

Resco Mobile CRM

Woodford (Rules Guide)

Document version 6.2.0.0

Resco.net



| | |
|--|----|
| 1 Form Rules..... | 2 |
| 2 Steps | 2 |
| 2.1 Function Step..... | 2 |
| 2.1.1 Selecting Variable, Use of the Property Selector | 3 |
| 2.1.2 Function Types..... | 3 |
| 2.1.3 Defining Argument | 5 |
| 2.2 Defining Branch Groups and Conditions | 5 |
| 2.2.1 Condition Editor | 6 |
| 2.2.3 Adding More Conditional Branches..... | 7 |
| 2.3 Defining Variables..... | 8 |
| 2.3.1 Editing Variable | 10 |
| 2.3.2 Deleting Variable | 10 |
| 2.3.3 StringList type | 10 |
| 2.4 Rule Validation | 11 |
| 3 Disabling the rule..... | 16 |
| 4 Rule Events | 16 |
| 5 Examples..... | 17 |
| Error Message | 17 |

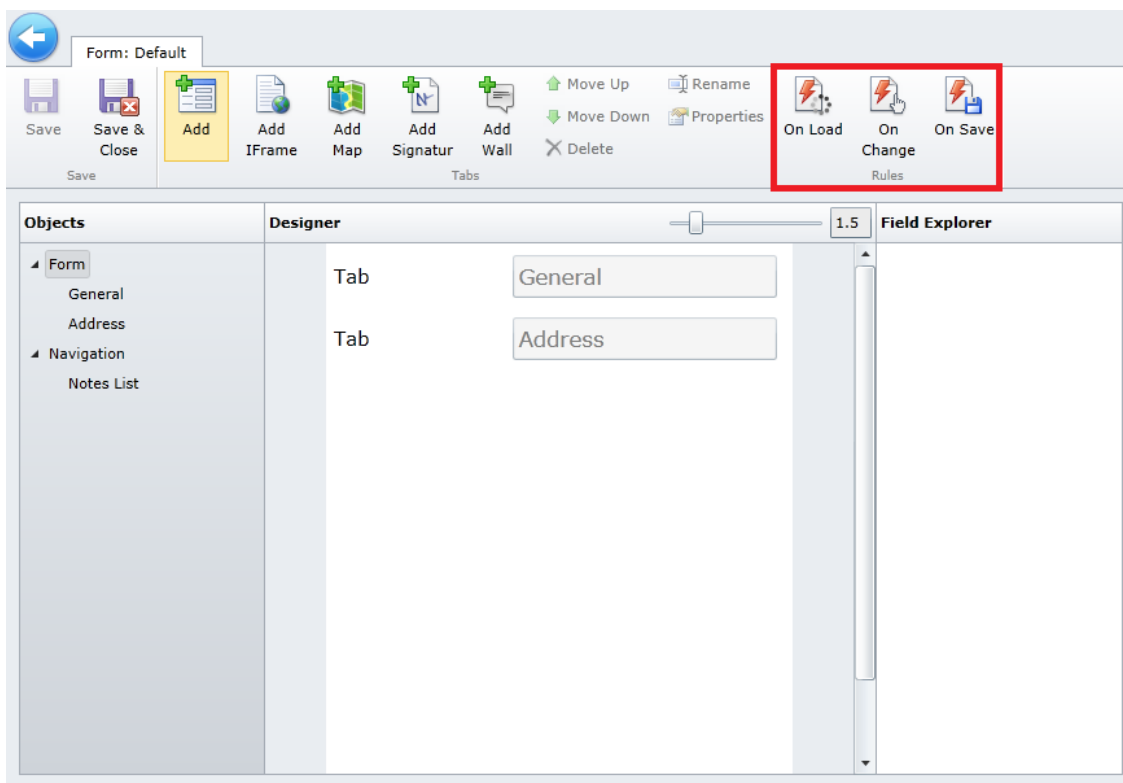
1 Form Rules

Form Rules describe sequences of steps that are executed on form related events *OnLoad*, *OnSave*, *OnChange*. These steps allow us to further customize edit form's UI (hide or disable form fields, assign values to them). Rules are bound to form and the main entity. To create a form rule open a Rule Editor by clicking on corresponding event button.

OnLoad rule is executed when the Edit Form is opened

OnChange rule is executed on opening of the Edit Form and also each time a value of the form's fields are changed

OnSave rule is executed when the Save button is selected.

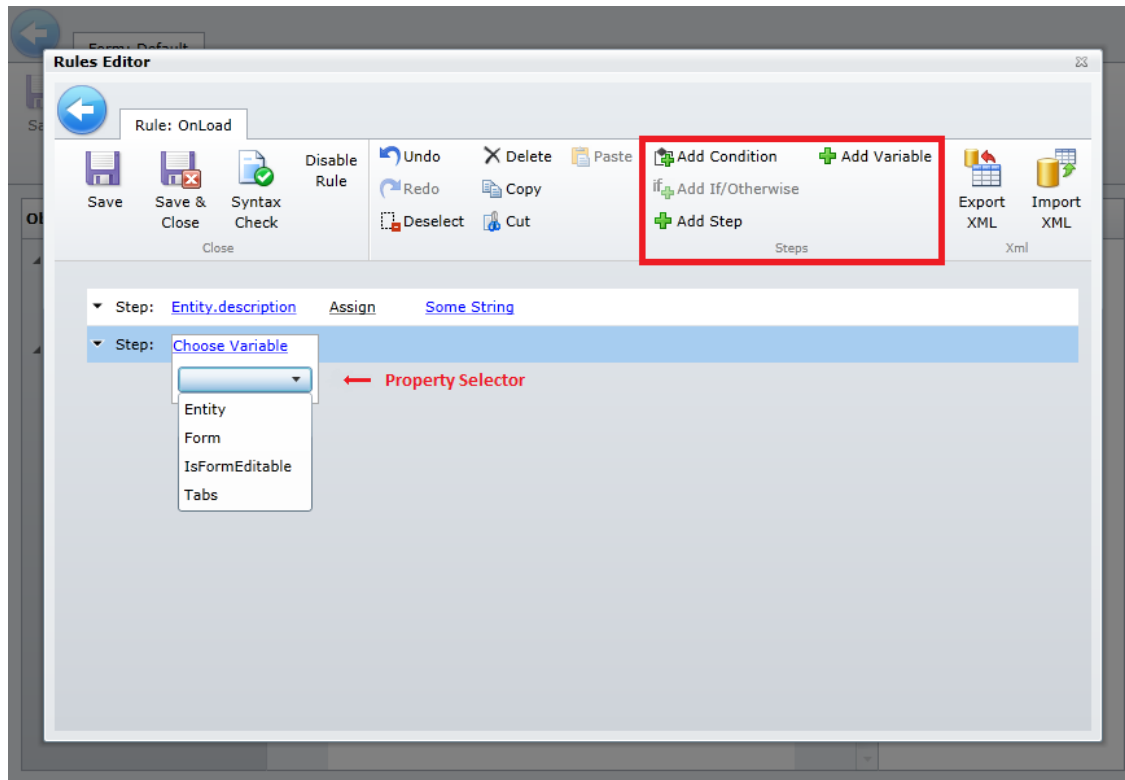


2 Steps

Each rule consists of steps. Every type of step can be described as an expression with the left side that we will refer as *variable*, right side which we will refer as *argument* and an *operation* (like function or comparison) between them. We can create these types of steps *Function Step*, *Condition step* and *Variable step*.

2.1 Function Step

To add a function step click on "Add step" button. Function step sets value of selected variable by using function.



