWitness version 11.5.1, you can automatically create the mappings without the need to manually enter the name and path of the mapping. For details, see .Auto Discover JSON Mappings.

When the system auto discovers mappings, the path is filled in automatically, based on the structure of the mappings. For simple name-value pairs, this is straightforward. For example, for this pair, **"host_ name": "WIN-OT2OAJHG9NN"** , the path is set to `/host_name`.

However, the rules for nested mappings and arrays are as follows:

- For a nested structure, names are separated with a forward slash (/).

```
{
  "parent": {
    "child": "value"
  }
}
```

The path is set to `/parent/child`.

- Arrays are accessed by omitting the index.

```
{
  "array": [
    "x",
    "y",
    "z"
  ]
}
```

The path is set to `/array/`.

```
{
  "array": [
    {
      "name": "value"
    }
  ]
}
```

The path is set to `/array//name`.

- Variable names are accessed by omitting them from the path.

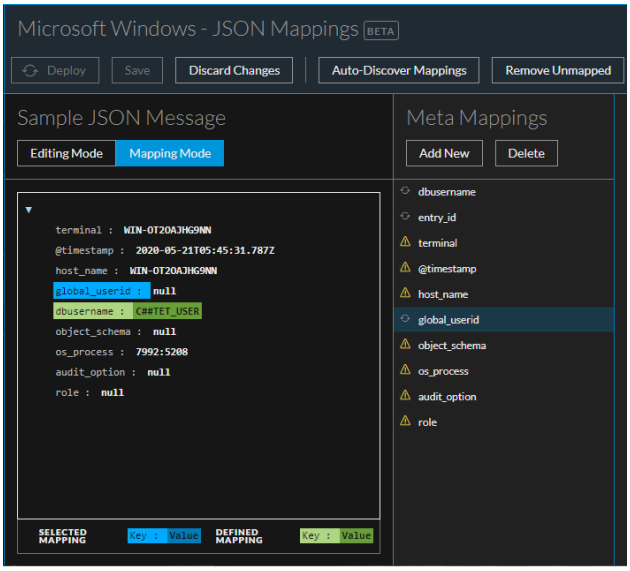
```
{
  "root": {
    "x": {
      "name": "value"
    },
    "y": {
      "name": "value"
    }
  }
}
```

The paths are set to `/root/x/name` and `/root/y/name`.

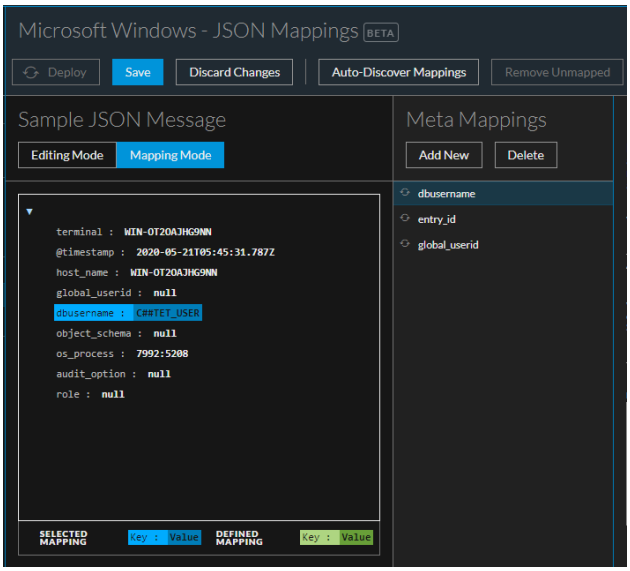
Remove Unmapped Entries

If there are mappings that you do not want to save, you can remove them. After you validate all of the mappings that you want to keep, you can click **Remove Unmapped** to remove all mappings that you have not yet validated.

For example, assume you have auto-discovered some mappings as shown here:



After you click **Remove Unmapped**, you can see that only mapped entries remain:



Note: You cannot save your work until all of the entries have either been mapped or removed from the list.

Meta Mappings

This panel lists the mappings that exist for this parser. You can add a mapping by using the **Add New** button, or delete an existing mapping by selecting it and clicking **Delete**.

Mapping Details

The Mapping Details pane displays the following information.

Field	Details
DISPLAY NAME	This name corresponds to the name displayed in the JSON Mappings panel.
PATH	The path to where the values for this portion of the log are stored.
DESCRIPTION	Optionally, you can enter a text description for this mapping.
META	Select a meta key to which this value from the log is mapped. Select a value from the drop-down menu. Optional if you choose a Value Format.
VALUE FORMAT	Choose a value format parser onto which to pass this JSON value. You can either select an existing meta or **Custom Regex Type**. If you select custom regex type, you must define the regex and capture to *fine* parse the value in the meta. Optional if you choose a Meta.
CUSTOM REGEX TYPE	Select Custom Regex Type from the Value Format drop-down, which allows you to add new custom regex type.
REGEX PATTERN	Specify a regex to identify different pieces of data contained within a JSON node value.
FIRST CAPTURE	Select a meta key that should be captured first based on the value defined in the Regex pattern.

Field	Details
ADD A CAPTURE	New capture field is added. By default, it is loaded with meta keys in the drop-down. You can add maximum of 20 captures and this option will be disabled once it reaches maximum.

Note: You need to select a meta or enter a Value Format, but you do not need to fill in values for both settings.

