



Purpose

Purpose here is to get to know different ways of backing up a validation rule, validation ruleset or validation project.

Description

Multiple ways exist to make backups and duplicates of the content we have created with Studio. When certain rule is made in a project and there's a need to reuse it in another project, it may be worth considering to copy/paste it directly, especially if the validation rule is complicated

Following is done here:

1. Copying a rule
2. Copying a ruleset
3. Backing up project
4. Duplicating project
5. Deleting project

This section concludes the advanced training stage.

Instructions

1: Copying a rule

At this point, we should have a rule limiting value of Transaction/Id to be either "Id1" or "Id2". Goal of this phase is to have this logic to be applied to Header/Id as well.

We can view the rule as XML, make a slight modification to the rule and then paste the rule as a ruleset, to minimize the amount of effort.

First, we'll locate our rule, open it and view it as XML. Button for viewing the rule as XML is in the upper right corner of rule modal:

Transaction_Id_valueRestrictions

(Antero's ruleset)

[Rule XML view](#)

Details Actions

MIG details: >

Rule basic details: ▾

Active

Name* Transaction_Id_valueRestrictions

Internal note Optional internal note of the rule

Wiki <http://wiki.xmlvalidation.com/> XMLdation wiki page

Rule location XPath /Message/Transaction/Id

Validation definition: ▾

Severity* ▲ Error ▾

Context* TransactionType1

Clicking the button will lead us to following-looking XML

Export rule XML

XML as text

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <myXMLExport xmlns="http://www.xmlvalidation.com" xmlns:default="http://www.xmlvalidation.com" >
3 <Header>
4   <exportedBy>testiantero</exportedBy>
5   <exportDateTime>2018-01-08T14:25:40+00:00</exportDateTime>
6   <sourceEnvironment>myxmldemo.xmlvalidation.com</sourceEnvironment>
7   <exportOriginRuleset>Antero's ruleset</exportOriginRuleset>
8 </Header>
9 <rules>
10 <rule>
11   <name>Transaction_Id_valueRestrictions</name>
12   <severity>ERROR</severity>
13   <isActive>true</isActive>
14   <description><![CDATA[]]></description>
15   <context>TransactionType1</context>
16   <ruleOcl><![CDATA[self.Id.matches("Id1") or self.Id.matches("Id2")]]></ruleOcl>
17   <message><![CDATA[Value of 'Id' does not match the pattern 'Id1'.]]></message>
18   <wikiAddress></wikiAddress>
19   <migData>
20
21

```

XML as file Save as file...

Close Save

If we wanted, we could make changes here directly as well. However, in this case we want to copy everything here and paste it as another rule. We'll select all of the content, and click copy

The screenshot shows the 'Export rule XML' dialog box. The XML code is as follows:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <myXMLExport xmlns="http://www.xmldata.com" xmlns:default="http://www.xmldata.com" />
3 <Header>
4   <exportedBy>testiantero</exportedBy>
5   <exportDateTime>2018-01-08T14:26:52+00:00</exportDateTime>
6   <sourceEnvironment>myxmldemo.xmldata.com</sourceEnvironment>
7   <exportOriginRuleset>Antero's ruleset</exportOriginRuleset>
8 </Header>
9 <rules>
10 <rule>
11   <name>Transaction_Id_valueF</name>
12   <severity>ERROR</severity>
13   <isActive>true</isActive>
14   <description><![CDATA[ ]]></description>
15   <context>TransactionType</context>
16   <ruleOcl><![CDATA[self.Id.n
17   <message><![CDATA[Value of
18   <wikiAddress></wikiAddress>
19   <migData>
20
21

```

The context menu is open, showing options: Cut (Ctrl+X), Copy (Ctrl+C), Paste (Ctrl+V), Paste as plain text (Ctrl+Shift+V), Select all (Ctrl+A), Print... (Ctrl+P), Spellcheck, Writing Direction, AdBlock, and Inspect (Ctrl+Shift+I). The 'Copy' option is highlighted with a red line.

Next, we shall choose a ruleset we want to have the new rule to be copied. We can select the normal, non-group ruleset here for example by pressing the settings icon and selecting "Import rule".

The screenshot shows the settings menu for 'Antero's ruleset'. The menu items are: Edit, Delete, Copy to library, Change type to, Ruleset XML view, and Import rule. The 'Import rule' option is highlighted with a red box.

A new modal will open where we can paste the rule we copied earlier.

However, after pasting, our rule is exactly the same as the original. Therefore, we shall make three modifications to the XML directly

```
<name>Transaction_Id_valueRestrictions</name>
needs to be changed to
<name>Header_Id_valueRestrictions</name>
```

Otherwise we would have two rules with the same name, which is not valid.

```
<context>TransactionType1</context>
needs to be changed to
<context>HeaderType1</context>
```

This ensures we are targeting the correct element.

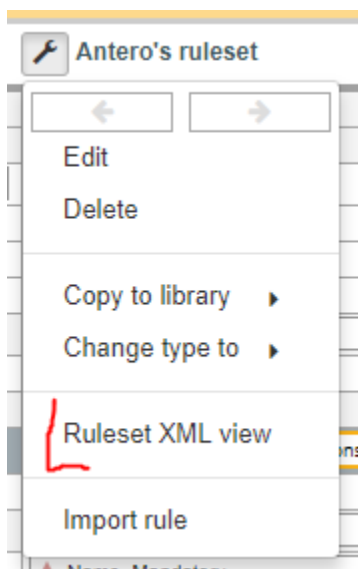
```
<ruleLocationXPath>/Message/Transaction/Id</ruleLocationXPath>
needs to be changed to
<ruleLocationXPath>/Message/Header/Id</ruleLocationXPath>
```

ruleLocationXPath is not used in rule execution, it is only used in where the rule is displayed in Studio. We should change this to be in the correct element, in Header instead of in Transaction.