2.1.1 Selecting Variable, Use of the Property Selector

Click on “Choose Variable” to open Property Selector. Property Selector allows us to choose from *Entity properties*, *Form fields*, event specific properties like *IsFormEditable* and *Tabs* which represents form views. Remember that if you want to modify value of a form field, you need to assign the value to entity property. Form properties are used only to hide/show, disable/enable or set an error message for form fields. All values of property selector need to be set to continue defining step.

After we have selected variable we can select function and argument.

2.1.2 Function Types

Each variable data type will have different set of functions. See the table for details.

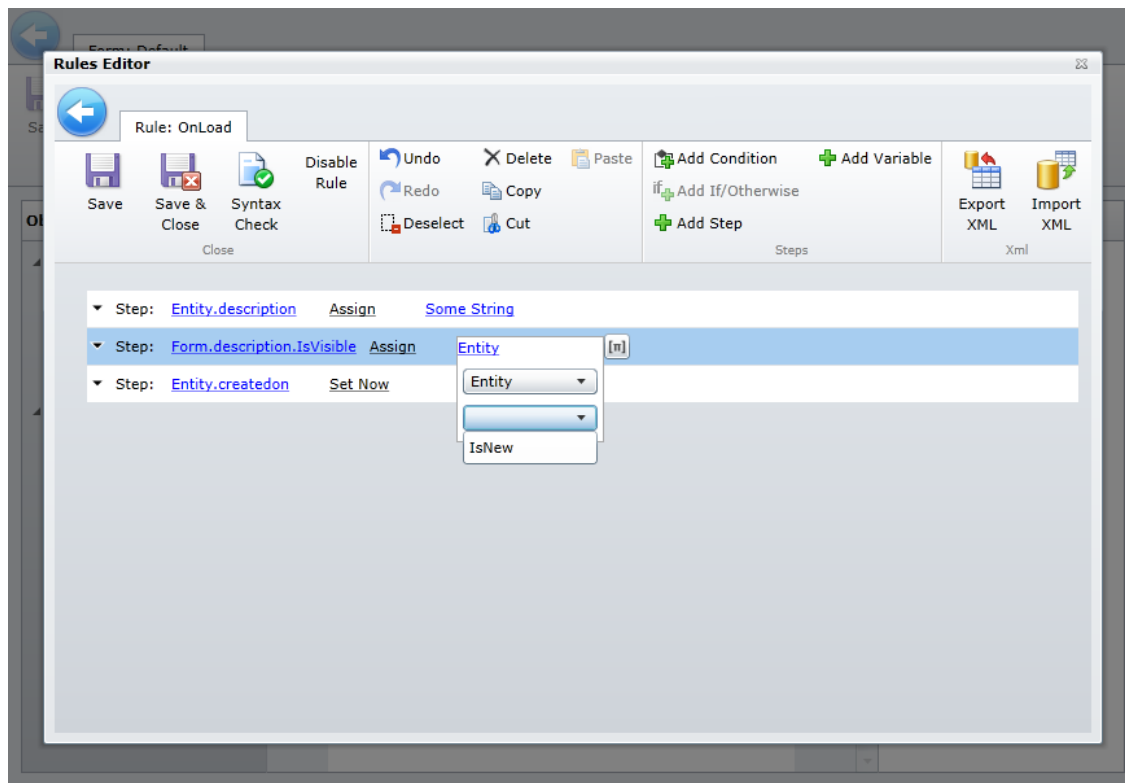
| Function Name | Description | Variable data types | Argument data types |
|----------------|---|-----------------------------|--|
| Assign | Assigns value to variable | Text, Number, Boolean, Date | Variable data type, all types for text variable |
| Clear | Assigns null or empty value to variable | Text, Number, Boolean, Date | |
| Convert | Converts value from string | Number, Boolean, Date | String value |
| Negate | Returns negated value of argument | Boolean | Boolean |
| AddSufix | Adds argument value as suffix to variable | Text | Text, Number, Boolean, Date |
| AddPrefix | Adds argument value as prefix to variable | Text | Text, Number, Boolean, Date |
| Concat | Returns concatenated string from two arguments | Text | 2 args: Text, Number, Boolean, Date |
| Concat3 | Returns concatenated string from three arguments | Text | 3 args: Text, Number, Boolean, Date |
| Format | Returns formatted string from StringList | Text | Arg1: Text Arg2: StringList |
| Addition | Returns addition of arguments | Number | Arg1:Number, Arg2:Number |
| Difference | Returns difference between first and second arguments | Number | Arg1:Number, Arg2:Number |
| Multiplication | Returns product of arguments | Number | Arg1:Number, Arg2:Number |
| Division | Returns division of arguments | Number | Arg1:Number, Arg2:Number |
| SetNow | Assigns current time to variable | DateTime | |
| SetToday | Assigns current day to variable | DateTime | |
| SetTomorrow | Assigns tomorrow to variable | DateTime | |
| AddMinutes | Returns DateTime argument plus X minutes | DateTime | Arg1:DateTime, Arg2:Number |
| AddHours | Returns DateTime argument plus X hours | DateTime | Arg1:DateTime, Arg2:Number |
| AddDays | Returns DateTime argument plus X days | DateTime | Arg1:DateTime, Arg2:Number |
| AddWeeks | Returns DateTime argument plus X weeks | DateTime | Arg1:DateTime, Arg2:Number |
| AddMonths | Returns DateTime argument plus X months | DateTime | Arg1:DateTime, Arg2:Number |
| AddYear | Returns DateTime argument plus X years | DateTime | Arg1:DateTime, Arg2:Number |
| AddItem | Adds argument to end of list. | StringList | Arg1:Number, Boolean, String, DateTime |
| InsertItem | Adds argument to specified index. Index starts at 0. | StringList | Arg1:Number, Boolean, String, DateTime Arg2: Number |

| | | | |
|---------------|--------------------------------------|---------------------|---------------------|
| GetItem | Returns item at specified index. | StringList | Number |
| RemoveItem | Removes item at specified index | StringList | Number |
| ClearItems | Removes all items from list variable | StringList | |
| Function Name | Description | Variable data types | Argument data types |

Text = String, Memo, Number = Integer,Float, Decimal, Number

2.1.3 Defining Argument

After selecting function argument data type will be set. Argument can be constant value or variable. By pressing [x] button u switch from constant editor to property selector which allows you to select any variable. Note that Property Selector displays only properties with correct type for function.

