

LUX 2.7 Rule Form Creation Guide

Last Update: 5/18/18

1 Rule Form Creation	2
1.1 Adding a new rule form	2
1.2 Initial Rule Form Details	2
1.2.1 Format Description	2
1.3 Matching Fields (formFields)	3
1.3.1 Format Description	3
1.4 Using Drop Downs for Match values	4
1.5 Form Sections	6
1.5.1 Sample Form Section Configuration	6
2 Alerting within the Rule Form	7
3 Analytic Rule Form	7
4 Deleting and Updating Forms	8
5 Time Correlated Rule Forms	8
6 Conflicts with Existing Rule Forms	8

1 Rule Form Creation

1.1 Adding a new rule form

1. Go to LUX Administration Console
 - a. Under “Forms, Data stores, Templates” select the “Rule forms” link.
 - i. This will show a list of the current forms in the system.
2. Select the “Add” link at the top right-hand side of the page.
 - a. You will be taken to a default xml form definition with a few basic LUX features specified.
 - b. This guide will individually address fields taken from the following xml form :

```
C. <?xml version="1.0" encoding="UTF-8"?>
<form name="new" type="basic" xmlns="http://www.icgsolutions.com/lux/forms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.icgsolutions.com/lux/form
  http://www.icgsolutions.com/lux/form/luxform.xsd">
  <description>add description</description>
  <classification>UNCLASSIFIED</classification>
  <supportTimeRanges>true</supportTimeRanges>
  <streamName>DefaultStream</streamName>
  <formFields dataType="TEXT" fieldType="NORMAL" label="NewLabel" name="NameID"
  operator="textOperator"
  path="/rss/channel/item/lux:attribute[@name='NameID']" tooltip="NameID"/>
</form>
```

1.2 Initial Rule Form Details

1.

```
<form name="new" type="basic" xmlns="http://www.icgsolutions.com/lux/forms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.icgsolutions.com/lux/form
  http://www.icgsolutions.com/lux/form/luxform.xsd">
  <description>add description</description>
  <template default="false">defaultTemplate</template>
  <classification>UNCLASSIFIED</classification>
  <supportTimeRanges>true</supportTimeRanges>
  <streamName>DefaultStream</streamName>
```

1.2.1 Format Description

- a. **form name**: the display name of the rule form, enter a value other than “new”
- b. **type**: basic or tcr.
- c. **description**: description of the rule form or data source, this will be used in the ui form selection page.
- d. **supportTimeRanges**: Boolean value. If true then the rule form will include the option to control when the rule will execute.
- e. **template default**: links the ruleform to a predefined display template, thereby setting the rule alert layout within the UI.

- f. **streamName**: the name of the stream in the rule engine for which this rule will be used.

Editing a '0-Default Rule Form Template' rule.

* Name: Icon: Change Status: Enabled

Description

Description:

Tags:

Alert Title Format:

Classification

* Classification:

* Justification:

Data Group:

Rule Scheduling

Run this rule during specified time of day (Zulu):
 From: To:

Run this rule during specified days of the week:
 Sunday Monday Tuesday Wednesday Thursday Friday Saturday

Alerting

Send to E-mail? To: Add KML? True False

Send to alert list? Priority: Display:

Figure 1- Simple rule form with no match properties.

1.3 Matching Fields (formFields)

1. `<formFields dataType="TEXT" fieldType="NORMAL" name="NewID" operator="textOperator"path="/rss/channel/item/lux:attribute[@name='New ID']" label="New ID" tooltip="New ID"/>`

1.3.1 Format Description

- a. **dataType**: (**TEXT/NUMBER**) indicates to the engine how matching should be performed.
- b. **name**: the internal name for the field on the form.

