TIBCO® MDM Studio
UI Builder User's Guide

Software Release 5.1
August 2017

Two-Second Advantage®

TIBCO®

Important Information

Contents

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, visit:

https://docs.tibco.com

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

TIBCO_HOME/release_notes/TIB_bstudio-mdm_*5.1.0*_docinfo.html

where *TIBCO_HOME* is the top-level directory in which TIBCO products are installed. On Windows, the default *TIBCO_HOME* is `C:\tibco`. On UNIX systems, the default *TIBCO_HOME* is `/opt/tibco`.

The following documents for this product can be found on the TIBCO Documentation site:

- TIBCO MDM Studio Release Notes
- TIBCO MDM Studio Installation Guide
- TIBCO MDM Studio Process Designer Tutorial
- TIBCO MDM Studio Process Designer User's Guide
- TIBCO MDM Studio Repository Designer Tutorial
- TIBCO MDM Studio Repository Designer User's Guide
- TIBCO MDM Studio Rulebase Designer Tutorial
- TIBCO MDM Studio Rulebase Designer User's Guide
- TIBCO MDM Studio UI Builder Tutorial
- TIBCO MDM Studio UI Builder User's Guide

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

  Entry to this site requires a user name and password. If you do not have a user name, you can request one.

How to Join TIBCO Community

TIBCO Community is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCO Community offers forums, blogs, and access to a variety of resources including product wikis that provide in-depth information, white papers, and video tutorials. In addition, users can submit and vote on feature requests via the Ideas portal. For a free registration, go to https://community.tibco.com.

Getting Started

This chapter explains how to get started using the MDM UI Builder.

MDM UI Builder Overview

MDM UI Builder allows you to quickly, easily, and uniformly develop custom UIs by using simple drag and drop methods.

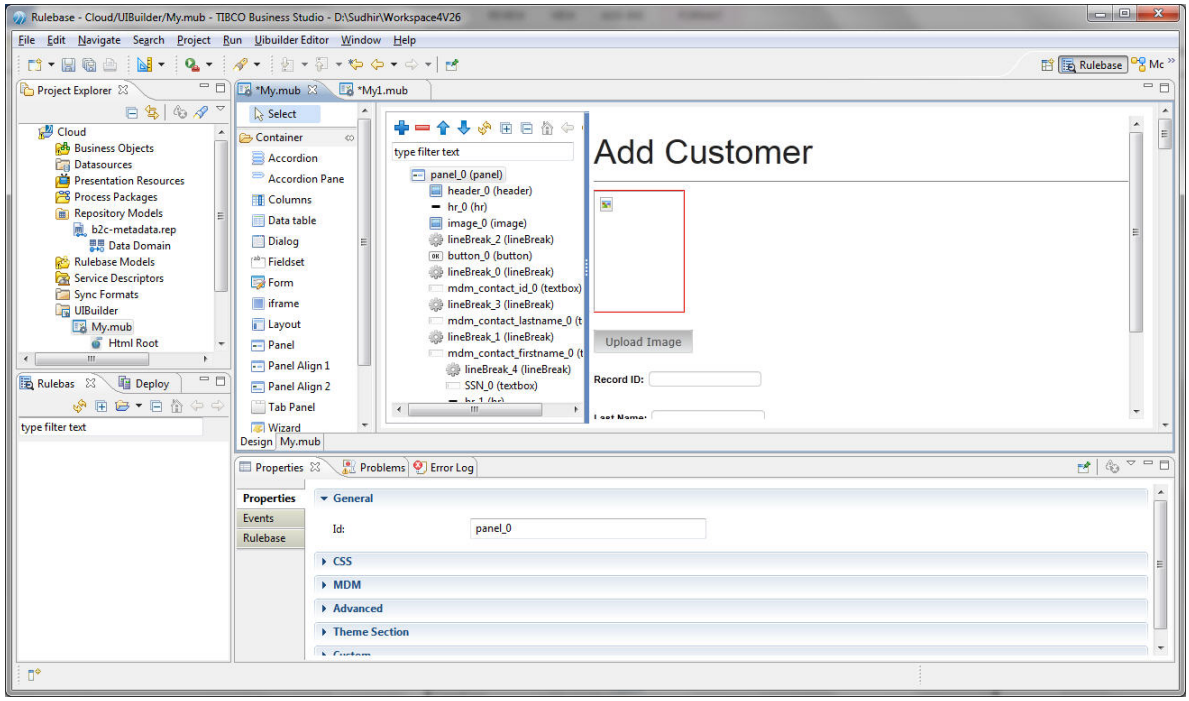
A new tool known as MDM UI Builder is developed to have a cleaner, simpler, more flexible UI in MDM. MDM UI Builder can be divided into three high level feature components:

- **User Interface Designer**
  User Interface Designer is part of MDM Business Studio. It is used to design HTML pages for the custom UI.
- **User Interface Widgets**
  User Interface Widgets are a collection of UI components which are developed in HTML, CSS, JavaScript and popular JavaScript libraries like jQuery and jQuery UI.
- **JSON Web Services**
  JSON Web Services are used by the generated page to communicate with the backend to perform various operations like find record, add record, delete record, modify record and so on.

MDM UI Builder allows customers and TIBCO Field Engineers to build custom UIs for the MDM application. Using UI Design tool you can simply drag and drop the UI widgets onto the tree outline viewer or canvas to build custom user interfaces for their own requirements. The tool supports TIBCO MDM by providing access to complete data models, individual repositories, and repository attributes and their attribute groups directly from the tool's GUI palette. Currently only two levels of relationship is supported in UI Builder. It also allows to include custom widgets which are not available in out-of-box widget list. This process reduces the custom UI development time compared to the exiting methods and practices.

Once the custom UI is created, you can either export or deploy the auto-generated UI artifacts (HTML, JS, and CSS) from the tool to MDM application.

MDM UI Builder Interface



Widget Palette: Widget Palette shows all the widgets that are present in the widget library. They are categorized in different groups based on their functions. Container widgets are used to create a group of controls. Form widgets are mainly input and other form related widgets such as textboxes, buttons, and labels. MDM widgets are pre configured widgets which are used for MDM specific actions such as Save Record, Cancel, Save and Process and so on.

Tree Outline Toolbar: The tree outline toolbar displays the toolbar icons for deleting, re-ordering, refreshing, expanding and collapsing the tree outline.

Tree Outline Viewer: Tree outline viewer shows the hierarchy of widgets which helps to understand the layout of the page. The widgets should be dragged and dropped onto the tree outline viewer in order to create the proper layout. Using the tree outline toolbar, you can make changes to the layout by deleting widgets or changing their order. You can also copy or cut any widget and paste it to a different location.

Canvas: The canvas in UI Builder is actually a web browser. Whenever you drag and drop widgets on the tree outline viewer UI Builder generates the code and it is rendered in the canvas (web browser).

Widget Properties: The property section displays the properties of the selected widget. There are various types of widget properties. They can be HTML properties of any element, CSS properties, MDM data binding properties, or user defined properties. The Property section also has a events tab where you can add the supported events.

Widget Palettes

The Widget Palette shows all the widgets that are present in the widget library.

They are categorized into different group based on their functions. Chart widgets is used to create charts and graphs. Container widgets are used to create a group of controls. Form widgets are mainly input and other form related widgets such as textboxes, buttons, and labels. MDM widgets are pre configured widgets which are used for MDM specific actions such as Save, Validate, Cancel.

The Palette (to the left of the screen) contains different artifacts to help you build your MDM UI Builder. Select and drop into the main drawing pane to create.

Chart Widgets

The Chart category contains different types of chart controls which help in building the UI.

To create a UI, you must select a chart control and drop in the tree outliner viewer.

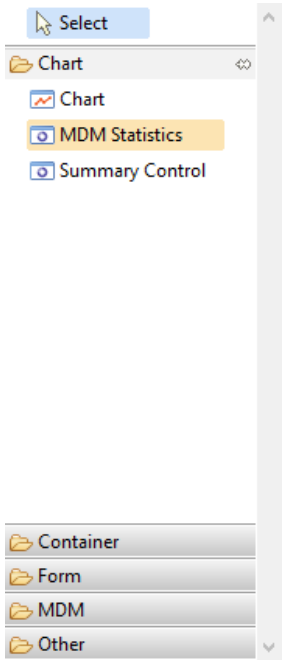
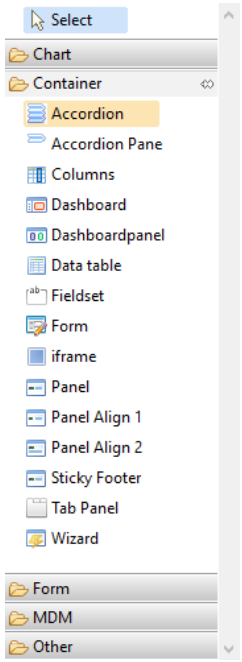


Chart	
	Chart Chart Widget provides an intuitive way to monitor the data quality. Using this widget, you can define a line chart to monitor the data quality of attribute
	MDM Statistics MDM Statistics widgets provides pie charts and number display to monitor the event summary, work item summary, and query summary.
	Summary Control Summary Control widget provides a radial progress chart to monitor the data quality of attribute.

Container Widgets

The Container category contains different types of group controls which help in building the UI.

To create a UI, you must select a group control and drop in the tree outliner viewer.

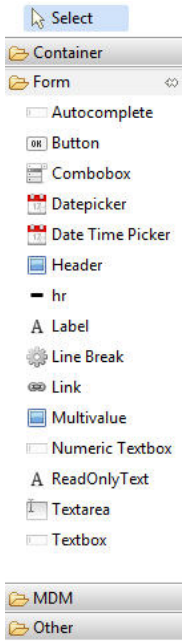


Container	
	Accordion The Accordion can be created using accordion and accordion pane widgets. Accordion is a Container where all the items in it are outlined in a segmented or accordion fashion.
	Accordion Pane The Accordion can be created using accordion and accordion pane widgets. Accordion Pane is a screen pane which outlines all the items in a segmented or accordion fashion.
	Columns The Columns widget is used to create vertical column layout. Panel (or panel align 1 or panel align 2) widget is used along with columns to create layout. Each panel represents a column if used under columns widget.
	Dashboard The Dashboard widget is the outer container that holds dashboardpanel widgets with different columns.
	Dashboardpanel The Dashboardpanel widget is the container that holds any other widgets. It is displayed as a small container with a blue header on the dashboard.

	Data table The Data grid control can be used when the data model has relationships associated with parent. If the model has any relationships they appear on the widget palette which can be selected and dropped on the outline viewer to create the data grid. After creating the data grid you can view all the properties in the property pane using which you can set different options for the grid.
	Fieldset The Field set can be used to group the widgets.
	Form The Form widget can be used to create a form.
	iframe The iframe can be used to embed another page within a page. The load URL property is used to load another page.
	Panel The Panel can be used as a container similar to html element in the web page.
	Panel Align 1 These panels automatically align the textbox and label widgets in the same line. It aligns when you put a label and a input widgets such as textbox.
	Pane Align 2 These panels automatically align the textbox and label widgets. However, the textbox is aligned under the label. It aligns when you put a label and a input widgets such as textbox.
	Sticky Footer It creates sticky footer for the page which stays at the bottom of the page at all the time. It can be useful to place buttons in the footer.
	Tab Panel Tabs can be created by selecting the Tab Panel widget. Multiple tabs can be created by putting the Panel widget inside the tab panel widget.
	Wizard The Wizard widget is for making multi-step workflows that do not require server request. You can drag and drop a data model into the wizard and the data model displays a wizard.

Form

The Form category contains widgets for input and other form related widgets such as button, combobox, date picker, multivalue, textarea, textbox and so on.



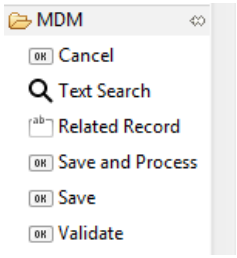
Form	
	Autocomplete The Autocomplete widget can be useful to provide suggestions as you type in the text field.
	Button The Button widget is used to define actions.
	Combobox The Combobox widget is used to specify the inputs where you can select a value from a list of values. The Source Data property is used to populate the values appearing in the combobox. The values should be specified as comma separated.
	Datepicker The Date picker widget can be used as input control for the date data types. Date can be selected using the calendar control.
	Fileupload The file upload widget is used to upload the selected file. The widget is created using html form, hidden iFrame object and jQuery without refreshing page. The Fileupload widget is part of a add record or modify record operation.

	Header The Header widget is used to define a header for the UI page.
	hr The Horizontal line widget is used to put a horizontal line in the panel.
	Label The Label can be used to display read only text.
	Line Break The LineBreak widget can be used to put vertical space between two widgets or elements. The height property can be used to specify the spacing.
	Link You can create hyperlink using link widget. Specify the URL property to create the hyperlink.
	Multivalue The Multivalue widgets allows you to enter multiple values for a data field. These values can be edited or deleted. If the data type of the multivalue field is date or timestamp then the appropriate widget is displayed to help you enter the value.
	Numeric Textbox It is used for the data fields which are numeric (For example, integers and decimals). It does not allow you to enter invalid string values. It can be configured in four modes in which values can be integer, positive integer, decimal, and positive decimal. It has all the properties of textbox and also additional properties of its own.
	Read Only Text The Read Only Text widget can be used if the value of the attribute cannot be changed. The label and value can be set similar to textbox.
	Textarea The Textarea widget is used for strings with large number of characters. It extends textbox control.
	Input The Input widget is a HTML textbox which can be used for strings. By default, MDM string data fields are mapped to the input widget.

	Timestamp Picker The Timestamp Picker widget is used for the fields that have timestamp data type. You can select date as well as time using the widget.
URL	URL The URL widget is used for the field that have url data type.

MDM

The MDM category widgets are pre configured widgets which are used for MDM specific actions such as Cancel, Related Record, Save and Process, Save and Validate.

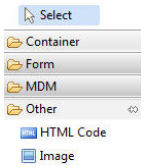


MDM	
OK	Cancel The Cancel button cancels the changes on page.
	Text Search The Text Search widget is used for quick search of records using TIBCO Patterns Engine. This widget is a textbox in which you can type and the records are retrieved as you type.
	Related Record The Related record widget is used when you have one-to-one relationship between root and child records. This widget requires manual configurations in the property and you need to put the attributes and relationship manually. Related record widget can also be configured automatically just like data grid. While dragging and dropping a relationship, in the dialog window select Group/Field Set option instead of Datagrid.
OK	Save and Process The Save and Process button is a specialized button widget which when clicked saves the data on the server.
OK	Save The Save is very similar to Save and Process except that it does not process the data in the MDM.

	Validate The Validate button is a specialized button widget which when clicked validates the data on the server.

Other

The Other category widget contains the HTML code for writing the HTML or JavaScript code and Image for adding images to your MDM UI Builder. To create, select and drop the controls into the canvas.



Other	
	HTML Code The HTML code widget is used to write HTML or JavaScript code which will be written within the body tag of the page. It must be valid HTML code.
	Image The Image widget is used for adding images to the MDM UI Builder.

Working with MDM UI Builder

This chapter introduces you to the widgets palettes in MDM UI Builder and explains the properties and usage of each widget palettes.

Widget Palette Properties

The property section displays the properties of selected widget.

There are various types of widget properties. They can be HTML properties of any element, CSS properties, MDM data binding properties, or user defined properties.

Common Properties

This sections explains some of the common properties of the widget palette available for Chart, Container, Form, MDM, and Other.

Common Properties

Properties	Description
General	
Id:	Specify the unique id for the widget.
Resource Key:	Specify the unique resource key for the widget.
CSS	
Class:	Specify the CSS class name for the widget.
Style:	Specify the CSS style name for the widget.

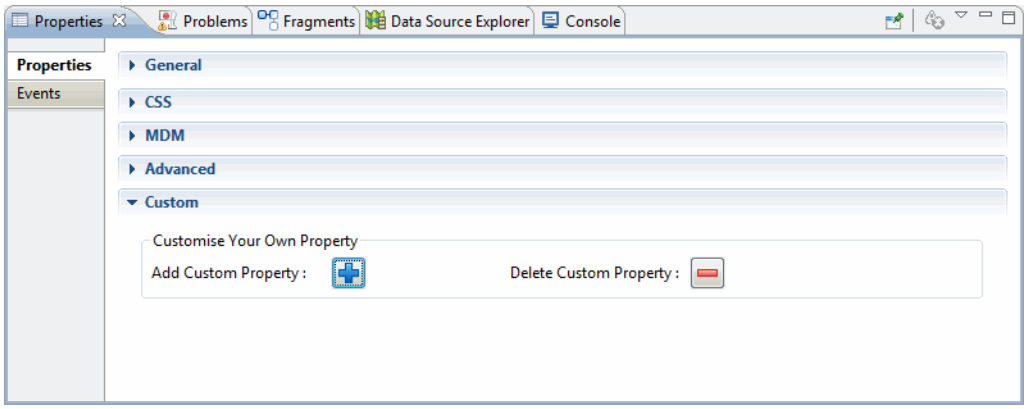
Property Section

Using Custom Property

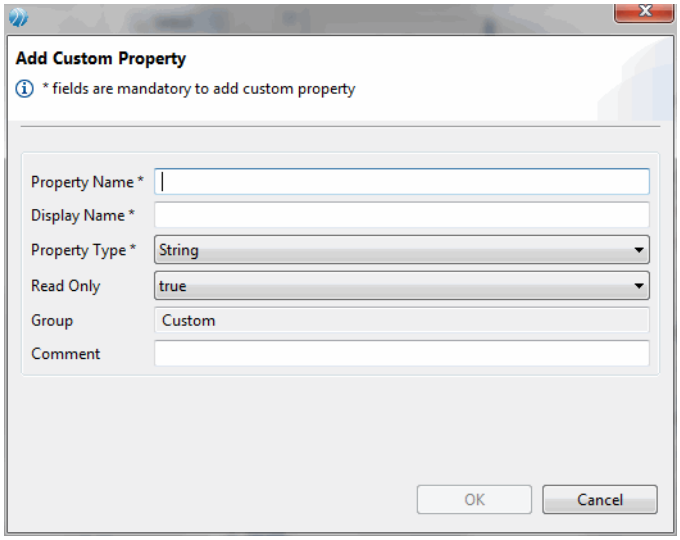
Using the Custom tab in the properties section you can add or delete a custom property.

Procedure

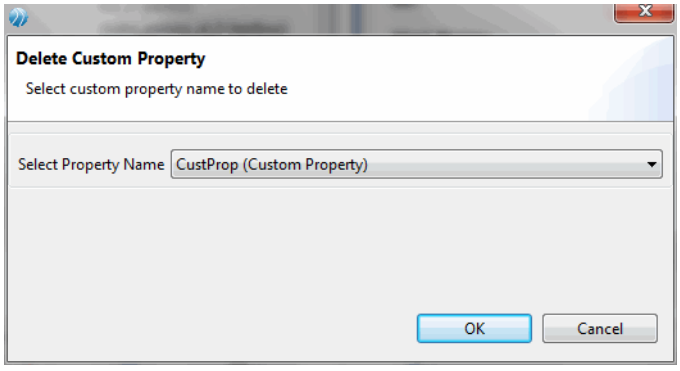
1. On the properties section, click **Custom**.



2. Click  to add new custom property. The Add Custom Property dialog is displayed.



3. Enter the custom property name and the display name in the **Property Name** and **Display Name** fields.
4. Select the property type from the **Property Type** drop-down list. Select whether the property should be read only from the **Read Only** drop-down list. The **Group** field in non-editable read only field. Specify any special instructions in the **Comment** field.
5. Click **OK**.
6. To delete a custom property click  . The Delete Custom Property dialog is displayed.



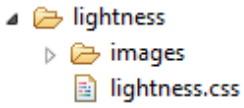
7. Select the custom property which you want to delete from the **Select Property** > **Name** drop-down list and click **OK**.

Adding Theme Section

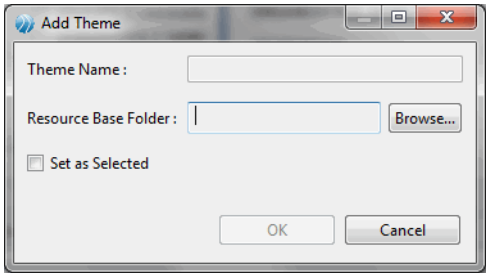
You can add custom theme to the UI pages. To add custom theme, follow these steps:

Procedure

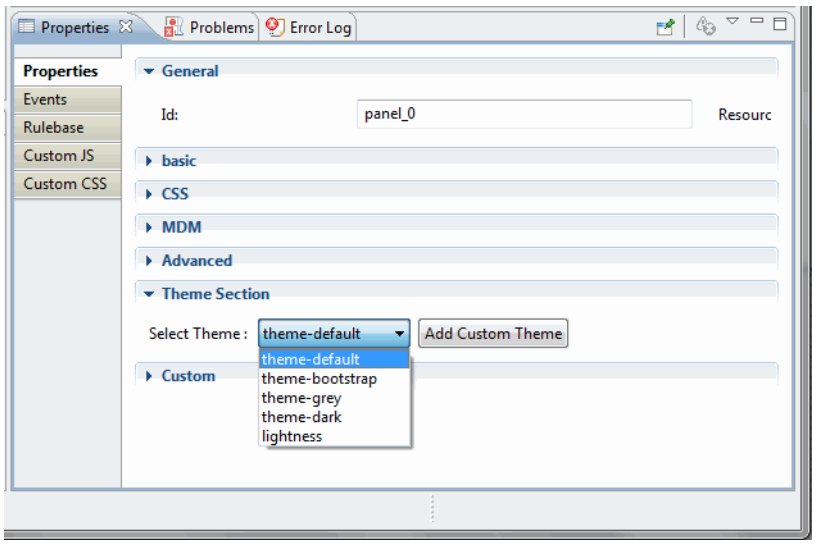
1. Create your own css theme or create one from jQuery theme roller. To access the jQuery theme roller, refer to http://jqueryui.com/themeroller/.
2. From the jQuery theme roller, download the theme as is or modify the theme for their fonts, colors, and texture and so on and then download it. Save the theme to your local folder. For example, a folder named lightness.
3. Open the folder and rename the jquery-ui.css name. Ensure that the css name and the folder name does not contain white spaces. For example, the `jquery-ui.css` is renamed to `lightness.css`.



4. From **Theme Section**, click Add Custom Theme. The **Add Theme** page is displayed.



5. Click **Browse** and navigate to the **Resource Base Folder**. and select the **.css** file. For example, `lightness.css`.
6. Select the **Set as Selected** check box to set the theme directly on the UI page from Add theme page.
7. The theme is added to the **Select Theme** drop-down list.



8. Select the newly added theme to add it the UI page.

Using Events

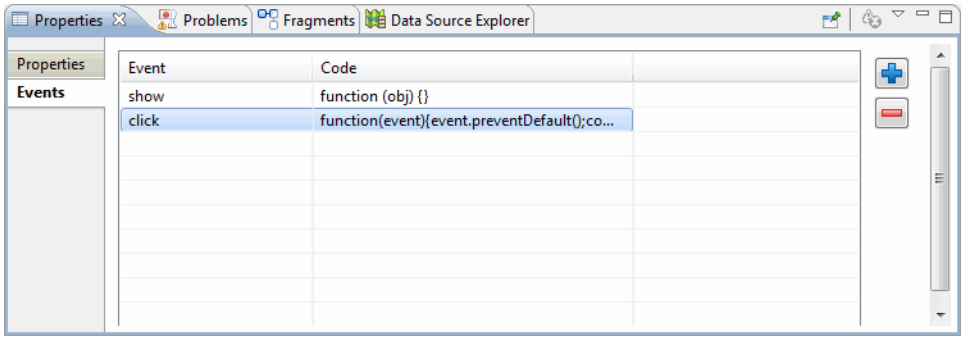
Just like properties, all items added to a page will contain some sort of events.

Events are functions that are called when an object does something or you have to perform certain action. For example, when you click a button, the “Onclick” event for the button is triggered. When a document is first loaded, the “Onload” event is triggered. You can intercept these events before or after they occur, allowing you control over what the events does.

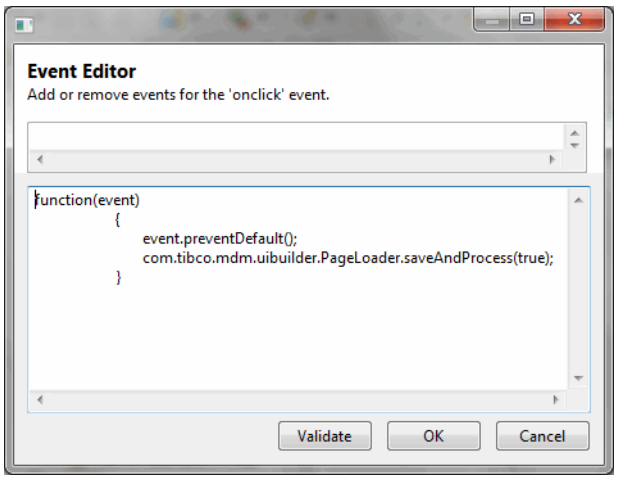
You can add, edit, or delete an event for a UI page.

Procedure

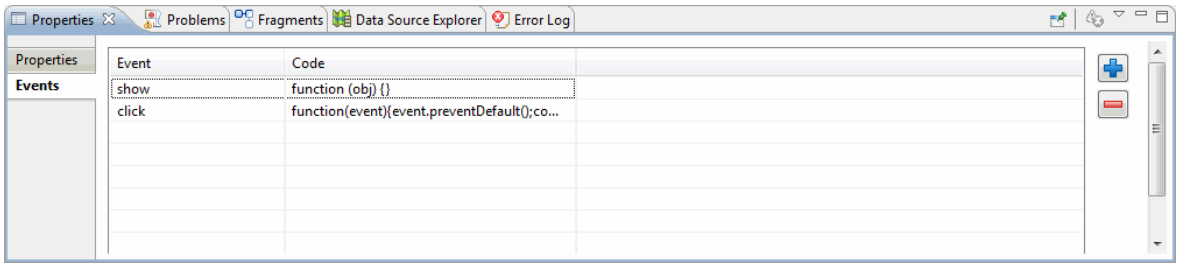
1. On the properties section, click **Events**.



2. Click . The Event Editor is displayed.



3. Enter the JavaScript function for the event. For events that do not include default functions, empty functions are created where all parameters defined in the event are passed into the empty function as arguments. This provides as a starting point when you want to override certain events.
4. Click **Validate** to validate that the JavaScript function is correct.
5. Click **OK**. Any code listed in the Event Editor will be displayed in-line in the MDM UI Builder Events view. In case you have defined a default JavaScript function in the widget XML file, the default function is displayed in-line in the events view.



6. Click  to add new a new event.

7. Select the event which you want to delete and click  . The event is removed from the list of events.

Creation of events is not supported for root panel.

Rulebase

Rulebase can be enabled for the UI page by clicking on the root panel in the tree outline viewer.

From the Property Section, select **Rulebase** tab and select **True** radio option. By enabling the rulebase, the UI invokes the rulebase services to ascertain the constraints on the repository fields.

If the rulebase is enabled, the deployed UI page identifies (by calling web service) which UI controls need to be shown differently based on the constraints on attributes. For example, if a particular attribute has valid value list then the associated UI widget with the attribute is replaced by a drop-down widget at runtime. However, this change will not be visible while designing the page.

The following rulebase constraints are supported in this release:

- **Assign constraint** - If the attribute has a constraint which has pre-assigned value to it then while adding a new record the attribute will get the value.
- **Valid values (uses drop-down UI control)** - If the attribute has drop-down (select action) in rulebase then on the deployed page the textbox is displayed as a drop-down with valid values list.
- **Hide constraint** - Attribute will be hidden. If the attribute group is hidden then associated attribute group tab or accordion pane will be hidden.
- **Read only constraint** - Attributes cannot be edited in the control.

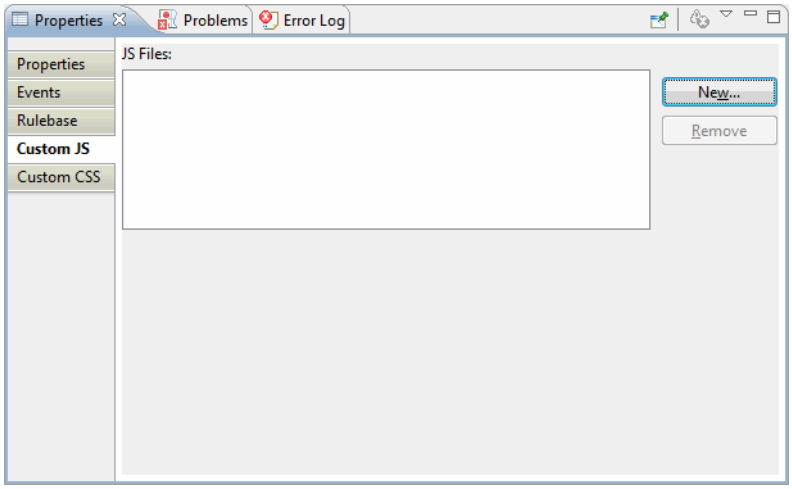
Using Custom JS

The Custom JS option is used for adding JavaScript files to the custom pages.

The JavaScript is more specific to the page and will not be shared with other custom pages. To add JavaScript file, follow these steps:

Procedure

1. On the **Custom JS** tab, click **New**.



2. Navigate to the folder where you have saved the JavaScript file and click **Open**.
3. The selected JavaScript file is displayed.

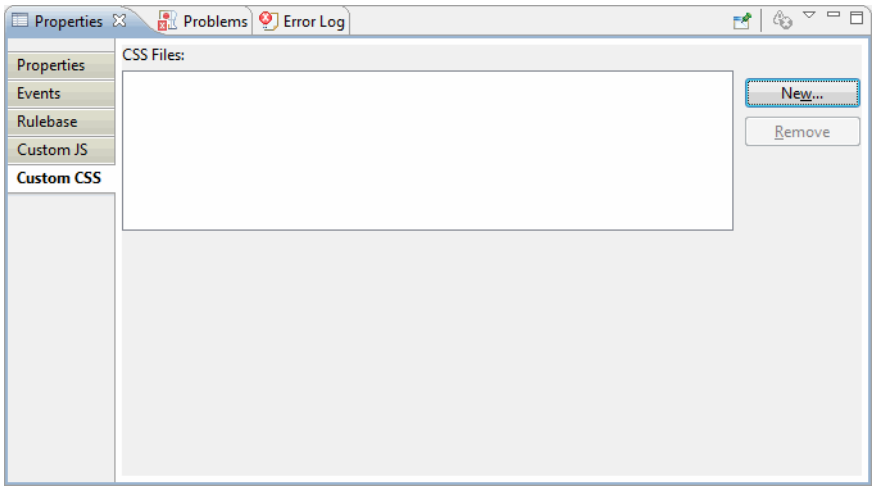
Adding Custom CSS

This is used for adding CSS files which are specific to the current page and will not be shared with other pages.

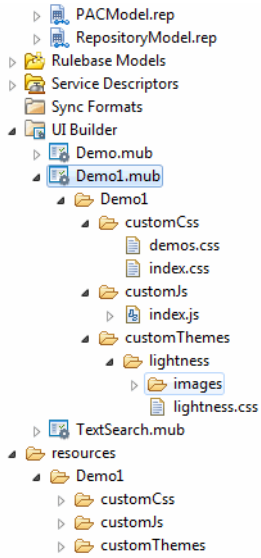
To add the CSS files, follow these steps:

Procedure

1. On the **Custom CSS** tab, click **New**.



2. Navigate to the folder where you have saved the CSS file and click **Open**.
3. The selected CSS file is displayed.
4. The newly created Custom CSS, Custom JS, Custom Themes, and Custom Images are stored in the folder that has same name as the UI page. You can view these folders in the project explorer. For example, **Demo1.mub** is the UI page and **Demo1** is the folder containing all the artifacts. A similar folder is created under the **resources** folder. For each UI page, there is a separate artifacts folder. You can modify the respective artifacts available for the selected UI page. The Custom CSS, Custom JS, Custom Themes, and Custom Images is also updated in the artifacts in the **resources** folder.

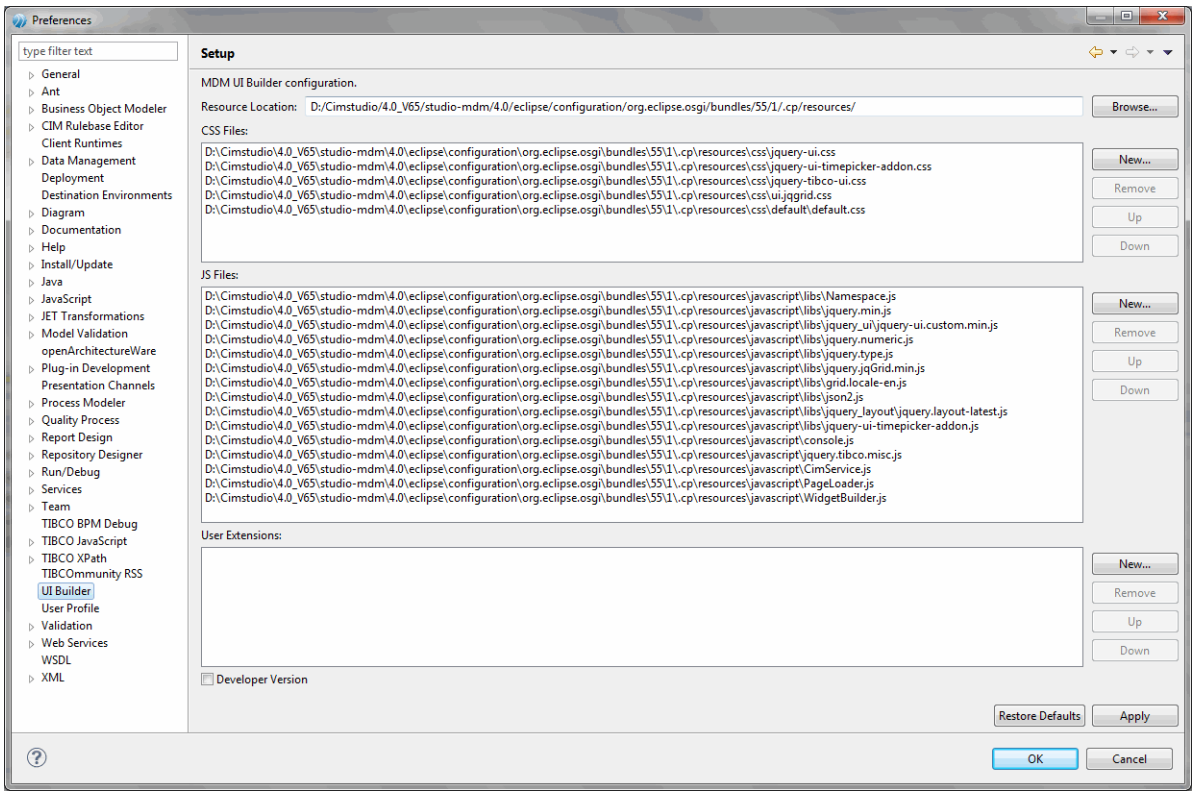


Configuring JavaScript and CSS files

Using the Preferences option, you can configure the JavaScript libraries and CSS files.

Procedure

1. On the **Windows** menu, click **Preference**.
2. The **Preferences** screen is displayed. Click **UIBuilder** > **Setup**.
3. The **Setup** screen for adding the MDM UI Builder Configuration is displayed.



4. Click **Browse** to locate the UI Builder resource folder. The selected resource folder is displayed in the **Resource Location** field.