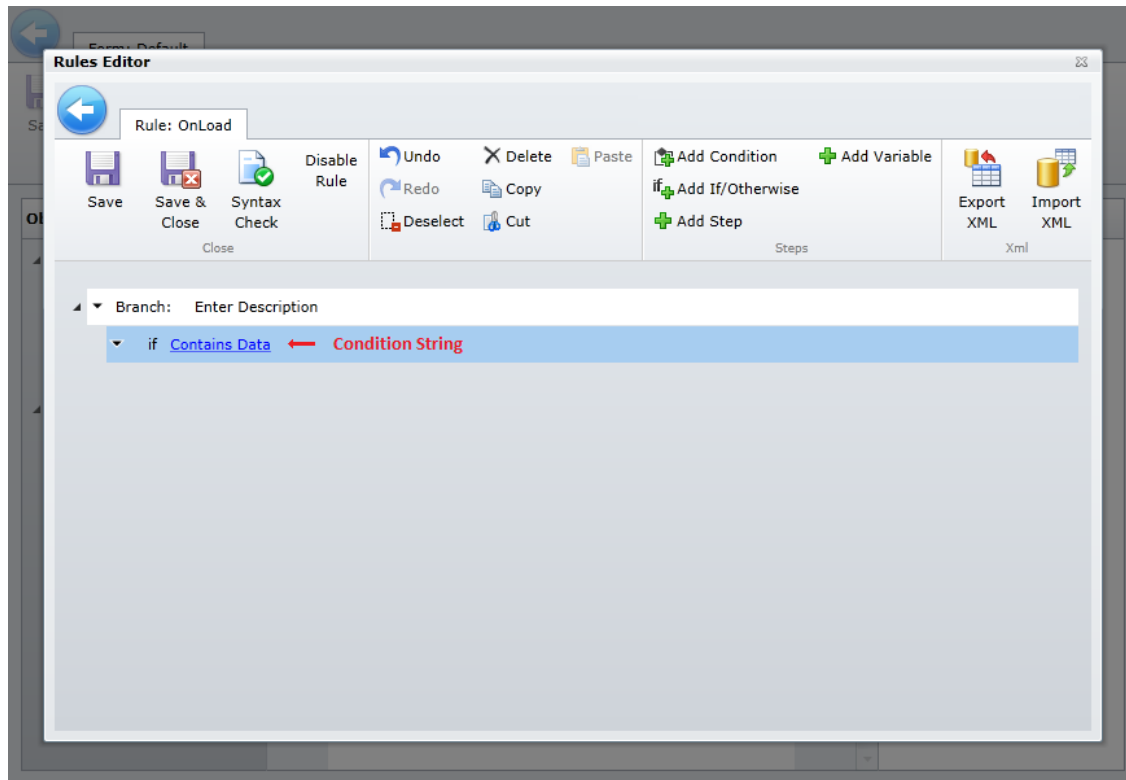


2.2 Defining Branch Groups and Conditions

Click "Add condition" to add new branch group into rule. As you can see in the picture two lines were added into editor.

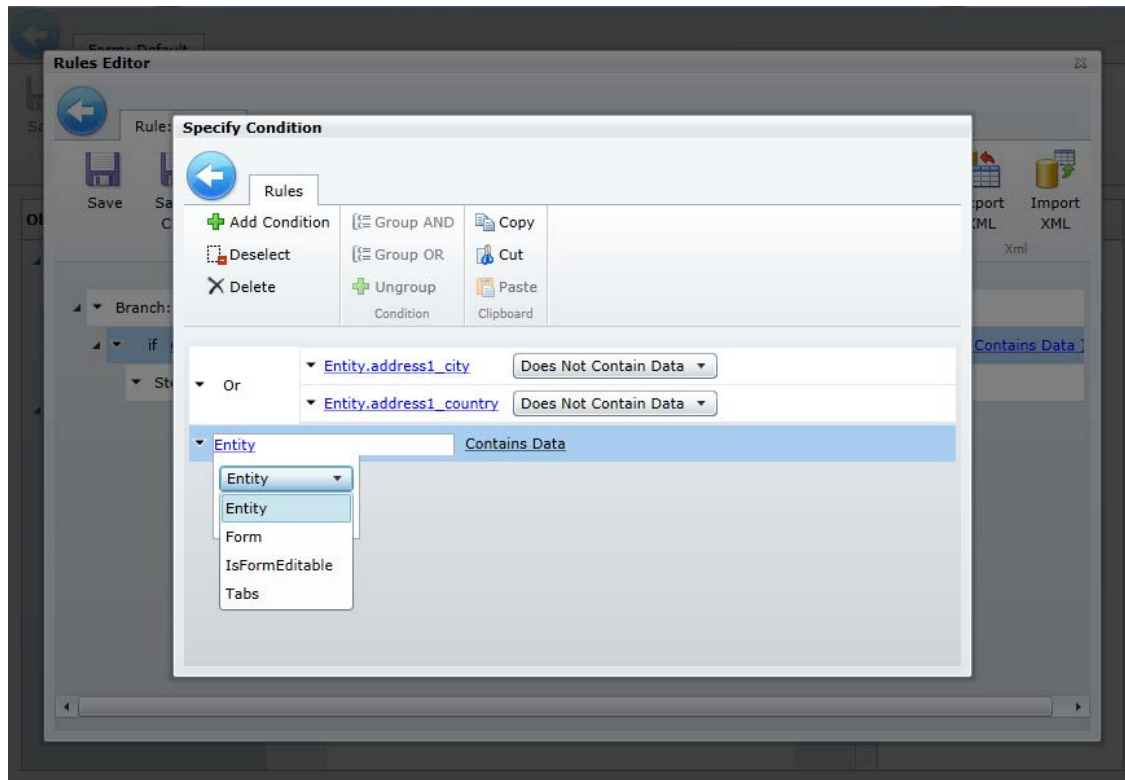
First line is labeled "Branch", defines **branch group** that can contain one or many conditional branches. For better orientation in rule you can add description to branch.

Second line defines empty conditional branch. You can see that first condition contains label "if" and condition string. Conditional branch contains condition and steps that are executed only when condition is valid. If you want to add step into conditional branch you have to select the conditional branch. To define condition, click on the condition string. This will open **Condition Editor**.



2.2.1 Condition Editor

In condition editor we can define condition steps (same as [2.1 Function Step](#)) and group them into “Or”, “And” groups. Conditions by default are grouped in “And” group. To add conditions into group, you have to select at least two conditions or groups on the same level (using Ctrl), or in the same group and click button “Group And” or “Group Or”.

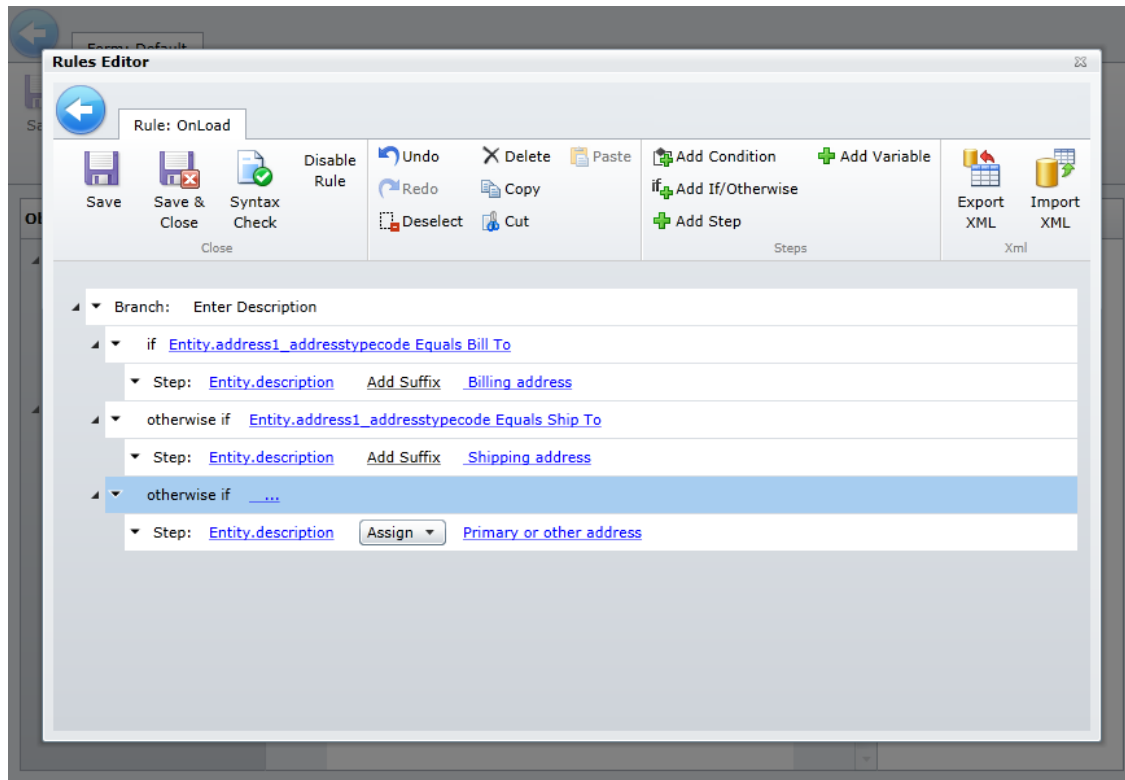


When you are done with creating condition, click the Back button and the condition will be saved.