- c. **operator**: the operator populates the match drop down. You can use one of the standard operator groups (textOperator,numOperator, geoOperator), or define your own using a data store.
- d. **path**: this is the path to the property in the event xml. It will generally take the form `${name}` or `/attributes/course_over_ground`.
 - i. When viewing a stream within the 'Stream Management' page, these paths can be found under the '**Path**' column.
- e. **label**: this is the display name of the field on the form. It can be the same as name, or a more user friendly version.
- f. **tooltip**: help label on mouse over.

The screenshot shows a web interface for editing a rule. At the top, it says 'Editing a '0-Default Rule Form Template' rule.' Below this, there are several sections:

- Name:** A text input field containing '0-Default Rule Form Template'. To its right is an 'Icon' field with a red diamond icon and a 'Change' link, and a 'Status' dropdown menu set to 'Enabled'.
- Description:** A section with a 'Description' text area (placeholder: 'Provide a description for your rule (optional)'), a 'Tags' text area (placeholder: 'i.e; Tag1, Tag2, Tag3, etc..'), and an 'Alert Title Format' text area (value: 'rule:name').
- Classification:** A section with a '* Classification:' dropdown menu (placeholder: 'Select a classification level for this rule.'), a '* Justification:' text area (placeholder: 'Provide a justification for your classification.'), and a 'Data Group:' text area.
- TopLabel:** A dropdown menu and a text input field with 'Add' and 'Remove' buttons.
- Rule Scheduling:** A section with a dropdown menu.
- Alerting:** A section with 'Send to E-mail?' (checkbox), 'To' (text input), 'Add KML?' (radio buttons for True/False), 'Send to alert list?' (checkbox), 'Priority' (dropdown menu with value 5), and 'Display' (dropdown menu with value (default)).

Figure 2 - Adding single match field to form.

1.4 Using Drop Downs for Match values

Drop downs can be used in place of text input for the match values to do this

- a. Create a data store with the match options. (see data store section for how to do this)
- b. Add the optional **dataProvider** attribute to the correct Match Field

```
<formFields name="field 2" operator="textOperator"
path="/rss/channel/item/lux:attribute[@name='test_dropdown']" label="Display Name field
2" dataType="TEXT" tooltip="some tool tip help" dataProvider="myValues"/>
```

The screenshot shows a web interface for editing a rule. At the top, it says "Editing a 'sample form 1' rule." Below this are several sections:

- Name:** A text input field with the placeholder "Enter a name for your rule." To its right is an "Icon" field with a location pin icon and a "Change" link, and a "Status" field set to "Enabled".
- Description:** A section with three sub-fields: "Description:" (text input with placeholder "Provide a description for your rule (optional)."), "Tags:" (text input with placeholder "i.e; Tag1, Tag2, Tag3, etc.."), and "Alert Title Format:" (text input with placeholder "##[{\$rule.name}]##").
- Classification:** A section with two sub-fields: "Classification:" (a dropdown menu with the text "Select a classification level for this rule.") and "Justification:" (text input with placeholder "Provide a justification for your classification.").
- Display Name:** A section with a dropdown menu, a text input field, and "Add Remove" links.
- Display Name field 2:** A section with a dropdown menu, a text input field, and "Add Remove" links. A dropdown menu is open below the text input, showing the items "apple", "bannana", and "carrot".
- Rule Scheduling:** A section with a dropdown menu.
- Alerting:** A section with a dropdown menu.

At the bottom right, there are "Save" and "Cancel" buttons.

Figure 3 - adding field with match values to current form definition.

Adding validation to form fields

- validator** – optional. Can be one of **EMAIL**, **EMAILLIST**, **REGEX**, or **NUMERIC**
- validationRegex**- optional. Required if **validator = REGEX**, this should be a java regular expression used to validate user input

```
<formFields name="field 2" operator="textOperator"
path="/rss/channel/item/lux:attribute[@name='test_dropdown']" label="Display Name field 2"
dataType="TEXT" tooltip="some tool tip help" dataProvider="myValues" validator="NUMERIC"/>
```

In the above example we are adding a numeric validator, which will require that the input not contain any characters.

```
<formFields name="field 2" operator="textOperator"
path="/rss/channel/item/lux:attribute[@name='test_dropdown']" label="Display Name field 2"
dataType="TEXT" tooltip="some tool tip help" dataProvider="myValues" validator="REGEX"
validationRegex="[a-z][a-z]"/>
```

In the above example we are adding a regex validator with a regular expression, which will require that the input be limited to 2 lower case letters.

