

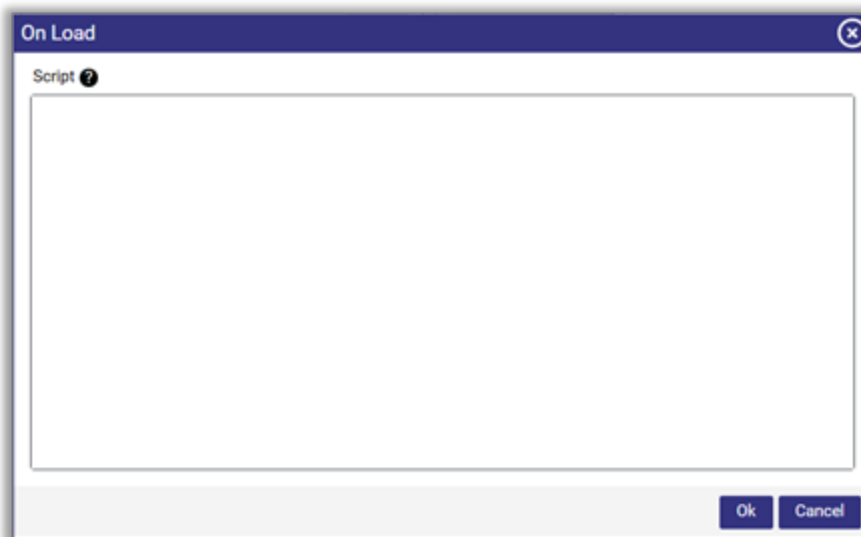
# Template Customization

## Events

Aras ProAppDesigner supports customization through various types of events. These events will be launched by Aras ProAppDesigner Wizard at various instances. As part of creating Aras ProAppDesigner template, admin can register custom script for each of the supported events. Following event types are supported on various controls in TemplateStudio. You can associate custom script with the Event using control settings.



The image shows two event configuration controls. The first is labeled "On Load" and the second is labeled "On Change". Each control consists of a text input field and a small button with a checkmark and a red 'x' icon, likely for enabling or disabling the event.



The image shows a dialog box titled "On Load" with a close button in the top right corner. Inside the dialog, there is a section labeled "Script" with a question mark icon, followed by a large empty text area for entering the script. At the bottom of the dialog, there are "Ok" and "Cancel" buttons.

### **<b>OnClick Event:</b>**

OnClick event is supported only on the button control. It will be launched when the button is clicked.

```
// on button click openina custom flvout  
appStudioWizard.showDialogFlyout(pageDialogName);
```



## <b>OnLoad Event:</b>

OnLoad event will be launched after loading the data for Wizard page or individual collection controls.

*//Setting list options on a column of the Table control:*

```
const itemId = appStudioWizard.getItemId();
```

*// calling a server method & passing the **itemId** to get the required options. Refer to **10.6 Examples** on how to call server method.*

*// building options map*

```
let options = {  
  optionValue: "OptionLabel",  
  optionValue2: "OptionLabel2"  
};  
appStudioWizard.setCollectionListOptions("table_name", "column_name",  
options);
```

## <b>OnChange Event:</b>

On Change event will be launched after changing the value of the control. It is supported both on simple controls and collection control columns.

*// On value change of control1. access its value and set the value of control2*

```
let control1Value = appStudioWizard.getControlValue("control1");  
if (control1Value) {  
  let value = control1Value === "Value1" ? "Discoverable" : "Hidden";  
  appStudioWizard.setControlValue("control2", value);  
}
```

## <b>OnBeforeLoad Event:</b>

OnBeforeLoad event will be launched before loading the data in the collection controls inside the page. It is supported only on table, worksheet, treeTable controls. OnBeforeLoad event can be used to set filterExpression on the collection control to load only required dataset.



```
//Setting filterExpression on Table control
```

```
// creating an expression string
```

```
let expressionString =  
"classification EqualTo Item AND name StartsWith AS_";
```

```
// Refer to section 10.3.4.9 filterCollectionItems on how to build your expression string
```

```
// calling filter API for table control
```

```
appStudioWizard.filterCollectionItems(controlName, expressionString);
```

### **<b>OnSelect Event:</b>**

OnSelect event will be launched when an item in the collection control is selected.

```
// getting the selected item to access its data
```

```
const selectedControlItems = appStudioWizard.getControlSelectedItems(  
"supplierParts");  
if (selectedItems) {  
const selectedItem = selectedItems[0];  
// accessing data of selected item
```

```
const itemId = selectedItem.getItemId();  
const relatedId = selectedItem.getRelatedId();  
const partWeight = selectedItem.getValue("weight");  
}
```

### **<b>OnSubmit Event:</b>**

OnSubmit event will be launched after selecting Submit button in the flyout. It is supported only for the flyouts.

```
// accessing form values in the flyout & saving them
```

```
let price = appStudioWizard.getControlValue("price");  
let date = appStudioWizard.getControlValue("purchased on");  
let managedBy = appStudioWizard.getControlValue("managed_by_id");
```

```
// calling a server method to save the data. Refer to 10.6 Examples on how to call server method.
```



```
if (onSuccess) {
  appStudioWizard.showSuccessMessage("Item saved successfully");
} else {
  appStudioWizard.showErrorMessage("Failed to save the Item.");
}
```

### **<b>OnInsert Event:</b>**

OnInsert event will be launched after Item has been inserted in the collection controls. Event handler will receive inserted item details via eventData, this object will be available in the scope of the event handler code.

```
// Accessing data of inserted item
const insertedItemCount = eventData.length;
for (let i = 0; i < insertedItemCount; i++) {
  // accessing data of selected item

  const relationshipId = eventData[i].getItemId();
  const relatedItemId = eventData[i].getRelatedId();
  const partWeight = eventData[i].getValue("weight");
  // setting a cell value on inserted item
  eventData[i].setValue("classification", "item");
}
```

### **<b>OnInsertFilter Event:</b>**

OnInsertFilter event will be launched while loading the ItemSelect flyout. Admin can configure a custom script and assign a simple expression string to the eventData, this object will be available in the scope of the event handler code.

```
// creating a custom expression string
let expressionString = "classification EqualTo Item AND id NotEqualTo
8D58C2D8DC1649279E64CCF5BD2FF7A7";
// Refer to 10.3.4.9 filterCollectionItems on how to build your expression string
```



```
// assignina expression string to eventData  
eventData = expressionString;
```

### **<b>OnSave Event:</b>**

OnSave event will be launched after saving the item inside the wizard or section / group controls if they have their own associated items.

```
const itemId = appStudioWizard.getItemId();  
const itemType = appStudioWizard.getItemType();
```

*// calling a server method to save custom data. Refer to **10.6 Examples** on how to call server method.*

```
if (onSuccess)  
appStudioWizard.showSuccessMessage("Data saved successfully");  
else  
appStudioWizard.showErrorMessage("Failed to save the custom data.");
```