2.2.3 Adding More Conditional Branches

To add more conditional branches click "Add If/Otherwise" button. The button is enabled only if Branch or Conditional Branch is selected. Conditional branches after first one are labeled "otherwise, if".

Remember that the first branch has to define condition and the condition for last branch is optional. If last branch doesn't define condition it is always evaluated as valid. Also note that execution of branches is exclusive. It means maximally one conditional branch in branch group can be executed.

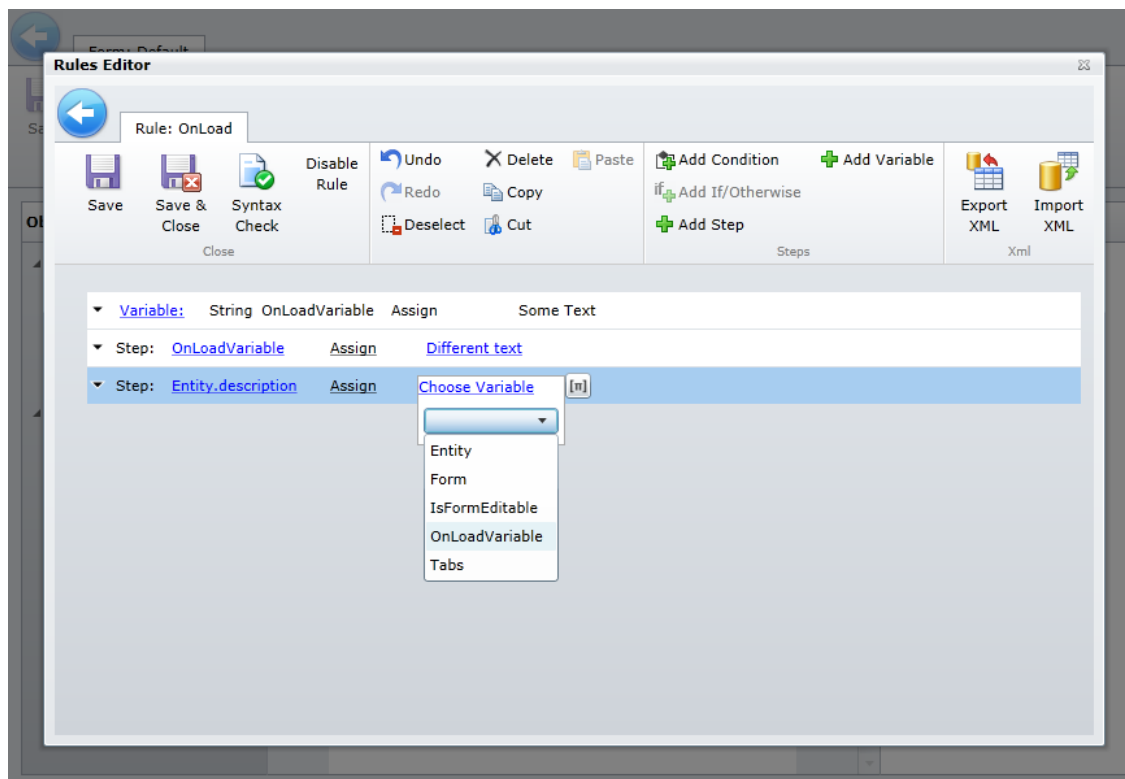
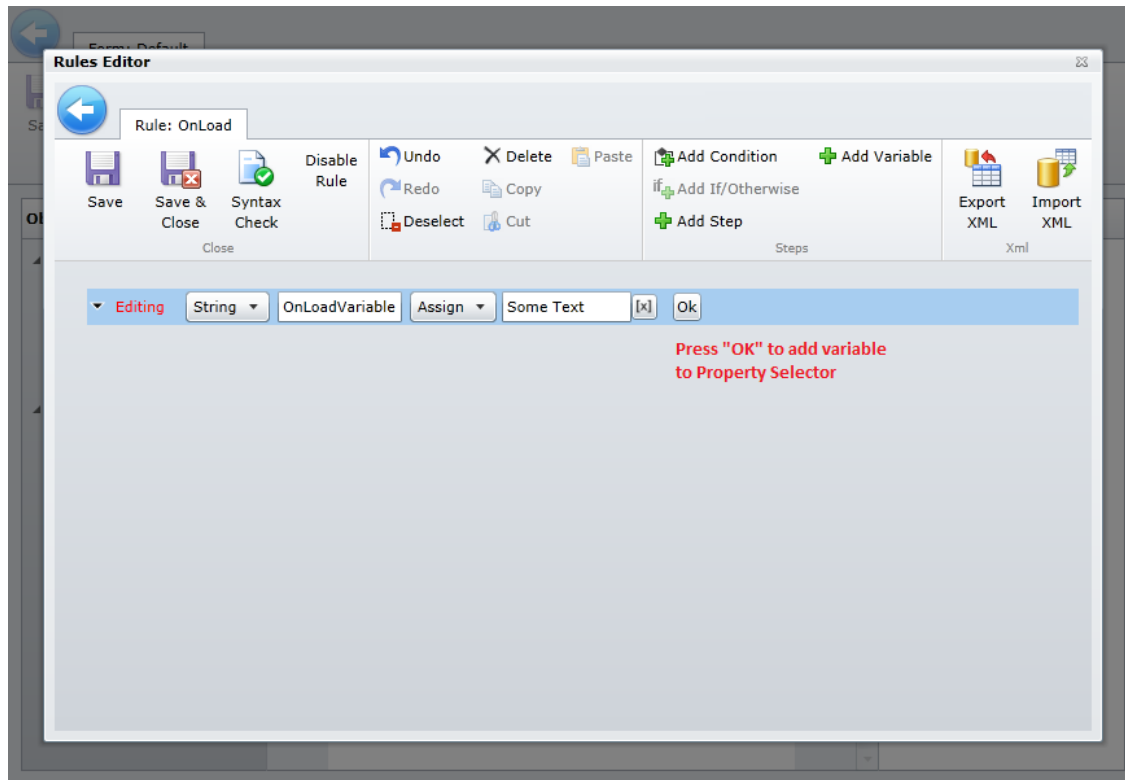


2.3 Defining Variables

By clicking "Add Variable" step you can define a variable that is added into the **Property Selector** of current rule. This variable can be only used in current rule. Defining a variable is done in two steps. As you can see in the picture, the step is currently in *editing state*. After you define a step (same as [2.1 Function Step](#)) you have to click the "Ok" button at the end of line. Only then the temporary is added into **Property Selector**.