After this is done, and when pressing save, we should now see the new rule in it's relevant location.

2: Copying a ruleset

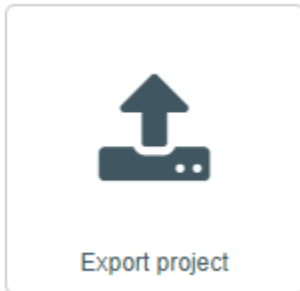
We can do the same to the whole ruleset as well, by selecting ruleset XML view.



This allows us to copy the ruleset elsewhere, or to save the ruleset to user's computer to make a local backup.

3: Backing up project

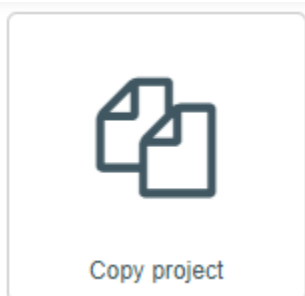
We can use the same logic for the whole project as well, this setting is under "project dashboard" -> export project



4: Duplicating project

Since the export project will be exporting the project exactly as it is, we cannot use it directly to make duplicates of a project. This is because the unique Id Studio assigns to a project will be the same if we try to do so.

In order to quickly duplicate a project, we can use copy project within dashboard:



This open following modal:

Create copy of project: ×

XMLdation myXML training > XMLdation reference project

Project group * +

Name *

Which allows us to set the destination and the name for the new project. This feature will automatically change Id's to be unique so no further tinkering is required when using this.

We can press the save here, so in next section we'll have a project to delete.

5: Deleting project

The project we created in previous phase should now be opened. It will also be listed in the left-hand-side navigation

XMLdation myXML training / Copy of: XMLdation reference project

Copy of: XMLdation reference project

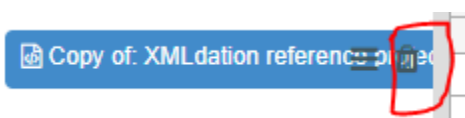
Project dashboard | Validation ru

Schema element

Message ▾
Header ▾
Id
TimeStamp

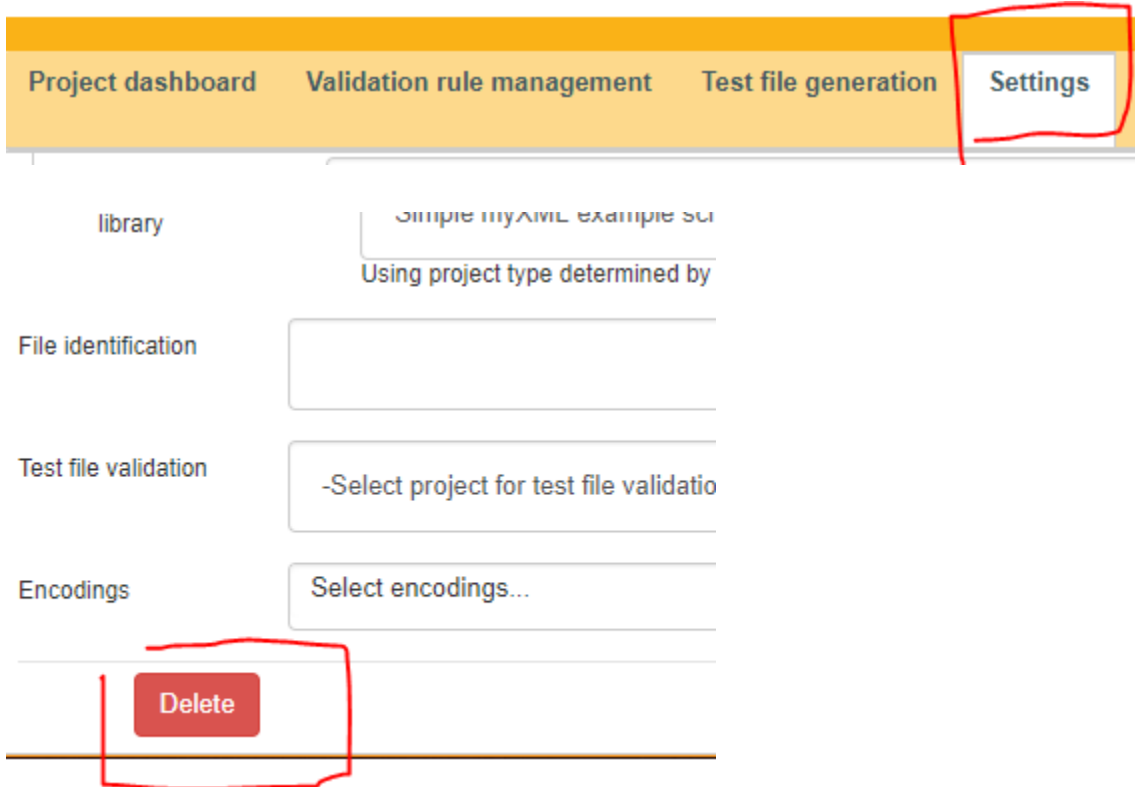
This navigation panel can be opened by clicking "project list"

There are two ways to delete a project, either by clicking trash bin here in the project list



(which in this case is somewhat hidden)

Or alternatively within project settings once the project is open



We should delete the project by using either of these options to keep the list more tidy. Whenever deleting content it's of course a good idea to be careful. As extra caution, backup of the project can be created first before deleting anything, good idea especially if there is actual and important validation rules present in the project.

This was the final section of the advanced training content. Next up will be expert level stuff, which should be fairly doable as you have made it this far.