5. The **CSS Files** field displays the list of CSS files in order to be included in each page creation. Click **New** to include more css files. The selected css file is displayed in the **CSS Files** field.
6. The **JS Files** field displays the list of JS files in order to be included in each page creation. Click **New** to include more JS files. The selected JS file is displayed in the **JS Files** field.
7. The U**ser Extensions** displays the custom widgets. Click **New** to in include more custom widgets.
8. Click **Apply**.
9. Click **Up** and **Down** to re-order the CSS, JS and User Extension files. Similarly click **Remove** to delete the CSS, JS, and User Extension files.
10. Click **Restore Defaults** to restore the default files.
11. Click **OK**.

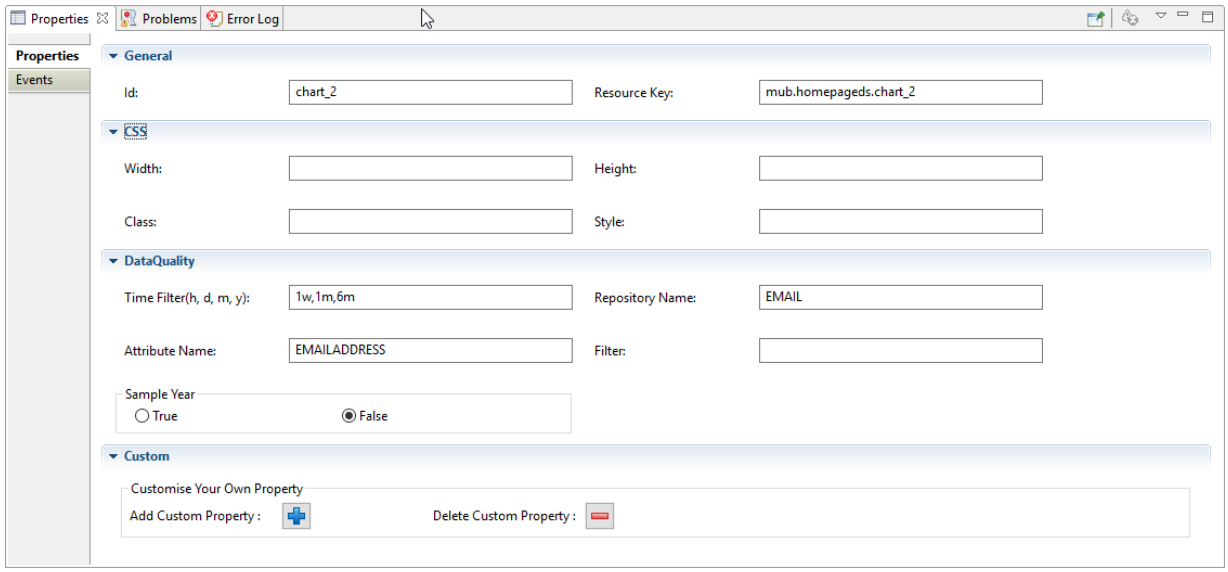
Chart Widget Properties

This section explains the properties for each widget palettes in the chart section.

- Chart
- MDM Statistics
- Summary Control

Chart

Chart Widget provides an intuitive way to monitor the data quality. Using this widget, you can define a line chart to monitor the data quality of attribute. In addition, you can also filter the data using time filter and category filter.

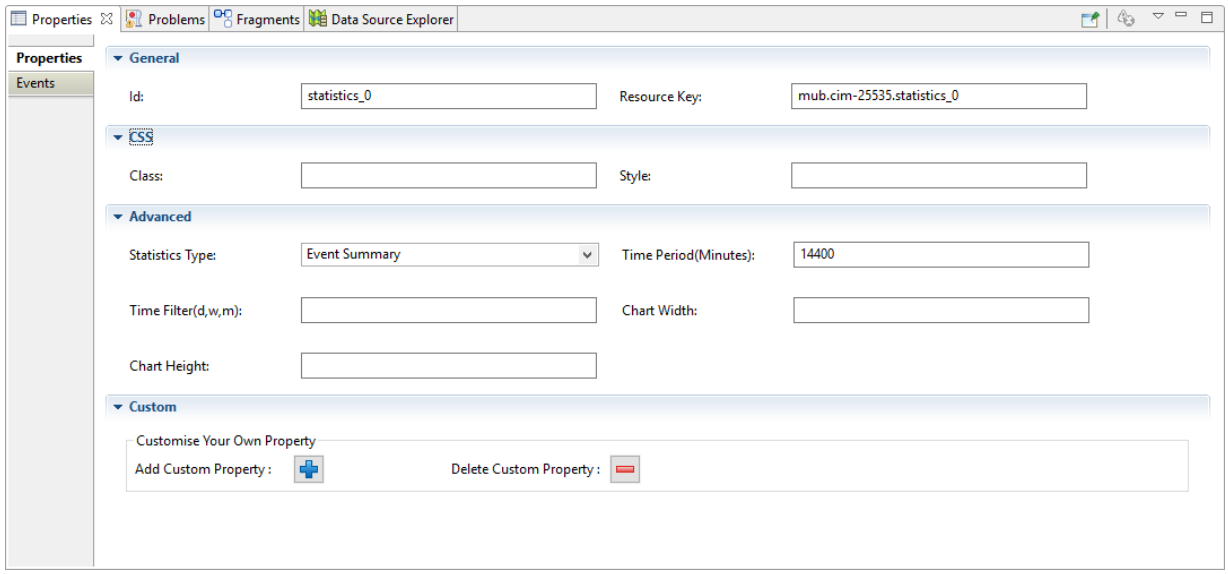


Properties	Description
DataQuality	
Time Filter (h,d,m,y):	Specify the time filter separated by comma. The symbols that can be used as time filters are 'h' (hour), 'd' (day), 'w' (week), 'm' (month), and 'y' (year). For example, "2h, 3d, 3w, 5m, 1y" defines five time filters. They are from two hours until current time, from three day until current time, from 3 weeks until current time, from five months until current time and from one year until current time.

Properties	Description
Repository Name:	The repository name of the MDM.
Attribute Name:	The name of the quality attribute.
Filter:	The value of the attribute in which quality attributes are categorized.
Sample Year	Specify whether the last one year attribute score info should be returned. The available options are *true* or *false*. Select *true* to return one year attribute score info.

MDM Statistics

The MDM Statistics widgets provides pie chart and number display to monitor the event summary, work item summary, and query summary. The event summary pie chart displays the number of success events and failure events. The query summary displays the total number of queries executed and average execution time in specified minutes. The work item summary pie chart displays the number of cancelled, closed, open and timeout events.

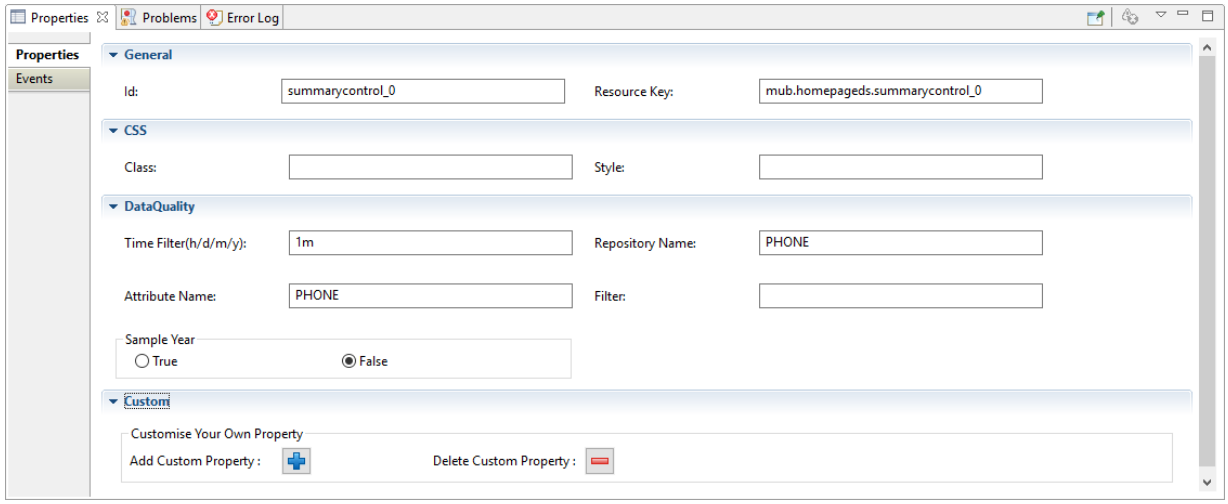


Properties	Description
Advanced	
Statistics Type:	Select the type of statistic summary. They are event summary, query summary and work item summary.
Time Period(Minutes):	Input time interval of event summary or query summary collection in minutes. The type of the value is integer. This value is used for event summary and query summary.

Properties	Description
Time Filter (d, w, m)	Input the time filter separated by comma. The symbols can be used as time filters are 'd', 'w' and 'm'. 'd' means a day; 'w' means a week; 'm' means a month. For example, "3d, 3w, 5m" defines five time filters. They are from three day until current time, from 3 weeks until current time and from five months until current time. This value is used for workitem summary.
Chart Width:	Input the width of the pie chart or number box. The type of value is integer such as 300.
Chart Height:	Input the height of the pie chart or number box. The type of value is integer such as 150.

Summary Control

Summary Control Widget provides a radial progress chart for the user to monitor the data quality of attribute.



Properties	Description
DataQuality	
Time Filter (h/d/m/y):	Specify the time filter separated by comma since the time the percentage data needs to be calculated until today. The symbols that can be used as time filters are 'h' (hour), 'd' (day), 'w' (week), 'm' (month), and 'y' (year). For example, '2d' means a radial progress chart will display the percentage data from two days until current time.
Repository Name:	The repository name of the MDM.
Attribute Name:	The name of the quality attribute.
Filter:	The value of the attribute in which quality attributes are categorized.

Properties	Description
Sample Year	Specify whether the last one year attribute score info should be returned. The available options are *true* or *false*. Select *true* to return one year attribute score info.

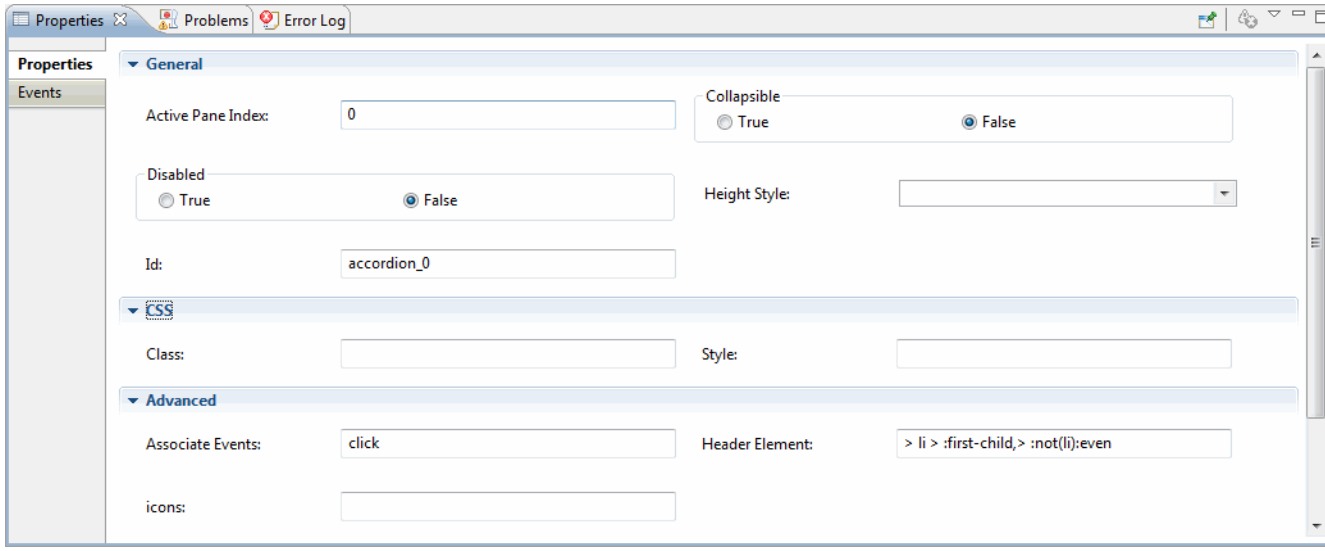
Container Widget Properties

This section explains the properties for each widget palettes in the container section.

- Accordion
- Accordion Pane
- Columns
- Dashboard
- Dashboard Panel
- Data Table
- Field Set
- Form
- iFrame
- Panel
- Panel Align 1
- Panel Align 2
- Sticky Footer
- Tab Panel
- Wizard

The properties which are common across widgets is explained in Common Properties section.

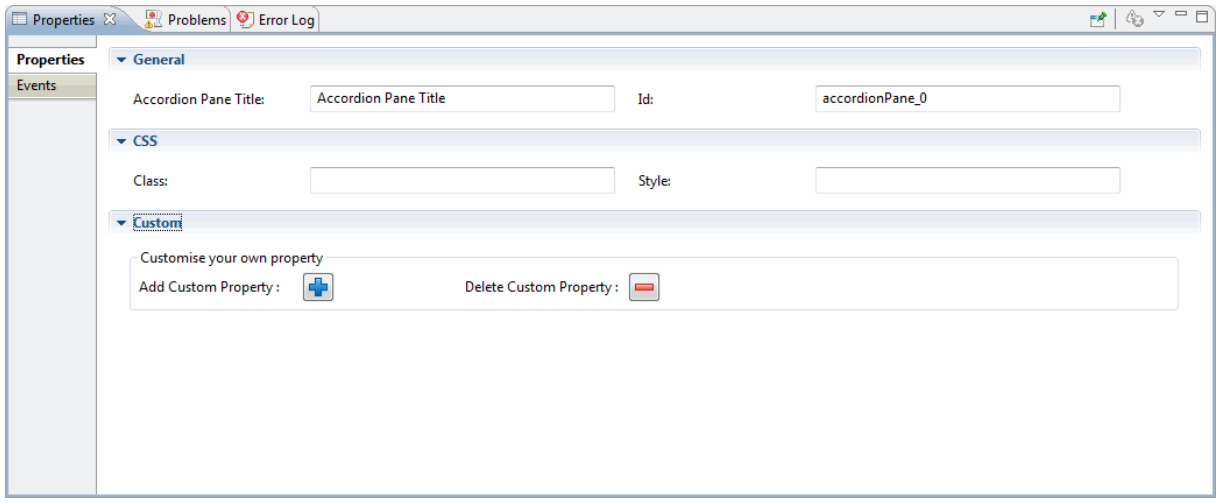
Accordion



Accordion Properties

Properties	Descriptions
General	
Active Pane Index:	The zero-based index of the panel that is active (open). A negative value selects panels going backward from the last panel. The default value is 0.
Collapsible	Whether all the sections can be closed at once. Allows collapsing the active section.
Disabled	Disables the accordion if set to true.
Height Style:	Controls the height of the accordion and each panel. Possible values: 'auto': All panels will be set to the height of the tallest panel. 'fill': Expand to the available height based on the accordion's parent height. 'content': Each panel will be only as tall as its content.
Advanced	
Associate Events:	The event that accordion headers will react to in order to activate the associated panel. Multiple events can be specified, separated by a space. For example, Click, mouseover and so on.
Header Element:	Selector for the header element, applied via.find() on the main accordion element. Content panels must be the sibling immediately after their associated headers.
Header Icon:	Specific the path of the header icon for the widget.
Active Header Icon:	Specific the path of the icon for the active header for the widget.

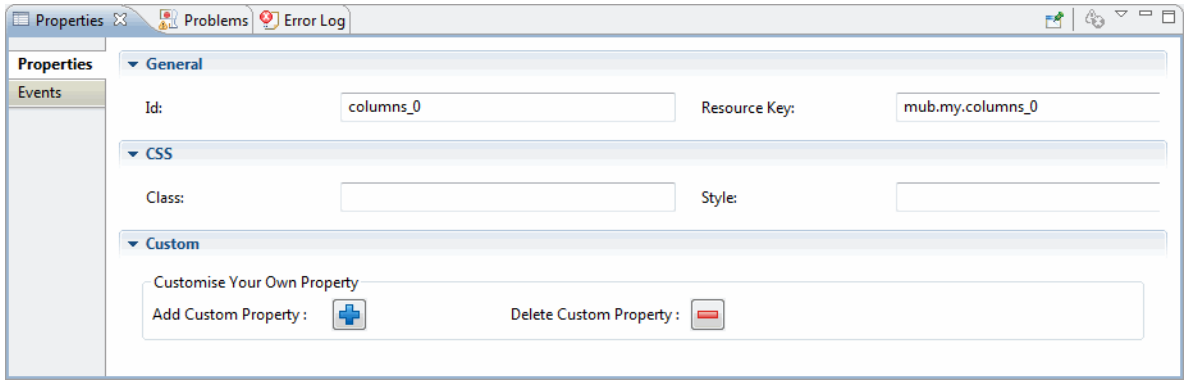
Accordion Pane



Accordion Pane

Properties	Descriptions
General	
Accordion Pane Title:	Set the accordion pane title.

Columns

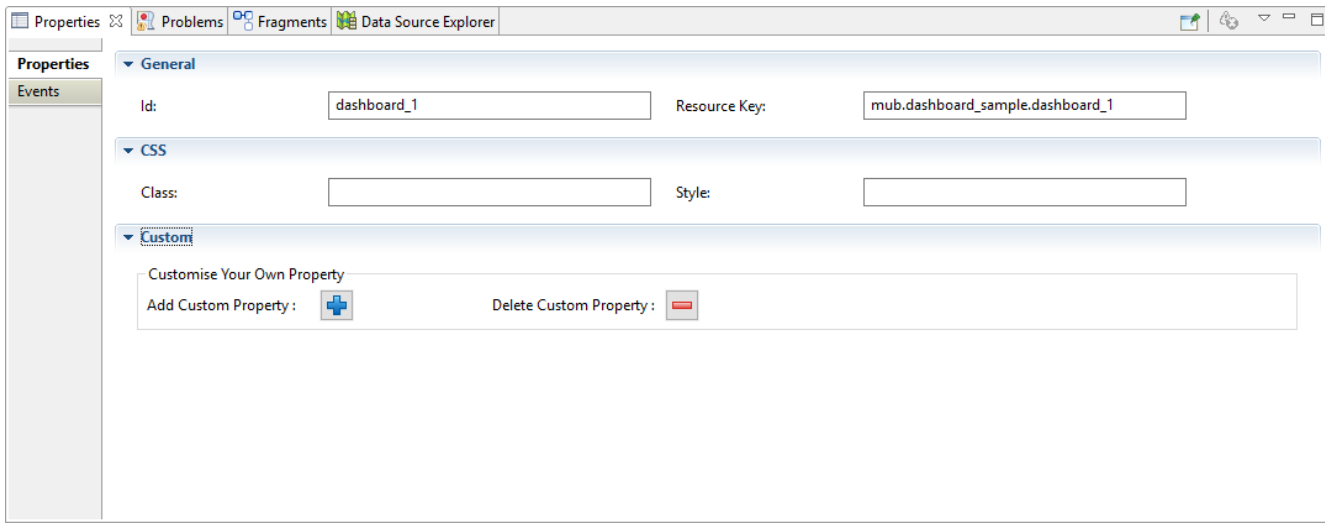


Columns Properties

Properties	Description
General	
Id:	The unique identification for the column.
Resource Key:	The unique resource key for the column.

Dashboard

The Dashboard widget is the outer container that holds dashboardpanel widgets with different columns.

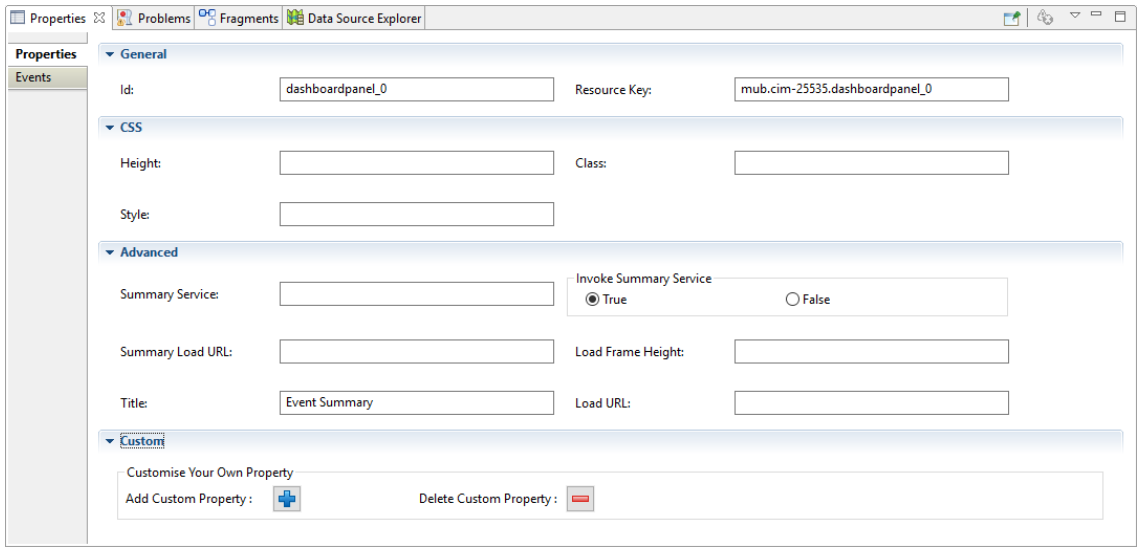


Dashboard Properties

Properties	Descriptions
General	
Id:	Specify the unique id for the widget.
Resource Key:	Specify the unique resource key for the widget.
CSS	
Class:	Specify the CSS class name for the widget.
Style:	Specify the CSS style name for the widget.

Dashboard Panel

The Dashboardpanel widget is the container that can hold any other widgets. It is displayed as a small container with a blue header on the dashboard. It automatically adjust the width within the columns.

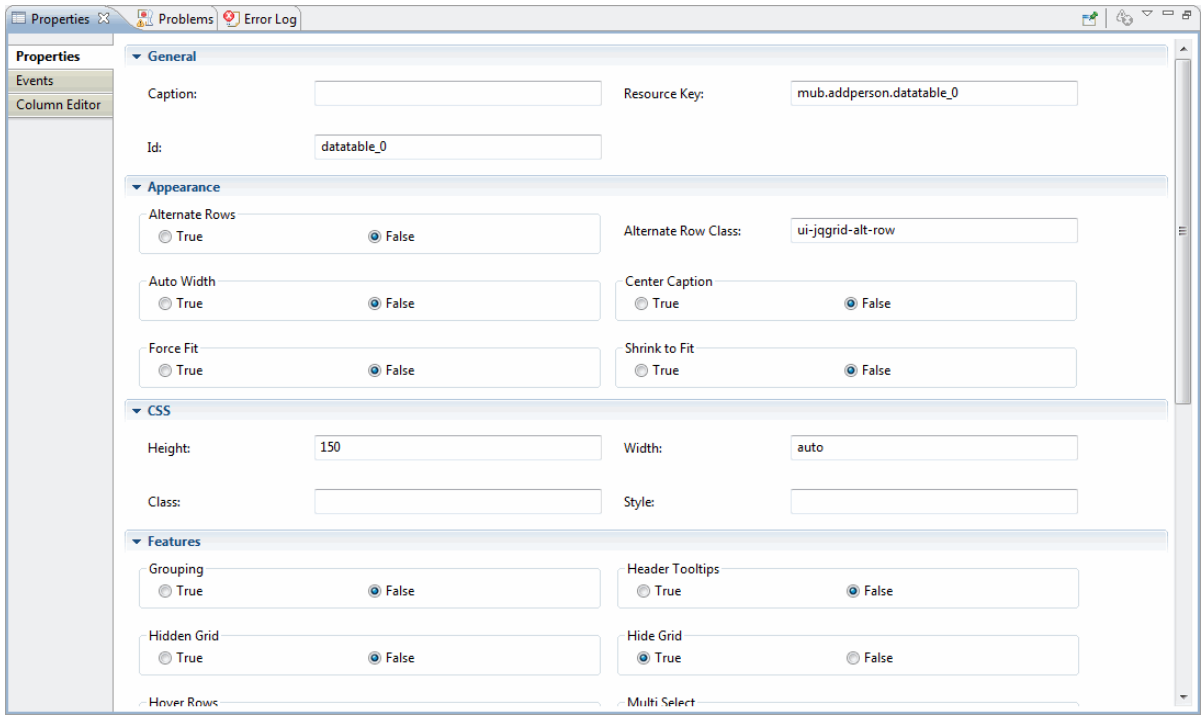


Dashboard Panel Properties

Properties	Descriptions
Advanced	
Summary Service:	Specify the name of the summary service you want to define.
Invoke Summary Service:	Select *True* if you want to control running the summary service for both dashboard view and detail view or not.
Summary Load URL:	The URL must start with "/eml/". If it is defined, the URL page is displayed in the dashboard view. By adding 'dashboard=true' value in the URL, removes the header and menu of the page. For example, '/eml/Inbox?dashboard=true'.

Properties	Descriptions
Load Frame Height:	Specify the height frame value of the dashboardpanel. You must define the width in pixels or percentage. For example, "100px" or "30%".
Title Info:	Specify the title name for the dashboard panel.
Load URL:	The URL must start with "/eml/". If it is defined, the URL page is displayed in the detail view. By adding 'dashboard=true' value in the URL, removes the header and menu of the page. For example, '/eml/Inbox?dashboard=true'.

Data Table



Data Table Properties

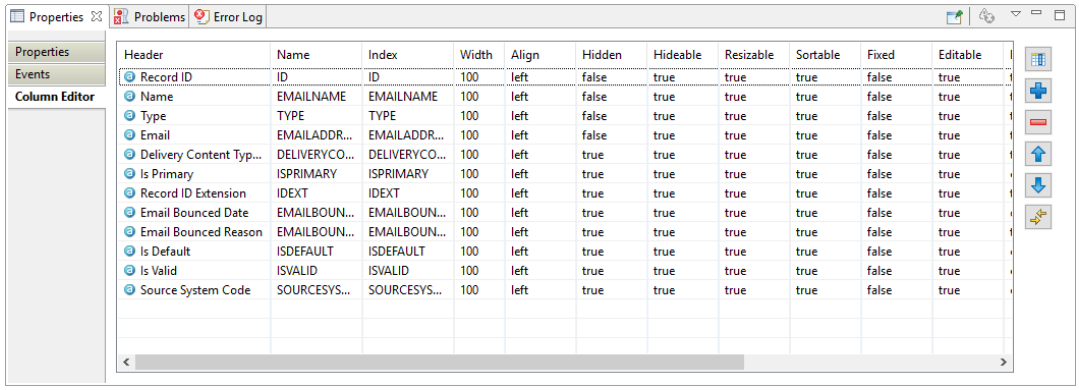
Properties	Descriptions
General	
Caption:	Defines the caption for the grid. This caption appears in the caption layer, which is above the header layer. If the string is empty the caption does not appear.
Appearance	
Alternate Rows	Set row backgrounds to alternate colors. The possible values are true and false. The default value is false.
Alternate Row Class	Specifies the alternate row CSS class. Default value is ui-jqgrid-alt-row.

Properties	Descriptions
Auto Width	When set to true, the grid width is recalculated automatically to the width of the parent element. This is done only initially when the grid is created. Default value is false.
Centre Caption	Set to true to display the caption of the table in the center. Default value is false.
Force Fit	If set to true, and a column's width is changed, the adjacent column (to the right) will resize so that the overall grid width is maintained (For example, reducing the width of column 2 by 30px will increase the size of column 3 by 30px). In this case there is no horizontal scroll bar. Default value is false.
Shrink to Fit	This option, if set, defines how the width of the columns of the grid should be re-calculated, taking into consideration the width of the grid. If this value is true, and the width of the columns is also set, then every column is scaled in proportion to its width. Default value is true.
Features	
Grouping	Enables grouping in the grid.
Header Tooltips	Enables header tooltips in the grid.
Hidden Grid	If set to true the grid is initially hidden.
Hide Grid	Enables or disables the show/hide grid button, which appears on the right side of the caption layer. Takes effect only if the caption property is not an empty string.
Hover Rows	When set to false the effect of mouse hovering over the grid data rows is disabled.
Multi Select	If this flag is set to true a multi selection of rows is enabled. A new column on the left hand side containing the checkboxes is added.
Column Reorder	When set to true, this option allows reordering columns by dragging and dropping them with the mouse.
Sort Columns	The column according to which the data is to be sorted when it is initially loaded. If this value is set and the index (name) matches the name from colModel, then an icon indicating that the grid is sorted according to this column is added to the column header. This icon also indicates the sorting is descending or ascending.
Sort Order	The initial sorting order (ascending or descending). The values are asc or desc.
View Records	If true, jqgrid displays the beginning and ending record number, out of the total number of records in the query. This information is shown in the pager bar (bottom right by default) in this format: 'View X to Y out of Z'.
Advanced	
Enable Pager	Set true, if you want to use a pager bar to navigate through the records.

Properties	Descriptions
Items per Page:	An array to construct a select box element in the pager in which we can change the number of the visible rows. Example: [10,20,30]
Default items per page:	Sets how many records you want to view in the grid.
Toolbar Position:	This option places a toolbar element at the specified location

Column Editor

After you create the datagrid you can change various settings of columns using column editor. The Column editor can be accessed from the properties section of the datagrid control.



Column Editor Properties

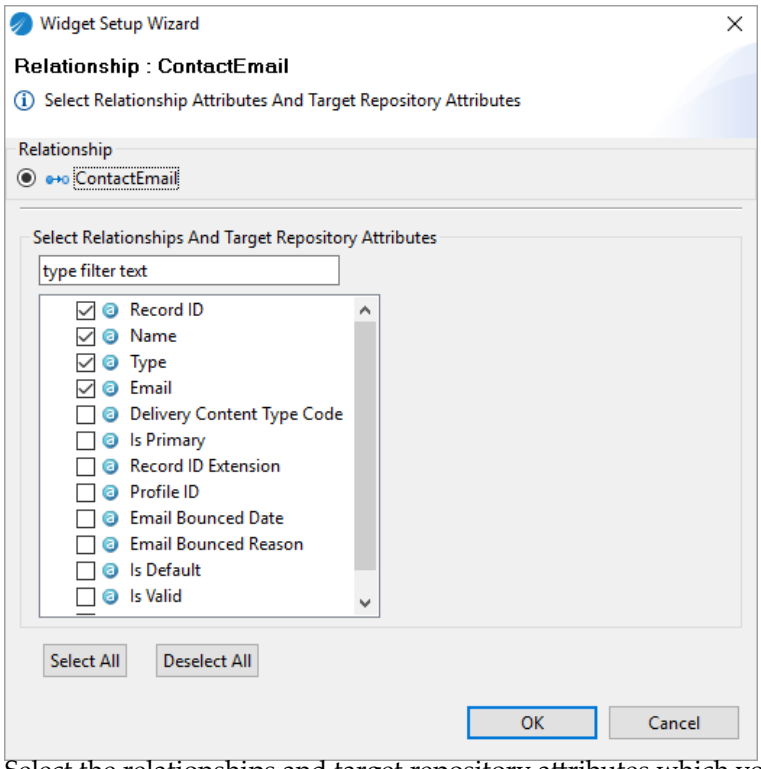
Properties	Descriptions
Header	Defines the heading for this column.
Name	Specify a unique name in the grid for the column. This is a mandatory column. Reserved words like the property name, event names cannot be used as names.
Index	Set the index name when sorting.
Width	Set the initial width of the column. This value supports only pixels. For example, 100 pixels. It does not support percentage.
Align	Specify the cell alignment in the Body layer. The supported values are left, center, and right.
Hidden	Specify this if the column is hidden at initialization. The default value is false.
Resizable	Specify this if you want the column to be resized.
Sortable	Specify this if you want the column to be sorted.
Fixed	Specify if you want the column width to be fixed. If the value is set to true, this option does not allow recalculation of the column width.

Properties	Descriptions
Editable	Specify if the field is editable. This option is used in cell, in-line and form modules
Edit Type	Specify the edit type for in-line and form editing. The possible values are text, textarea, select, checkbox, password, button, image, and file.
Edit Options	It is used for adding events to the column. For example, {"dataEvents": [{"type":"keypress", "fn":"function(){alert('key pressed');}"}]} It is mainly used for validation. For more information refer to http://www.trirand.com/jqgridwiki/doku.php?id=wiki:common_rules
Formatter	The predefined types (string) or custom function name that controls the format of column field. For more information refer to http://www.trirand.com/jqgridwiki/doku.php?id=wiki:colmodel_options

Working with Column Header

Procedure

1. To add a new column Click  and add column details. The newly added column is displayed in the canvas.
2. To delete a column, select the column and click  . The deleted column is removed from the datagrid in the canvas.
3. To hide or unhide a column, Click  . Select the column which you want to hide or unhide. Based on action the column is either hidden or visible in the canvas.
4. To re-order the column you can either use the  up arrow or  down arrow. In addition, you can also re-order the cloumn by drag and drop of the attribute. The column is re-ordered in the canvas.
5. To sync the datagrid columns with the repository, Click  . A wizard with target repository attributes and relationships attributes is displayed.



Select the relationships and target repository attributes which you want to configure as column when the datagrid is created for the relationship and click **OK**.

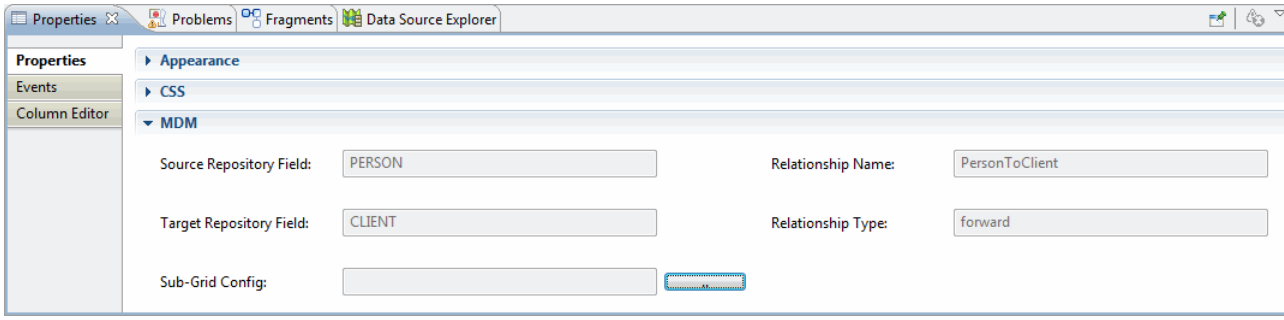
Subgrid

The first level is displayed as Datagrid and the second level as Subgrid.

The Subgrid supports multi level relationships. The Subgrid is displayed as a grid within a selected row and is configurable. The Subgrid can be configured using the property section.

Limitations of a Subgrid are as following:

- You cannot drag and drop any widget or repository to a datatable or subgridtable.
- You can cannot re-order (move up or move down) a subgridtable. It is disabled for subgrid. However, you can only re-order the datatable.