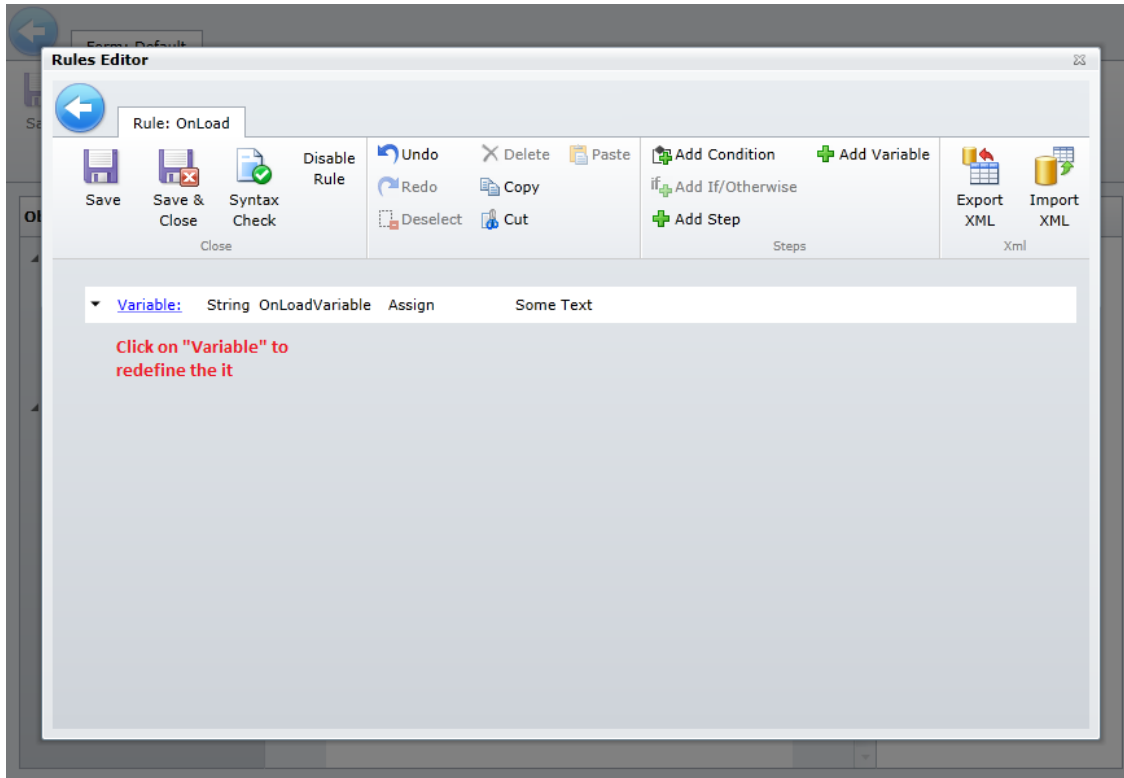
Remember that you can only use variable after the step where it was defined.

Please note that when editing or creating variable, tree and ribbon are disabled, until you press button Ok or Cancel. Ok adds the variable to workflow. Cancel just cancels creation or changes made to variable.



2.3.1 Editing Variable

Click on "Variable" to open the editing state. Now you can redefine the variable. If you have already used the variable in rule, changing the name will automatically update variable name everywhere it was used. However changing the type will reset all steps, where the type was used.

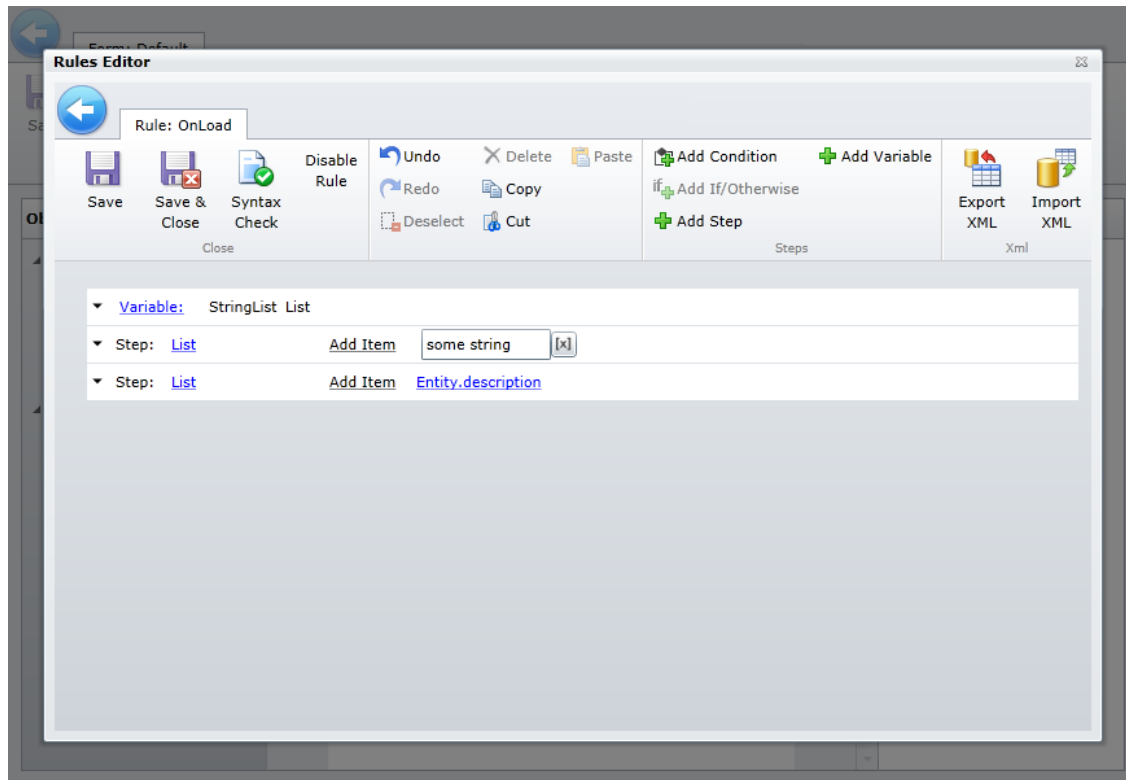


2.3.2 Deleting Variable

When you delete the variable all steps that are using the variable on the right side will be also deleted. Steps using the variable on the left side will reset and you have to define them again.

2.3.3 StringList type

StringList type is special type that you can define. It represents a list of variables. In editing state you can't initialize it right away, you have to add items in the StringList using *Function Step.StringList*. See the function type table for more details.