1.5 Form Sections

Fields on the form can be grouped into named groups for display using the `formSections` element. The `formSections` element can contain one or more `matchFields`.


```
<formSections>
  <name>General</name>
  [1 or more matchFields]
</formSections>
```

1.5.1 Sample Form Section Configuration

- Example configuration to update the form .xml to include `Group One` and `Group Two`:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <Form name="sample form 1" type="basic">
    <description>sample form description</description>
    <supportTimeRanges>true</supportTimeRanges>
    <streamName>add stream name</streamName>
    <formSections>
      <name>Group One</name>
      <formFields name="display name" operator="textOperator"
        path="/rss/channel/item/lux:attribute[@name='test']" label="Display Name" dataType="TEXT"
        tooltip="some tool tip help" />
    </formSections>
    <formSections>
      <name>Group Two</name>
      <formFields name="field 2" operator="textOperator"
        path="/rss/channel/item/lux:attribute[@name='test_dropdown']" label="Display Name field
        2" dataType="TEXT" tooltip="some tool tip help" dataProvider="myValues"/>
    </formSections>
  </Form>
```

Editing a '0-Default Rule Form Template' rule.

* Name: Icon:  Change Status:

▲ Description

Description:

Tags:

Alert Title Format:

▲ Classification

* Classification: ▼

* Justification:

Data Group:

▲ Group One

Display Name ▼ Add Remove

▲ Group Two

Display Name field 2 ▼ ▼ Add Remove

▼ Rule Scheduling

▲ Alerting

Send to E-mail? To Add KML? True False

Send to alert list? Priority Display ▼

Figure 4 - Form with Form Sections

2 Alerting within the Rule Form

The method of which an alert is configured within the LUX UI is determined by the display template. These can be configured within the LUX Admin Portal, under “Display Template”, by referencing the xml schema and LUX help documentation.

3 Analytic Rule Form

Within LUX, a rule form must be manually defined for each analytic. On the manage analytic screen, once an analytic is “viewed”, there is a helper function that will allow you to create a starter analytic rule form. This starter analytic rule form will use the contents of the analytic.xml form and the properties posted to LUX from the engine.

4 Deleting and Updating Forms

- a. Within LUX, you can delete a rule form from the form management page. However the rule form must not be in use in order to be deleted.
- b. When updating a form from the admin pages, you will need to refresh your LUX ui window in order to load the new form configuration.

5 Time Correlated Rule Forms

Starting within LUX, time correlated rule (TCR) forms must be defined like other rule forms, and the system can support multiple TCR rule forms. This allows the definition of custom TCR forms, for example limited to AIS data, in addition to the generic TCR form.

```
<timeCorrelatedField commonStore="aisTcrMatchProps" countDataProvider="x"
ruleFilter="AIS Rule" uniqueStore="aisTcrMatchProps"/>
```

- a. `commonStore` – name of the data store to use for the common match properties
- b. `uniqueStore` – name of the data store to use for the unique match properties
- c. `ruleFilter` – optional comma separated list of rule forms to include in the rule drop down
- d. `countDataProvider` – required field.

6 Conflicts with Existing Rule Forms

Changes to a rule form, such as changing a rule form definition, can cause issues for existing rules.

- a. EX) If a rule form is updated to reference a new stream name, any existing rules based on the rule form will continue to reference the old stream name, until which point they are edited and saved anew.

This is also true for any attributes that are stored in the rule syntax (stream Name, xpaths, data types, names); however, this does not apply to form specific data that is not stored in the rule syntax (display names, dropdown contents, ordering, adding new fields).