Setting Form Elements Dynamically

Elements can have default values and/or options to select from. These can be specified in the **Attributes** panel as described on the documentation pages of the respective form element (see **Prepopulation** for each supported form element).
There may be cases, however, where values of form elements need to be set dynamically at service runtime. Doing this is possible for all form elements.

Select Field Example

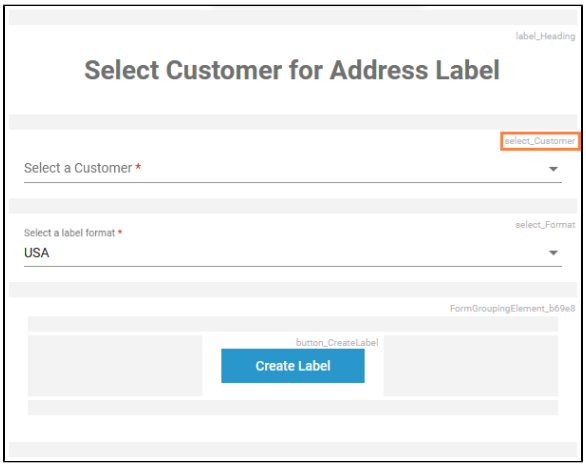
The following example shows how to set options to a select field.

UseLib_CustomerData_Example

Click the icon to download a simple example model that uses the **CustomerAddress_Lib** library.

Building the Data Model Structure for a Select Field

The mentioned **UseLib_CustomerAddress_Example** contains the following form **Form_SelectCustomer**:



To be able to set a list of values to a select field, you need to build a certain structure in your data model in the **Implementation** folder. This structure is based on PAS data types that are provided with the **Base Types**.
Wanting to prepopulate the customer select field **select_Customer** with custom data dynamically, you need to prepare the the following structure in your data model:

On this Page:

UseLib_CustomerData_Example

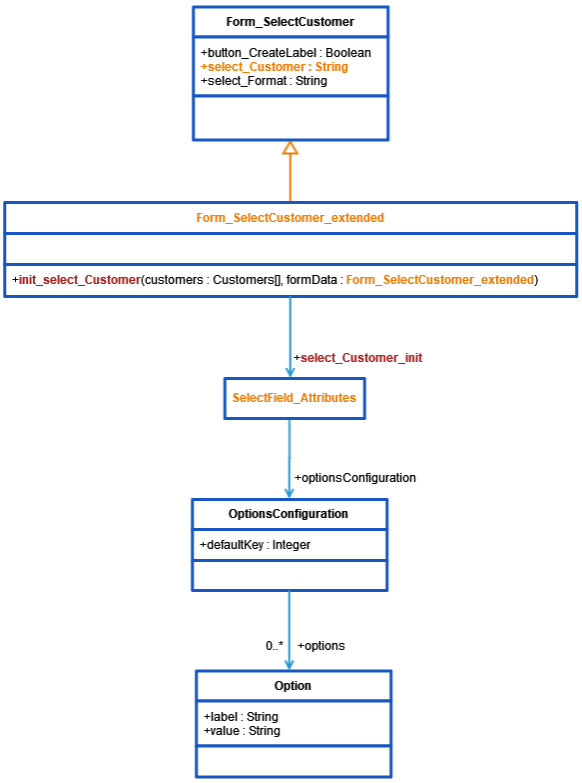
Click the icon to download a simple example model that uses the **CustomerAddress_Lib** library.

Form_InitialDataLoading_Checkbox_Example

Click the icon to download a simple example model that shows how you can **initialize checkbox elements dynamically** in a form.

