

Text Control

Text control displays properties of datatypes String, Text, Integer, Float, Decimal, Date, Sequence and Classification. Text control honors Format Specifier set on the ItemType definition. Text control honors the default value set through ItemType definition. You can also set default value for Text control by defining text expression with properties from other controls from the current page or previous pages. Text control has additional settings like Display Condition, Validation Rules, Conditional Formatting etc. that are covered in the subsequent sections.

The screenshot shows the 'Text Settings' dialog box. The 'Name' field is 'supplier_name', the 'Label' is 'Supplier Name', and the 'Data Type' is 'String'. The 'Control Style' is set to 'Rule', and the 'Rule Scope' is 'Context Type'. The 'Access' section shows 'Show' and 'Edit' permissions are both checked.

- **Name:** It is used to uniquely identify control inside the template. For text control, name is derived from the property bound to the control.
- **Label:** It is used for showing title on the text control. For text control, label is derived from the property bound to the control. If property label is localized inside innovator, at runtime localized label is pulled from the property definition.



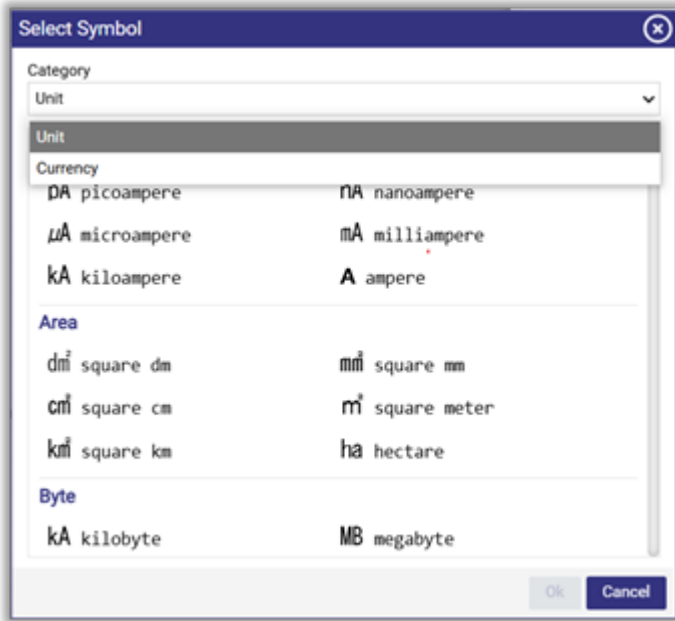
- **Data Type:** It allows type of data that you can entry in the text control. It is derived from the property bound to the control.
 - **Tooltip:** Tooltip will be shown when you hover the mouse on the information icon on the form field. It is derived from the property bound to the control.
 - **Help Text:** Help Text flyout will be shown when you click on the information icon on the form field label. It is derived from the property bound to the control.
 - **Format Specifier:** It shows the format in which data should be rendered in the control. It is derived from the property bound to the control.
- **Unit:** Unit setting will be shown only for the numeric properties associated with the control. Using this setting, you should be able to set units for Weight, Length, Area etc. through the Select Symbol dialog.

The image shows a 'Text Settings' dialog box with the following fields and values:

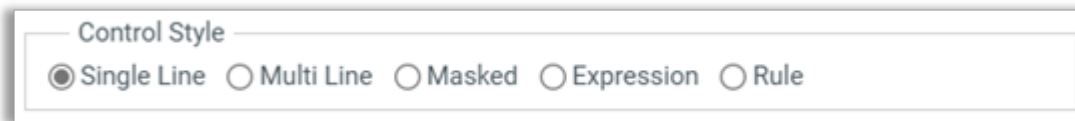
- Name (maximum 32 characters): total_weight
- Label: Total Weight
- Data Type: Decimal
- Tooltip: (empty)
- Help Text: (empty)
- Format Specifier: (empty)
- Unit: kg (highlighted with a red box)
- Access: Show (empty)
- Computed Value: Max Order Quantity*Weight
- Read Only: Yes, No, Expression, Script

Buttons: Ok, Cancel

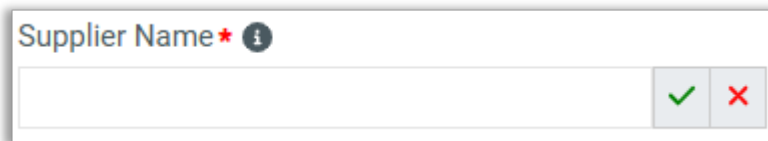




- **Control Style:** Text control can be rendered as textbox, text area, rule by setting its style as single line, multi-line, masked, expression, or rule.



Control Style: Single Line



Control Style: Multi Line



Supplier Name * ⓘ

- **Control Style:** Masked

Supplier Name * ⓘ

.....

Control Style: Expression

With expression style, editing the control will show expression editor to create expression with properties from the contextual ItemType.

Supplier Name * ⓘ

✓ ✗

Supplier Name ⓘ

Expression Editor

Property	Operator	Value
<input type="text"/>	<input type="text"/>	<input type="text"/>

Expression

AND OR () ✕

Control Style: Rule

With rule style, editing the control will show rule editor to create rule with properties from the contextual ItemType.