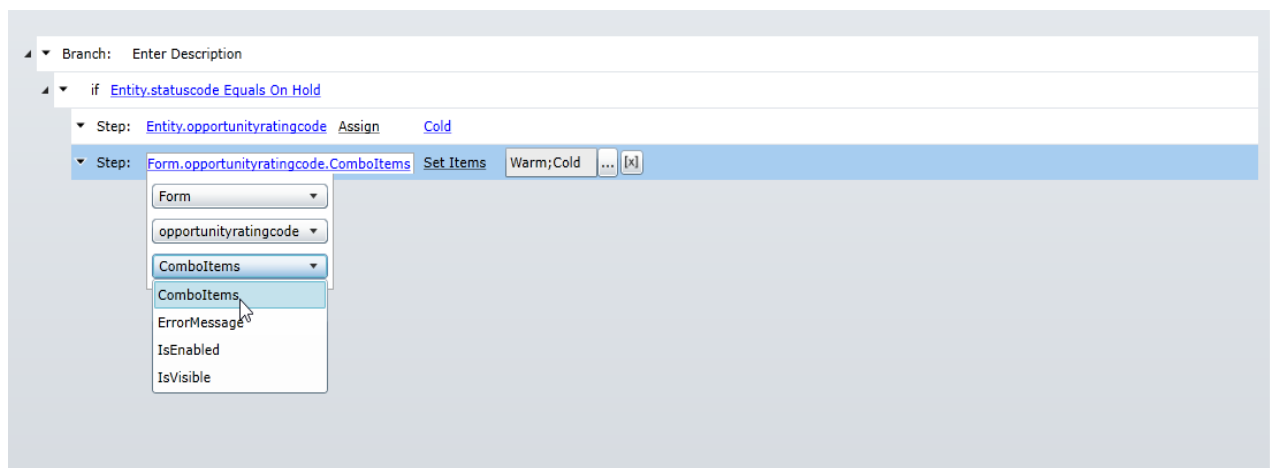


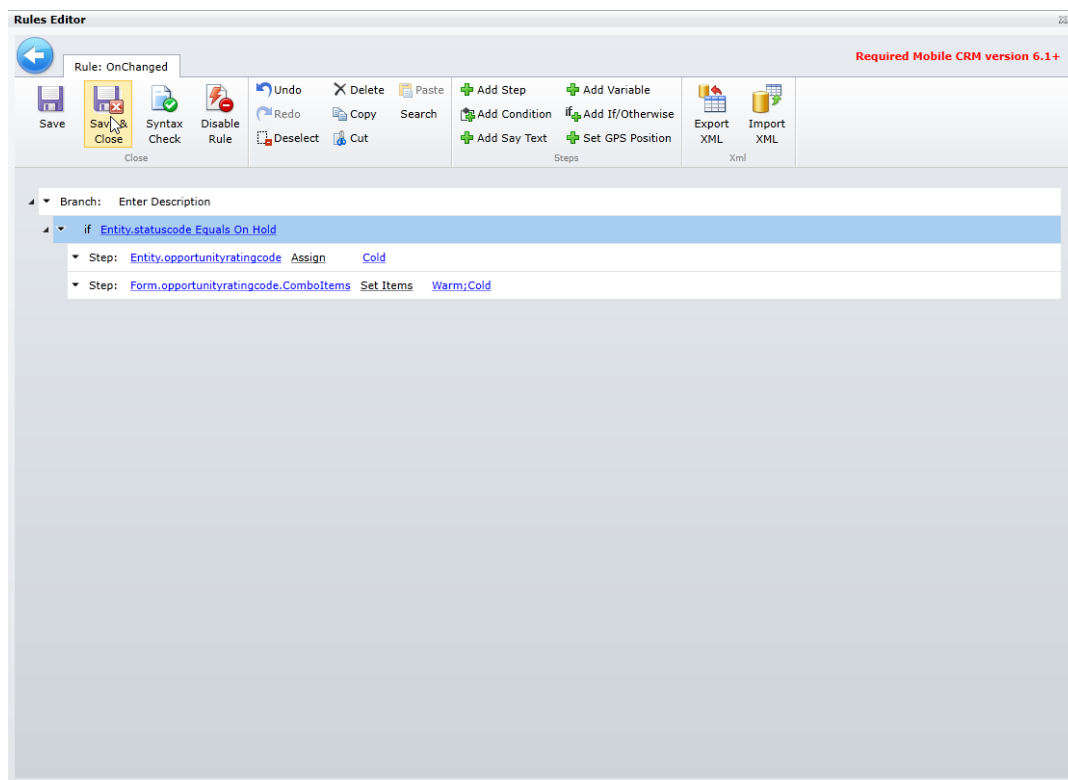
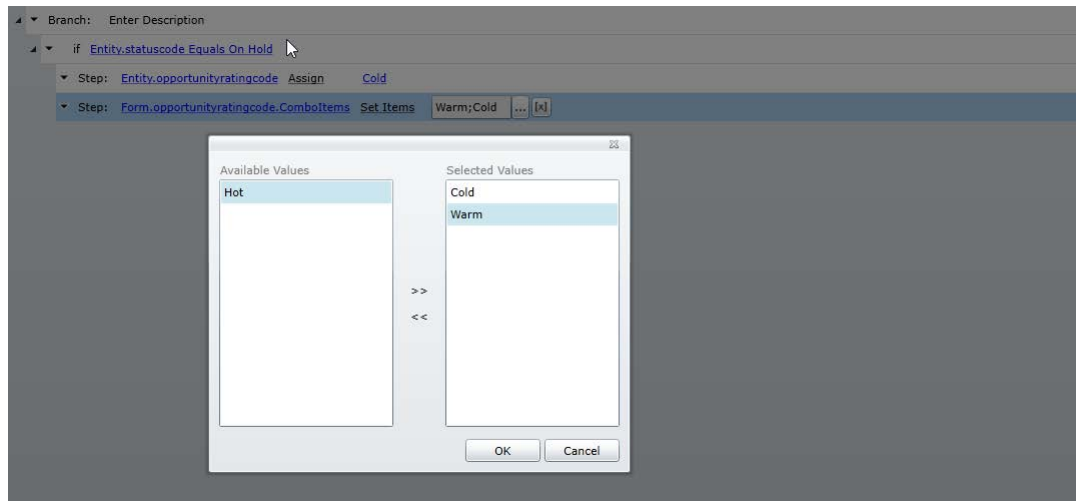
2.3.4 Option set filtering

New property on OptionSet types named ComboItems

Function SetItems to set which items should be available.

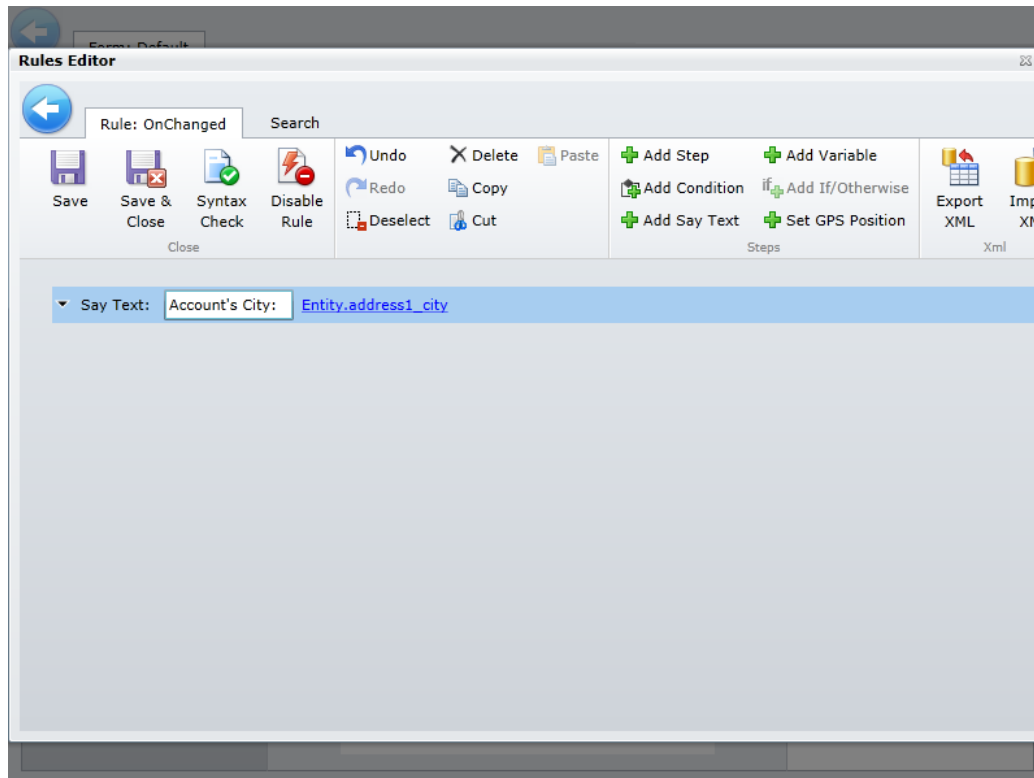
Please note that when you set options when condition is met in OnChange event, you had to define also which options are visible when condition is not met.