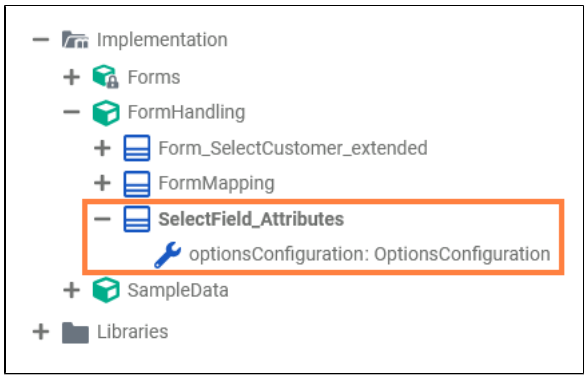
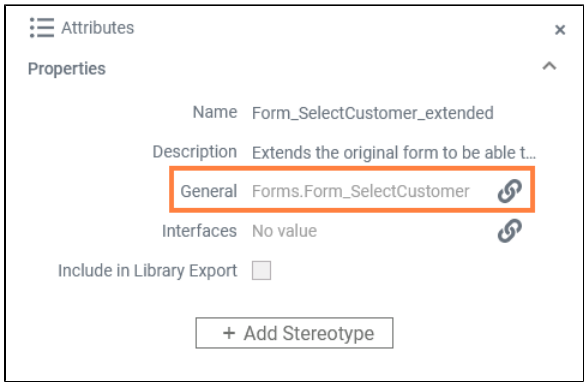
Related Pages:

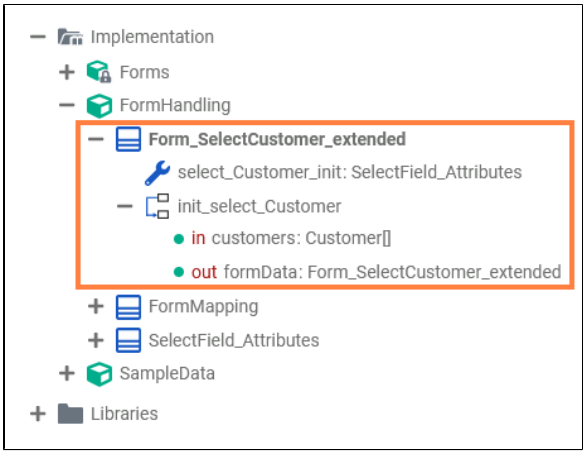
- Supported Form Elements



Elements marked in **orange** need to be created, elements marked in **red** must have a name that conforms to certain rules.

In the example service, a select field **select_Customer** on form **SelectCustomer** is populated from an array of customers.
For creating the needed structure, proceed as follows:

	In your **Implementation** folder, create a class that defines the attributes of a select field. In the example that would be **SelectField_Attributes**. Add a property of type **Base Types.PAS Form Elements.pas.form.elements.select.OptionConfiguration** to this class. This property will contain the options.
	Create an extended class that derives from your form class. In the example, that would be **Form_SelectCustomer_extended** which derives from **Form_SelectCustomer**.



This new form class must extend the form class by two elements having specific names:

- a property called **<name of the select field>_init** This property must be of the type you created earlier, the type that defines the attributes of a select field (e.g. property **select_Customer_init** of type **SelectField_Attributes**).

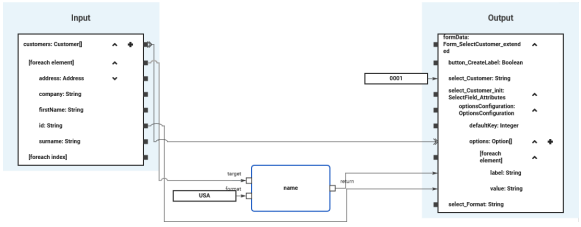
- an static operation called **init_<name of the select field>**

  This operation must have exactly one output parameter having the type of the owning class. An example for that is operation **init_select_Customer**, having the output parameter **formData** of type **Form_SelectCustomer_extended** .

	You can add as many input parameters to this operations as you need for setting the field options. The example operation has one input array of type **Customer** that contains the customer data that should be used to prepopulate the customer select field.

Setting the Options for a Select Field

To actually set the options, you need to implement the **init_...** operation. The example operation **init_select_Customer** is implemented with a mapping diagram.



Using a **foreach** loop, values from the **customers** array are set to the **options** array of the select field.
You can use the select field itself (**select_Customer**) to initialize the default selection of the select field.

Available Properties for Dynamic Access

Every form element has a set of default and additional properties that you can initialize dynamically in your execution. Find below a list of all element properties that can be set dynamically.

Property	Type	Available for	Possible Values		Example
Label (label)	String	• all form elements	any string		
Placeholder (placeholder)	String	• Text Box • Text Area	any string		
Read-only (readonly)	Boolean	• all interactive elements	true	Element is read-only.	
			false	Element is not read-only (default).	

