



Form Validations

Built-in Validations

Custom Validations

Validation Messages

Validating User Inputs

- Applications should limit user mistakes
 - But they are inevitable
- Input fields may...
 - Have different data types
 - Depend on other user input values
 - Require business-specific validations
- OutSystems provides mechanisms to help implement input validations

The image shows a web form with the following elements:

- Description:** A large text input field with a placeholder and a small icon in the bottom right corner.
- Due Date:** A text input field with a placeholder "mm/dd/yyyy".
- Priority:** A dropdown menu with "Low" selected and a downward arrow.
- Status:** A dropdown menu with "New" selected and a downward arrow.
- Requested By:** A dropdown menu with "Jay Silver" selected and a downward arrow.
- Buttons:** A "Cancel" button with a red border and a red 'x' icon, and a red "Save" button with a white checkmark icon.

Built-in Validations

- Built-in validations for Input Widgets are performed automatically
 - **Mandatory fields** must be filled
 - Input values must **comply with the data types** of the Variables bound to the widgets

The screenshot shows a form with several input fields. A properties window is open for the 'Input_DueDate' widget, which is highlighted with a red box. The properties window has two tabs: 'Properties' and 'Styles'. The 'Properties' tab is active, showing a table of properties for the widget. The properties are:

Input_DueDate	
Name	Input_DueDate
Variable	GetOrderById.List.Current.Order.DueDate
Prompt	
Input Type	Date
Mandatory	True
Enabled	True
Style Classes	"form-control"

The form itself has the following fields:

- Description: A text input field.
- Due Date: A date input field with a placeholder 'mm/dd/yyyy' and a 'Low' dropdown menu.
- Status: A dropdown menu with 'New' selected.
- Requested By: A dropdown menu with 'Jay Silver' selected.
- Buttons: 'Cancel' (with a red 'x' icon) and 'Save' (with a green checkmark icon).

Built-in Validations

- Built-in validations for Input Widgets are performed automatically
 - **Mandatory fields** must be filled
 - Input values must **comply with the data types** of the Variables bound to the widgets
- Buttons / Links inside a Form have a Built-in Validations Property
 - Set to **Yes** to perform built-in validations

The image shows a form with the following fields:

- Description:** A text input field.
- Due Date:** A date input field with the placeholder text "mm/dd/yyyy".
- Priority:** A dropdown menu with "Low" selected.
- Status:** A dropdown menu with "New" selected.
- Requested By:** A dropdown menu with "Jay Silver" selected.

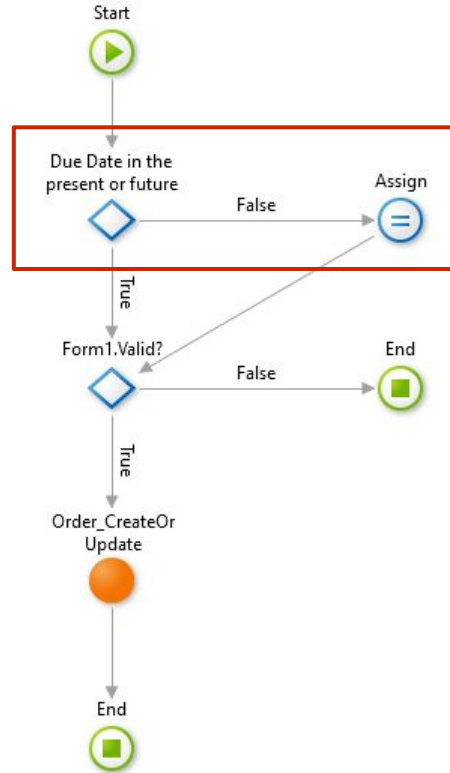
At the bottom of the form are two buttons: "Cancel" with a red 'X' icon and "Save" with a green checkmark icon.