Disable log Parser Rules

You can disable log parser rules, so that none of them are processed by the Log Decoder. You might have your log parsers working as you like, and do not want any extra processing that you do not need.

You disable them from the reference Log Decoder.

1. Go to  **(Admin) > Services**.
2. In the **Administration Services view**, select the Decoder and  **> View > Config.**

   The Services Config view is displayed with the General tab open.
3. Under **Parsers Configuration**, look at the Config Value for **PARSERULESCAN**.

   If it is **Enabled**, log parser rules are processed. If it is **Disabled**, they are not processed.
4. If the rules are Enabled, click Enabled and select Disabled to disable the log parser rules.

   To save the changes, click **Apply**.

Add VARTYPE Support to CEF Parser

VARTYPE added in 11.0 and later versions provide type and pattern validation for variables. While the variables are parsed, NetWitness has added the ability to condition them to match a certain format.

CEF Parser format logs with **CEF:0** and will use CEF parsers without any further identification. Some CEF sources have structured data embedded in variables. This data requires further processing. To direct parse rules scanning of the data, support for scanned variables is required for the CEF parser. Scanned variables are supported during the VARTYPE implementation.

To add VARTYPE support to CEF parser:

1. Load VARTYPE from CEF parser (cef.xml and cef-custom.xml) successfully.
2. Load all existing CEF parsers and parse the data successfully.
3. Enable VARTYPE to validate the header information.
4. Enable VARTYPE to fine parse CEF variable (MetaName).
5. Ensure that VARTYPE is working fine with custom CEF parser.

Defining Log Decoder Parse Rules

A single rule is defined by the following XML elements:

- **RULE** (element, required) - Each rule is enclosed in a RULE element.
  - **id** (attribute, required) - A name associated with the rule.
  - **stop** (attribute, optional) - An attribute to stop the rule.
  - **order** (attribute, optional) - An attribute to order the rule.
- **LITERAL** (child of RULE, requires one or more) - The literal defines the string inside each log message for which Log Decoder will search. Multiple literals can be defined.
  - **value** (attribute, required) = The string value for which to search.
- **PATTERN** (child of RULE, requires exactly one) - A pattern is defined as either a **regex** or a **format**.
  - **regex** (attribute, optional) - A regular expression that can be used to extract one or more values from the matched string.
  - **format** (attribute, optional) - A built-in format type for which to scan and extract values (see **Built-In Format** below).
  - **range** (attribute, optional) - Determines the range to be searched for the pattern before and after the found LITERAL token. It supports the following values:

Value	Description
after	Applies regex after the anchor till the end of the log. This is the default value.
before	Applies regex from the beginning of the log till the starting position of the anchor.
all	Applies regex to entire log.
before, after	Applies regex to entire log.
Negative Integer (-X)	Applies regex to X or available characters before the found LITERAL token.
Positive Integer (+X)	Applies regex to X or available characters after the found LITERAL token.
-X, +Y (Combination of Positive and Negative Integers)	Applies regex to X or available characters before the found LITERAL token to Y or available characters after the found LITERAL token. - range=”-15,+12”
-X, after (Combination of Negative Character, and After)	Applies regex to X or available characters before the found LITERAL token to the end of the log. - range=”-15,after”

- **CAPTURE** (child of PATTERN, optional) - A capture is used when extracting data from the log that was captured by the regular expression defined in PATTERN.
  - **index** (attribute, required) - The index of the regex match to capture.
  - **key** (attribute, required) - The meta key into which to assign the captured value.
  - **format** (attribute, optional) - The NwType of the key to be created (For example, Text, IPv4, UInt32 and so on). Default is Text.
- **META** (child of RULE, optional) - A meta element defines meta values that get created when there is a match for the pattern.
  - **key** (attribute, required) - The meta key into which to store the value.
  - **value** (attribute, required) - The value to store.
  - **format** (attribute, optional) - The NwType of the key to be created (For example, Text, IPv4, UInt32 and so on). Default is Text.

Note: The **format** defined in the **CAPTURE** and **META** elements determines the meta type of the corresponding key. If this meta key is defined elsewhere in Log Decoder (in another parser, a language file and so on) and the types do not match, then an error will be generated and parse rule could potentially be disabled.

Built-In Formats

Log Decoder Parse Rules also offer several built-in formats for the more commonly used types. These formats are:

Format	Description	Example
IPv4	ipv4	192.168.1.1
IPv6	ipv6	2607:f0d0:1002:51::4
MAC	physical Mac address	01:23:45:67:89:ab
UInt8	unsigned 8-bit integer	0 to 255
UInt16	unsigned 16-bit integer	0 to 65535
UInt32	unsigned 32-bit integer	0 to 4294967295
UInt64	unsigned 64-bit integer	0 to 18446744073709551615
Int8	signed 8-bit integer	-128 to 127
Int16	signed 16-bit integer	-32768 to 32767
Int32	signed 32-bit integer	-2147483648 to 2147483647
Int64	signed 64-bit integer	-9223372036854775808 to 9223372036854775807
Float32	decimal numbers	2.71818
Float64	decimal numbers	2.71818

Format	Description	Example
Email	valid email address	bob@company.com
URI	universal resource identifier	http://www.google.com/path/script?query=param
Hostname	RFC-1123 compliant hostname	abc.xzy.com