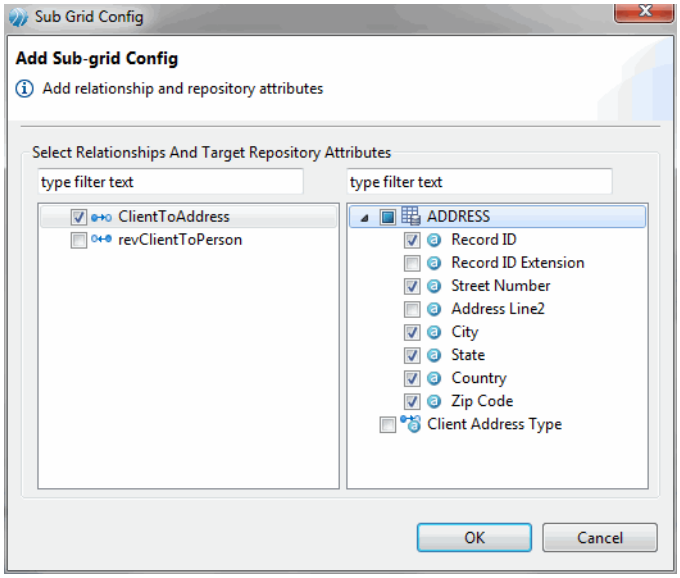


Configuring Subgrid

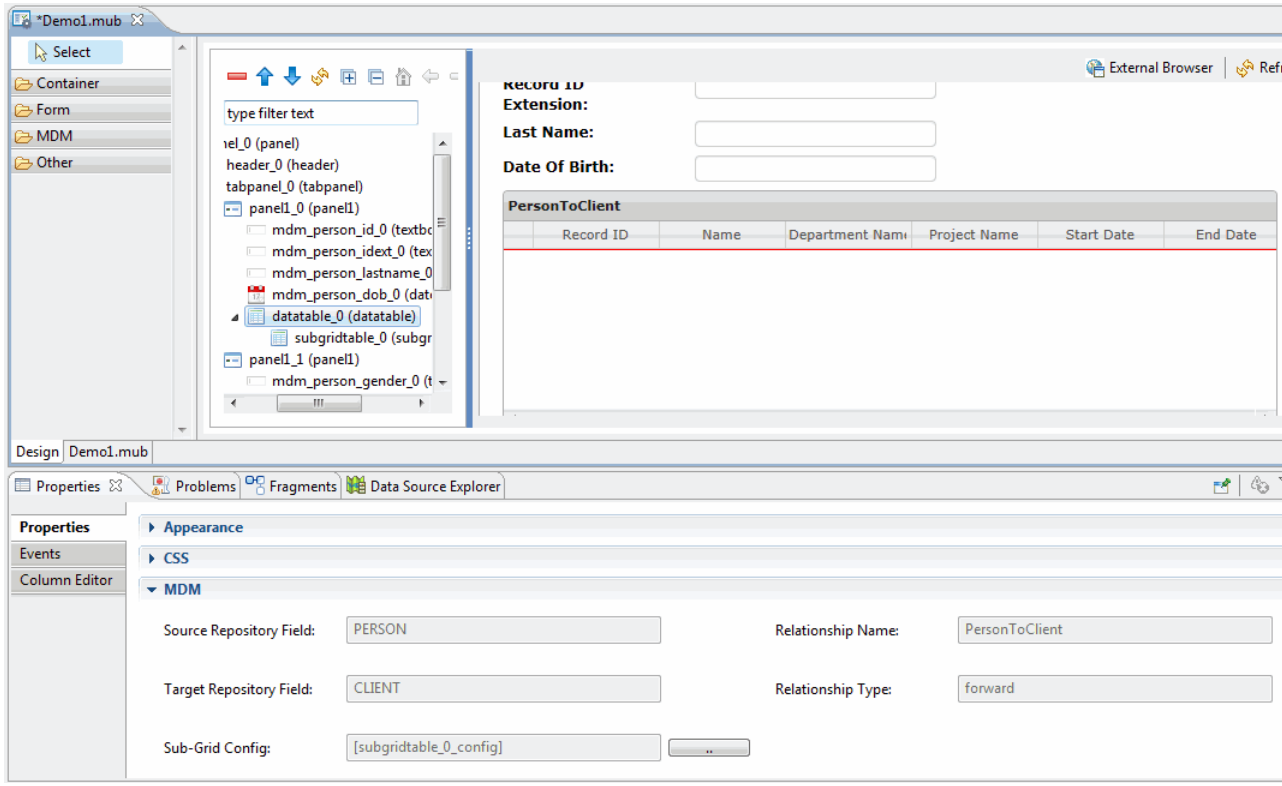
Procedure

1. To configure the subgrid, click on the Datagrid property section, click on **MDM** tab.
2. Click the button corresponding to **Sub-Grid Config** field.

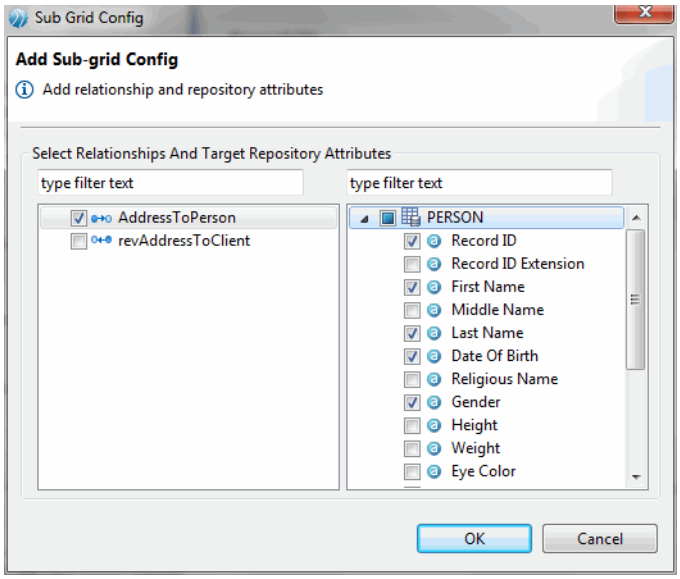
3. The **Add Sub-grid Config** wizard is displayed.



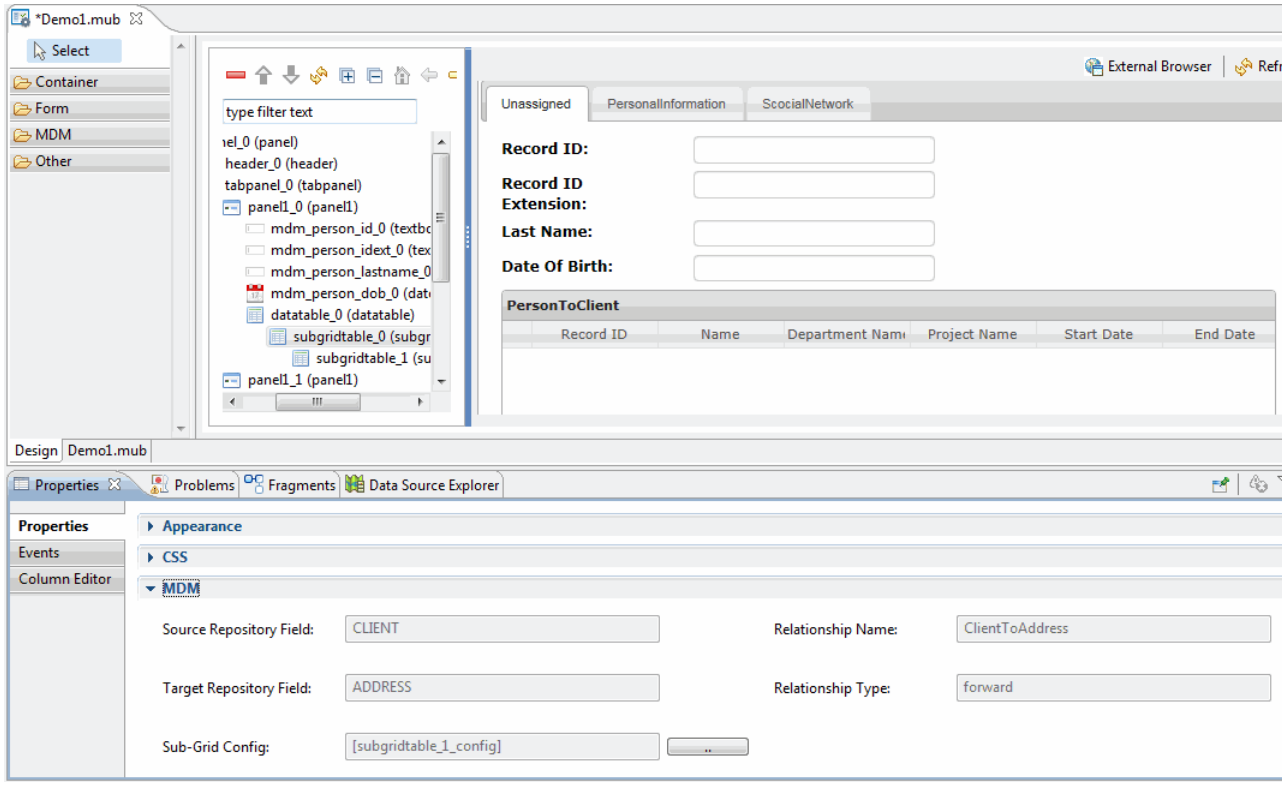
4. Select the relationship (forward or reverse) and attributes and click **OK**.
5. The newly added **subgridtable_0** is displayed under the **datatable_0**.



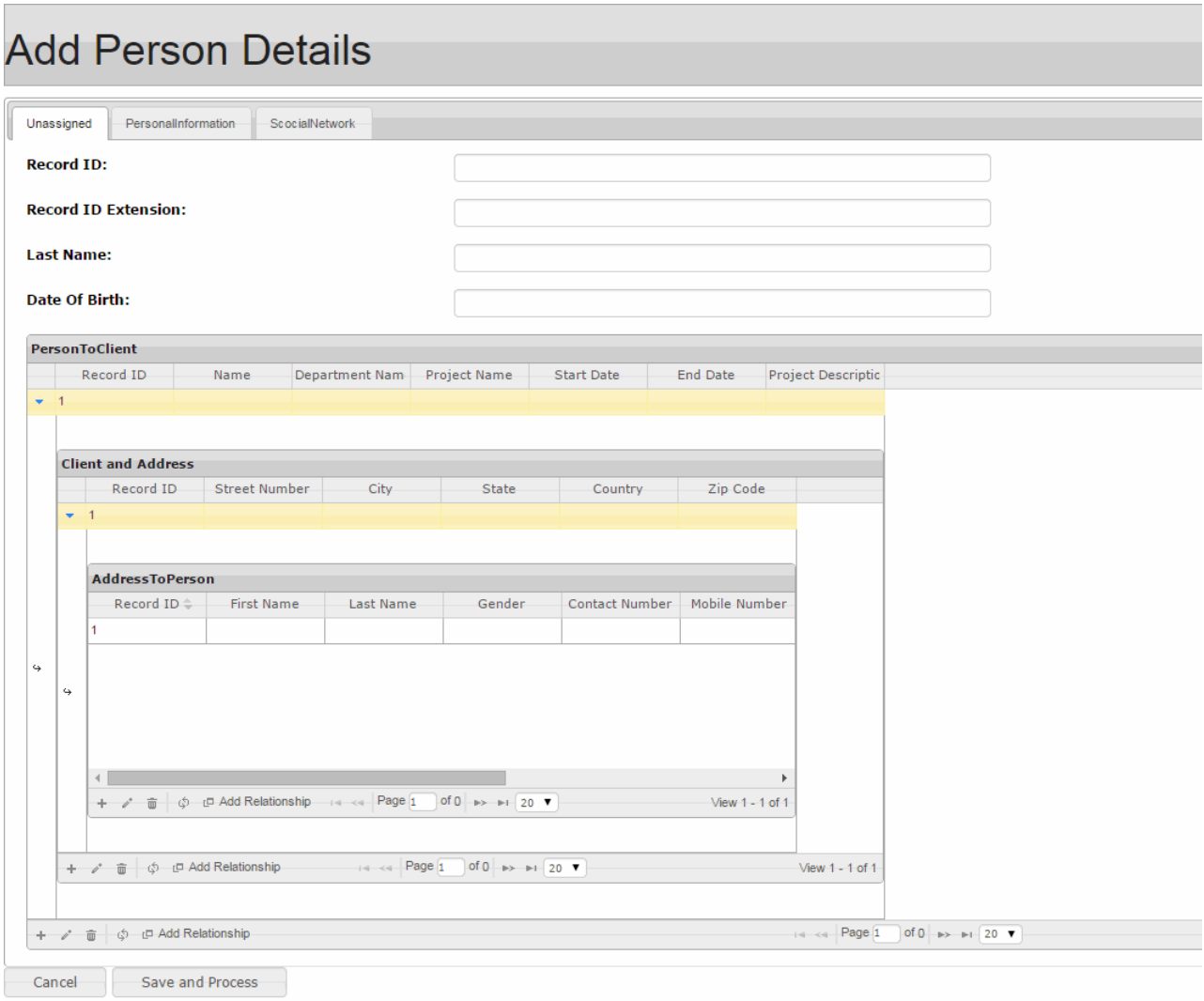
6. To add multi level relationship in the subgrid, click on the Subgrid property section, click on **MDM** tab.
7. Click the button corresponding to **Sub-Grid Config** field.
8. The **Add Sub-grid Config** wizard is displayed.



9. You can add relationship attribute on **Add Sub-grid Config** wizard by selecting the relationship attribute. Select the relationship (forward or reverse) and attributes and click **OK**.
10. The newly added **subgridtable_1** is displayed under the **subgridtatable_0**.

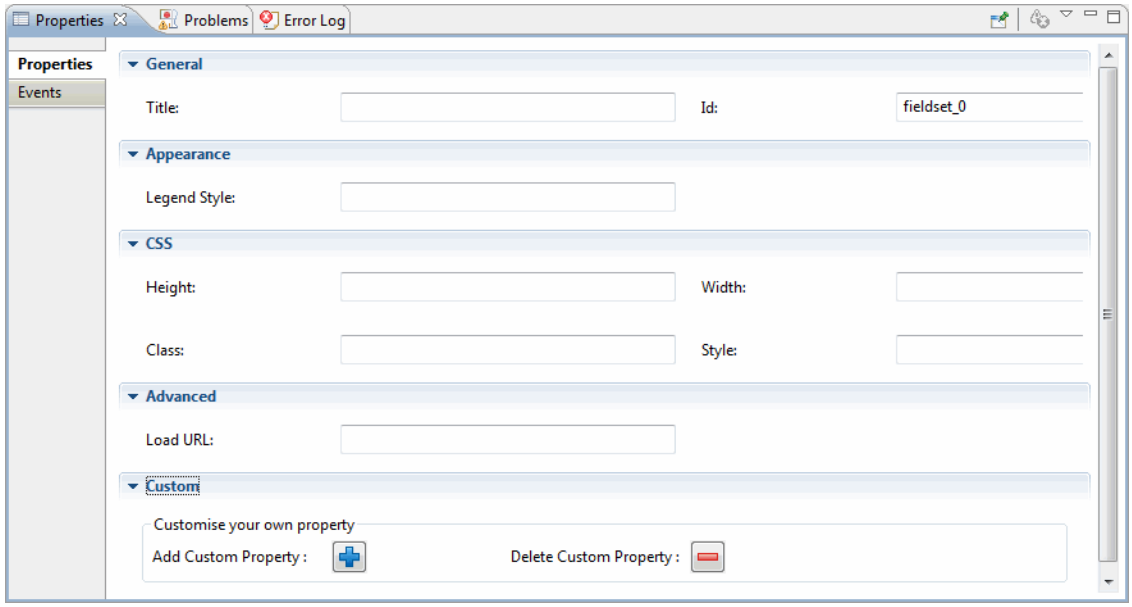


11. You can modify the caption of the subgrid using the **General** tab in the property section. For example, the **ClientToAddress** subgrid is modified to **Client and Address**.
12. After the page is deployed, the subgrid can be displayed by clicking  on the first cell of the row.



You can add multiple relationships at the same level and at different levels.

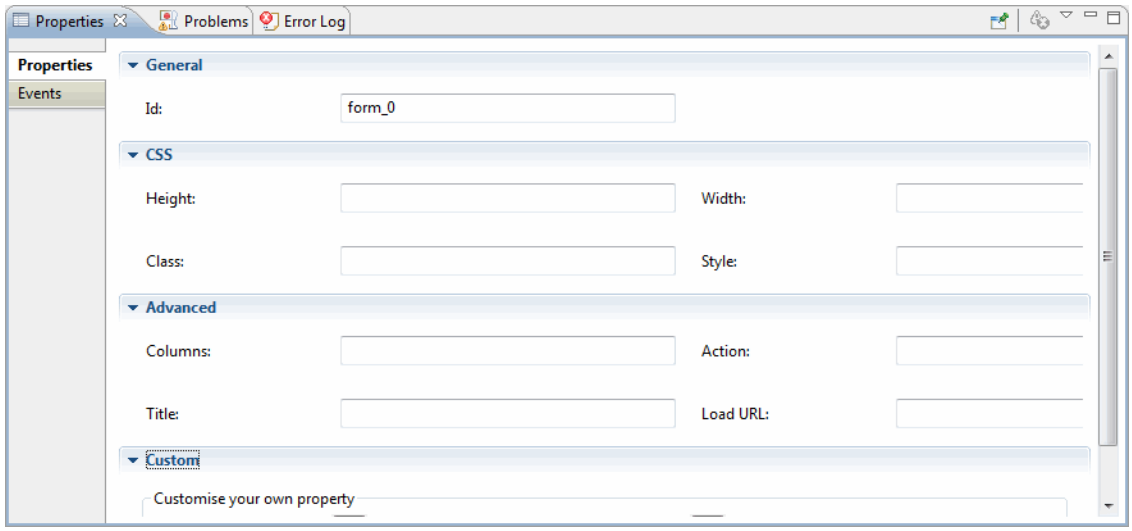
Fieldset



Fieldset Properties

Properties	Descriptions
Appearance	
Legend Style	Specify the CSS for the header.

Form

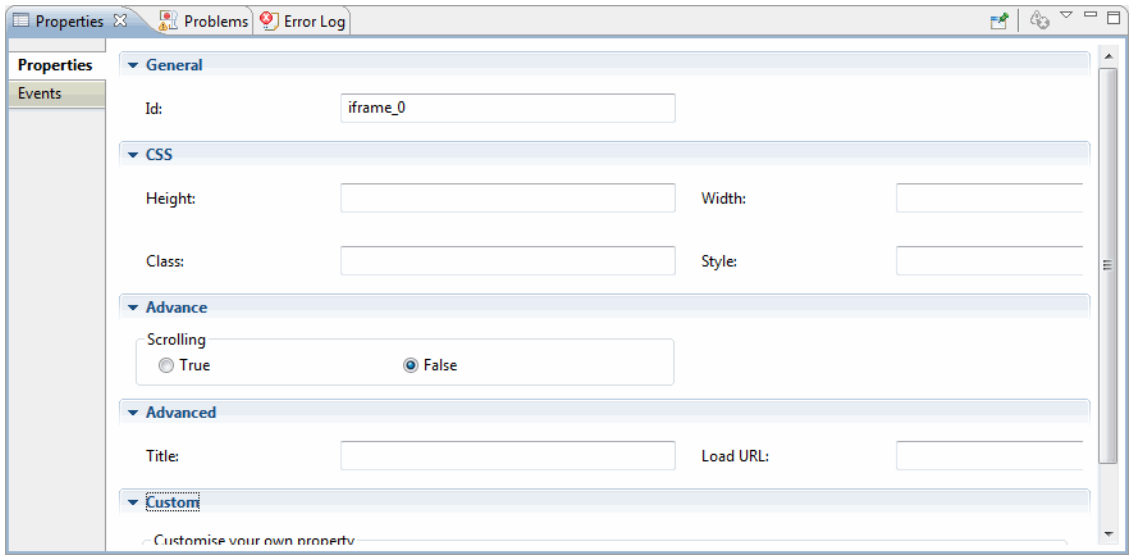


Form Properties

Advanced

Actions	Specifies where to send the form-data when a form is submitted. Normally it will be an url or the deployed servlet name.
Title	Specify the title for the Form.

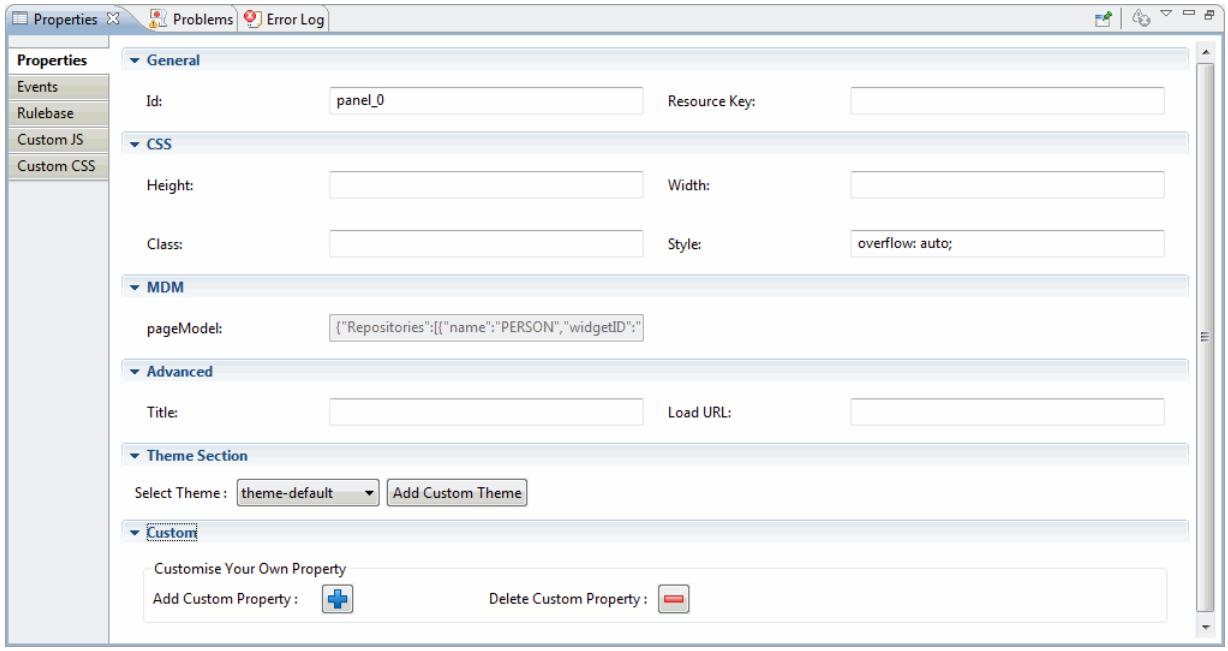
iframe



iframe Properties

Properties	Descriptions
Advance	
Scrolling	To enable or disable the scroll bars inside the iframe. Default value is false.
Advanced	
Title	Specify the title for the iframe.
Load URL	Specifies the address of the document to embed in the <iframe>. For example, http://www.tibco.com

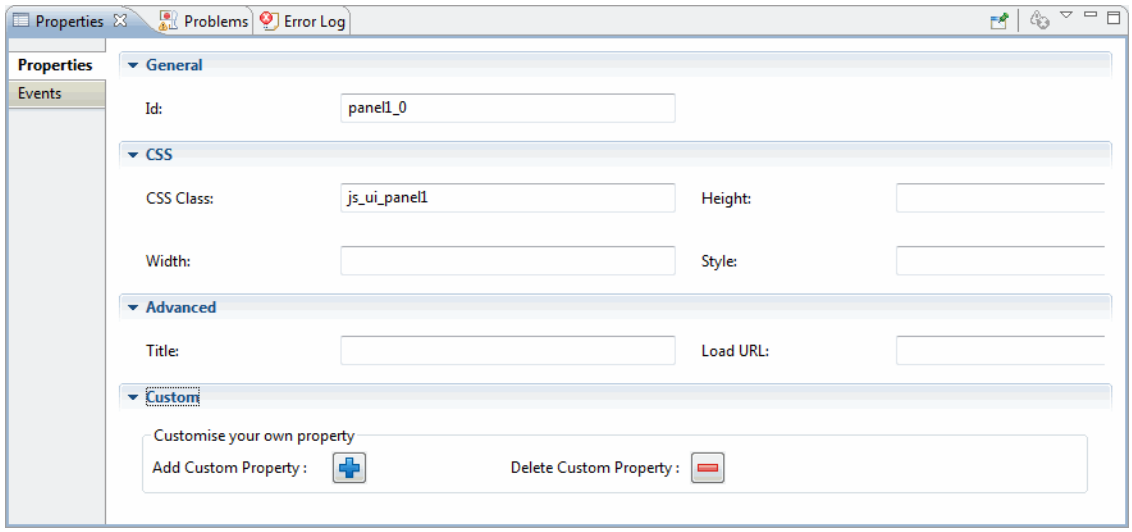
Panel



Panel Properties

Properties	Descriptions
Advanced	
Title:	Specify the title for the panel.

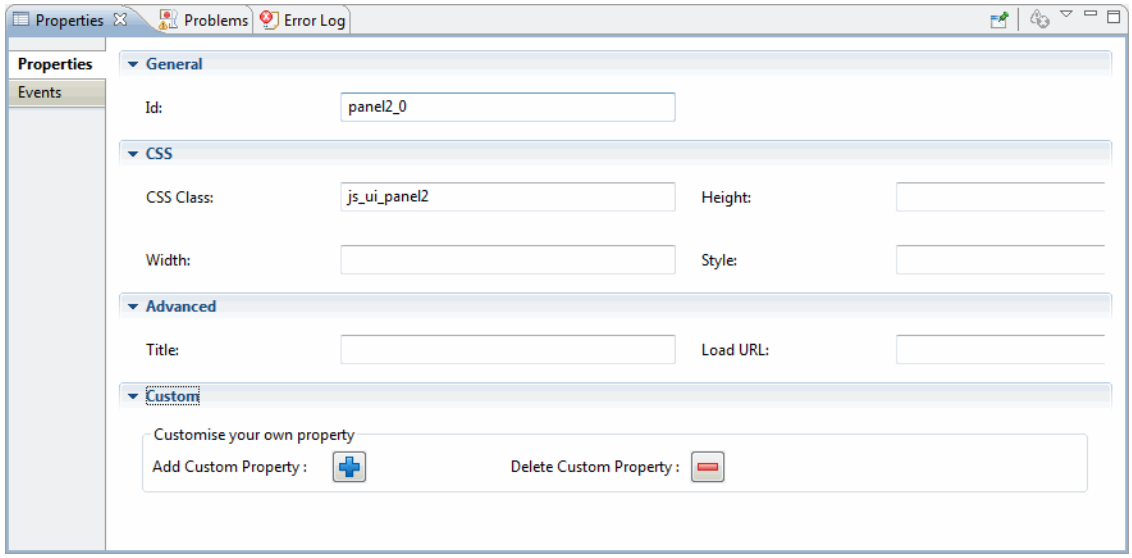
Panel Align 1



Panel Align 1

Properties	Descriptions
CSS	
CSS Class:	Specify the widget class. Default value is js_ui_panel1.

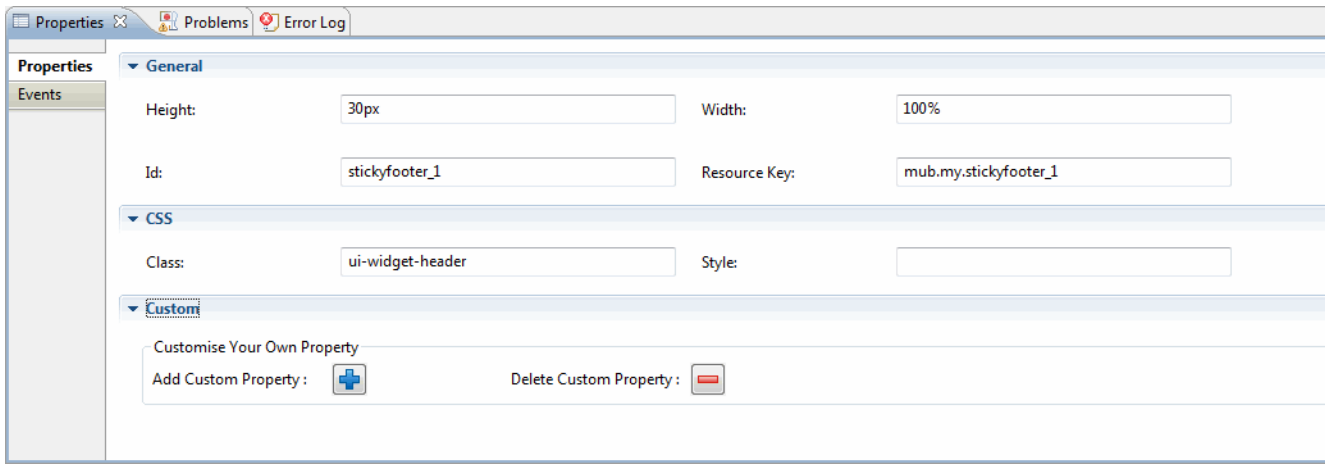
Panel Align 2



Panel Align 2

Properties	Descriptions
CSS	
CSS Class:	Specify the widget class. Default value is js_ui_panel2.

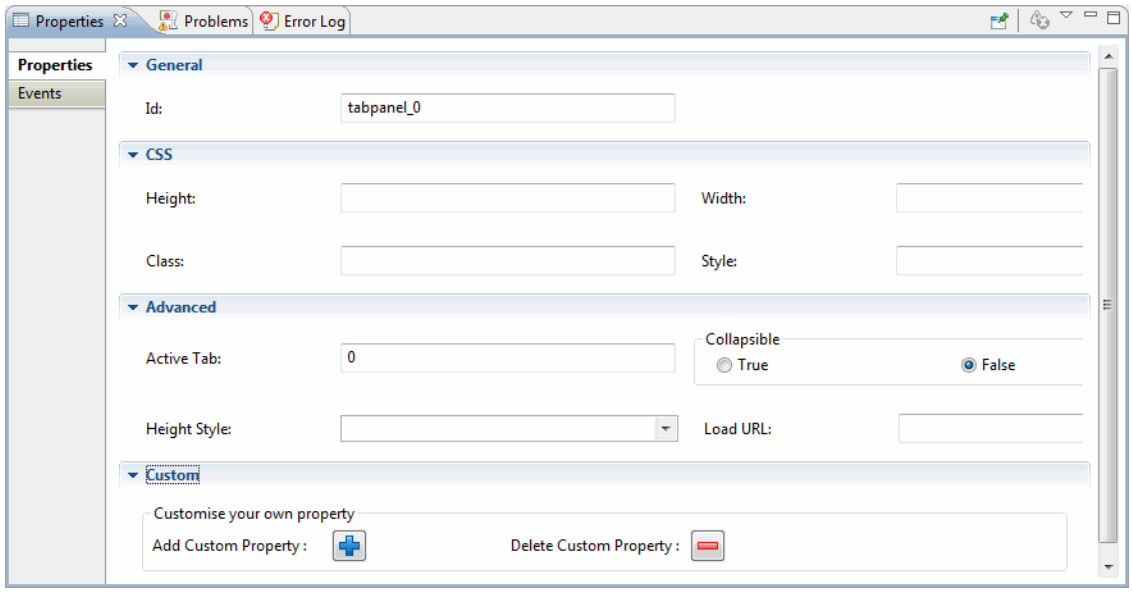
Sticky Footer



Sticky Footer Properties

Properties	Description
General	
Height:	Specify the height of the sticky footer. Default value is 30px.
Width:	Specify the width of the sticky footer. Default value is 100%
Resource Key:	Specify the resource key. Default value is mub.PageName.WidgetID. For example, mub.my.stickyfooter_1.
CSS	
Class:	Specify the widget class. Default value is ui-widget-header.

Tab Panel

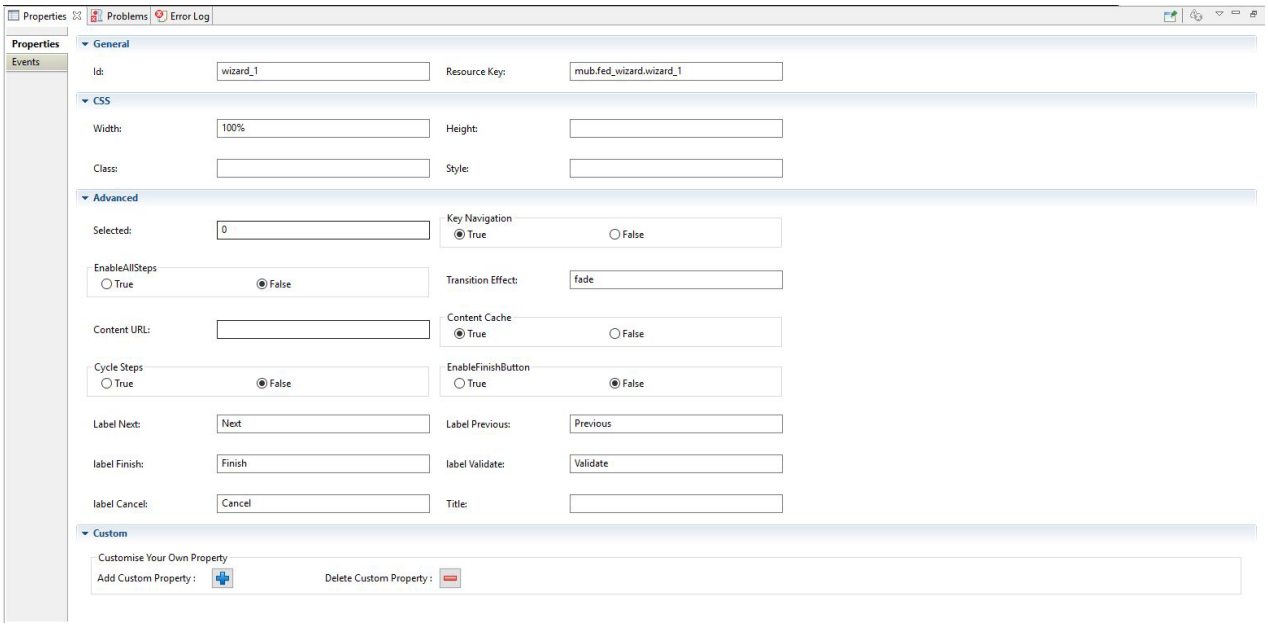


Tab Panel Properties

Properties	Description
Advanced	
Active Tab:	Specify the tab which has to be active. The default value is 0
Collapsible	When set to true, the active panel can be closed.
Height Style:	Controls the height of the accordion and each panel. Possible values: 'auto': All panels will be set to the height of the tallest panel. 'fill': Expand to the available height based on the accordion's parent height. 'content': Each panel will be only as tall as its content.

Wizard

The Wizard widget is for making multi-step workflows that do not necessarily require any server request. You can drag data model into it and the data model displays a wizard. The width of wizard is automatically shaped according to the browser's resolution. You can customize the content of buttons and that of the event handler.



Wizard Properties

Properties	Descriptions
Advanced	
Selected:	Specify a numberic value as a step to which you want to navigate. The indexing starts with 0, so if you specify a value 2, then the selection displays the Step 3 in the wizard.
Key Navigation:	Specify *True* if you want to navigate to Previous and Next step using the keyboard arrow keys.
EnableAllSteps:	Specify *True* if you want to enable all the steps in the wizard. In addition, you can directly navigate to any steps by clicking on it. If you set this field to *False*, all steps are disabled and you have to click the Next button to enable the step.
Transition Effect:	Specify the transistion effect you want to set when you navigate from one step to another. The supported transistion effects are "fade", "slide", and "sideleft".
Content URL:	Specify if you want to enable Ajax content loading. If you do not specify any value, the Ajax content loading is disabled.