Below the form is a table showing the events for the "Save" button:

Events	
On Click	SaveOnClick
(New Argument)	
Built-in Validations	Yes

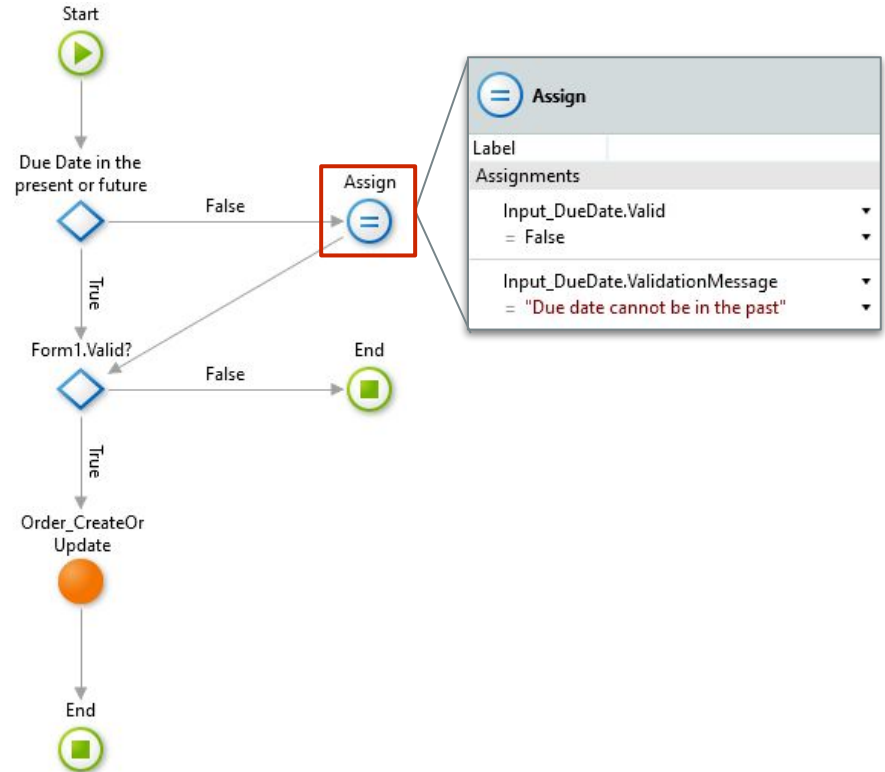
Custom Validations

- Inputs can have custom validations
 - Must be performed in the Action flow
 - For invalid Form Input fields
 - Set the **Valid** property to False
 - Set the **ValidationMessage**