2.3.5 Set Gps

Click on Set Position from ribbon. Select longitude and latitude fields. The arguments are precision and age.



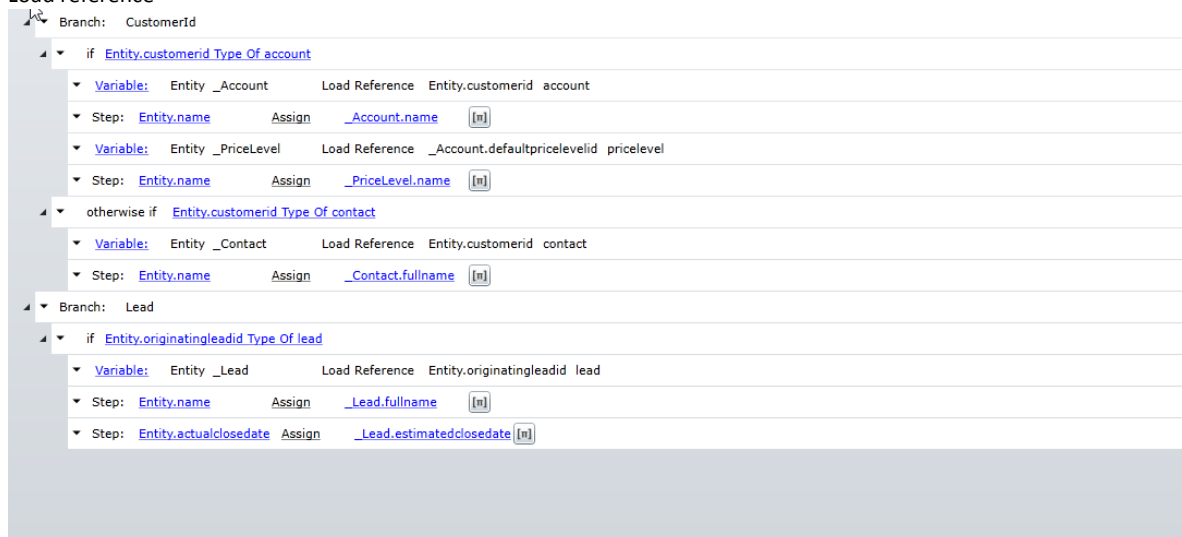
2.3.6 Load lookup

First you need to check the type of entity lookup is pointing to. That is done via condition step with operator Type Of.

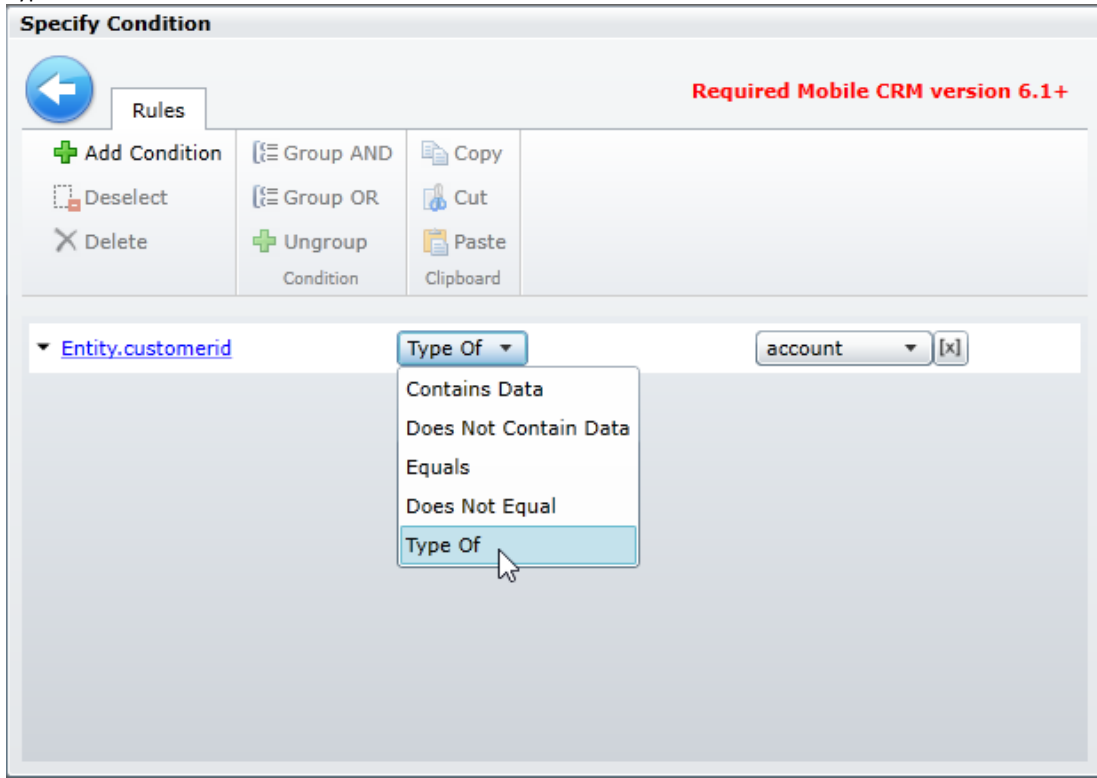
Load lookup is enabled via Temporary Variable. Select type Entity and function Load Reference and set arguments.

Please note that after editing temporary variable of type Entity all steps using this variable will reset.