Properties	Descriptions
Content Cache:	Specify *True* if you want to enable caching of the content on ajax content mode. The contents are fetched from the content url on the first load of the step.
Cycle Steps:	Specify *True* if you want to navigate back to step one from the last step on click of the next button.
EnableFinishButton:	Specify *True* if you want to enable the Finish button from step one. If you set it to False, the Finish button is enabled only after reaching the last step.
Label Next:	Specify the label for the Next button.
Label Previous:	Specify the label for the Previous button.
Label Finish:	Specify the label for the Finish button.
Label Validate:	Specify the label for the Validate button.
Label Cancel:	Specify the label for the Cancel button.
Title:	Specify the title for the page. This title is displayed corresponding to the Repository Name. For example, :{Repository Name:} {Title}

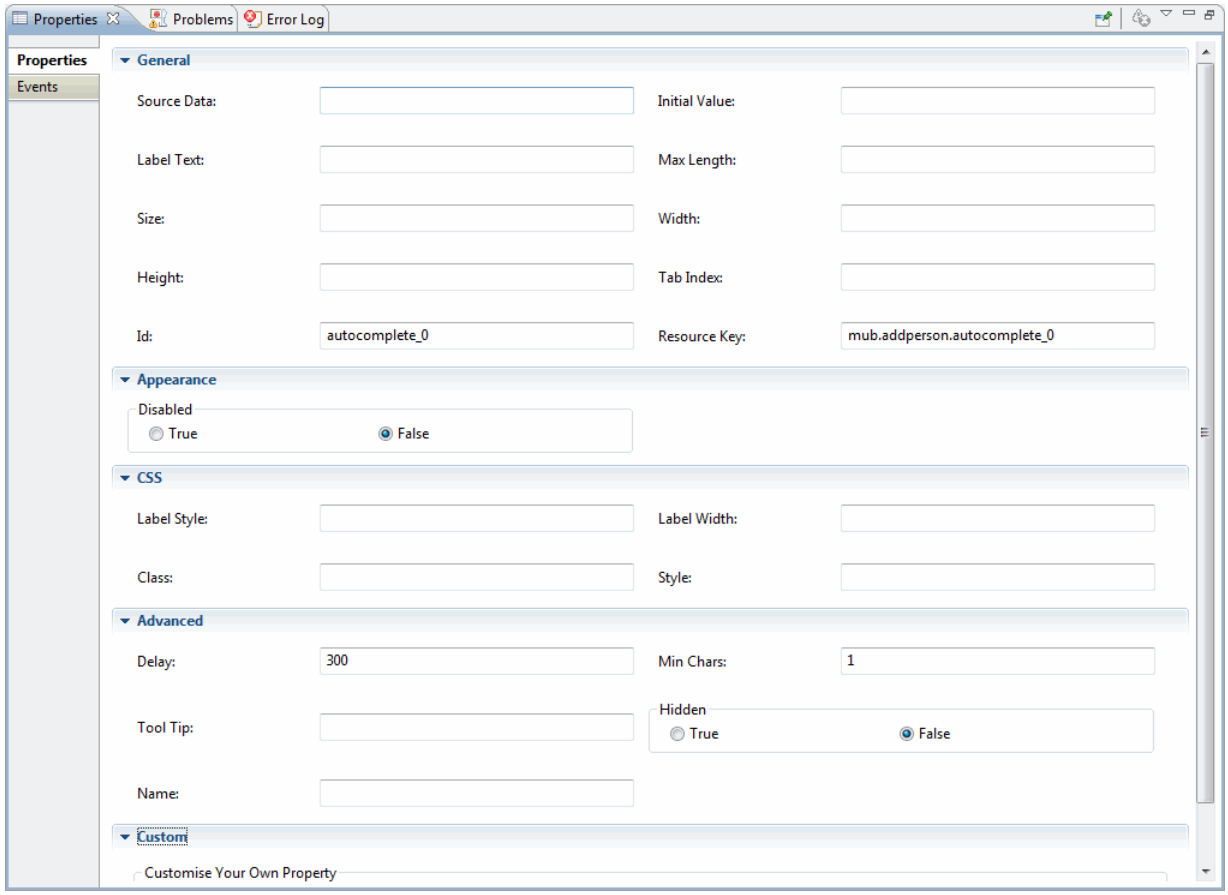
Form Widget Properties

This section explains the properties for each widget palettes in the form section.

- Autocomplete
- Button
- Combobox
- Datepicker
- Fileupload
- Timestamp Picker
- Header
- Horizontal Line (hr)
- Label
- Line Break
- Link
- Multivalue
- Numeric Textbox
- ReadOnly Textbox
- Textarea
- Textbox
- Link
- URL

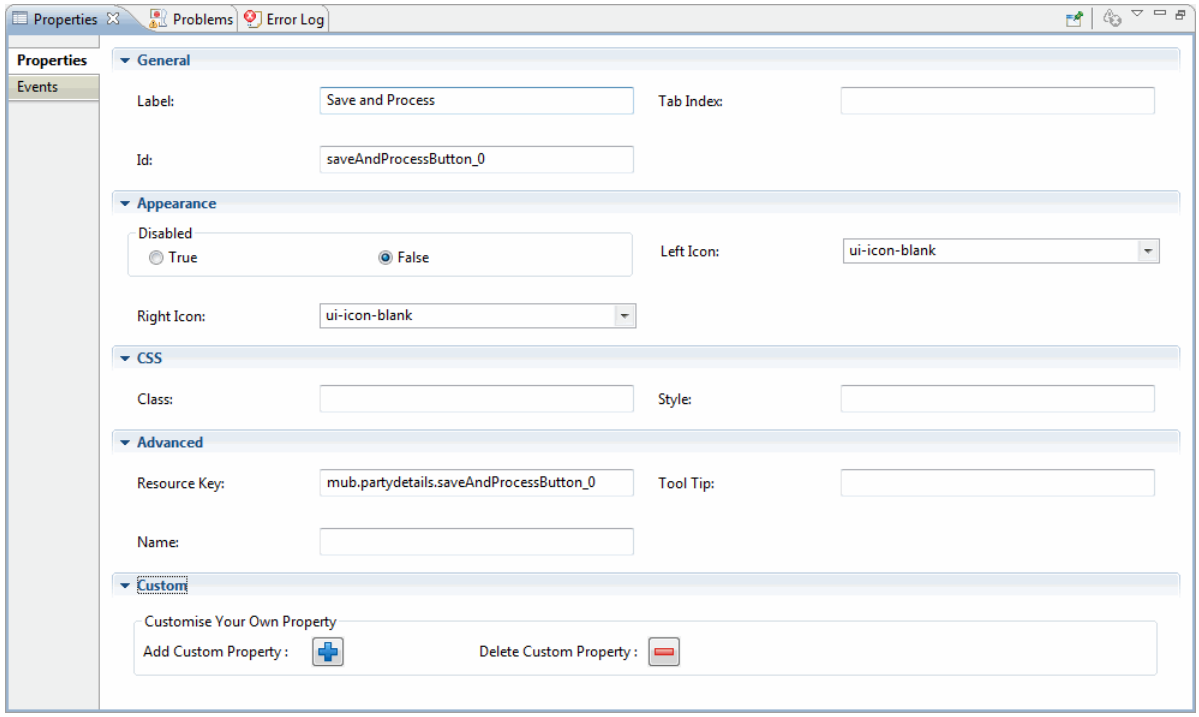
Autocomplete

The autocomplete widget helps you quickly find and select from a pre-populated list of values as you type ahead. This helps in leveraging searching and filtering.



Properties	Description
General	
Source Data:	Defines the data to use, must be specified. Types allowed: Array, url or closure functions to retrieve source data.
Appearance	
Disabled:	Render this component disabled.
Advanced	
Delay:	The delay in milliseconds between when a keystroke occurs and when a search is performed. A zero-delay makes sense for local data (more responsive), but can produce a lot of load for remote data, while being less responsive. Default value is 300 milliseconds.
Min Chars:	Specify the minimum number of characters to be type before a search is performed. Default value is one character.

Button

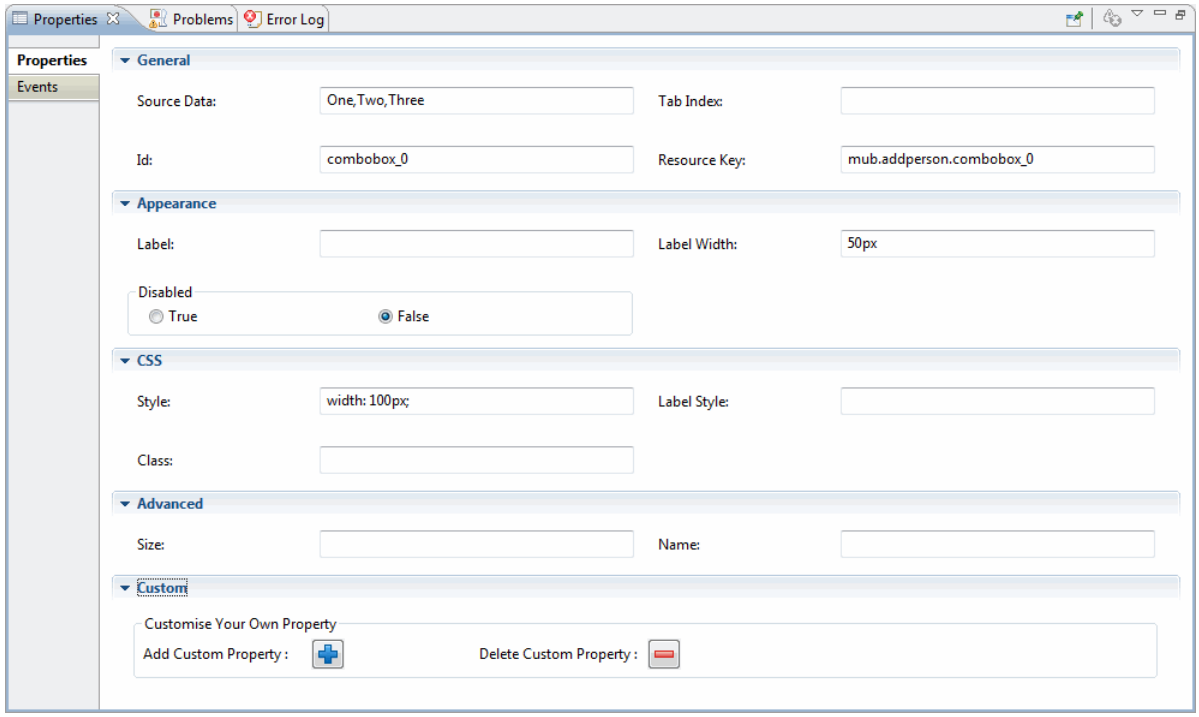


Properties	Description
General	
Label:	Specify the display text on the widget.
Appearance	
Disabled:	Render this component disabled. The default value is false.
Left Icon:	CSS class name to display an icon on the left side on the button. The default value is empty.
Right Icon:	CSS class name to display an icon on the right side on the button. The default value is empty.

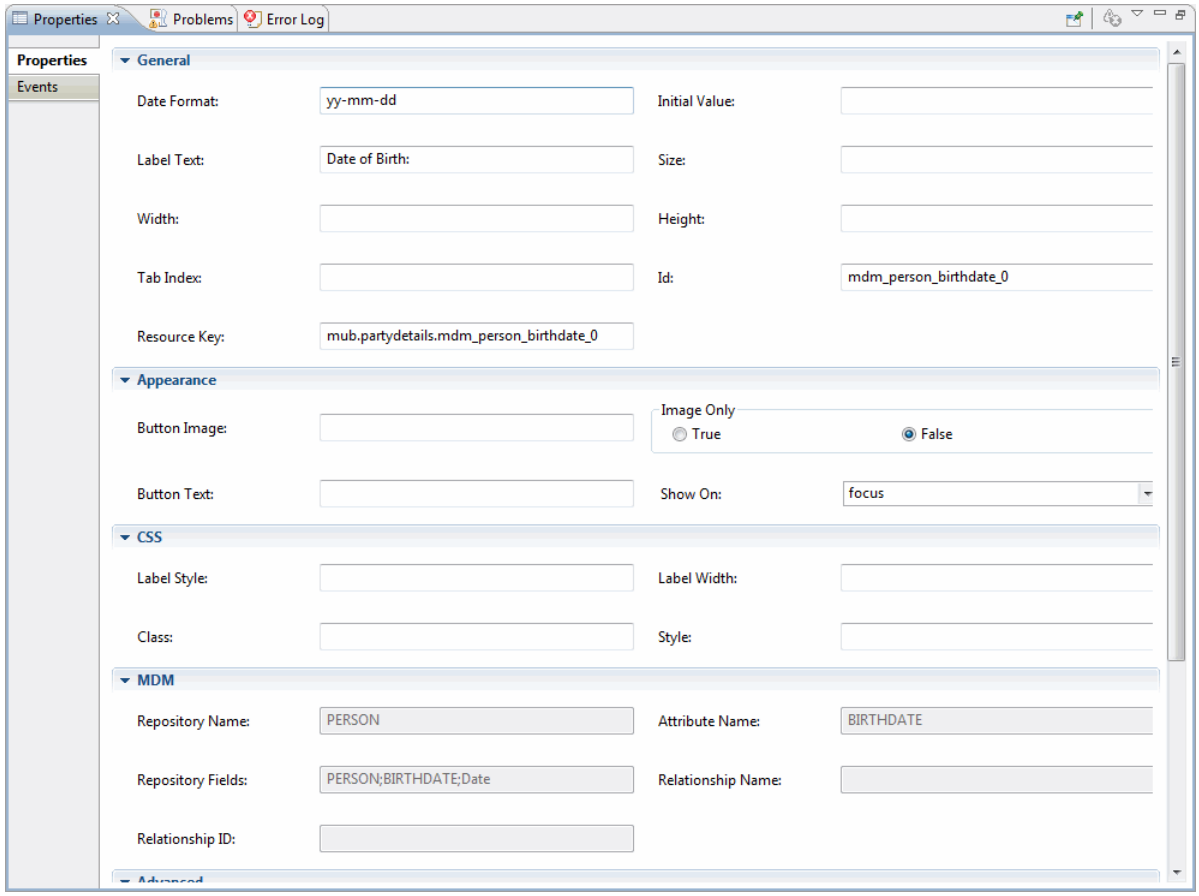
Combobox

Properties	Description
General	
Source Data:	Available values in the combo box. Default values are One, Two, and Three.
Appearance	
Label:	Specify the display text for the combo box.

Properties	Description
Label Width:	Set the width of the combo box label. The default value is 50pixels.
Disable:	Render this component disabled.
Advanced	
Size	Size (width) of the combo box in terms of number of characters.



Datepicker

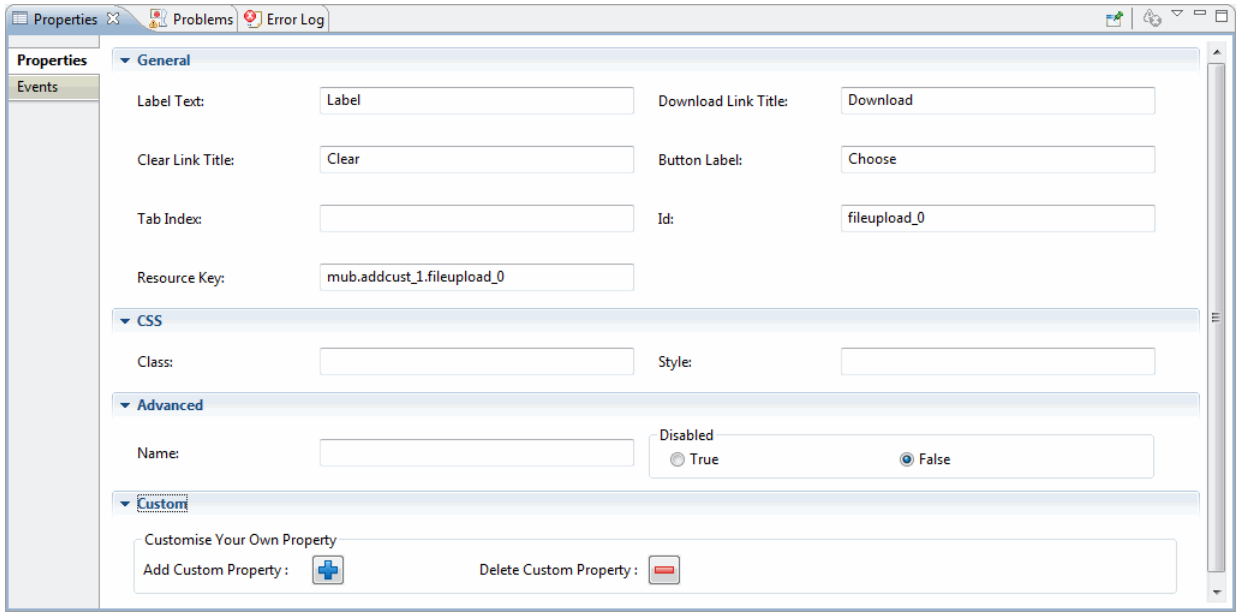


Properties	Description
General	
Date Format:	The format for parsed and displayed dates. The default date format is yy-mm-dd. Even though you select "yy' the year is displayed as "yyyy".
Initial Value:	The initial value to be displayed in the field.
Label Text:	Specify the display text for the datapicker field.
Size:	Specify the size of the datapicker field.
Width:	Specify the width of the datepicker field.
Height:	Specify the height of the datepicker field.
Appearance	
Button Image	The URL for the popup button image. If set, the Button Text option becomes the alt value and is not directly displayed.

Properties	Description
Image Only	Whether the button image should be rendered by itself instead of inside a button element. The default value is false.
Button Text:	The text to display on the trigger button. Use in conjunction with the showOn option set to 'button' or 'both'.
Show On:	Set to 'focus', 'button' or 'both'.

Fileupload

The Fileupload widget is used in Add Record and Modify Record operation. It is used to upload selected files.

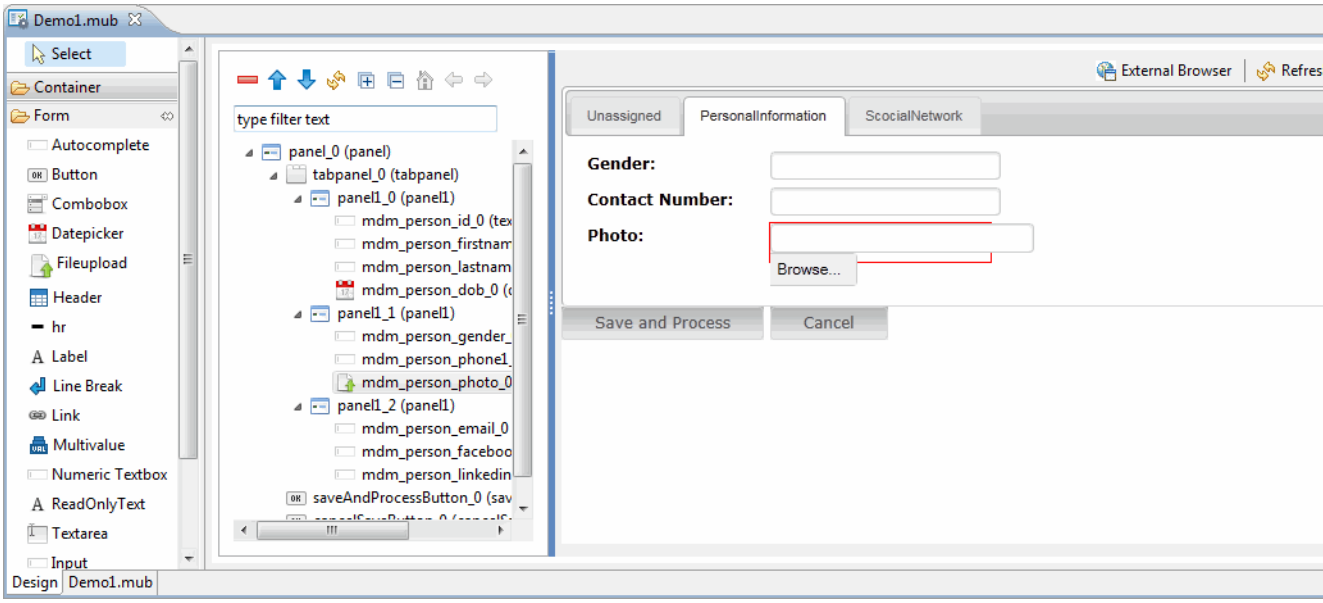


Properties	Description
General	
Label Text:	Specify the display text for the fileupload field.
Download Link Title:	Specify the **Download Link** title for the fileupload field. The uploaded file can be downloaded using the Download link.
Clear Link Title:	Specify the **Clear Link** title for the fileupload field. The Clear link clears the fileupload text box and removes the selected file and the Download link.
Button Label:	Specify the button label text for the fileupload field.
Tab Index:	Specify the tab order of an element (when the 'tab' button is used for navigating)
Button Text:	Specify the text to display on the trigger button.

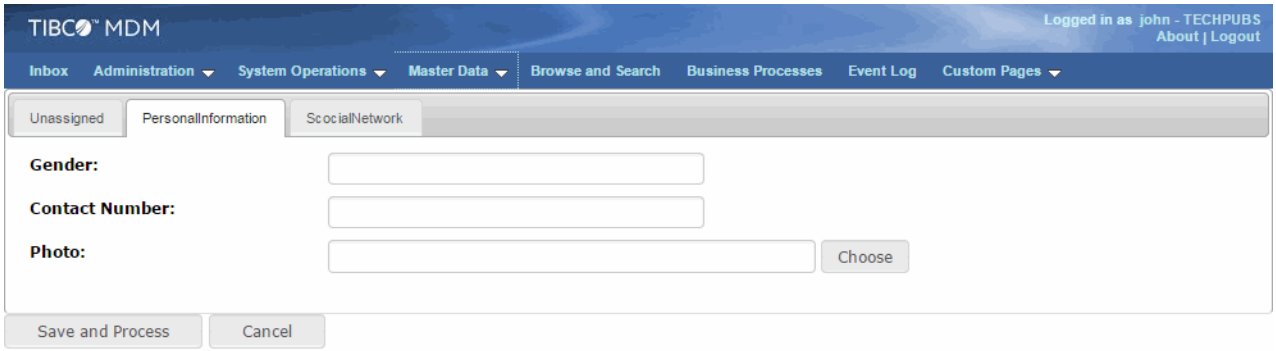
The Fileupload widget is used in Add Record and Modify Record operation. It is used to upload selected files. The Fileupload supports File type attributes.

The limitations of the Fileupload widget are the following:

- You cannot upload multiple files simultaneously.
- You cannot cancel the fileupload action after it is initiated.
- Multivalue fileupload is not supported.
- Relationship fileupload is not supported.
- You can either drag and drop the repository having File type attributes to the Tree Outliner Viewer or drag and drop the Fileupload
- 
- widget.

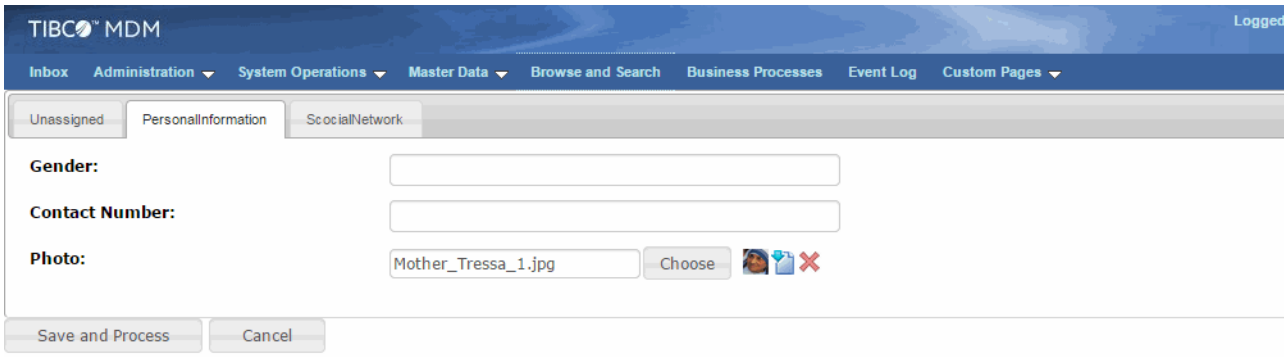


- Using the Fileupload you can select an individual file by clicking the **Choose** button.

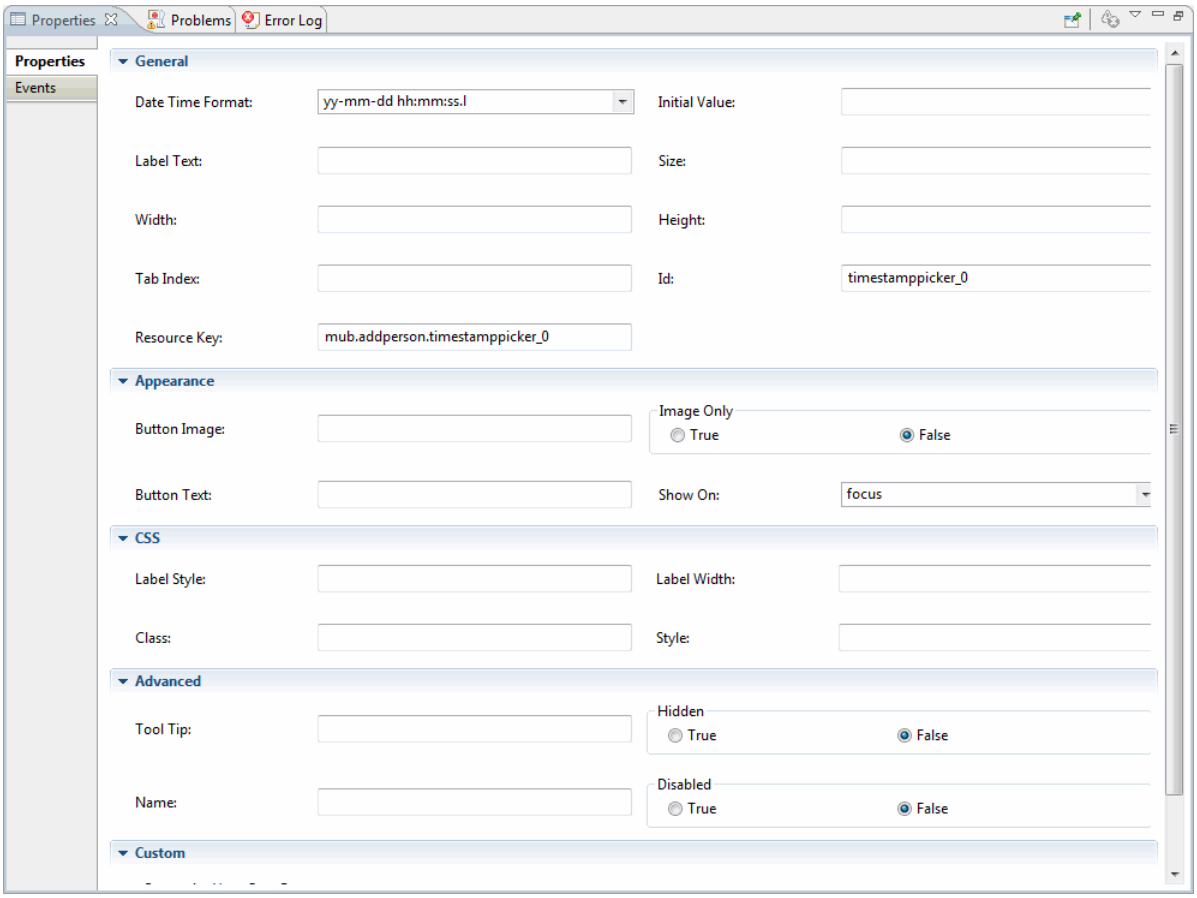


- The selected file name is displayed in the field. If you select an image for uploading, the thumbnail view of the upload file is displayed on the screen.
- If you want to download the selected file click
- 
- and to clear the current selection click
- 

- .

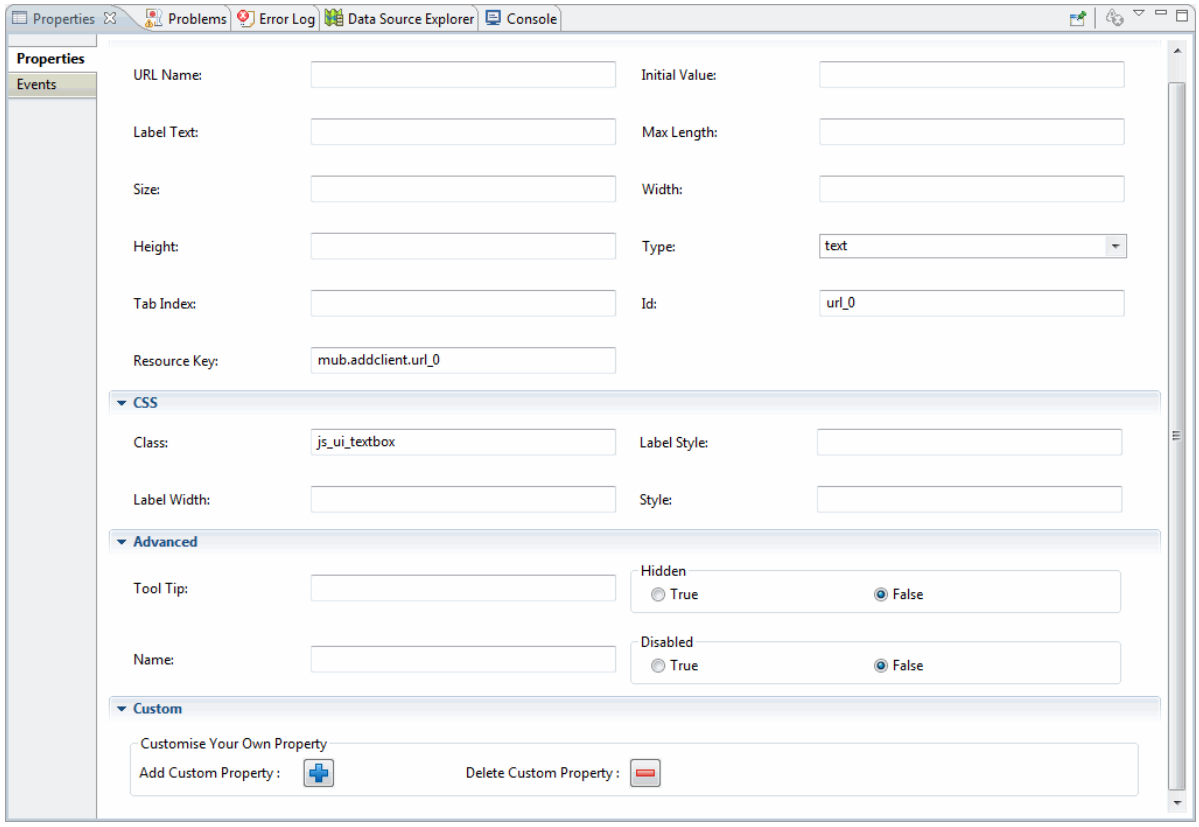


Timestamp Picker



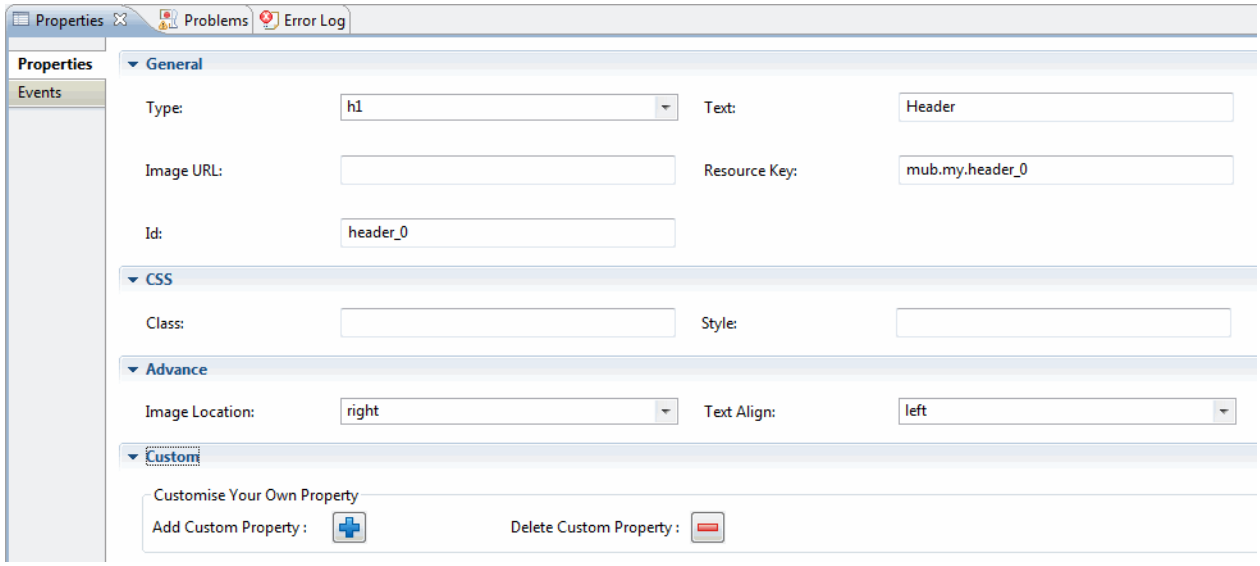
Properties	Description
General	
Date Time Format:	The format for parsed and displayed dates. Default date format is yy-mm-dd and the time format is HH:mm:ss. The formats shown in the drop-down list are the same as the timestamp format in the TIBCO MDM user profile.

URL



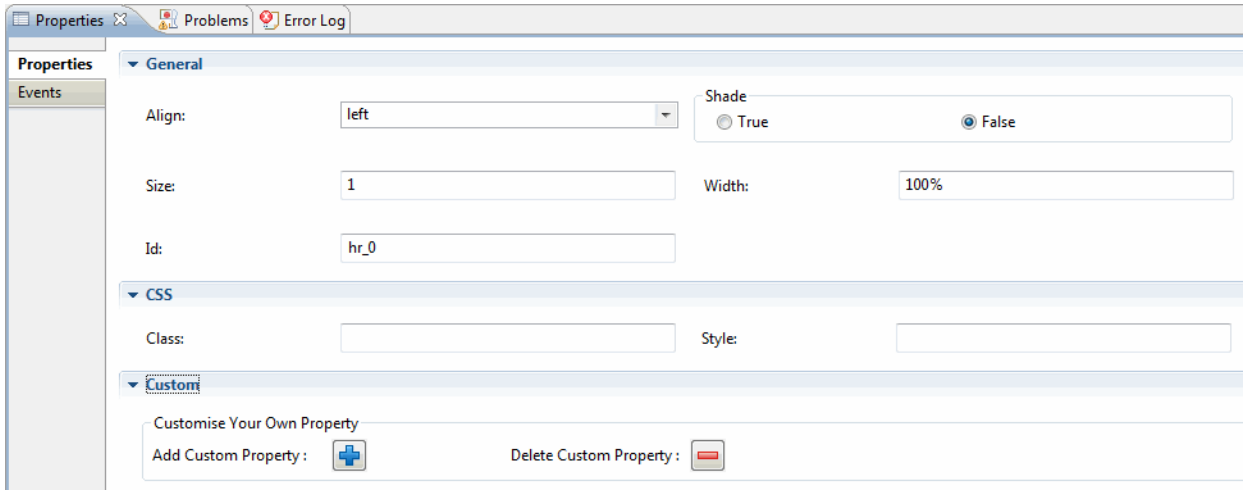
Properties	Description
General	
URL Name:	Specify the URL name.
Type:	The supported types are text, checkbox, file, password, and radio option.
Advanced	
Disabled	Render this component disabled. The default value is false.