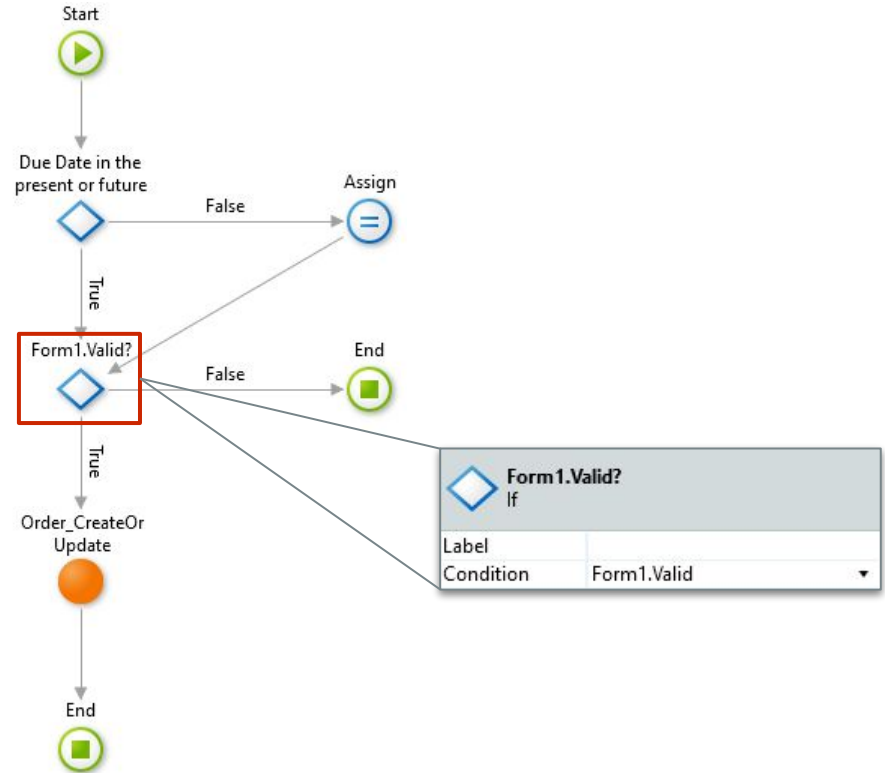
Custom Validations

- Inputs can have custom validations
 - Must be performed in the Action flow
 - For invalid Form Input fields
 - Set the **Valid** property to False
 - Set the **ValidationMessage**

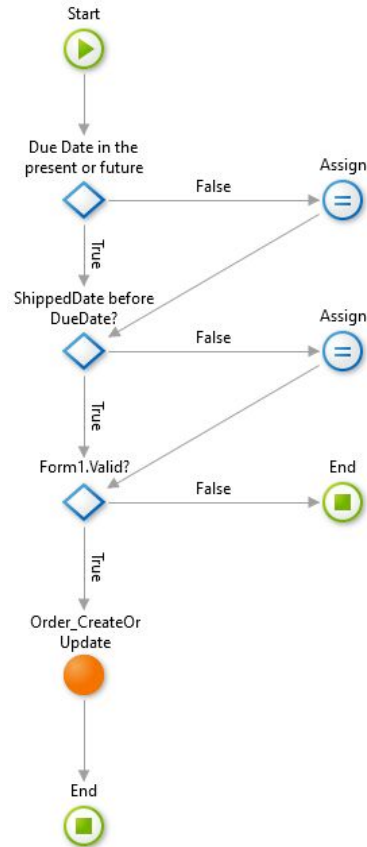


Custom Validations

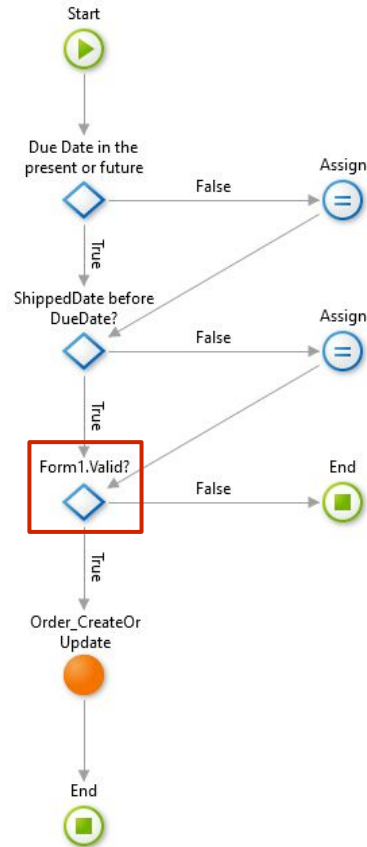
- Inputs can have custom validations
 - Must be performed in the Action flow
 - For invalid Form Input fields
 - Set the **Valid** property to False
 - Set the **ValidationMessage**
- The **Form.Valid** property should be checked after the last custom validation
 - If one Input is not Valid, the Form is automatically not Valid
 - The Form.Valid cannot be explicitly changed (e.g: Assign)