Mandatory (mandatory)	Boolean	• all form elements	true	Element is mandatory.	
			false	Element is not mandatory (default).	
CSS Class (cssClass)	String	• all form elements	a valid CSS class from your associated CSS file		
Regular Expression (regex)	String	• Text Box	a valid regular expression		
Image Element					
Horizontal Alignment (horizontalAlign)	Integer	• Image	**0**	Align the image to the left (default).	
			1	Align the image to the horizontal center.	
			2	Align the image to the right.	
Description (description)	String	• Image	any string		
Caption (caption)	String	• Image	any string		
Target URL (targetUrl)	String	• Image	a valid URL		
Image (source)	String	• Image	a valid URL		
Select Field / Radio Button Group / Checkbox					
Options (optionsConfiguration)	OptionConfiguration	• Select Field • Radio Button Group	an array of label/value pairs		**UseLib_CustomerData_Example**  Click the icon to download a simple example model that uses the **CustomerAddress_Lib** library. **Form_InitialDataLoading_Checkbox_Example**  Click the icon to download a simple example model that shows how you can **initialize checkbox elements dynamically** in a form.
Layout Orientation (orientation)	Integer	• Radio Button Group	0	Distribute the radio buttons horizontally.	
			1	Distribute the radio buttons vertically.	
Label Position (labelPosition)	String	• Checkbox	**before**	Display the label before the checkbox.	
			after	Display the label after the checkbox (default).	
File Upload Element					
Allowed File Types (allowedFileTypes)	String	• File Upload	an array of values		
Allowed Mime Types (allowedMimeTypes)	String	• File Upload	an array of values		
Max File Size (maxFileSizeMB)	Float	• File Upload	any float value		
Upload Type (uploadTyp	String	• File Upload	**Single**	Only single file upload (default).	

e)			Multiple	Allow multi-file upload.	
profileName	String	• File Upload	a valid Designer namespace		
Date Picker					
dateFormat	String	• Date Picker	one of • **DD.MM.YYYY** (default) • MM.DD.YYYY • YYYY.MM.DD		
UseCurrentDate (useCurrentDate)	Boolean	• Date Picker	true	Display the current date.	
			false	Provide a date picker to select a date (default).	
Data Table					
items	Array of data table item	• Data Table	an array of valid table item objects		
columnModel	Column Model	• Data Table	a valid column configuration object		
Rows per Pages (rowsPerPage)	String	• Data Table	a valid count of rows per page		
Remove Row (actionRemoveRow)	Boolean	• Data Table	true	Removing table rows is allowed.	
			false	Removing table rows is not allowed (default).	
Add Row (actionAddRow)	Boolean	• Data Table	true	Adding new tables rows is allowed.	
			false	Adding new table rows is not allowed (default).	
No Data Text (noDataTxt)	String	• Data Table	any string		
preSelection	String	• Data Table	name of a **Boolean** column key that should be selected from the table		
selectable	Boolean	• Data Table	true	Display a checkbox before each data row to be able to select rows.	
			false	Data rows are not selectable (default).	
emptyCell	String	• Data Table	any string		

Property Types

Column Model

Property	Type	Description	Allowed Values
name	String	Specify a column name.	a string that conforms with the element naming rules
emptyCell	String	Specify a text for empty cells.	any string
label	String	Specify a column label.	any string

cssClass	String	Specify a CSS class that contains styling information for the column.	a valid CSS class	
hidden	Boolean	Define the visibilty of the column.	true	Column is visible.
			false	Column is hidden (default).
editable	Boolean	Specify whether the colum should be editable.	true	Column is editable.
			false	Column is read-only (default).
filterable	Boolean	Specify wheter the column should be filterable.	**true**	Column is filterable (default).
			false	Column is not filterable.
resize	Boolean	Specify wheter the column should be resizable.	true	The width of the column is sizable.
			false	The column has a fixed calculated width.
sorting	String	Define the sorting direction of the column contents if sorting is allowed (see **disableSort** below).	**no value**	Do not sort the table contents by this column (default).
			ascending	Sort table contents by this column ascending.
			descending	Sort table contents by this column descending.
disableSort	Boolean	Disable sorting for this column. The sorting direction can be defined with **sorting** (see above).	**true**	Column is not sortable (default).
			false	Table contents can be sorted by column contents.