Header



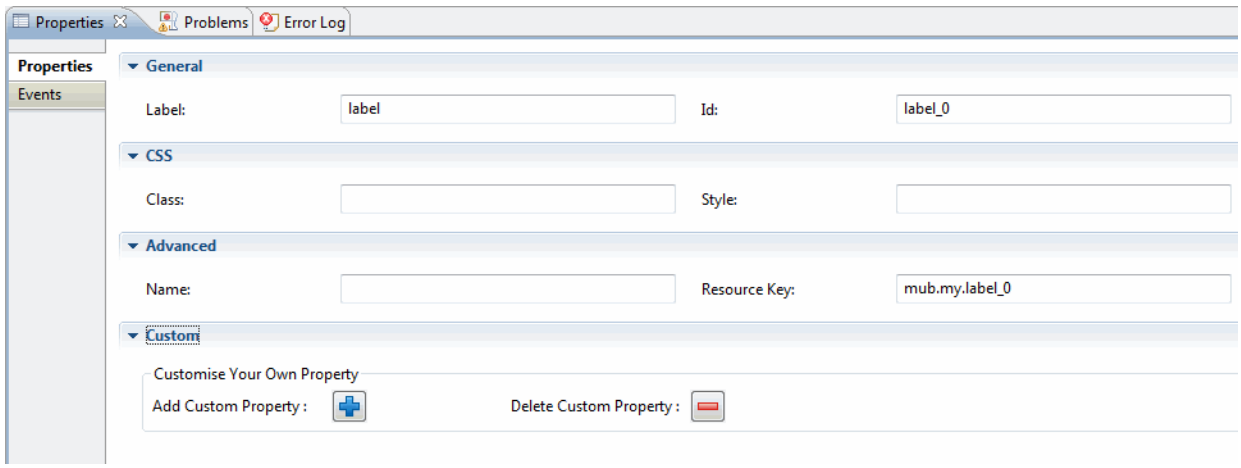
Properties	Description
General	
Type	Select the heading type for the header text. The default value is h1.
Text	Specify the header text for the UI page. The default value is Header.
Image URL:	Specify the URL of the header image.
Advanced	
Image Location:	Specify the location of the header image. The default value is right.
Text Align:	Specify the text alignment of the header image. The default value is left.

Horizontal Line (hr)



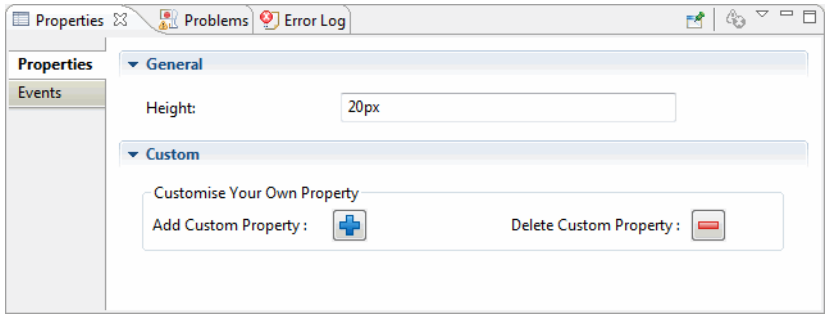
Properties	Description
General	
Align:	Specify the alignment of the horizontal line. The default value is left.
Shade:	Specify whether you want to have shade for the horizontal line. The default value is false.

Label



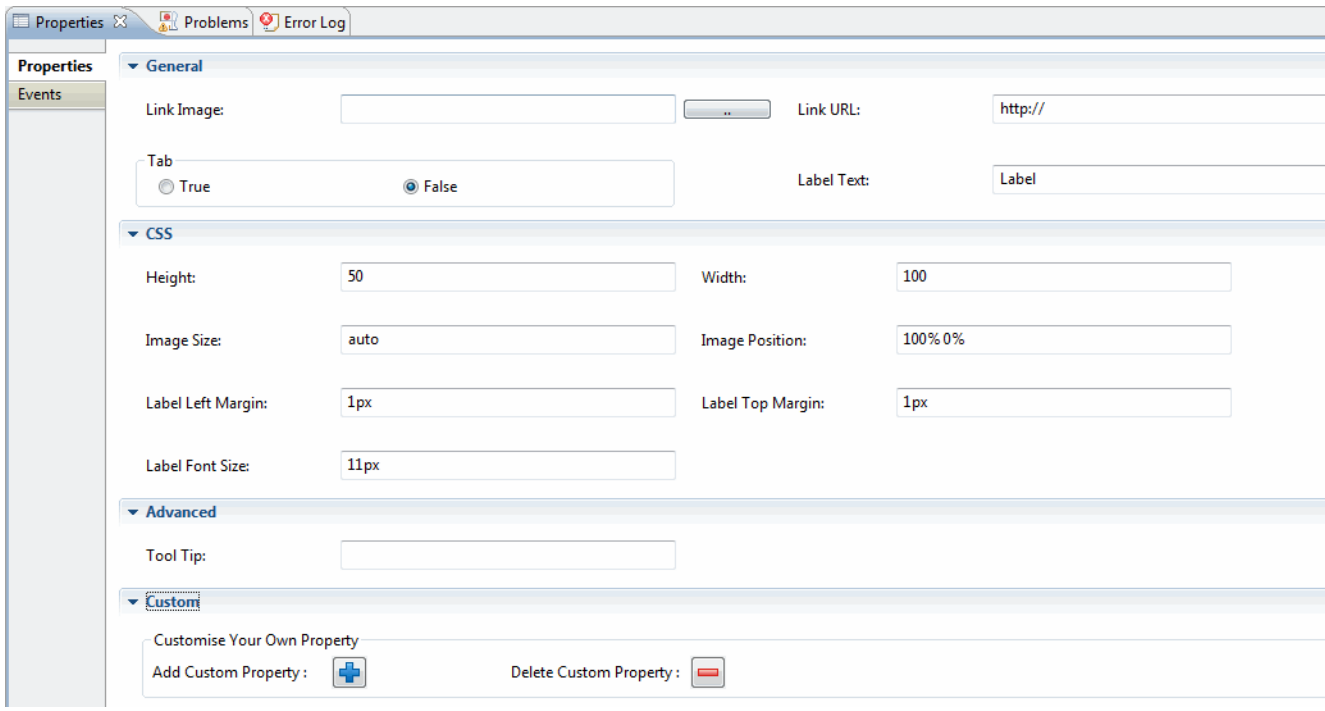
Properties	Description
General	
Label:	Specify the text you want the label to display.
Id:	Specify the ID for the label. The default value is label_0.

Line Break



Properties	Description
General	
Height:	Specify the height of the line break. The default value is 20 pixels.

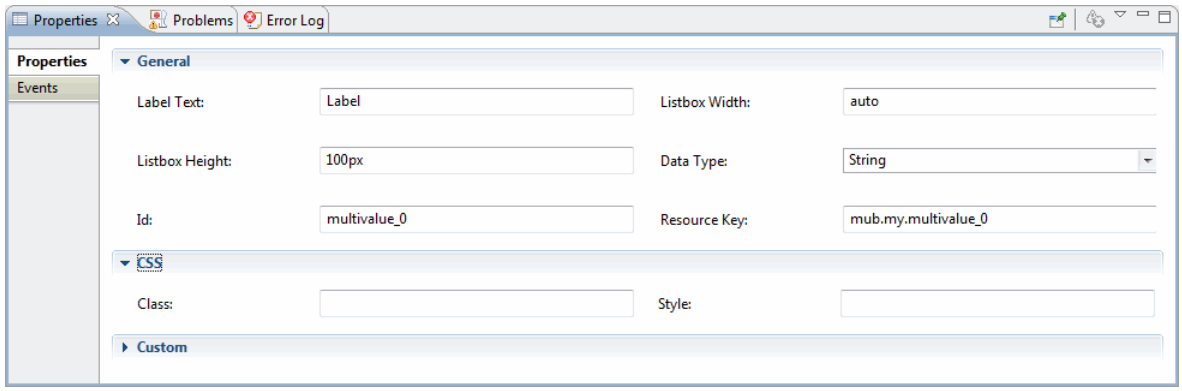
Link



Properties	Description
General	
Link Image:	Specify the path of the image to be linked.
Link URL:	Specifies the URL of the page the link goes to.
Tab	The link opens in a new browser window or a new tab. The default value is false.

Properties	Description
CSS	
Image Size:	Specifies the size of the image. The default value is auto.
Image Position:	Sets the starting position of the image. The default value is 100%0%
Label Left Margin:	Specify the size of the label left margin. The default value is 1px.
Label Right Margin:	Specify the size of the label right margin . The default value is 1px.
Label Font Size:	Specify the size of the label font. The default value is 11px.

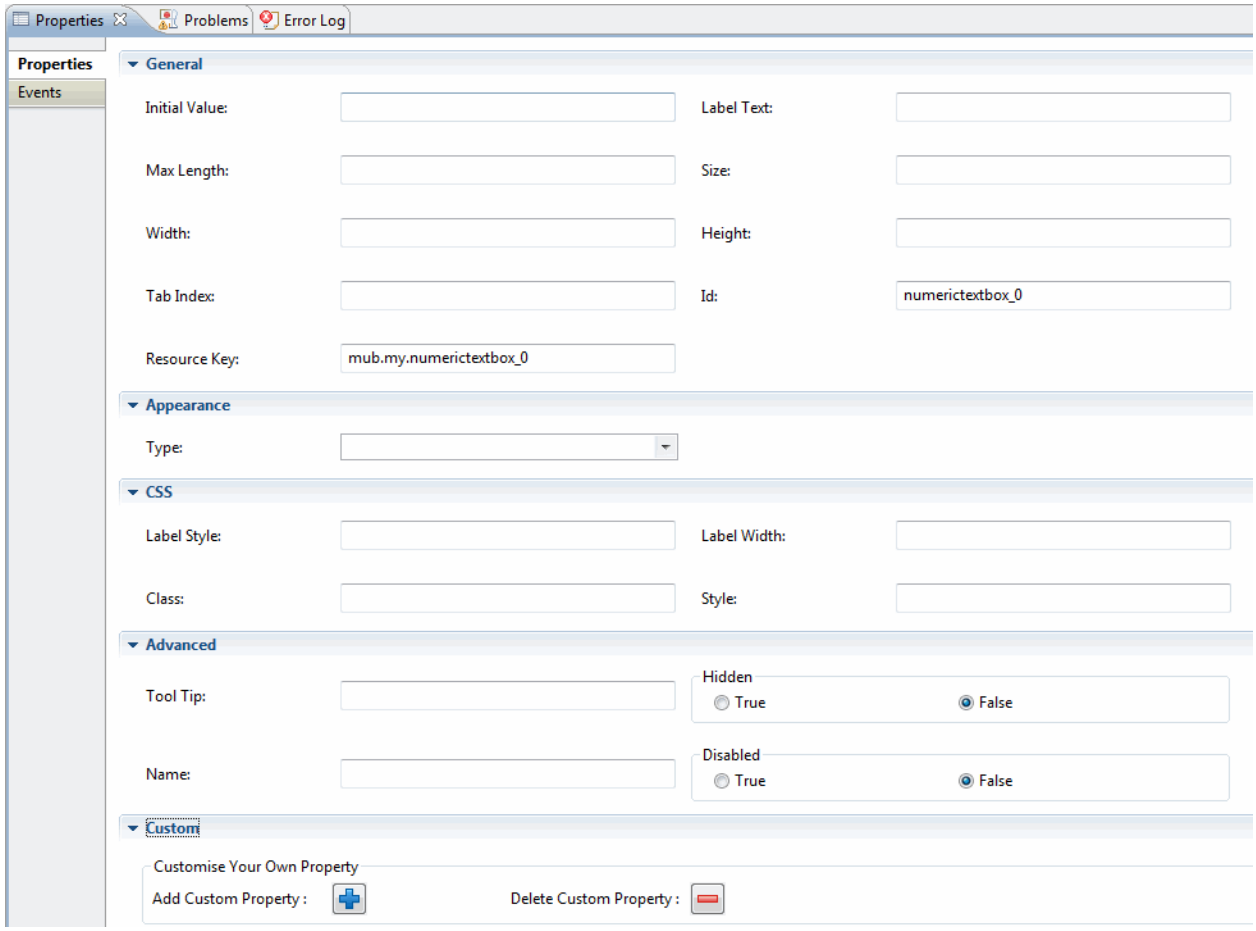
Multivalue



Properties	Description
General	
Label Text:	Specify label for the textbox.
Listbox Width:	Specify the width of the listbox. The default value is auto.
Listbox Height:	Specify the height of the listbox.
Data Type:	Specify the data type from the drop-down list. The default value is String.

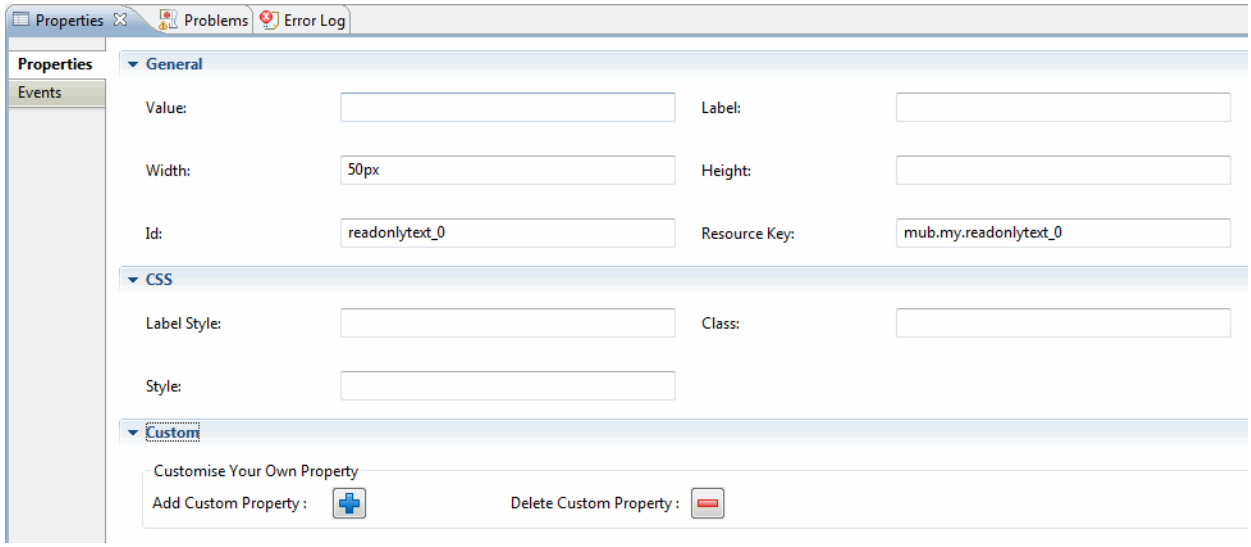
Apart from the General, CSS, and Custom sections, the MDM section with MDM properties are displayed when you drag a repository within the multivalue.

Numeric Textbox



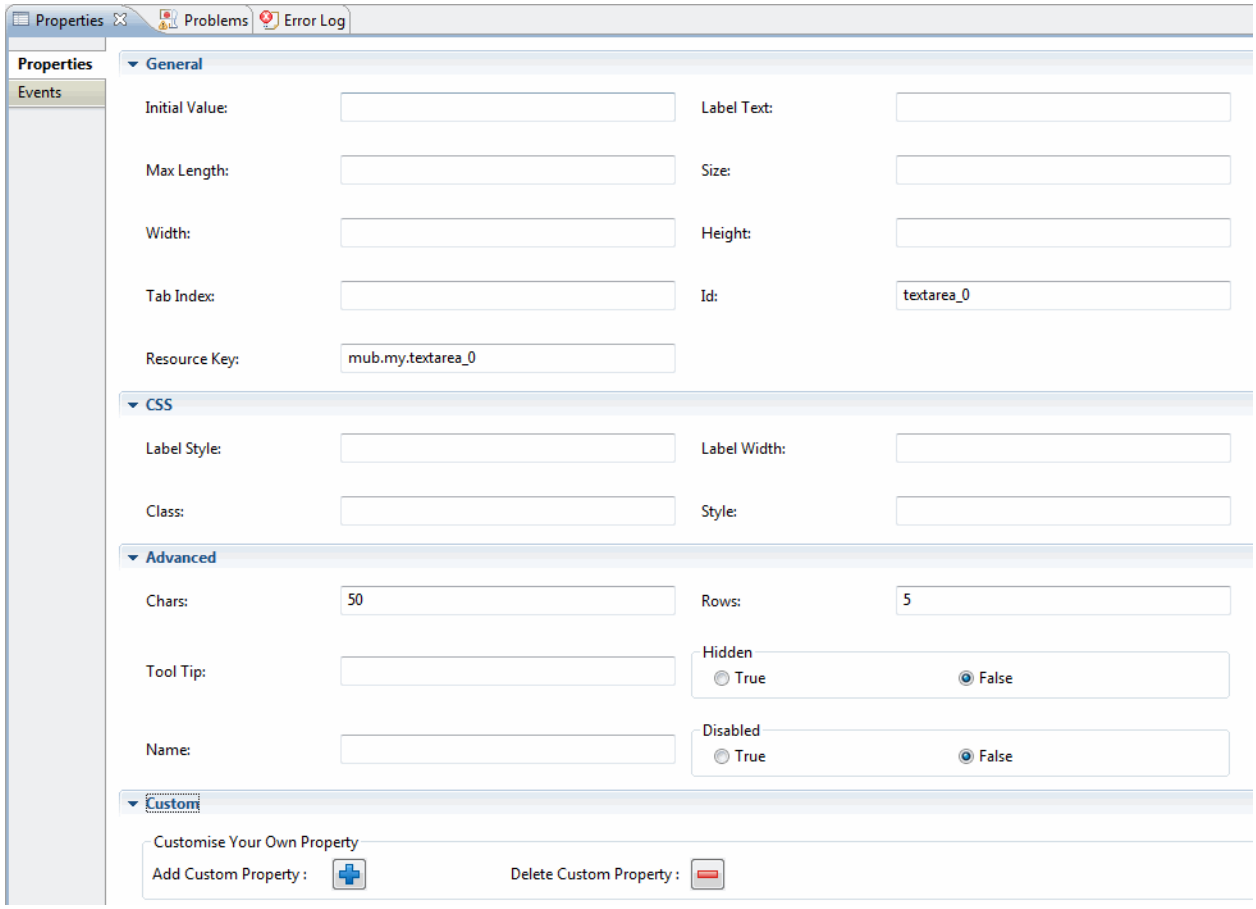
Properties	Description
General	
Initial Value:	Specify the initial value to be displayed in the widget.
Label Text:	Display label for the widget.
Max Length:	The maximum number of characters to be entered in the widget.
Size	Specify the size of the widget in terms of number of characters.
Width	Specify the width of the widget.
Height:	Specify the height of the widget
Tab Index:	Specify the tab order of an element (when the 'tab' button is used for navigating)
Appearance	
Type	Specify the type of the values you want to restrict the user from entering. The possible values are integer, postiveInteger, decimal, and postiveDecimal.

ReadOnly Textbox



Properties	Description
General	
Value:	Specify the read only value in this field.

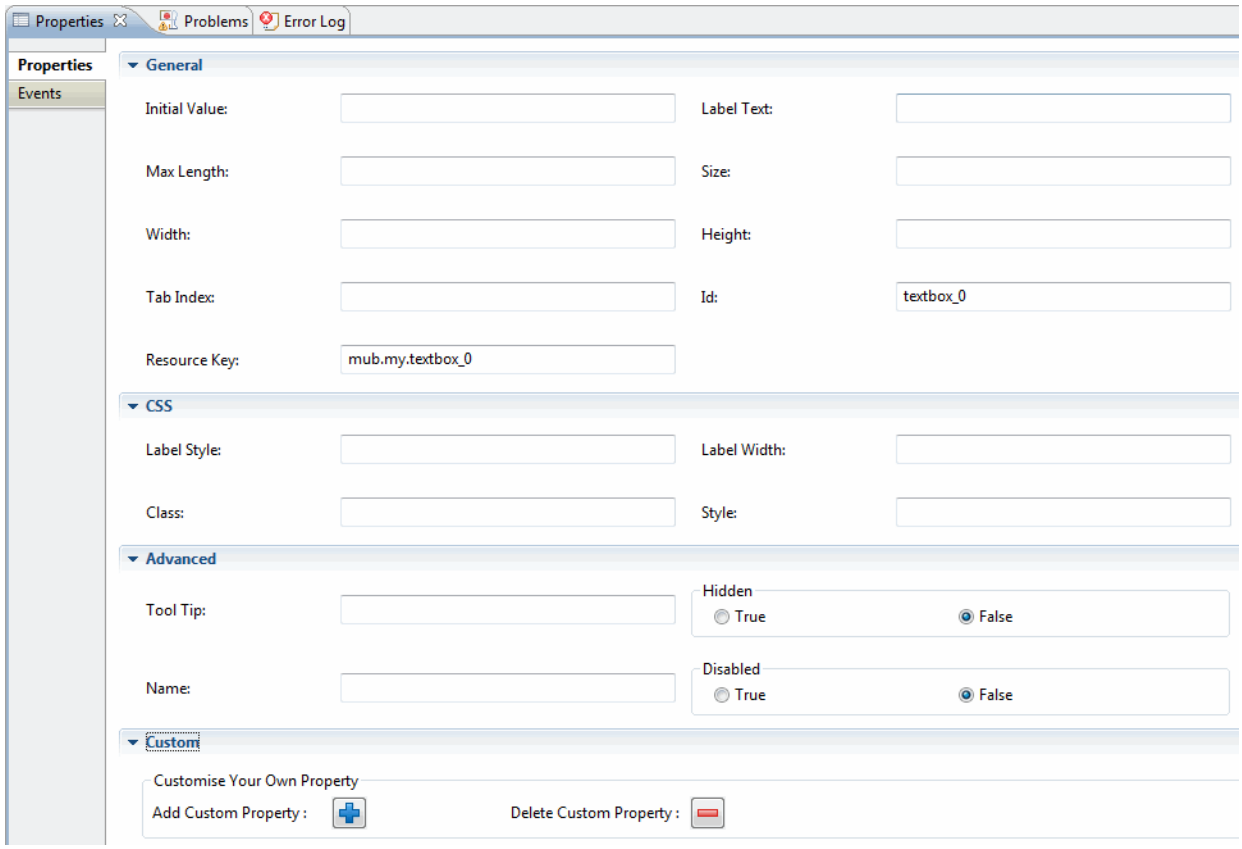
Textarea



Properties	Description
General	
Initial Value:	Specify the initial value to be displayed in the widget.
Label Text:	Display label for the widget.
Max Length:	The maximum number of characters to be entered in the widget.
Size	Specify the size of the widget in terms of number of characters.
Width	Specify the width of the widget.
Height:	Specify the height of the widget
Tab Index:	Specify the tab order of an element (when the 'tab' button is used for navigating)
Advanced	
Chars:	Specify the number of characters to be type in this field. The default value is 50 characters.

Properties	Description
Rows:	Specify the number of rows. The default value is 5 rows.
Hidden	Specify whether you want to hide the widget. Default value is false.
Name:	Specify the name of the widget.
Disabled:	Specify whether you want to disable the widget. Default value is false.

Textbox



Properties	Description
Advanced	
Tool Tip:	Specify the tool tip.
Hidden	Specify whether you want to hide the widget. Default value is false.
Name:	Specify the name of the widget.
Disabled:	Specify whether you want to disable the widget. The default value is false.

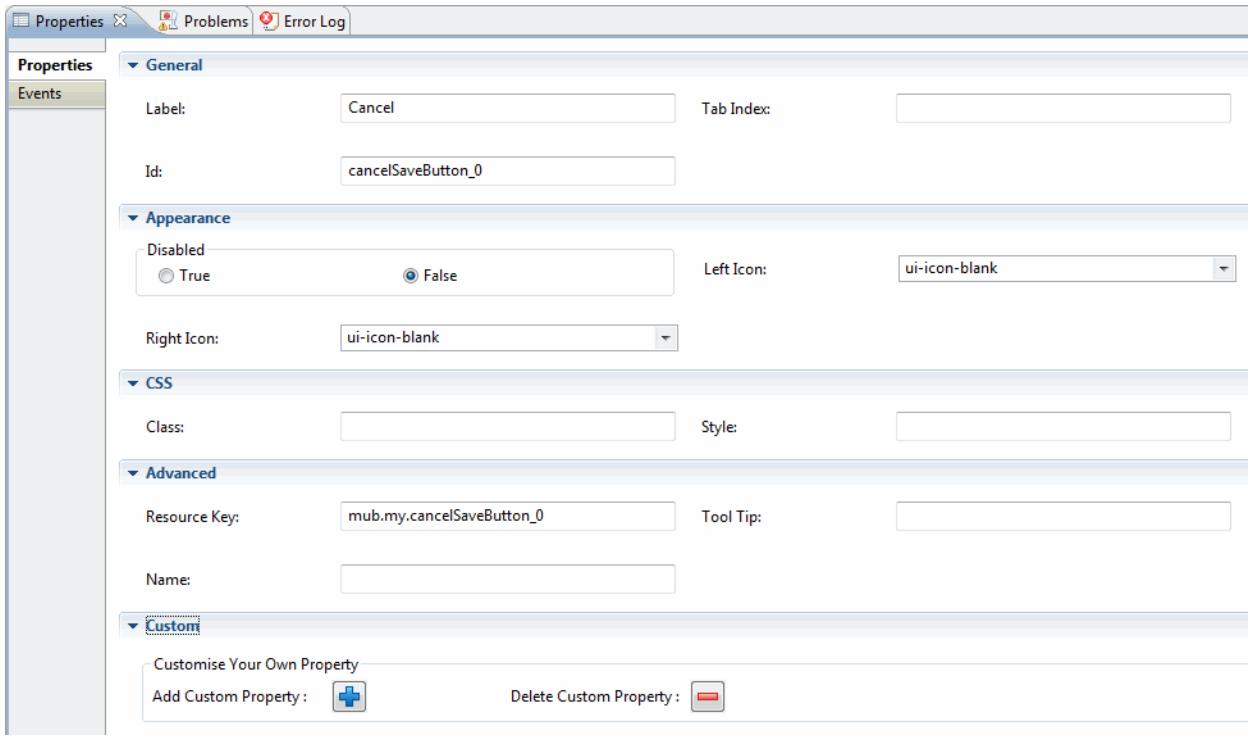
MDM Widget Properties

In this section explains the properties for each widget palettes in the MDM section.

- Cancel
- Text Search
- Related Record
- Save and Process
- Save
- Search
- Validate

Cancel

Cancel button cancels the changes on page.



Properties	Description
Appearance	
Disable:	Specify whether you want to disable the widget. The default value is false.
Left Icon:	Specify the icon which you want to appear inside the button on the left hand side. For example, you can set an icon of an arrow, line, dotted line, horizontal and so on.
Right Icon:	Specify the icon which you want to appear inside the button on the right hand side. For example, you can set an icon of an arrow, line, dotted line, horizontal and so on.

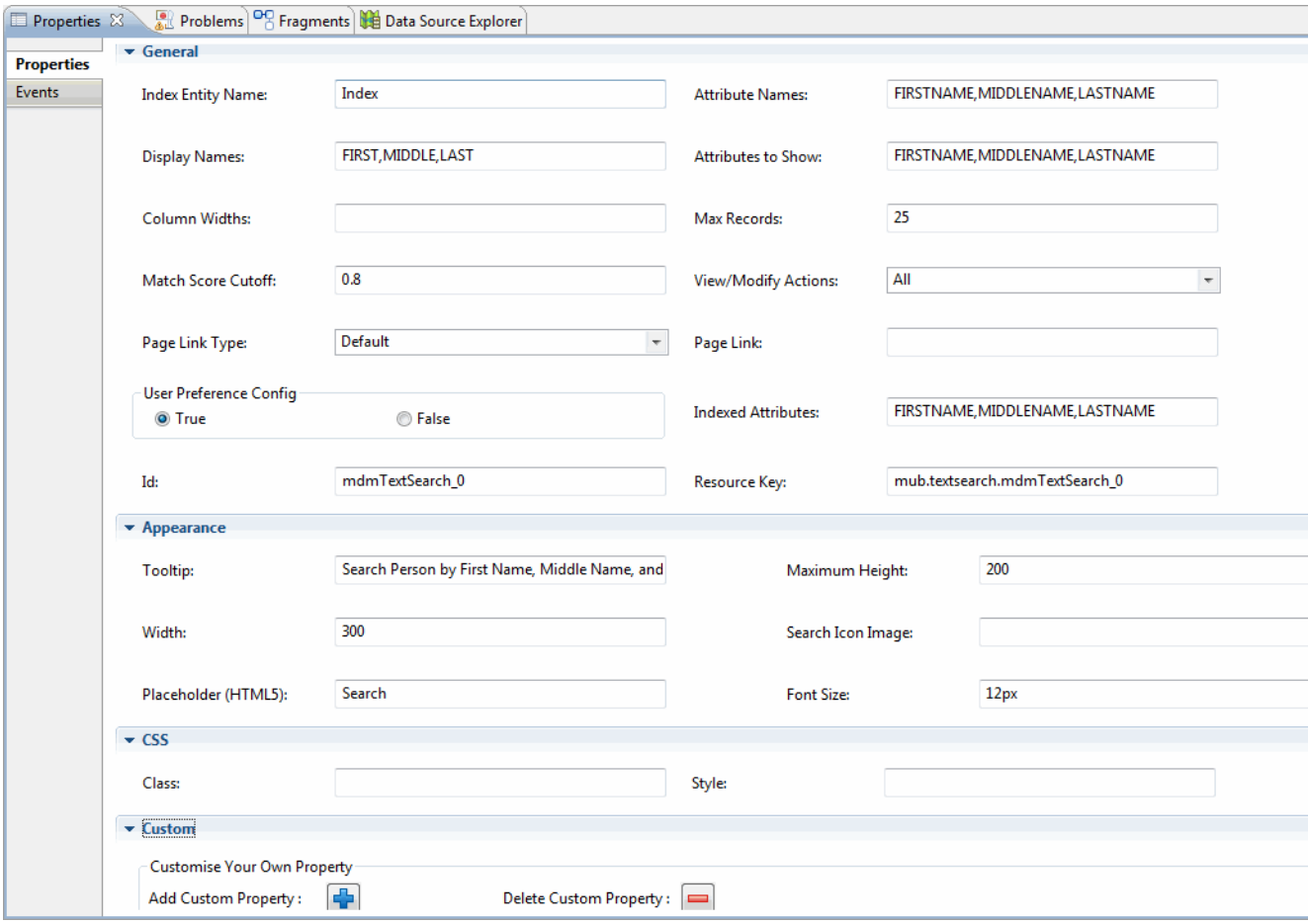
Text Search

The Text Search widget uses the TIBCO Patterns Engine for quick searching of records.

The Text Search widget supports a maximum of three columns. You can search using this widget when you have an index entity configured in the `IndexerConfig.xml` file.

In order to use text search widget on custom page, set the environment variables for WebLogic server. Set the following environment variable to override the rest modules supplied in WebLogic.

```
export EXT_PRE_CLASSPATH=$MQ_HOME/lib/external/jackson-all-1.9.8.jar:$MQ_HOME/lib/external/jackson-databind-2.2.2.jar:$MQ_HOME/lib/external/jackson-annotations-2.2.1.jar:$MQ_HOME/lib/external/jackson-jaxrs-1.9.13.jar:$MQ_HOME/lib/external/jackson-core-2.2.2.jar:$MQ_HOME/lib/external/jackson-mapper-asl-1.9.13.jar:$MQ_HOME/lib/external/jackson-core-asl-1.9.13.jar:$MQ_HOME/lib/external/jackson-xc-1.9.13.jar:$MQ_HOME/lib/external/jettison-1.3.3.jar
```



Properties	Description
General	
Index Entity Name	Specify the index entity name for which you want to perform search. This is a mandatory field.

Properties	Description
Attributes Names	Specify the comma-separated indexed attribute names to display in the search results in a given order. The indexed attribute names are case-sensitive. You can specify maximum three indexed attribute name values. Ensure that there are no white spaces in the comma-separated indexed attribute names. For example, FIRSTNAME,LASTNAME,DOB
Display Names	Specify the comma-separated attribute display names in the search results in a given order. You can specify the same number of display names as the attribute names. Maximum three display name are allowed. For example, First,Last,DateofBirth
Attributes to Show	Specify the comma-separated attribute display names in the search results in a given order. You can specify the same number of display names as the attribute names. Maximum three display names are allowed. The other attributes are hidden. The **Attributes to Show** and **Column Widths** corresponds to the **Attributes Names**. For example, FirstName,LastName,DateofBirth
Column Widths	Specify the comma-separated numbers that represent the width in pixels of each column in the result set. If the column width is not specified, it is set to equal width. The column width must match the number of attributes names.
Max Records	Specify the maximum number of records to be displayed in the search result. The default value is 25.
Match Score Cutoff	Specify the patterns cut off score. If you do not specify the cut off score, the default cut off score is defined. The default cut off score is 0.8.
View/Modify Actions	Select the appropriate value from the drop-down list to enable or disable the View or Modify actions. The available options are **All**, **View only**, **Modify only**, and **None**. The default value is All.
Page Link Type	Select the appropriate value from the drop-down list to indicate how the View and Modify links should be displayed. The available options are: • **Default**: The record is viewed or modified in the out-of-the-box UI. • **Self**: The record is viewed or modified on the same page. To view the record on the same page, design an appropriate UI page to view or modify the record. • **Custom**: The record is viewed or modified in another page designed in UI Builder. If you have selected the **Custom** option, provide the page link (another page link).
Page Link	If you have selected **Custom** in the **Page Link Type** field, specify the page link for the custom page. For example, /eml/CustomForm?componentName=ModifyRecord&pageName=ModifyRecord
User Preference Config	Specify whether you want to enable or disable the configuration option. By default, the configuration option is enabled. Select **False** to disable the option.

Properties	Description
Indexed Attributes	If you select **True** in the **User Preference Config** field, specify the comma-separated indexed attributes names.
Appearance	
Tooltip	Specify the tooltip message to indicate on which attributes the records should be searched.
Maximum Height	Specify the maximum height of the search result.
Width	Specify the width of the widget.
Search Icon Image	Browse the search icon that you want to associate with the widget.
Placeholder (HTML5)	Specify a hint word that helps in understanding what can be entered in the control. By default, the value is Search.
Font Size	Specify the CSS property of the textbox to change the font-size. Specify the values in pixels. For example, 11px.

Configuring Text Search

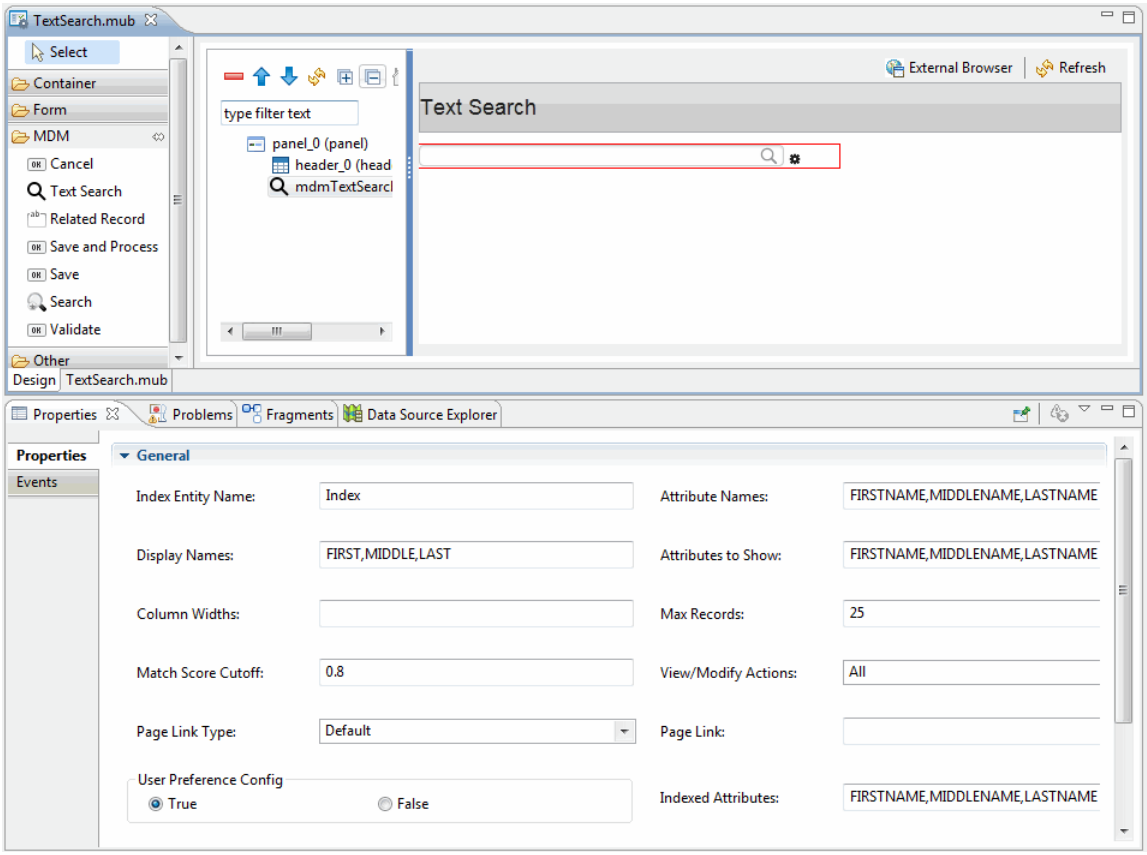
The main configurations for the Text Search widget is the Index Entity name and attributes to display in the search results.