- **Access:** It allows you to define roles which can access control. Roles (in the form group identities) can be at Show and Edit level. When control is hidden because you do not have defined role, property associated with this control is not even fetched from server.
- **Computed Value:** It allows you to show default value when control is rendered. Default value can be computed based on the expression; in the expression you can reference other properties from the current page or previous pages. If you create an expression for default value and make control as read-only, it can dynamically change its value based on what you enter in other controls on the page.
- **Read Only:** Based on the option selected for Read Only, control can become read-only even if you can edit the properties used inside the page with lock. You can select one of the options like 'Yes', 'No', 'Expression', or 'Script'. Expression allows you to define a Boolean expression using properties from the current and previous pages (in case of multi-page application) to conditionally make the control read-only if expression is evaluated to true. Expression will be evaluated based on the cached property values on the client. Script allows you to write custom JavaScript code that should return Boolean value. If returned value is true, the control will be read-only.
- **Display Condition:** Based on the option selected for Display Condition, the control will be shown or hidden in the wizard at runtime. You can select one of the options like 'Always Show', 'Always Hide', 'Expression', or 'Script'. Expression allows you to define a Boolean expression using properties from the current and previous pages (in case of multi-page application) to conditionally display the control if expression is evaluated to true. Expression will be evaluated based on the cached property values on the client. Script allows you to write custom JavaScript code that should return Boolean value. If the returned value is false, the control will be hidden.
- **Validations:** By default, a row is displayed under the Validations section. By clicking the cell under Rule, a condition or script can be set to validate the data entered in the control. Normally validations are executed when you click on Save, Done, Next buttons from the wizard. While you are defining the rule, you need to set the message that you want to display to user when rule is not satisfied. Message can also be a text expression; you can use current control or other control values to display dynamic text to the user. You can also set locale specific messages for the languages you configured. While defining the rule, you also need to specify

Type of the rule as Error or Warning. Additional rules can be added by clicking on Add (+) action on the previous row after which you need to add the rule. If you define multiple rules, all rules should be satisfied to move forward. You can remove or reposition the rules using the respective icons from actions column.

The following example shows validation rule of type Error. When validation rule fails, error will be shown in the popup dialog with options to ignore and move forward or fix the issues before moving forward. If you move forward without fixing validation errors for that page, in the navigation panel, page title will be shown with red circle with tick icon to indicate there are validation errors in the page.

Validations			
Rule	Message	Type	Actions
Supplier Name...	*Supplier name can't b...	Error	+ ✎ ✖ +

Validation Rule

Rule

Expression Script ?

Control: Search Operator: Contains Value: [] +

Expression: Supplier Name NotEqualTo "" [AND OR ()]

Message

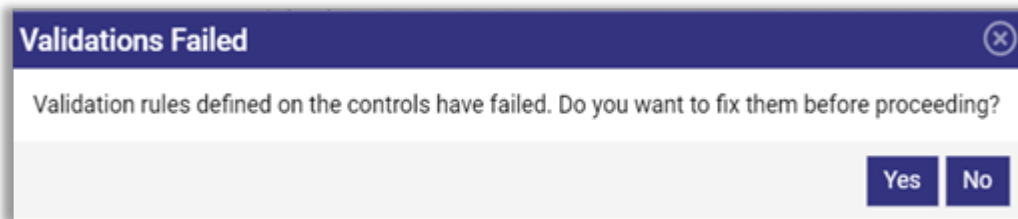
Language: English

Expression: Control: [] +

*Supplier name can't be empty"

Ok Cancel





The following example shows validation rule of type Warning. When validation rule fails, warning will be shown in the flash notification with orange background color. Warnings can be ignored by clicking on Save, Done, Next buttons second time to move forward with the warning.

Validations			
Rule	Message	Type	Actions
Contact Number...	"Contact number can't..."	Error	+ ✎ ✖ +
Contact Number...	"Contact number shou..."	Warning	+ ✎ ✖ +

A dialog box titled "Validation Rule" with a close button (X) in the top right corner. It is divided into two main sections: "Rule" and "Message".

Rule Section:

- Radio buttons for "Expression" (selected) and "Script" (with a help icon).
- Fields for "Control" (Search), "Operator" (Contains), and "Value" (empty).
- Buttons for logical operators: AND, OR, (,), and a red trash icon.
- Text area containing: "Contact Number Contains -"

Message Section:

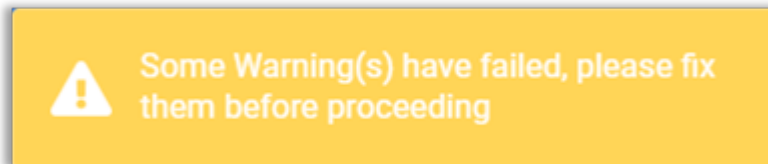
- Language dropdown set to "English".
- Fields for "Expression" (empty) and "Control" (empty).
- Buttons for logical operators: + and a red trash icon.
- Text area containing: "Contact number should contain '-' to separate area code"

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



Contact Number ⓘ

2345679874



Validation rule can be defined using Script option. This will be useful for defining validation based on some complex logic rather than simple expression. Script can be written in JavaScript by accessing control values using API. Script can contain the logic to access data from the server to perform complex validation. Script should return a Boolean value to indicate whether the validation is successful.