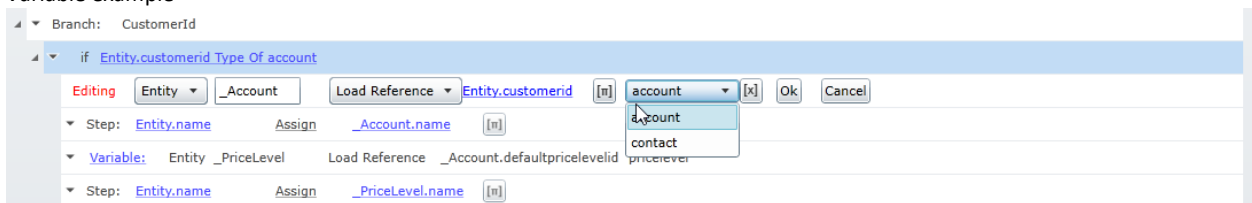
Load reference



Type of condition

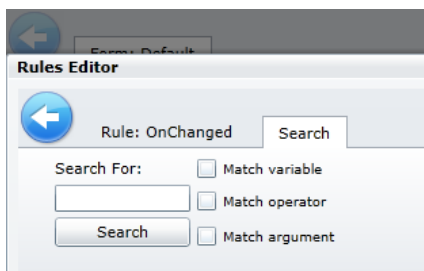


Variable example



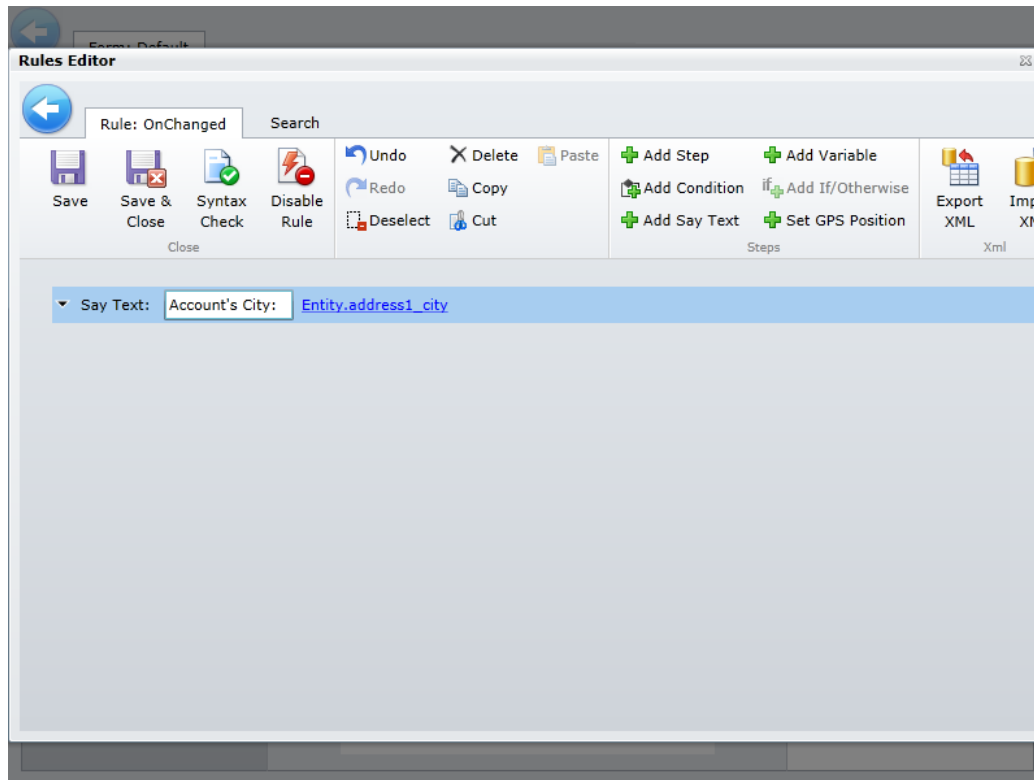
2.3.7 Search tab in ribbon

Click on Search in Ribbon to open display search tab. Enter text and click search. Searched items are highlighted with yellow color in workflow tree. If none of the search options is selected, searching acts as if all options were selected.



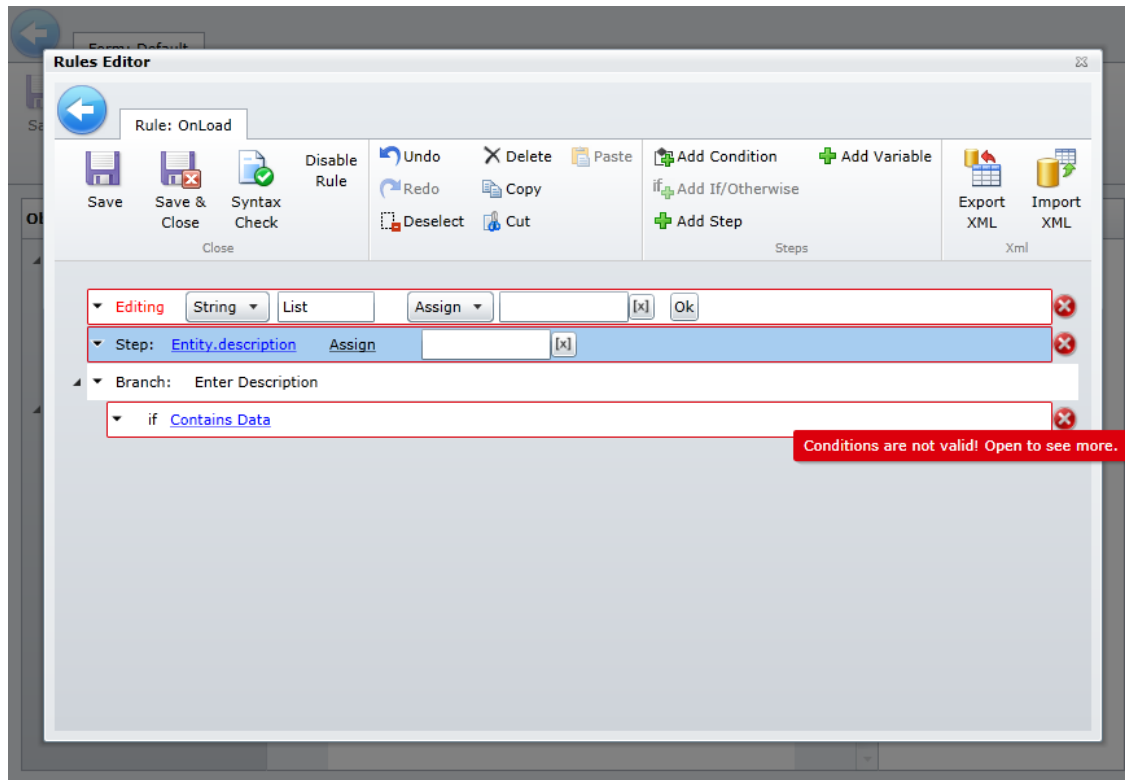
2.3.8 Say Text

Click on Add Say Text and enter the static text and choose a variable. The message will show the static text and the value assigned to the field selected.



2.4 Rule Validation

Before you can save function it needs to be valid. To Validate function step, click on "Syntax Check" button. If rule is invalid error prompt is displayed and icons will be displayed next to invalid steps. You can check error message by hovering the mouse cursor over the error icons.



3 Disabling the rule

By clicking “Disable rule” button in **Rule Editor** you can disable rule. Disabled rule won’t be published and synchronized to mobile client.

If the rule is already in published project, you need to publish the project after disabling the rule and synchronize the Mobile CRM application to apply the form without rule.

4 Rule Events

As we mentioned, rules can be defined for form events. Each event has some unique variables.

- *OnLoad* Event occurs when form is loaded
 - *isFormEditable* – is a Boolean variable, setting it false you can disable editing the whole form
- *OnChange* Event occurs when some field in form is edited (gains/losses focus, input entered) and when the form is loaded

changedItem – is a variable that can be assigned the changed entity field and can be used for the condition, so the steps are performed only when this field is changed.

- *OnSave* Event occurs when form is being saved.
 - *ErrorMessage* – is a String variable where you can set ErrorMessage for form. When an error message is shown, the saving process of the form is canceled.

5 Examples

Error Message

The screenshot displays the 'Rules Editor' window with the following components:

- Toolbar:** Includes icons for Save, Save & Close, Syntax Check, Disable Rule, Undo, Redo, Delete, Copy, Paste, Deselect, Cut, Add Condition, Add Variable, Add If/Otherwise, Add Step, Export XML, and Import XML.
- Rule Name:** 'Rule: OnChanged'.
- Branch:** 'Enter Description'.
- Condition:** 'if (Entity.address1_city Does Not Contain Data And Entity.address1_country Does Not Contain Data)'.
- Steps:**
 - Step: 'Form.address1_city.ErrorMessage' with action 'Assign' and value 'City or Country cannot be empty'.
 - Step: 'Form.address1_country.ErrorMessage' with action 'Assign' and value 'City or Country cannot be empty'.
 - Step: 'Form.address1_city.ErrorMessage' with action 'Clear'.
 - Step: 'Form.address1_country.ErrorMessage' with action 'Clear'.