To define Text Search, you must provide the index entity and minimum of one column. The Text Search also supports record view and record modify operation. The search result displays the View and Modify icons using which you can perform a quick view or modify operations for the selected record.

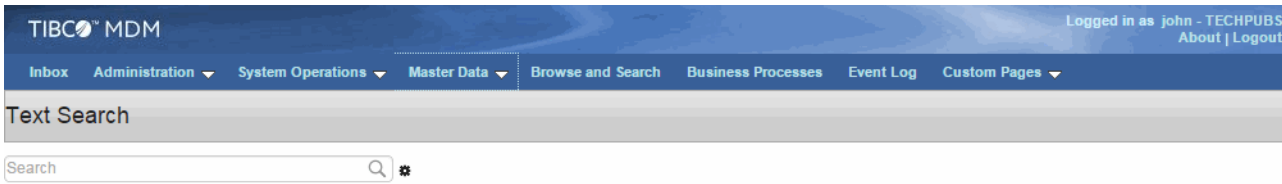
The "Text Search property" section is used to define the various parameters. The Text Search is not supported for related records. For more information on Text Search, refer to the chapter "Search and Matching" in *TIBCO MDM System Administration Guide*.

Procedure

1. To create Text Search, drag and drop the Text Search  widget to the Tree Outline Viewer.

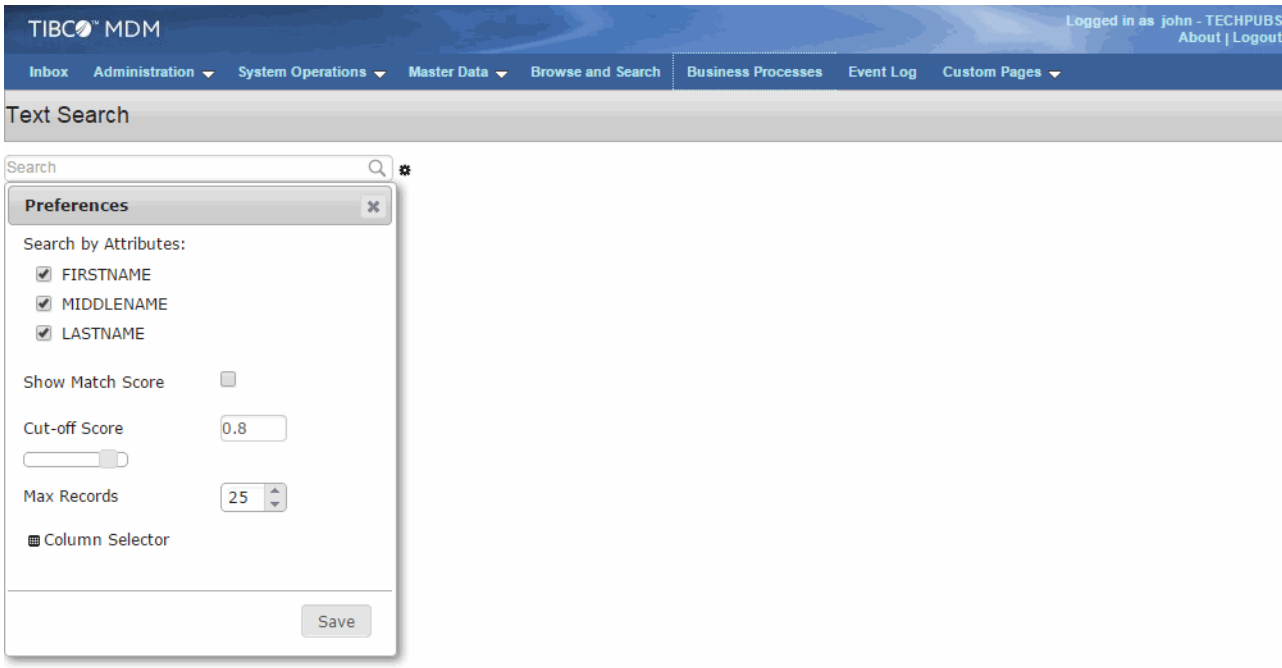


2. In the **Properties** section, specify the index entity name, display attributes, display names, and other parameters.
3. On deploying, Text Search can be accessed from the TIBCO MDM server.



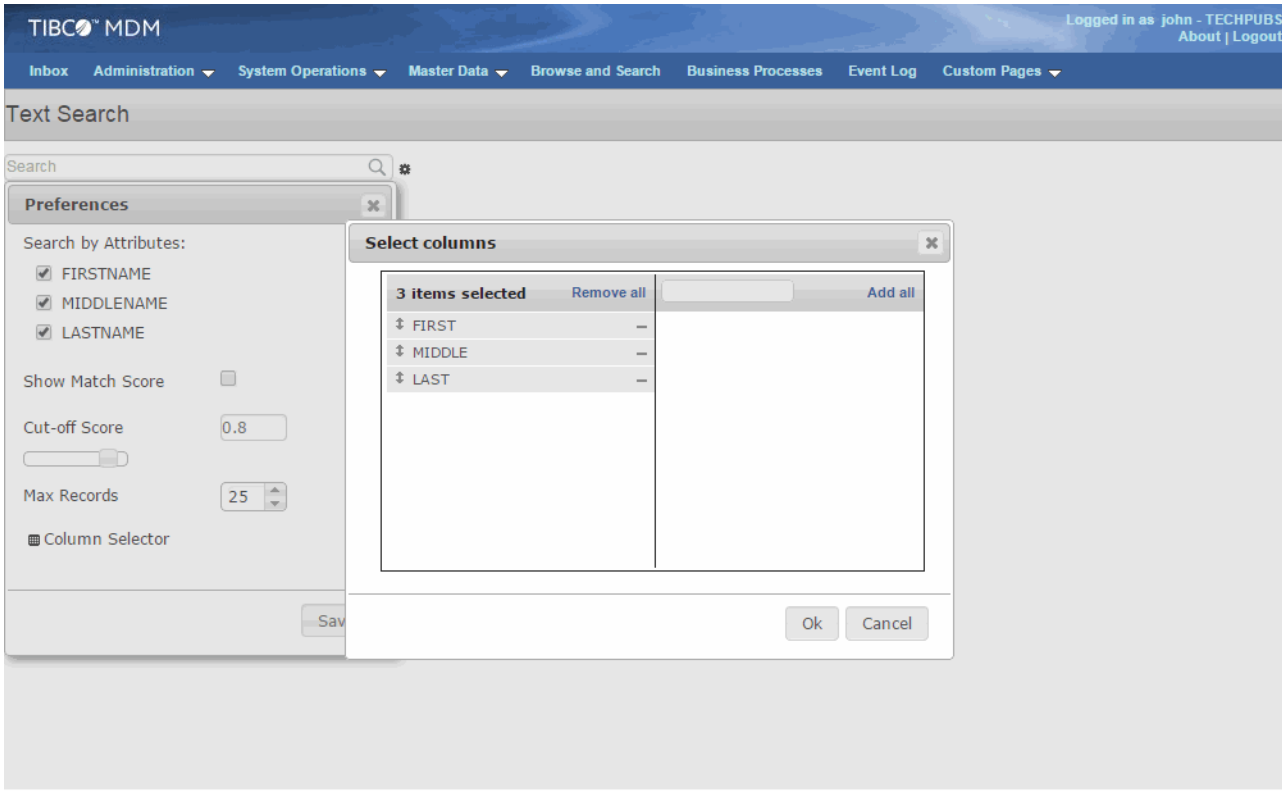
4. The Text Search widget has configuration options to configure the search. The configured search persists and is applied to the Text Search widget each time you visit the same page.

5. Click ⚙ . The Preference drop-down list is displayed with various parameters.

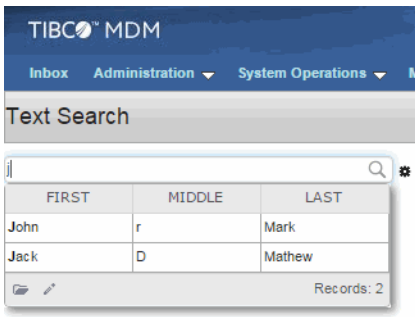


Properties	Description
Preferences	
Search by Attributes	Select the check boxes corresponding to the search attributes, on which you want to perform search.
Show Match Score	Select the check box if you want the match score to be displayed in the search results in a column next to the action button columns.
Cut-off Score	Specify the Patterns cut off score from 0.0 to 1.0.
Number of records	Specify the number of records to be retrieved in the search result.
Column Sector	Select the columns to be displayed in the search result. You can define the list of columns to be displayed in the search result.

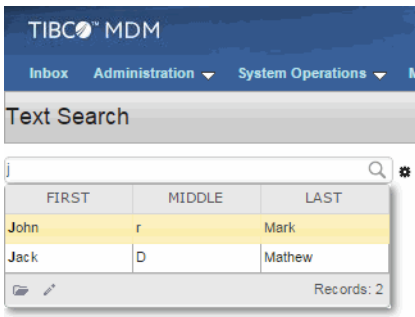
6. On clicking the Column Selector, the Select columns dialog is displayed. Click **Add all** to add the columns to the search result. To select a specific column click the respective attributes and they are displayed on the right.
7. Click **Ok**.



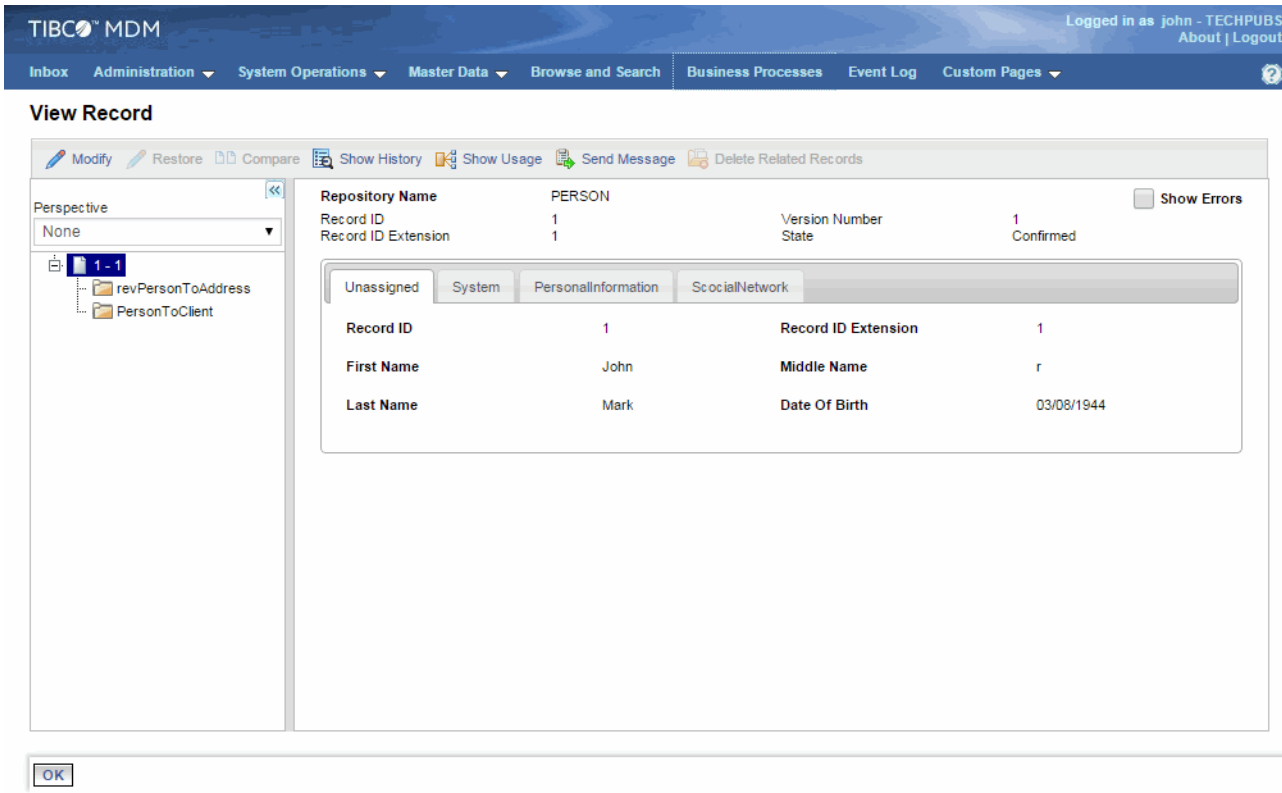
8. Click **Save**. The selected configuration is saved.
9. To search for a value, type the initial text of the search value, and the search result matching the search text is displayed. You can see the display attributes that you have defined in the **Properties** section of the text search page.



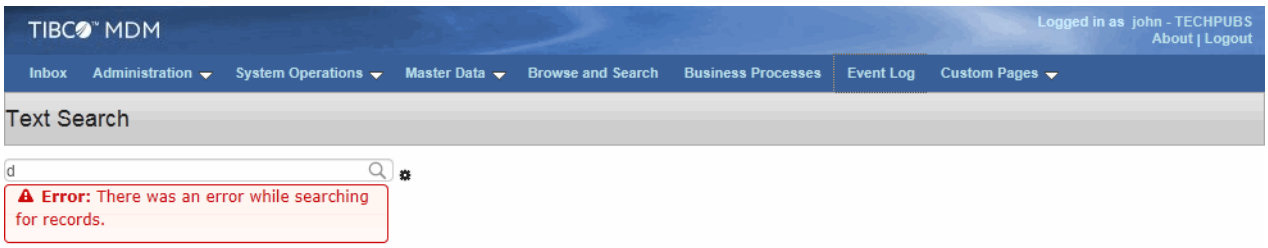
10. You can edit or view the selected record. Select the record that you want to view or modify. The last column of the search result displays the **View** icon and the **Edit** icon .



11. For example, click the **View** icon displayed on the first row. The selected record details are displayed in the view only mode on the same page.



12. While defining the properties of Text Search, if you have entered incorrect index entity name or did not configure the indexer config or Patterns config files, an error message is displayed while typing the search value.

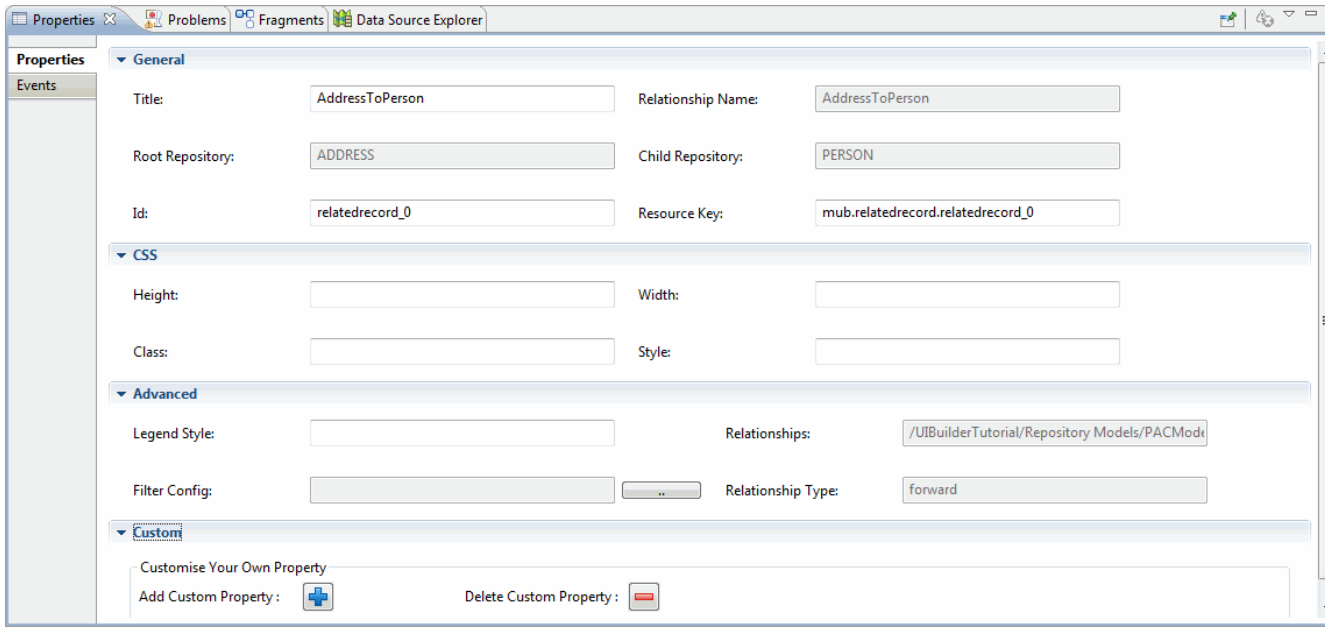


13. While defining the properties of Text Search, if you have entered a tooltip for the text search, the tooltip is visible when you hover the mouse on the **Search** field.

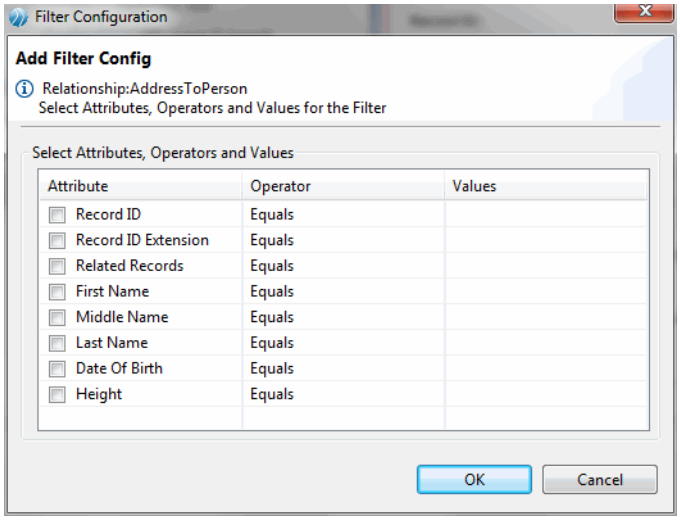
Related Record

Related record widget is used when you have one-to-one relationship between root and child records.

This widget requires manual configurations in the property and you need to put the attributes manually.



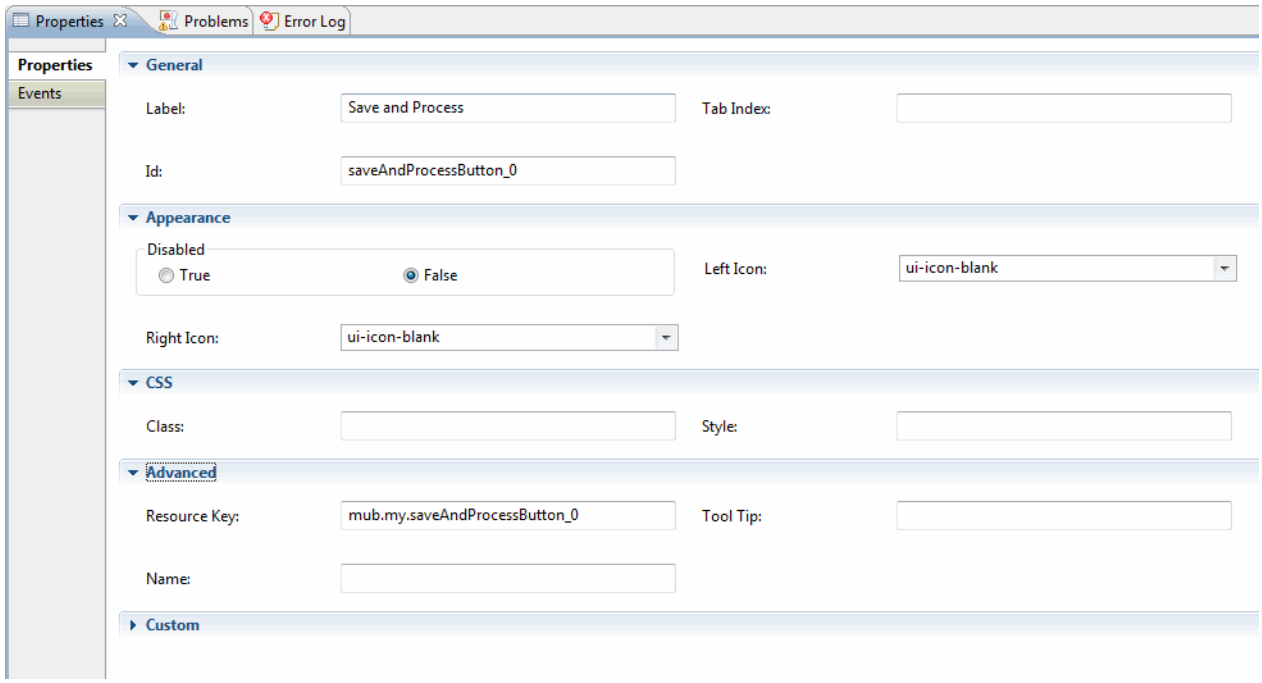
Properties	Description
General	
Title:	Specify the title of the fieldset of the widget.
Relationship Name:	Displays the MDM relationship name.
Root Repository	Displays the parent repository name.
Child Repository	Displays the child repository name.
Advanced	
Legend Styles:	Specify the CSS for the header.
Relationships	Displays the path of the selected relationship. This is an non editable field.

Properties	Description
Filter Config	The Filter Config button allows you to configure the filters. • On clicking the **Filter Configure** button, the **Filter Configuration** page is displayed.  • Select the checkbox corresponding to the attributes for which you want to define the filters from the **Attribute** cloumn. • Select the appropriate operator for a filter from the **Operators** column. For a multi-value attribute operator, specify the list of values as comma separated. • Specify the filter values in the **Values** cloumn. • Click **OK**.
Relationship Type	Displays the relationship type. For example, forward or reverse. This is an non editable field.

The Relationship Name, Root Repository and Child Repository needs to be added manually when you drag the Related Record widget from MDM container.

Save and Process

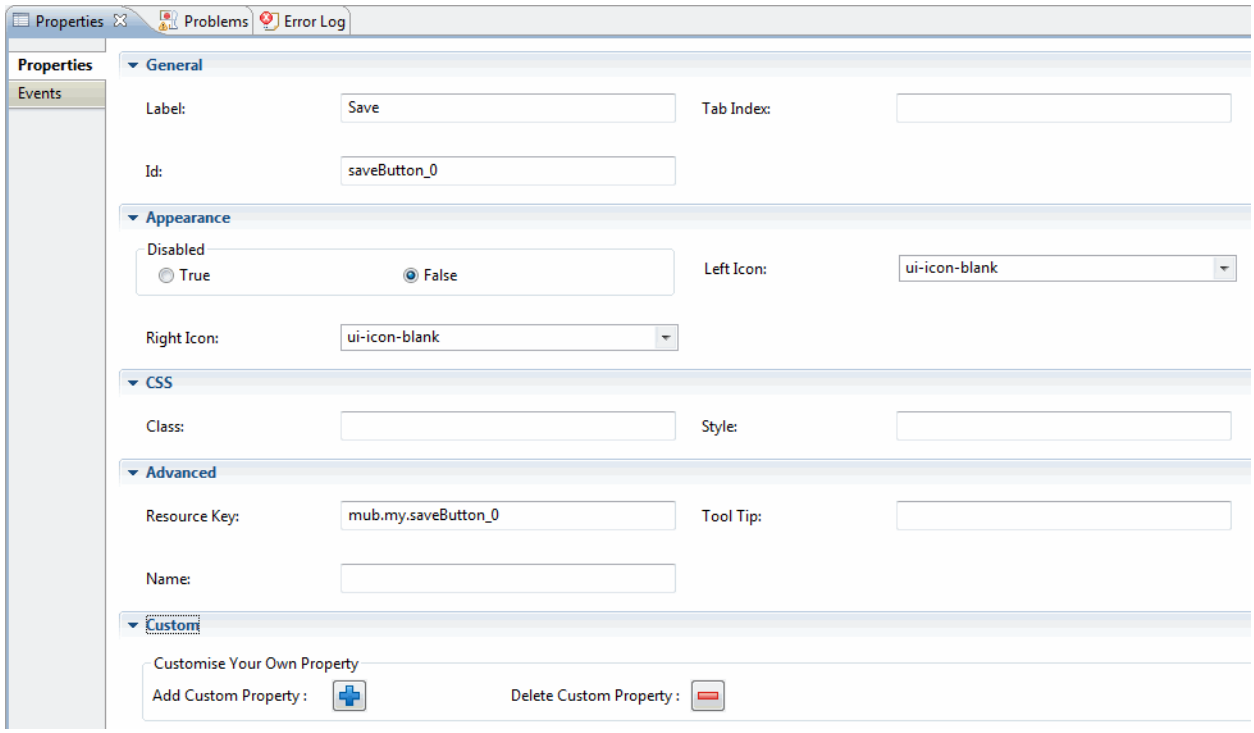
The Save and Process button is a specialized button widget which when clicked saves the data on the server. It has the pre configured event handler.



Properties	Description
Appearance	
Disabled:	Specify whether you want to disable the widget. The default value is false.
Left Icon:	Specify the icon which you want to appear inside the button on the left hand side. For example, you can set an icon of an arrow, line, dotted line, horizontal and so on.
Right Icon:	Specify the icon which you want to appear inside the button on the right hand side. For example, you can set an icon of an arrow, line, dotted line, horizontal and so on.

Save

Save is very similar to Save and Process except that it does not process the data in the MDM.



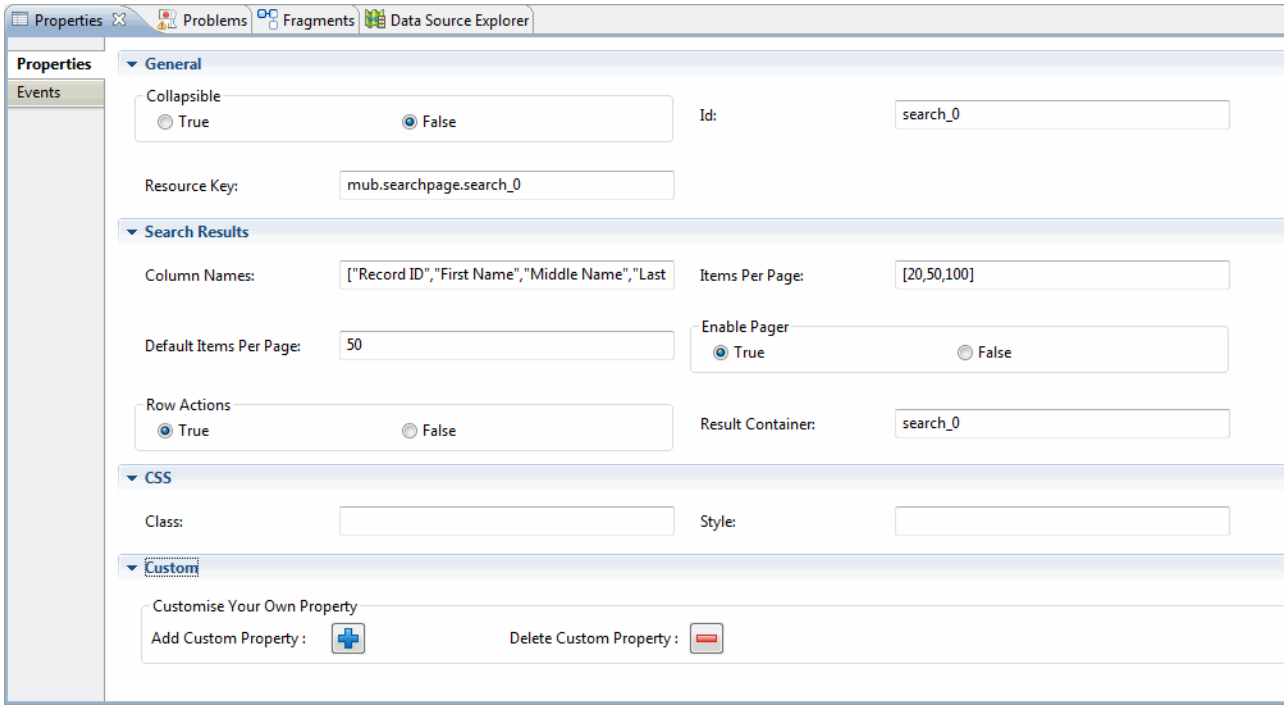
Properties	Description
Appearance	
Disabled:	Specify whether you want to disable the widget. The default value is false.
Left Icon:	Specify the icon which you want to appear inside the button on the left hand side. For example, you can set an icon of an arrow, line, dotted line, horizontal and so on.
Right Icon:	Specify the icon which you want to appear inside the button on the right hand side. For example, you can set an icon of an arrow, line, dotted line, horizontal and so on.

Search

The Search widget is used to search the repository records based on attribute values and operators.

It is similar to the Browse and Search page available in the out-of-the-box UI. Using this widget you can control both the searchable attributes to be displayed on the screen and the records to be displayed in the search results. Using the Search widget you can do the following:

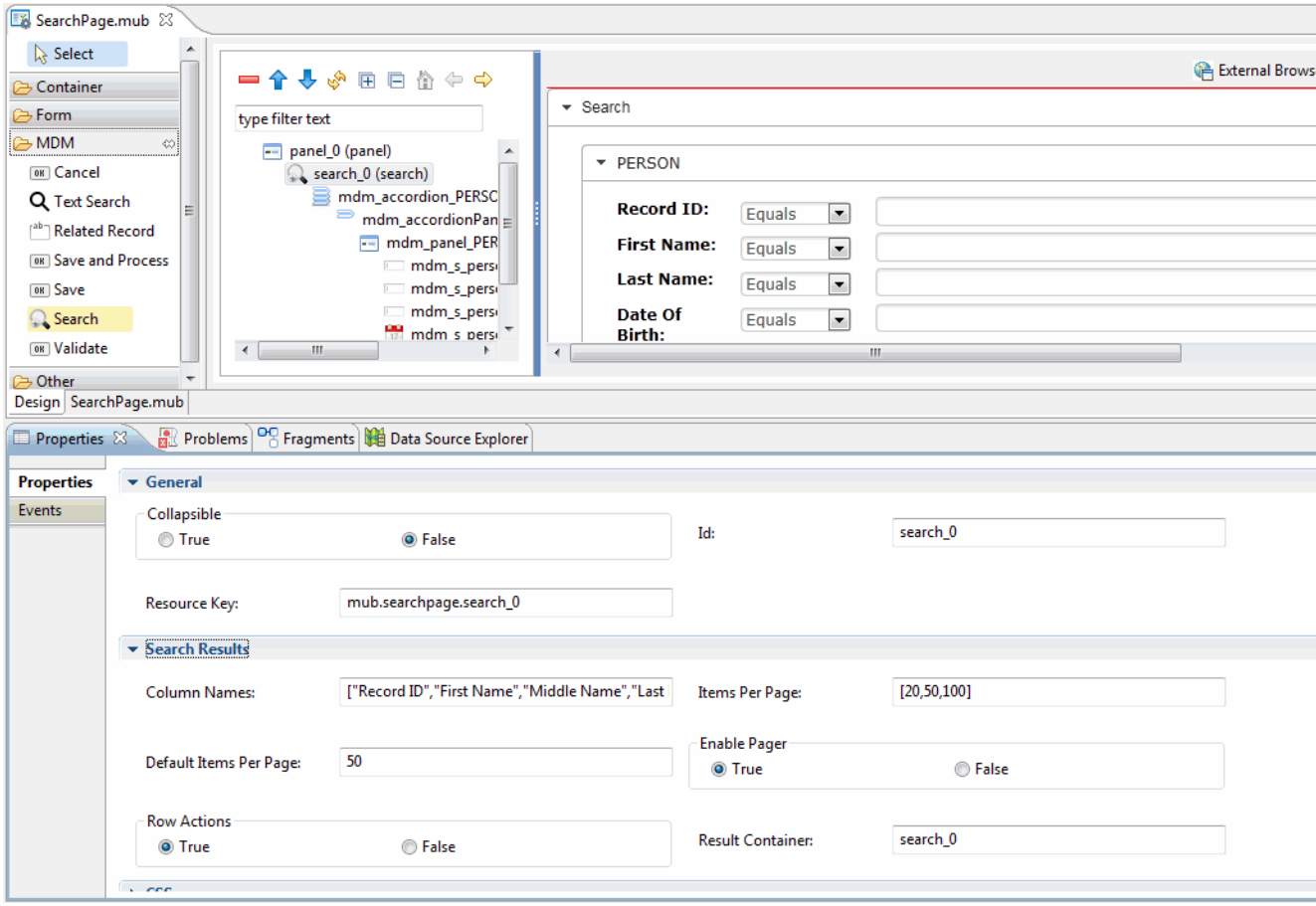
- Choose root repository attributes, relationships attributes, and child repository attributes to search records.
- After creating the search widget, reorder the attributes on the tree outline viewer.
- Choose the attributes to be displayed as columns for the search result data grid. The data grid is configurable.
- View the search result records on a custom page or in the out-of-the-box UI depending on the custom UI configurations.
- View or modify the search result records in the out-of-the-box UI or on a custom page. Ensure that the custom page is linked to the metadata operation.



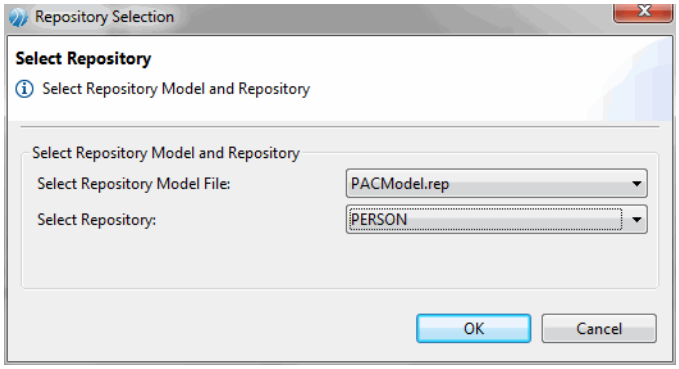
Properties	Description
General	
Collapsible	Specify if you want to make the search filter collapsible. By default it is not collapsible and is set to **False**. Select **True** to make the search filter collapsible.
Search Results	
Column Names	The column names are populated with the selected attributes of a particular repository in the Search widget wizard.
Items Per Page	An array to construct a select box element in the pager in which we can change the number of the visible rows. Example: [10,20,30].
Default Items Per Page	Specify the number of records you want to view in the search result grid. For example, 20.
Enable Pager	Specify if you want to use the pager bar to navigate through the search result records. By default the pager bar is displayed. Select **False** if you do not want the pager bar.
Row Actions	Specify if you want the Add, Modify, View, and Delete options to be enabled for the searched rows. By default it is enabled and set to **True**. Select **False** if you want to disable it.