- **Conditional Formatting:** By default, a row is displayed under Conditional Formatting section. By clicking the cell under Rule, a condition or script can be set to display specific styles for the data displayed in the control if the condition is evaluated to true. Additional rules can be added by clicking on Add (+) action on the previous row after which you need to add the rule. Condition formatting rules are evaluated at runtime in the order in which they are defined from first to last, whenever the first rule is satisfied it stops and won't execute remaining rules. Because of that, extra care should be taken while defining the order of the rules. You can edit the existing rule by clicking on action icon null . You can remove or reposition the rules using the respective icons from actions column.



Text Settings [Close]

Read Only
 Yes No Expression Script

Display Condition
 Always Show Always Hide Expression Script

Validations

Rule	Message	Type	Actions
Supplier Name...	*Supplier name can't b...	Error	+ ✎ ✕ ⛶

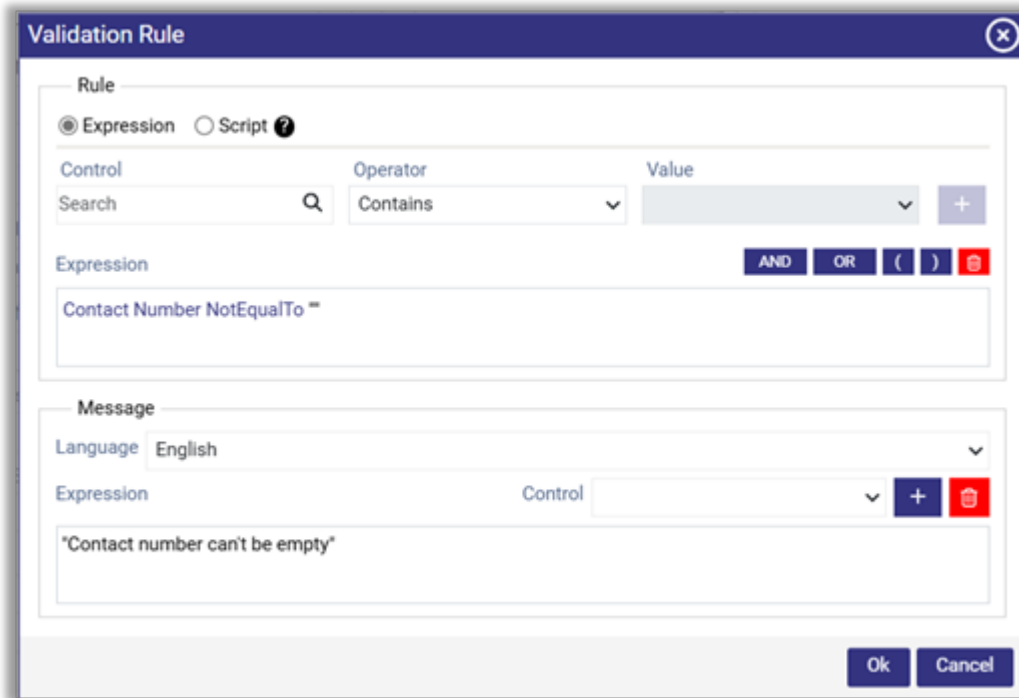
Conditional Formatting

Rule	B	I	U	A	Color	Actions
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	■	+ ✎ ✕ ⛶

Conditional Formatting

Rule	B	I	U	A	Color	Actions
Total Weight LessThanOrEqualTo 1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	■	+ ✎ ✕ ⛶
Total Weight LessThanOrEqualTo 2000 AND Total...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	■	+ ✎ ✕ ⛶
Total Weight GreaterThan 2000.00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	■	+ ✎ ✕ ⛶





Conditional Formatting rule can be defined using Script option. This will be useful for defining the rule based on some complex logic rather than simple expression. Script can be written in JavaScript by accessing control values using API. Script can contain the logic to access data from the server to execute complex rule. Script should return a boolean value to indicate whether the rule is successful.

- **Reset Dependency:** It allows you to reset value of the control to the default if any of the controls defined in “Reset Dependency” have their value changed at runtime.
- **Font:** It allows you to set the required font and size for the value in the control. If any of the settings are changed, it shows Reset icon on the border, using which you can reset the settings.
- **Margin:** It allows you to set required top, right, bottom, left margins with respect to adjacent controls. If any of the settings are changed, it shows Reset icon on the border, using which you can reset the settings.
- **On Load:** OnLoad event allows you to write custom logic that will be executed after the value in the control is loaded. Please refer to the Template Customization section for more details.
- **On Change:** OnChange event allows you to write custom logic that will be executed after the value in the control is updated. Please refer to the Template Customization section for more details.