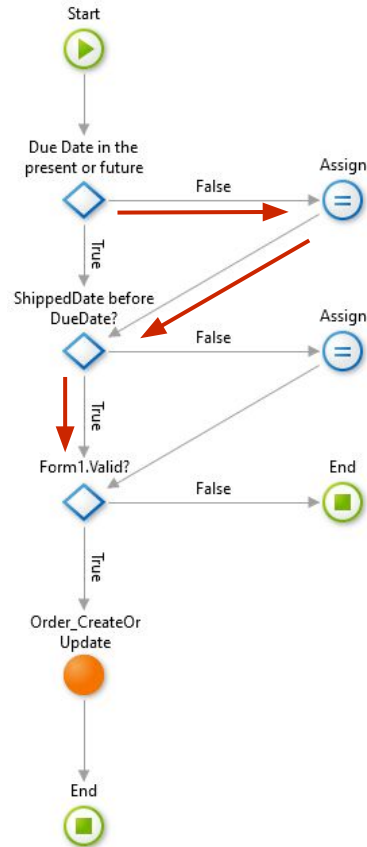
Multiple Validations



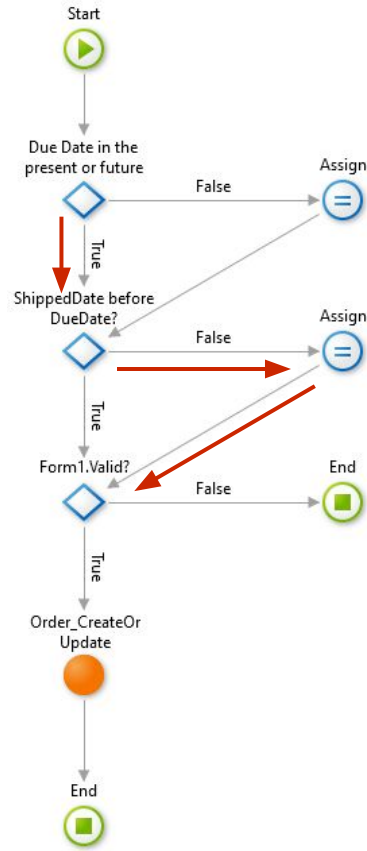
Multiple Validations



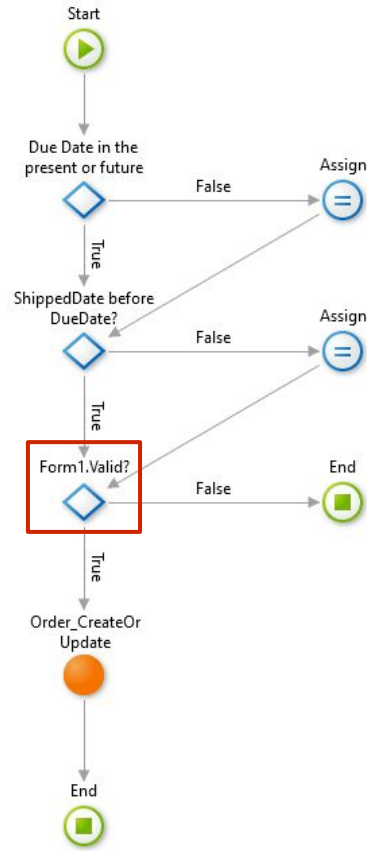
Multiple Validations



Multiple Validations

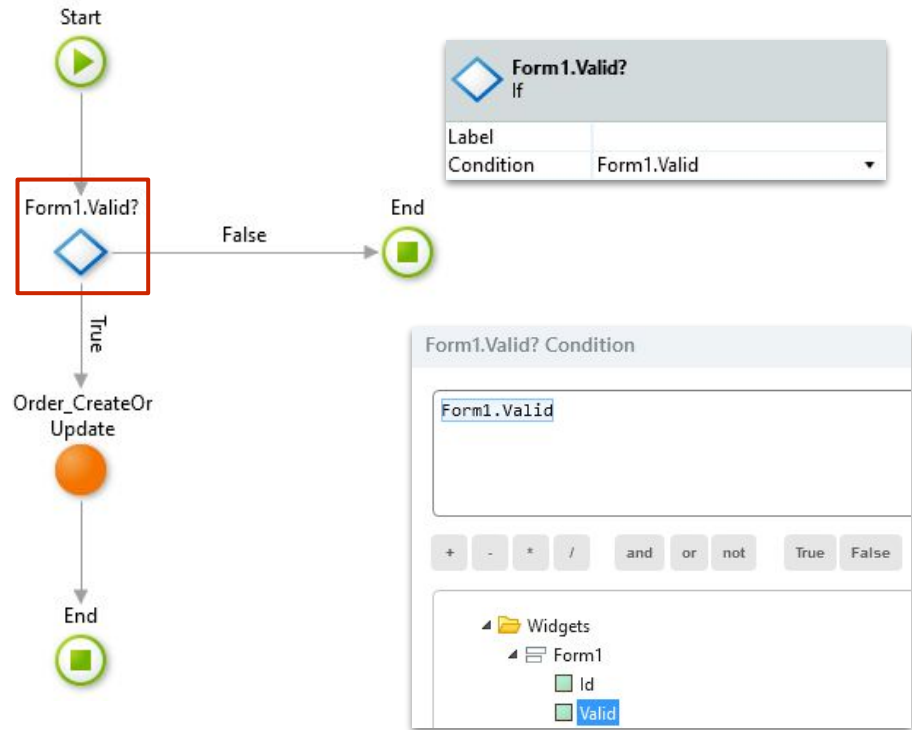


Multiple Validations



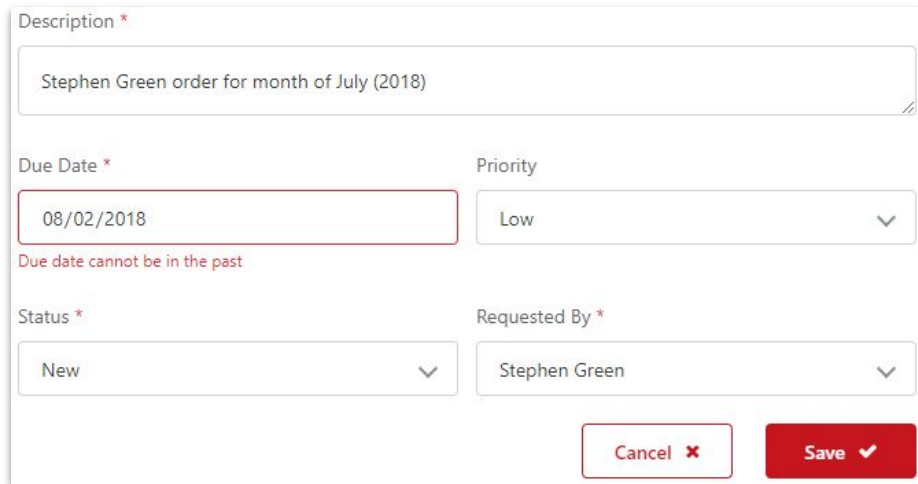
Checking Built-in Validations

- The *Form.Valid* property should be checked even if there are no custom validations
 - To avoid storing / manipulating invalid data
- OutSystems automatically performs the validations
 - But it is up to the developer to leverage that in the logic



Validation Messages

- The Valid property of the Inputs are checked when rendering the Screen
- If Valid property is set to True:
 - Displays the regular widget
- If Valid property is set to False:
 - Displays the regular widget
 - Applies the specific styling
 - e.g. a red border



The screenshot shows a form with several fields. The 'Description' field contains the text 'Stephen Green order for month of July (2018)'. The 'Due Date' field contains '08/02/2018' and has a red border and a red error message below it: 'Due date cannot be in the past'. The 'Priority' field is a dropdown menu with 'Low' selected. The 'Status' field is a dropdown menu with 'New' selected. The 'Requested By' field is a dropdown menu with 'Stephen Green' selected. At the bottom right, there are two buttons: 'Cancel' with a red 'x' icon and 'Save' with a green checkmark icon.

Description *

Stephen Green order for month of July (2018)

Due Date * Priority

08/02/2018 Low

Due date cannot be in the past

Status * Requested By *

New Stephen Green

Cancel ✕ Save ✓

Questions?

Thank you