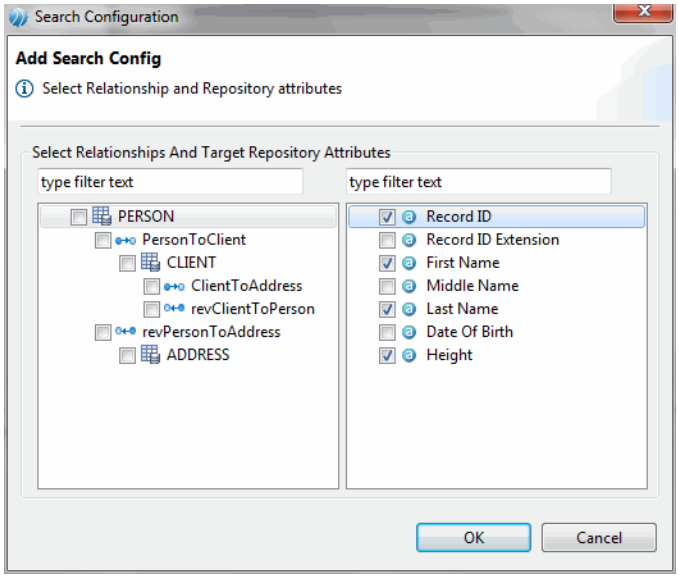
Properties	Description
Result Container	Specify the result container value. By default the result container value is search_0. The search results data grid can be created in another container such as a panel. Provide the HTML DOM ID of the new container, in which you want to put the search result data grid. For example, if you defined the new container property as "panel_1", it creates a datagrid inside panel_1, which is on top of default search_0.

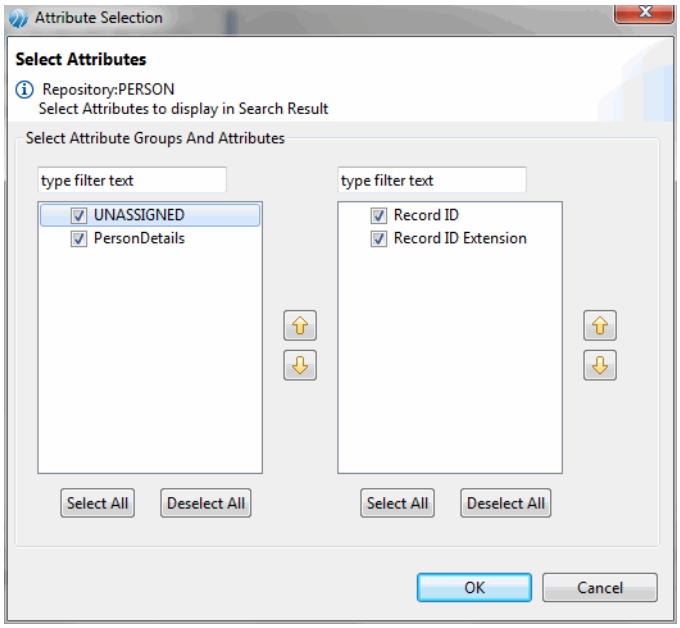
- To create Search, drag and drop the Search widget to the Tree Outline Viewer.



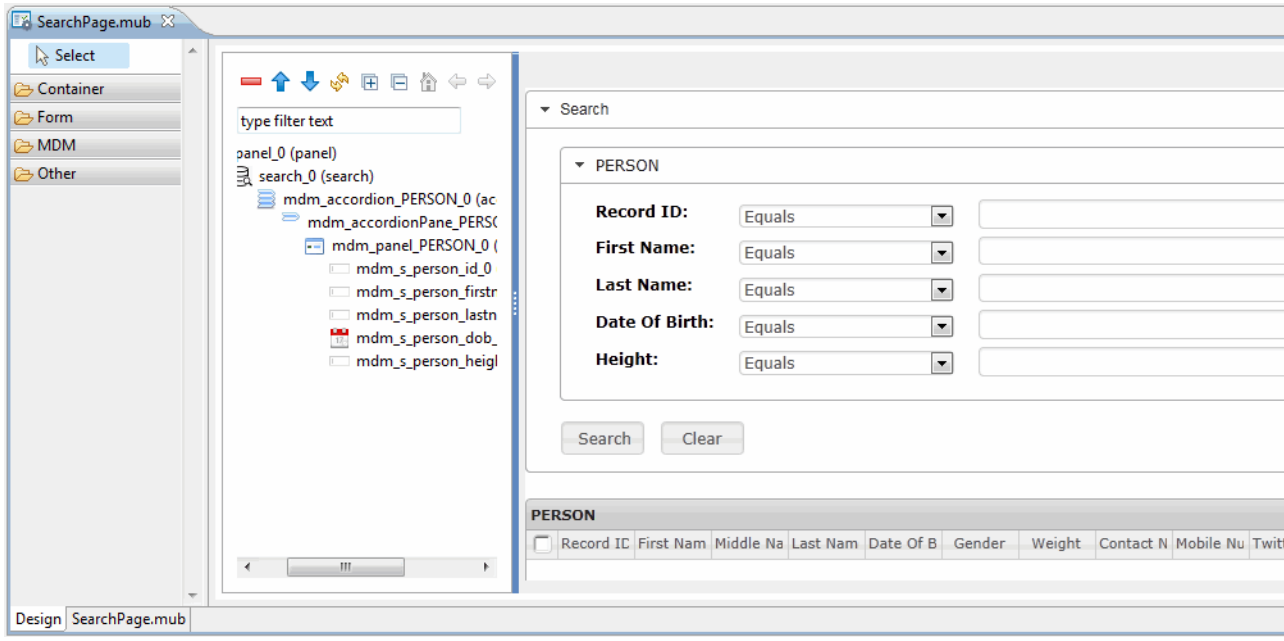
- The Repository Selection page is displayed.



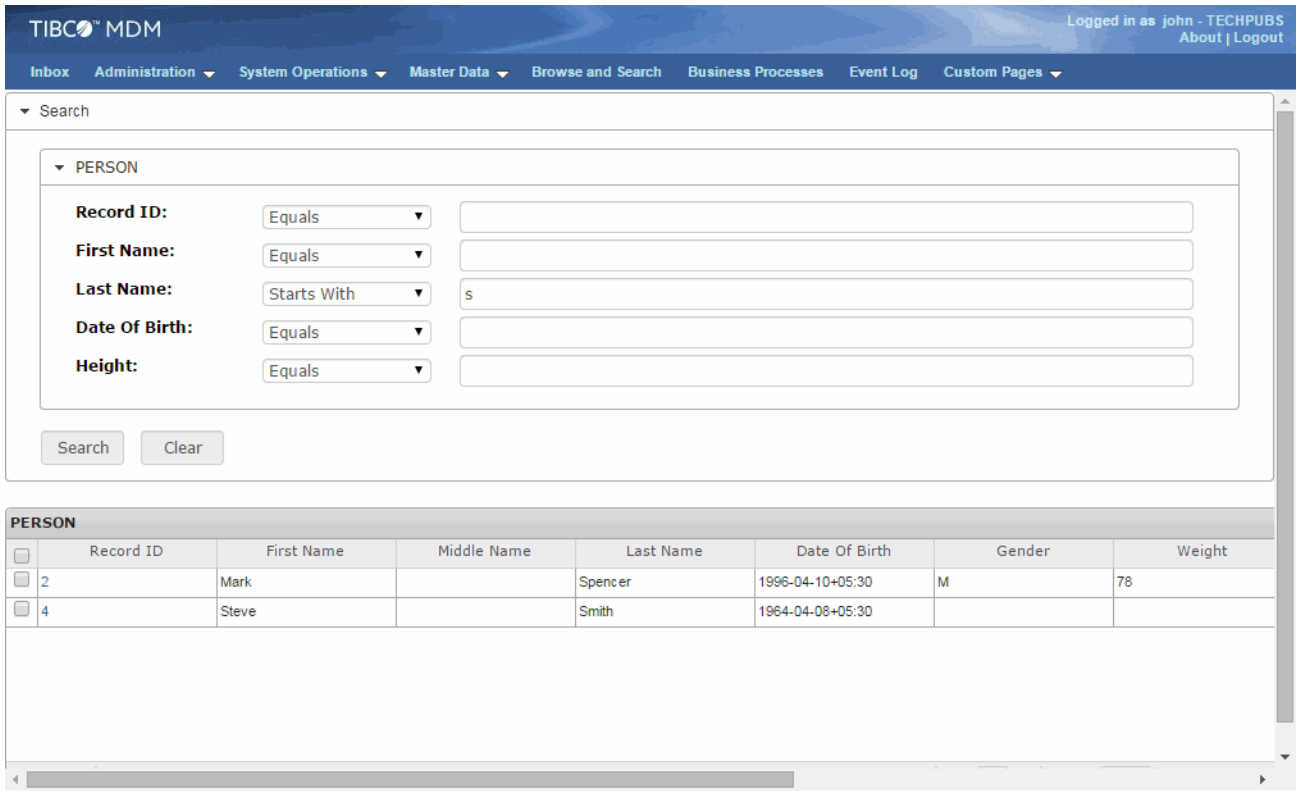
- Select the appropriate repository model file and repository from the **Select Repository Model File** and **Select Repository** drop-down list for which you want to create Search.
- Click **OK**.
- 

- Select the Relationship and Repository attributes for which you want to perform the Search operation and click **OK**.
- 

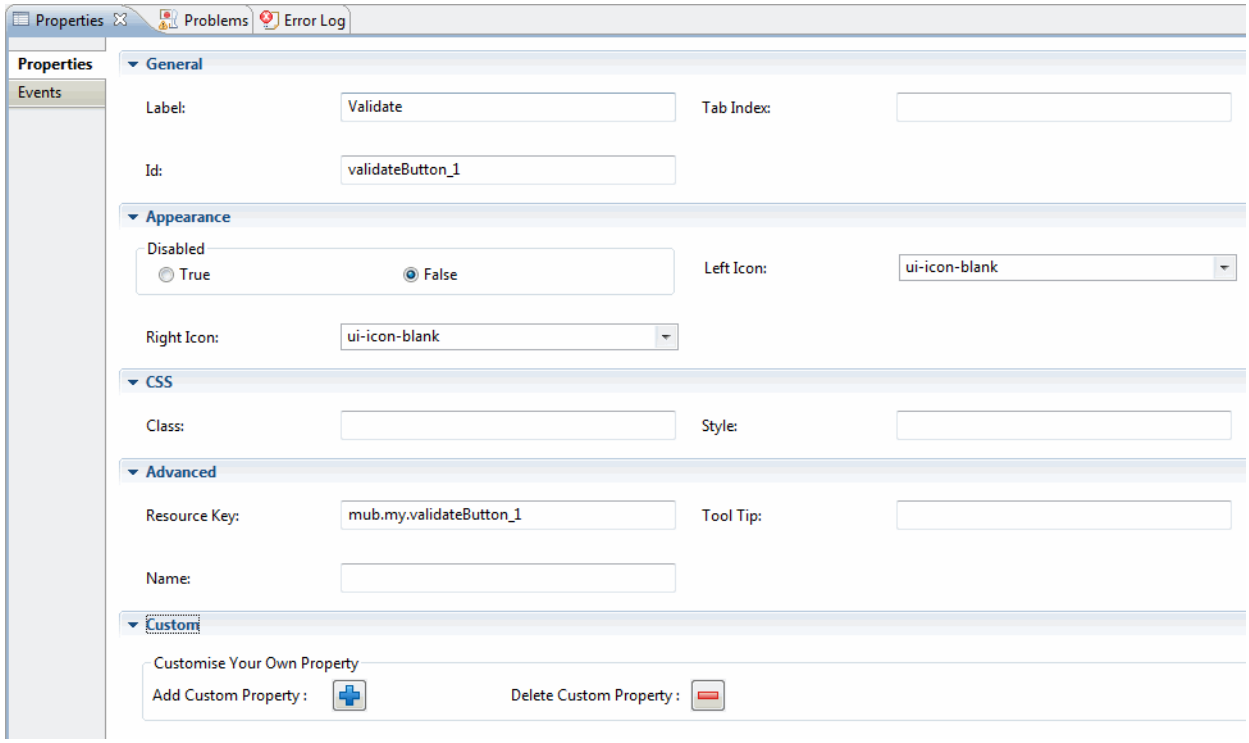
- For the selected repository, all the available attributes are displayed. Select the attribute groups and attributes that you want in the search result column.
- Click **OK**.

- 

- Based on the selected attributes, the Search page is displayed in the canvas.
- On deploying, the Search can be accessed from the custom page on TIBCO MDM Server.



Validate



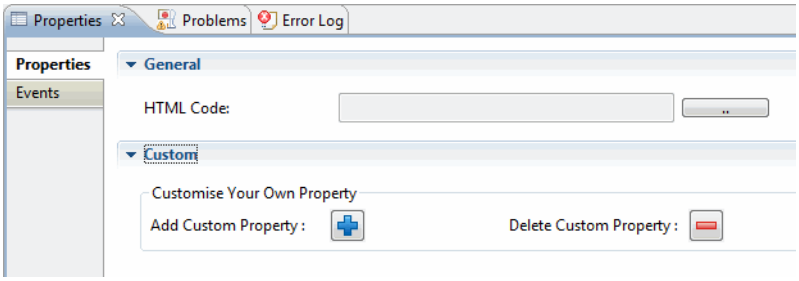
Properties	Description
Appearance	
Disabled:	Specify whether you want to disable the widget. The default value is false.
Left Icon:	Specify the icon which you want to appear inside the button on the left hand side. For example, you can set an icon of an arrow, line, dotted line, horizontal and so on.
Right Icon:	Specify the icon which you want to appear inside the button on the right hand side. For example, you can set an icon of an arrow, line, dotted line, horizontal and so on.

Other Widget Properties

In this section explains the properties for each widget palettes in the container section.

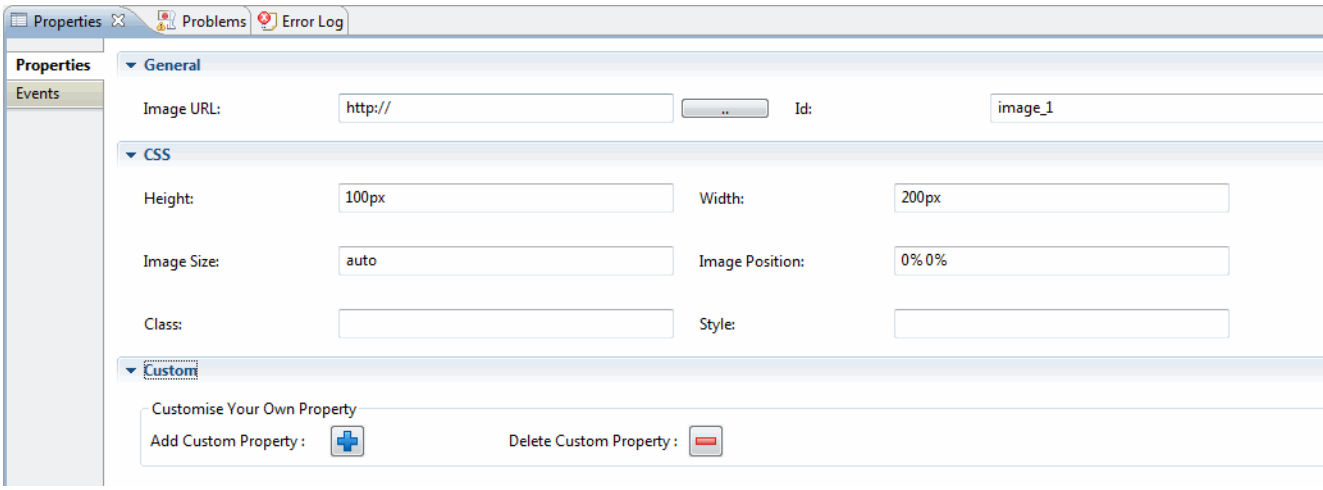
- HTML Code
- Image

HTML Code



Properties	Description
General	
HTML Code:	This property can be used to write HTML or JavaScript code which will be written within the body tag of the page. It must be valid HTML code.

Image



Properties	Description
General	
Image URL:	Specify the path of the image.
CSS	
Image Size:	The property specifies the size of the image. Default value is auto
Image Position:	Sets the starting position of the image. Default value is 0% 0%.

Custom Widgets

The custom widget allows you to add your own widgets to accomplish the tasks that cannot be done with the out-of-the-box widgets.

You must define the following requirements to configure the custom widgets in order to extend within the MDM UI Builder.

- Directory Structure
- Library.xml details
- Widgetname.xml details
- Widgetname.js details
- Widgetname.css details
- Exporting the artifacts

Directory Structure

The directory structure to be followed for a custom widget.

The directory structure will be similar to the sample structure explained below:

```
<base-dir>/
            library.xml
      <widgetname-dir>/
               <widgetname>.png
                                <widgetname>.xml
               css/
                <widgetname>.css
               javascript/
                                  <widgetname>.js
```

Library.xml Details

The `library.xml` list all the custom widgets relative to the base directory. This file needs to be include in **MDM Studio** > **Window** > **Preferences** > **UI Builder** > **Setup** > **User Extensions**.

The sample `library.xml` is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
<component file="mywid1/mywid1.xml" />
<component file="mywid2/mywid2.xml" />
<component file="mywid2/mywid3.xml" />
</library>
```

Each widget is encapsulated using the component element. The sample base-directory structure is as follows:

```
<base-dir>/
           library.xml
     mywid1/
```

```
                mywid1.xml
        mywid1.png
        css/
         mywid1.css
       javascript/
                      mywid1.js
    mywid2/
              mywid2.xml
      mywid2.png
      css/
        mywid2.css
     javascript/
                      mywid2.js
    mywid3/
             mywid3.xml
      mywid3.png
      css/
          mywid3.css
      javascript/
                    mywid3.js
```

widgetname.xml Details

The `<widgetname.xml>` defines the widget definition, which includes the name, basename, tagname, icon, properties to be captured during design time and events.

The sample xml is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>

<component label="myid1" name="myWid1" extends="base" icon="mywid1/mywid.png"
category="Container" tagName="div">

    <properties>

      <!-- property name="disabled" type="boolean"

       default="false"

     comment="Render this component disabled (default is false)."

/ -->

     <property name="Check This" type="string" default=""

       displayName="Check This" group="" comment="" />

      <property name="height" type="string" default=""

       displayName="Height" group="CSS" comment="" />

      <property name="width" type="string" default=""

       displayName="Width" group="CSS" comment="" />
```

```
    </properties>
   <events>
    </events>
</component>
```

Component Element

- **label** - Defines the display name used at widget gallery.
- **name** - Defines the unique identification name of the widget.
- **icon** - Specifies the image file to be used for graphical representation with in widget gallery. It is mandatory to have a icon image in the widget folder as specified in the folder relative to base directory.
- **extends** - Specifies the inheritance of the widget. The Default value must be 'base'.
- **category** - Specifies the group in which the widget appears in the widget gallery.
- **tagName** - Specifies the html tag to be used to generate the skeleton markup.

Properties

The list of properties is used to capture the input during design time and use the same information during runtime for rendering the widget or to do specific tasks based on the property value.

Each property can be accessed in a standard way the jQuery works. With the widget id you can access the property as `$(#widgetId).prop('propertyname')`.

List of design time properties are encapsulated using property element as shown in above example. Property element has following attributes.

- **name** - Specifies the Property name
- **type** - Specified the type of property. The valid property types are string, boolean, integer, and float.
- **default** - Specified the default value of the property.
- **comment** - Displays the help text on hovering the mouse cursor on the property.

Widgetname.js Details

The `<widgetname>.js` defines the rendering component definitions and other JavaScript functions, which are required for the widget. The default create function is called from the UIBuilder framework. The syntax of the create function is as follow:

```
com.tibco.mdm.uibuilder.WidgetBuilder.createMywid1 = function(selector, config)
{
    // Write widget creation and other configuration code
}
```

- **com.tibco.mdm.uibuilder.WidgetBuilder** - is a unique name space used to access the UI Builder's widgetbuilder object.
- **create** - is the method which is invoked during the html dom ready state.
- **Mywid1** - is the widget name for which the create method is invoked. Note that, the create method name will be of the form create<widgetname>, where widgetname's first character will be in uppercase. For example if the widget name is 'mywid1' then method name will be 'createMywid1'. Similarly if the widget name is 'myWid1' then method name will be 'createMyWid1'.
- **selector** - It is the widget id specified in the designer. At run time this will be used to access the container generated as an initial markup.
- **config** - It is a json object specified in the properties and its values which are captured during design time.

After completing the widget js file it needs to be include in **MDM Studio** > **Window** > **Preferences** > **UI Builder** > **Setup** > **JS Files**.

Widgetname.css Details

The `<widgetname>.css` defines the styles for the widget. All cascading styles are defined in this and this file needs to be include in **MDM Studio** > **Window** > **Preferences** > **UI Builder** > **Setup** > **CSS Files**.

Export Artifacts

All custom widget artifacts will be part of form specific JavaScript and css folder.

The sample of a form directory structure is as follows:

EML.war->components/<componentName>/

javascript/

mywid1.js

<formname_config>.js

css/

mywid1.css

<formname>.css

Sample Widget

The sample of google map widget definition allows rendering the google map in the specified container.

```
com.tibco.mdm.uibuilder.WidgetBuilder.createMywid =
function(selector, config){
// Set property values
if(config.width) {
$(selector).width(config.width);
}
if(config.height) {
$(selector).height(config.height);
}
var url = 'https://www.google.com/jsapi';
$.getScript(url, function(){
var container = $(selector).get(0);
google.load('maps', '3',
{
other_params: 'sensor=false',
callback: function(){
var mapOptions = {
zoom: 8,
center: new google.maps.LatLng(-34.397, 150.644)
};
var map = new google.maps.Map(container, mapOptions);
}
});
});
}
```

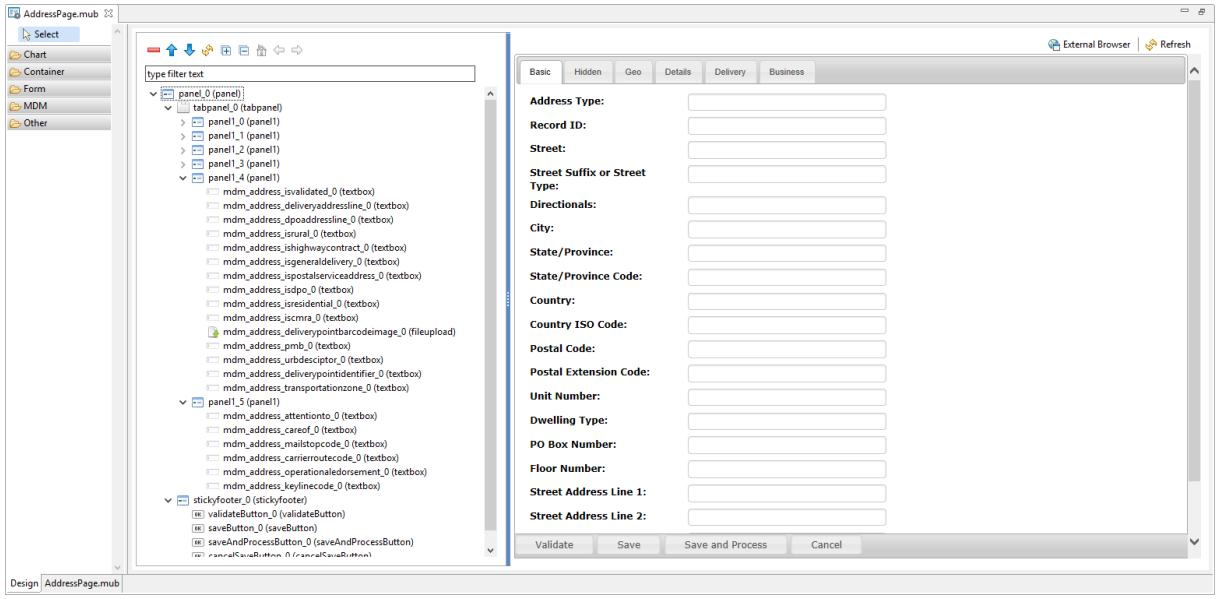
Samples

The UI builder samples are bundled in the application, these samples are based on the Customer Model in which the widgets are used. You can download these samples to your project and view their properties.

There are 8 new UI Builder samples created based on the Customer Model. These samples describe how to use new and existing widgets in the UI Builder. The sample UI Builder pages (.mub files) are available under **UI Builder** folder of the **Customer Model**. Import the **Customer Model** zip from the

MDM Model Templates available in the **TIBCO MDM Samples 5.1. Samples** folder to your studio project to view the samples.

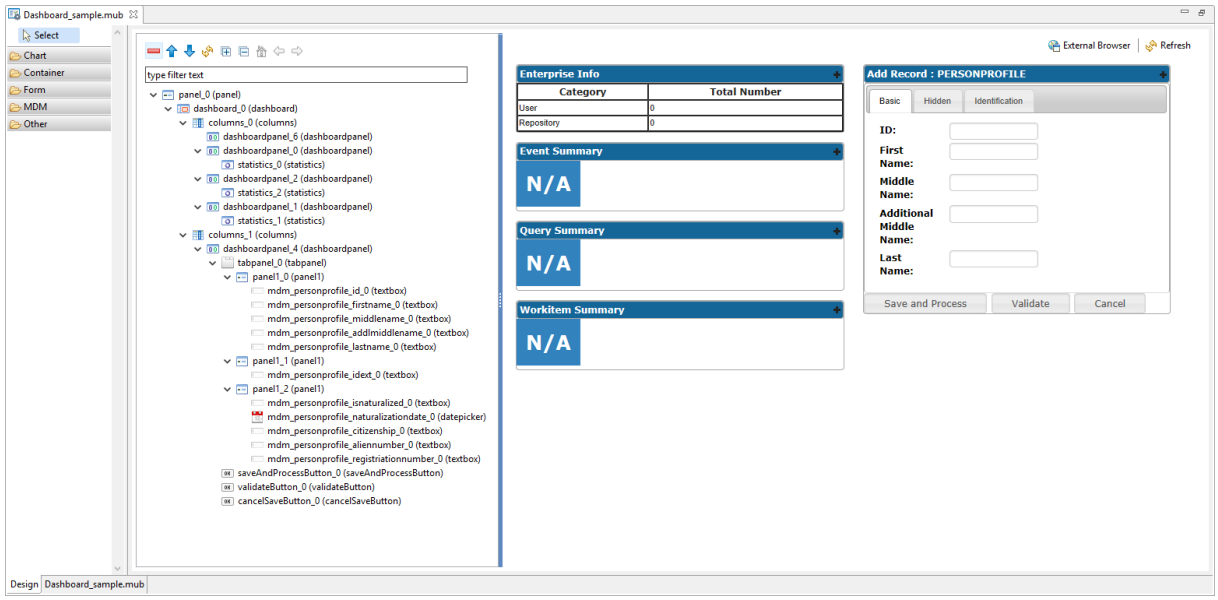
The **Address Page Sample** covers the usage of the following widgets:



- Panel
- Tab Panel
- Text area
- Numberix Text box
- File Upload

View the widget details from the properties tab in the **AddressPage.mub** page of your project.

The **Dashboard sample** covers the usage of the following widgets:

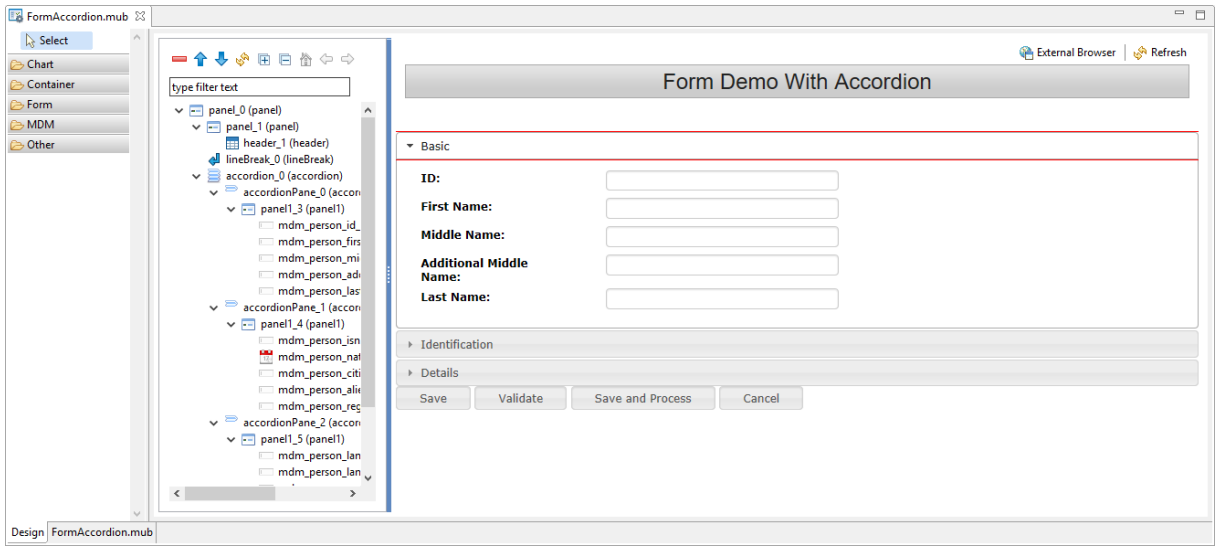


- Panel
- Dashboard

- Columns
- Dashboardpanel
- MDM Statistics
- Tab Panel
- Save and Process Button
- Validate Button
- Cancel Button

View the widget details from the properties tab in the **Dashboard.mub** page of your project.

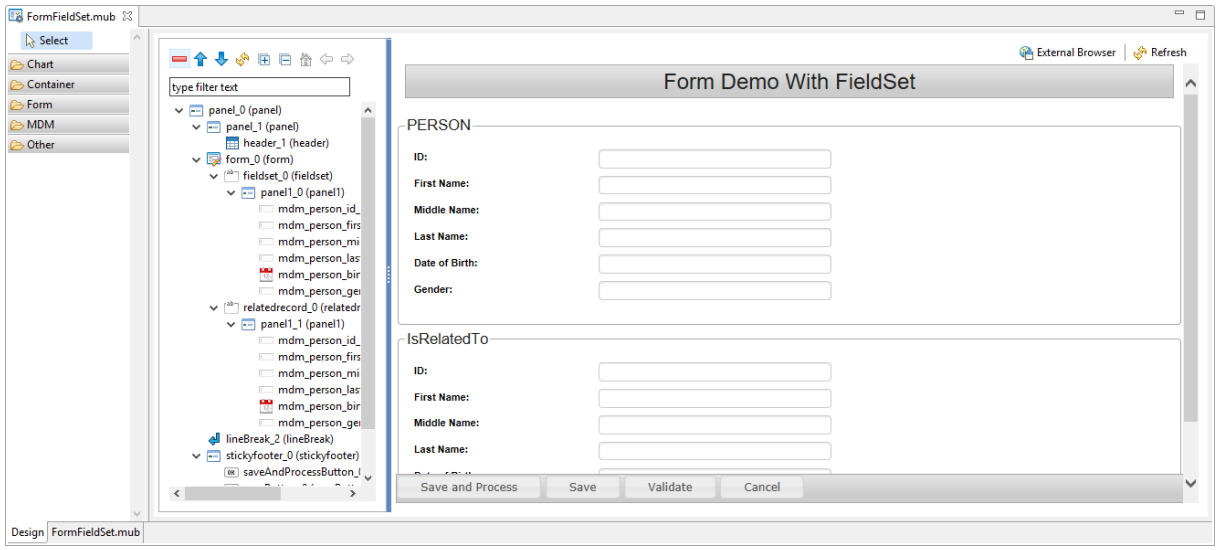
The **Accordion Form Sample** covers the usage of the following widgets:



- Accordion
- Panel
- Header
- AccordionPane
- Text Box
- Date Picker
- Timestamp Picker
- URL
- Line Break Panel
- File Upload

View the widget details from the properties tab in the **FormAccordion.mub** page of your project.

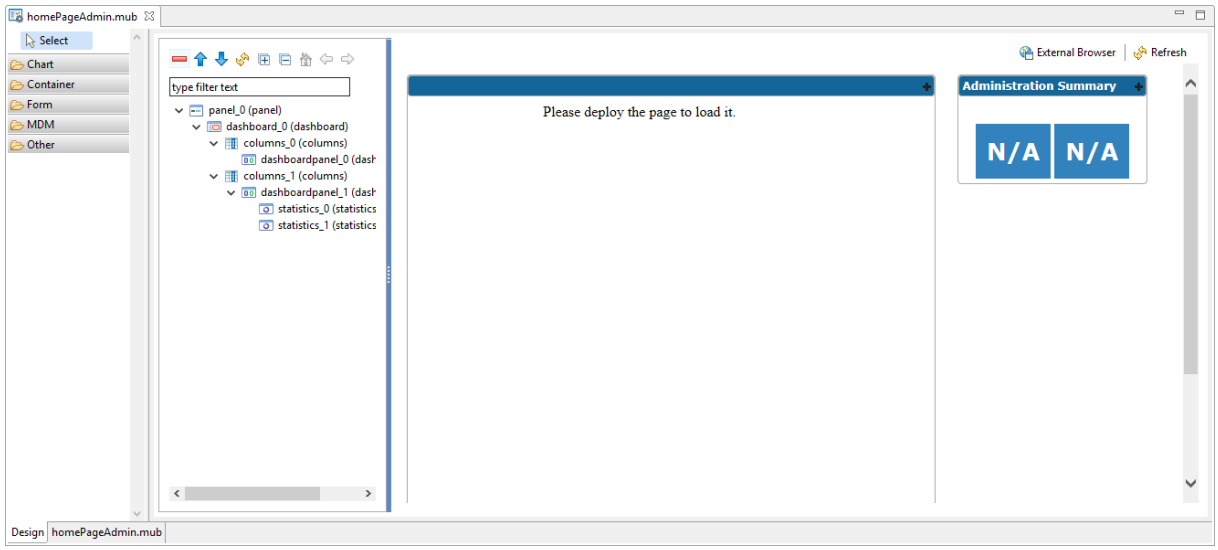
The **FieldSet Form Sample** covers the usage of the following widgets:



- Panel
- Header
- Form
- Fieldset
- Numeric Text Box
- Text Box
- Date Picker
- URL
- Line Break
- Sticky Footer

View the widget details from the properties tab in the **FormFieldSet.mub** page of your project.

The **Admin Home Page Sample** covers the usage of the following widgets:

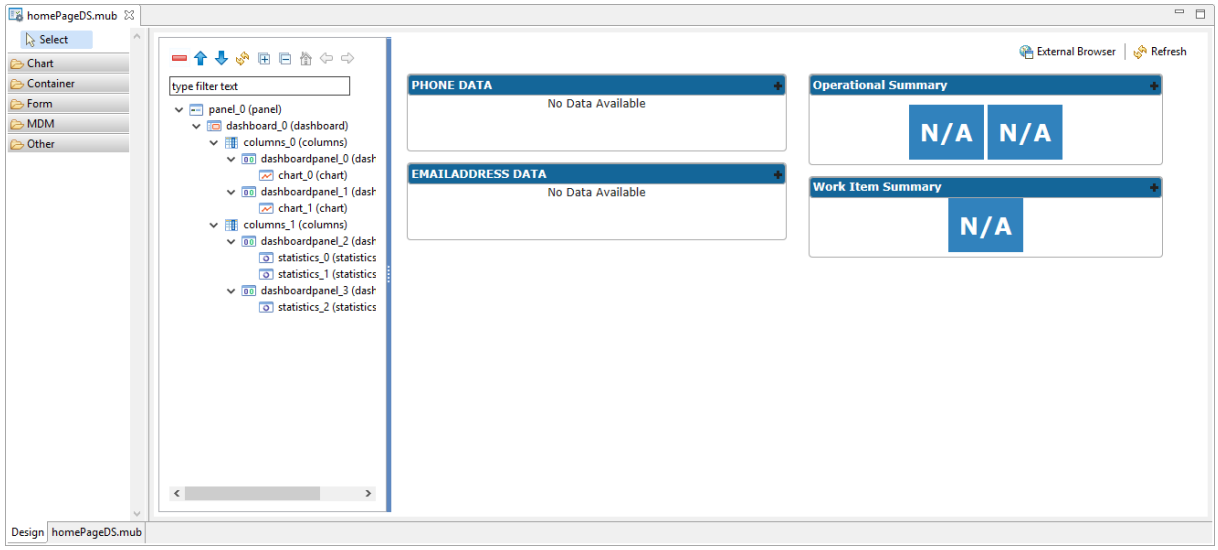


- Panel
- Dashboard

- Columns
- DashboardPanel
- Statistics

View the widget details from the properties tab in the **homePageAdmin.mub** page of your project.

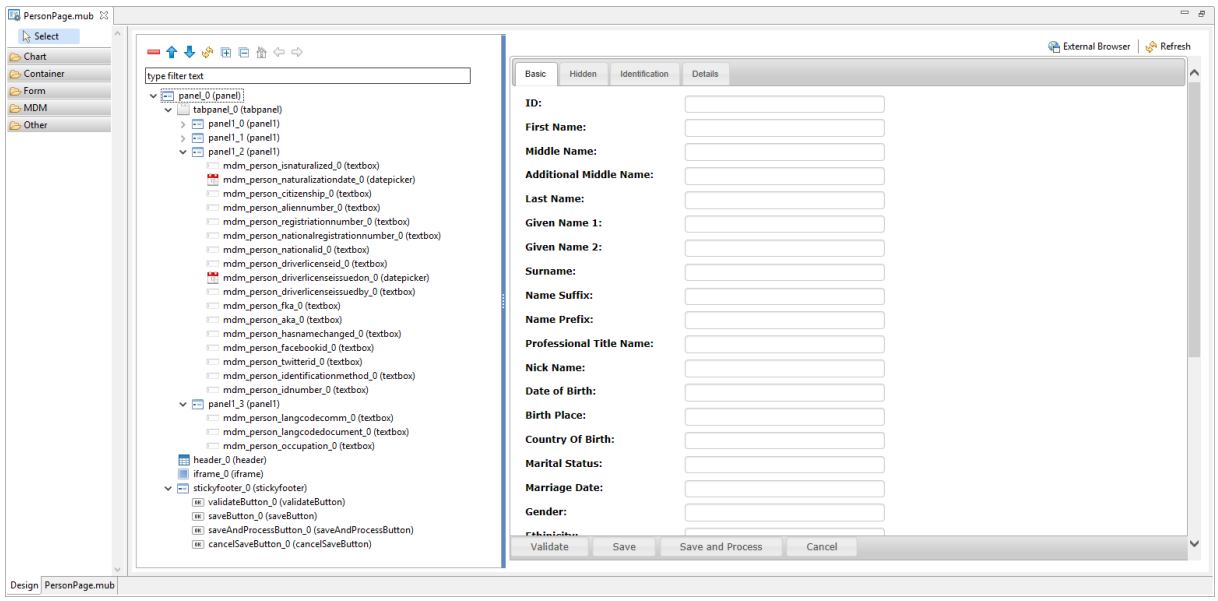
The **Data Steward Role Home Page Sample** covers the usage of the following widgets:



- Panel
- Dashboard
- Columns
- DashboardPanel
- chart
- Statistics

View the widget details from the properties tab in the **homePageDS.mub** page of your project.

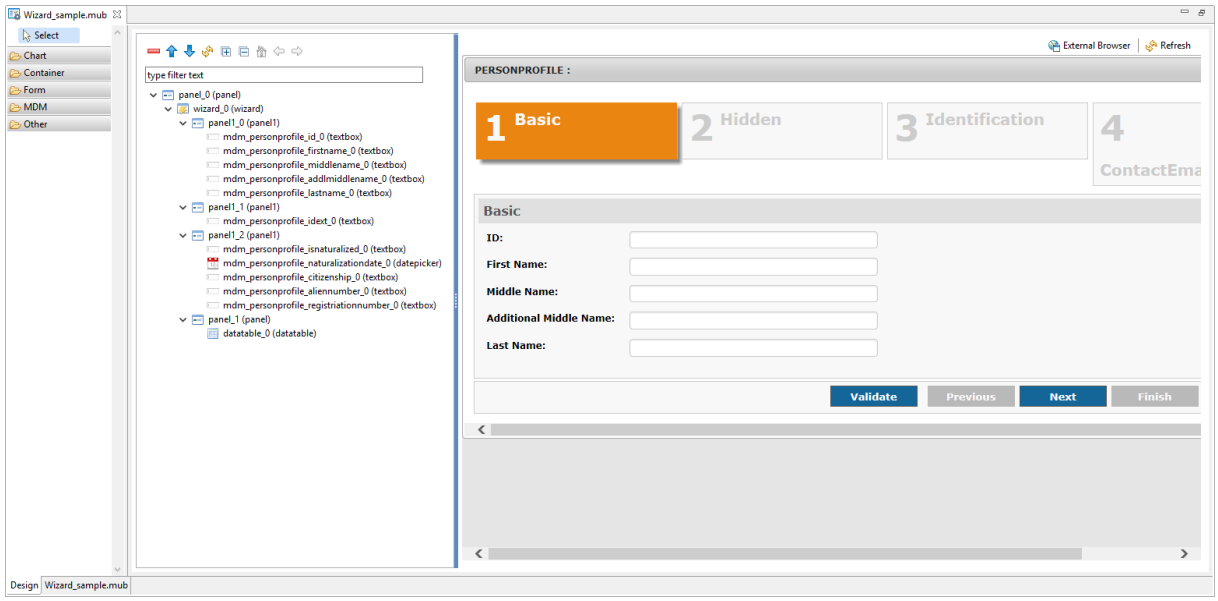
The **Person Page sample** covers the usage of the following widgets:



- Panel
- Tab Panel
- Text area
- Date Picker
- Header
- iframe
- Sticky Footer
- Validate Button
- Save Button
- Save and Process Button
- Cance Button

View the widget details from the properties tab in the **PersonPage.mub** page of your project.

The **Wizard sample** covers the usage of the following widgets:



- Panel
- Wizard
- Text area
- Date Picker
- File Upload
- Data Table
- Cloumn Editor
- Various Events like LeaveStep, ShowStep, Finish, Validate, Cancel, Show

View the widget details from the properties tab in the **Wizard_sample.mub** page of your project.

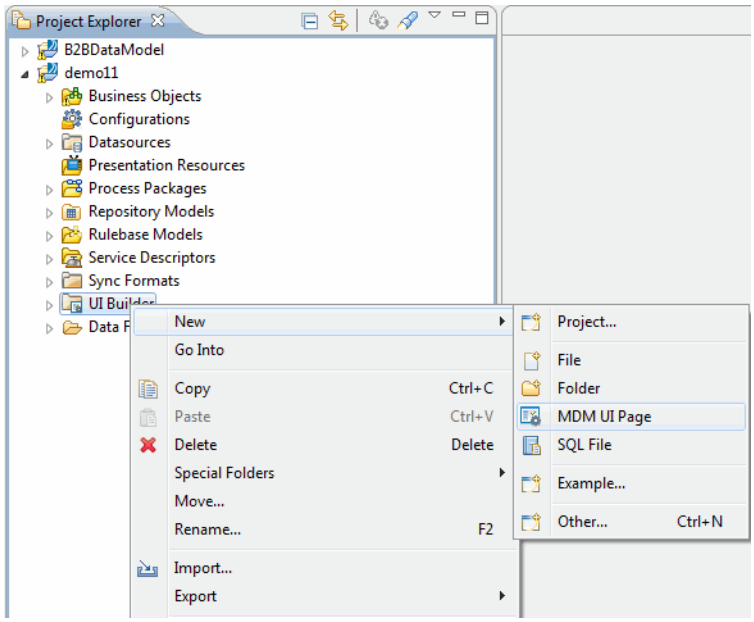
Creating a MDM UI Page

The scenario covers creating a new UI page to add person details. Use the B2B model available in TIBCOmmunity site. The person repository has multiple fields. The custom page will only display the

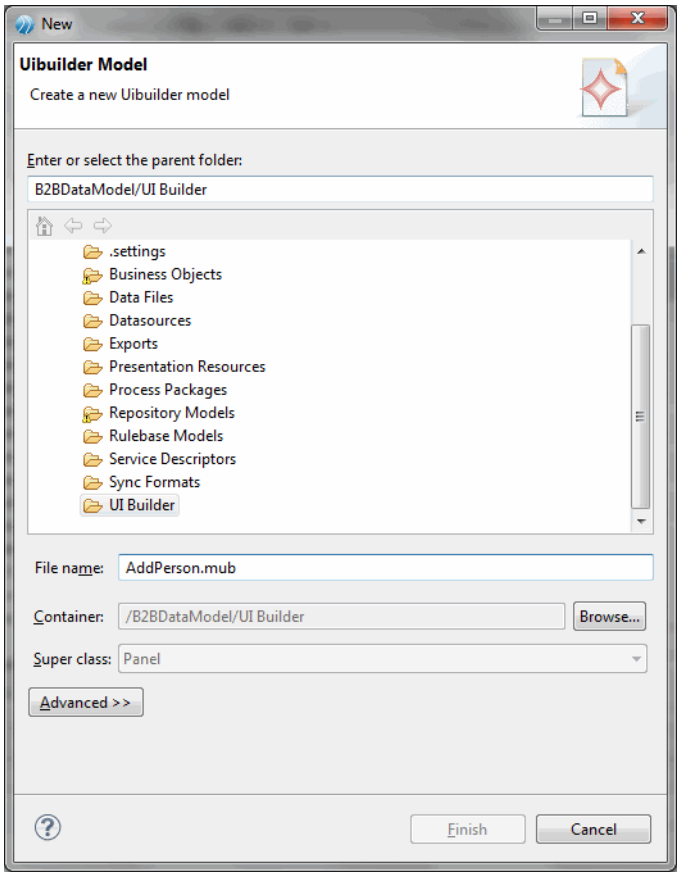
selected fields from the person repository. The new UI pages can be used as the new Home Page on server side.

Procedure

1. In the Project Explorer, Right click the **UI Builder** special folder and select **New** > **MDM UI Page** .

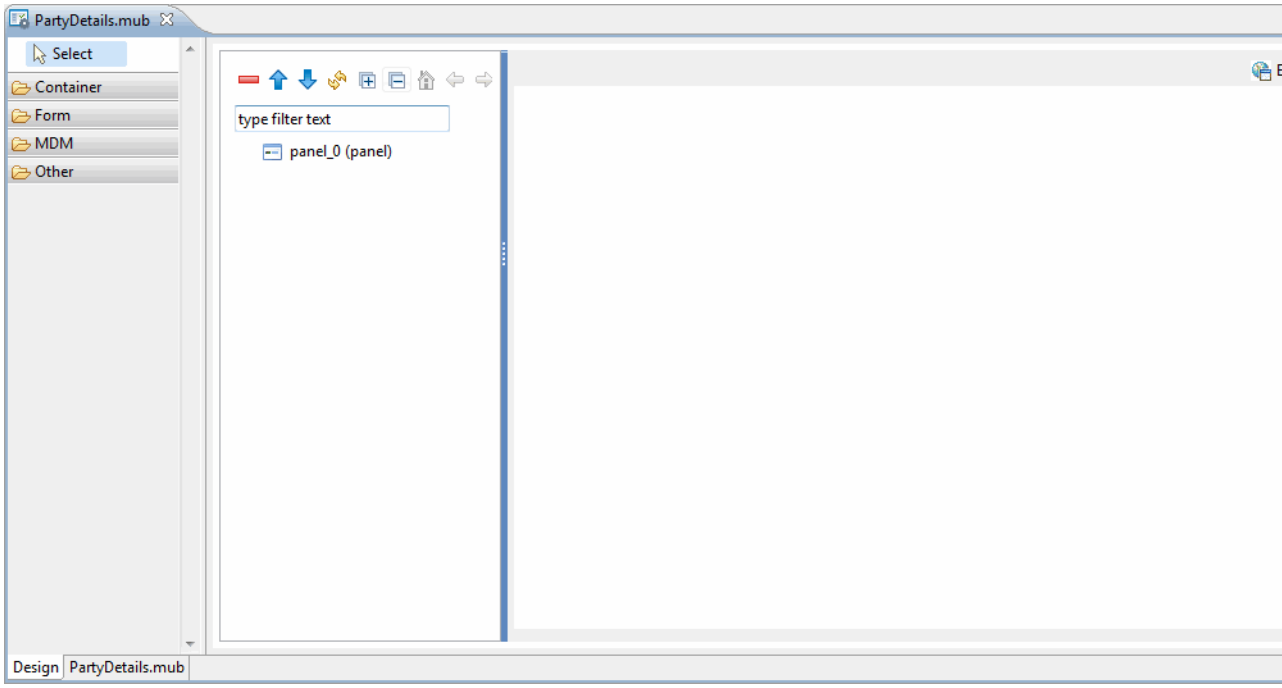


2. The **Uibuilder Model** wizard is displayed.

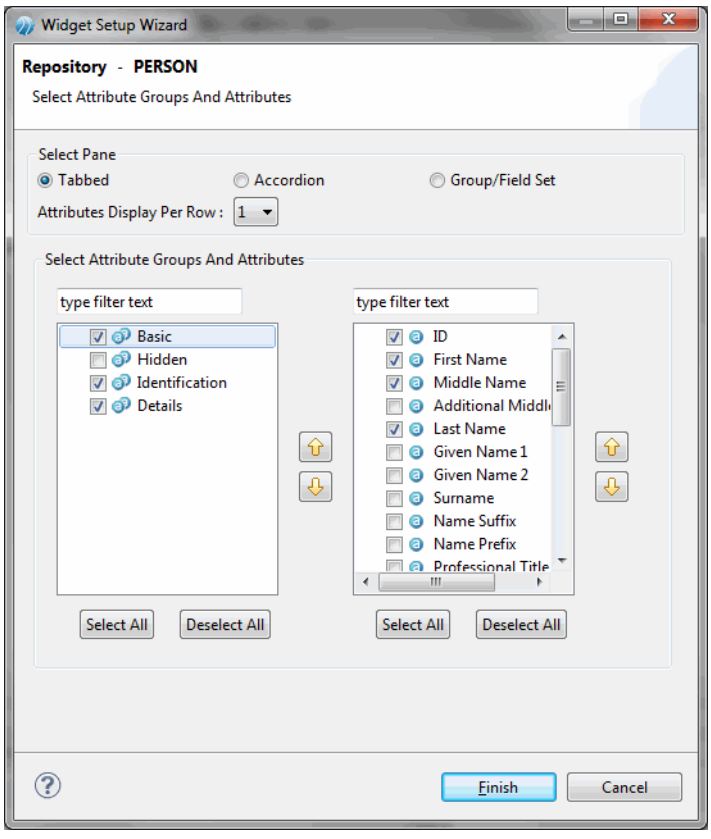


3. Enter the UI Builder model name in the **File name** field and click **Finish**.

4. The UI Builder canvas is displayed.

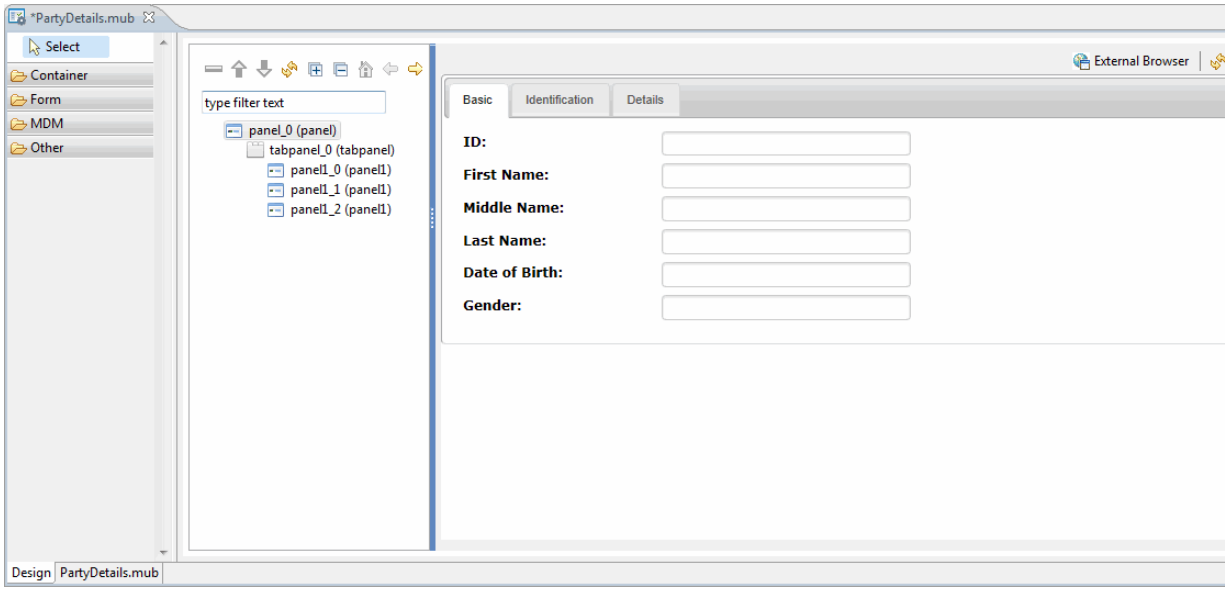


5. Drag and drop the **Person** repository on the panel. The Widget Setup Wizard is displayed.

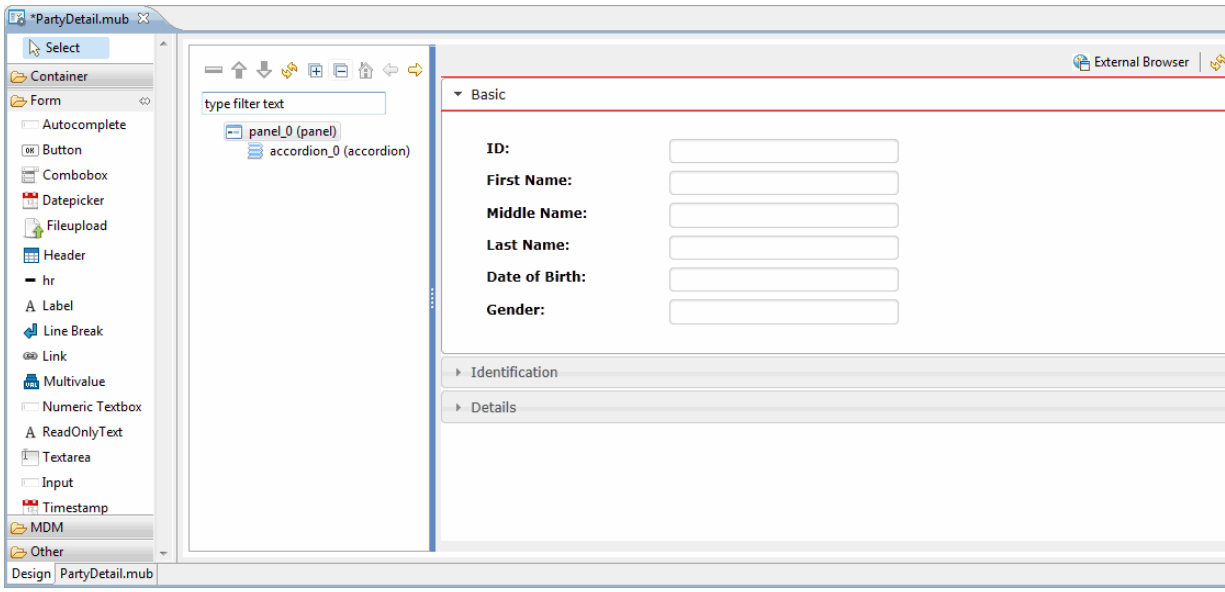


6. The **Select Pane** section displays three options, **Tabbed**, **Accordion**, and **Group/Field Set**.

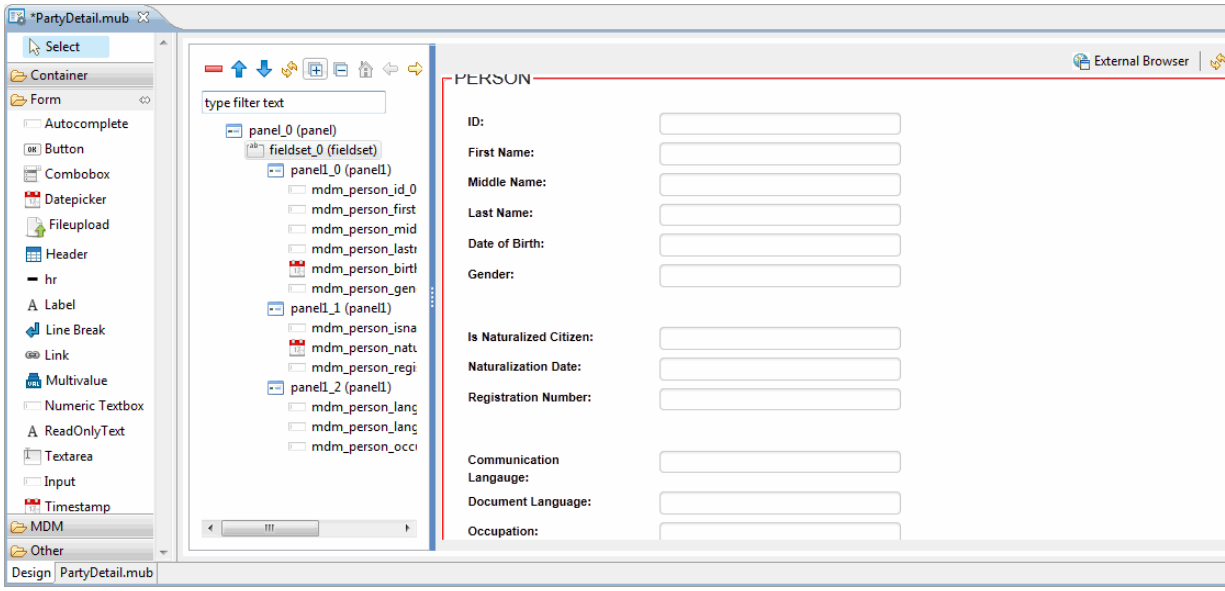
   - On selecting **Tabbed** check box the following screen is displayed.



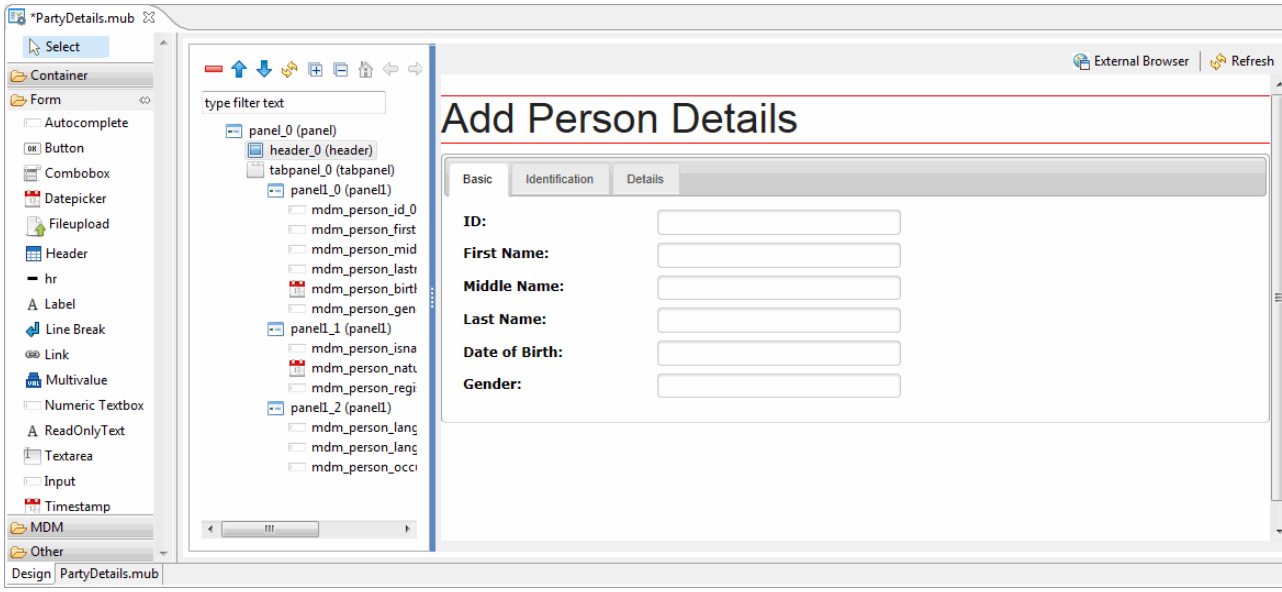
- On selecting **Accordion** check box the following screen is displayed.



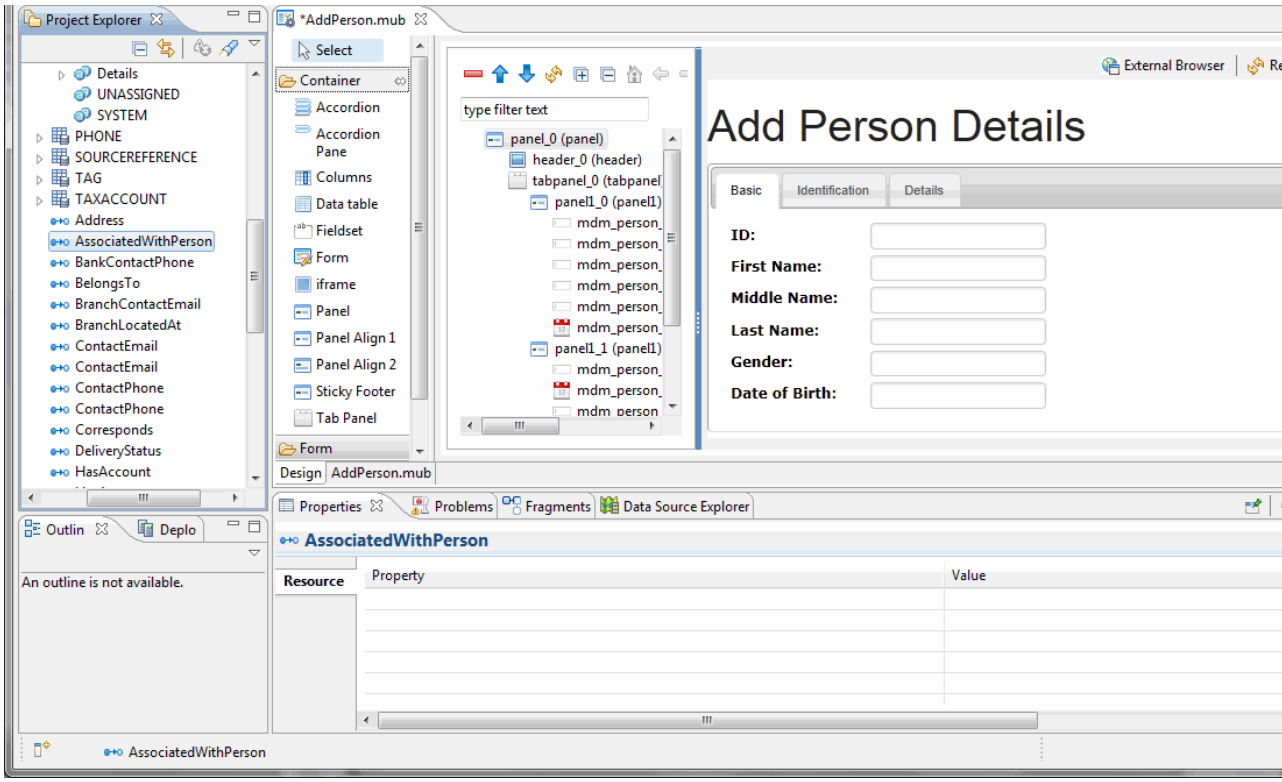
- On selecting **Group/Field Set** check box the following screen is displayed.



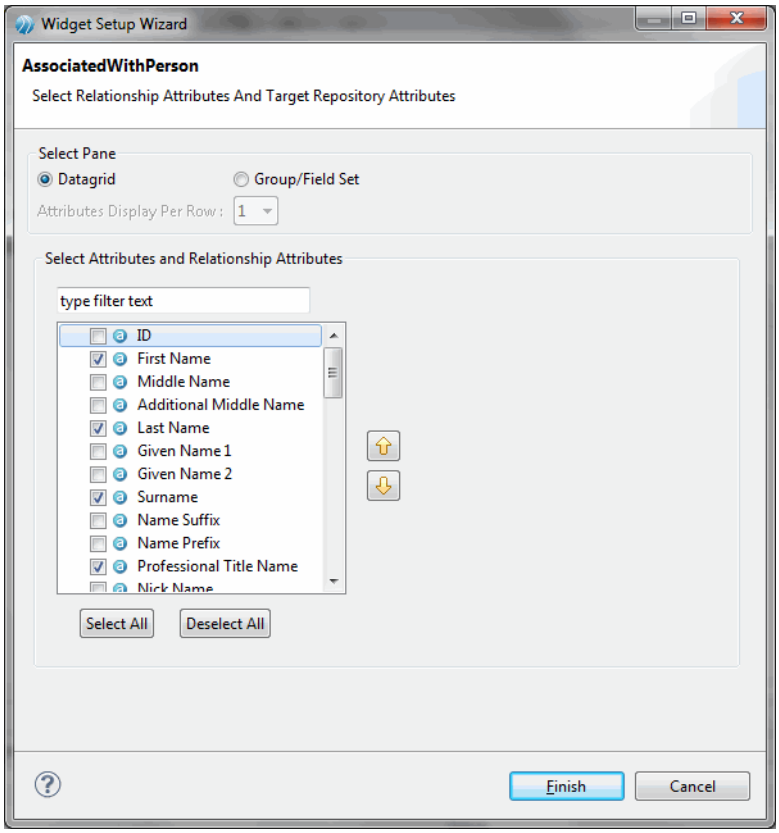
7. From the **Select Pane** section select the **Tabbed** check box.
8. Select the checkbox corresponding to the repository attributes and click **Finish**.
9. From the **Form** Widget, select **Header** and click on the panel. Specify the header value in the **Text** field in the properties section. By default, the header widget appears at the end, you can re-order it by clicking on the Tree Outline toolbar and move it before **panel_0**.



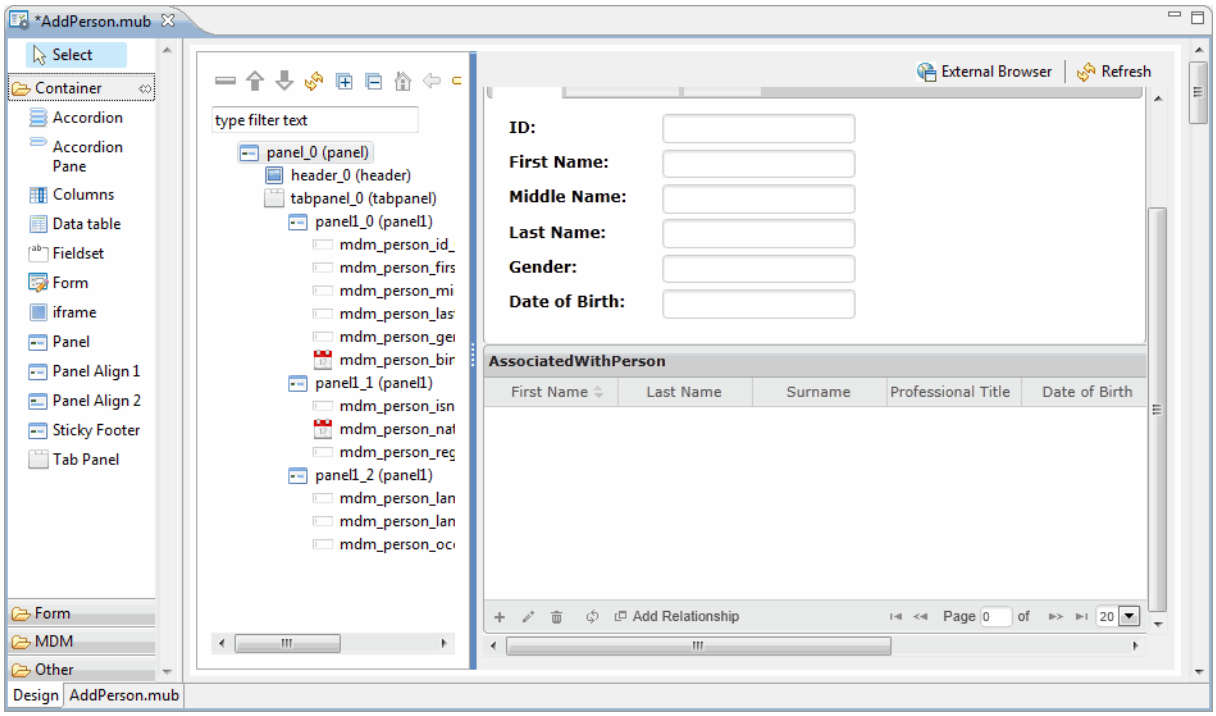
10. You can create repository relationship by dragging the relationships under repository model in the project explorer. For example, drag the AssociatedWithPerson relationship. This relationship shows that Party (client) is associated with person.



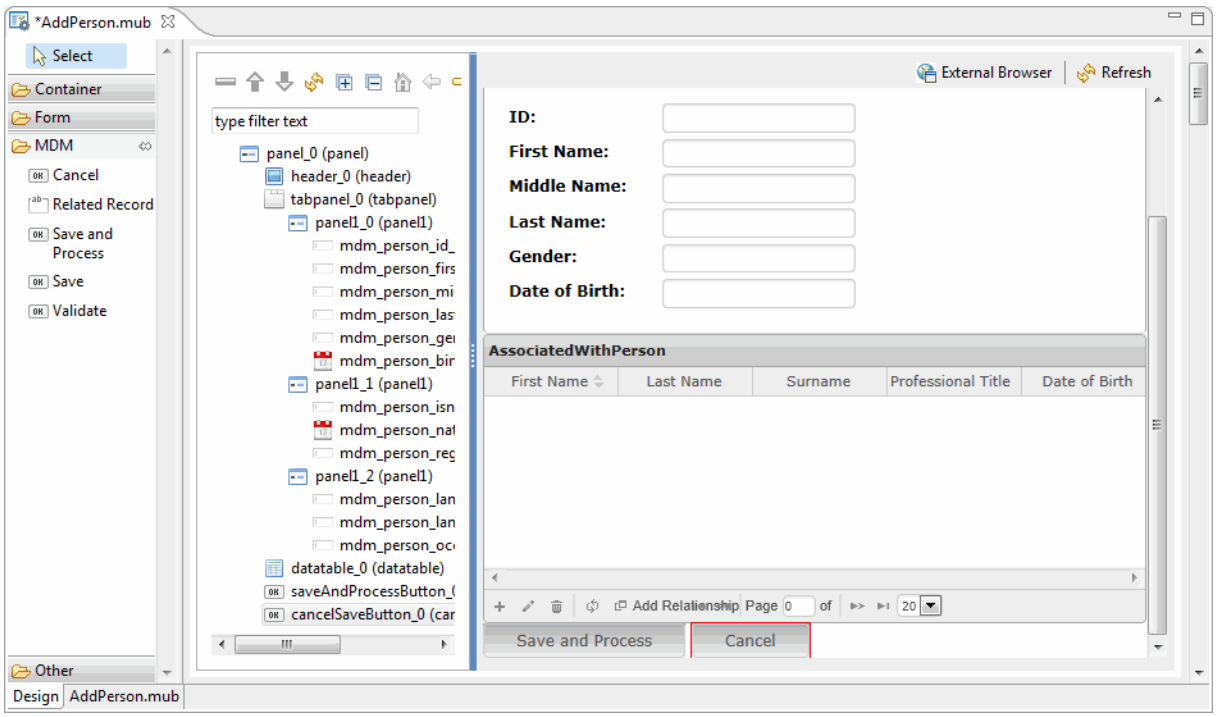
11. The **Widgets Setup Wizard** for selecting the relationship attributes is displayed.



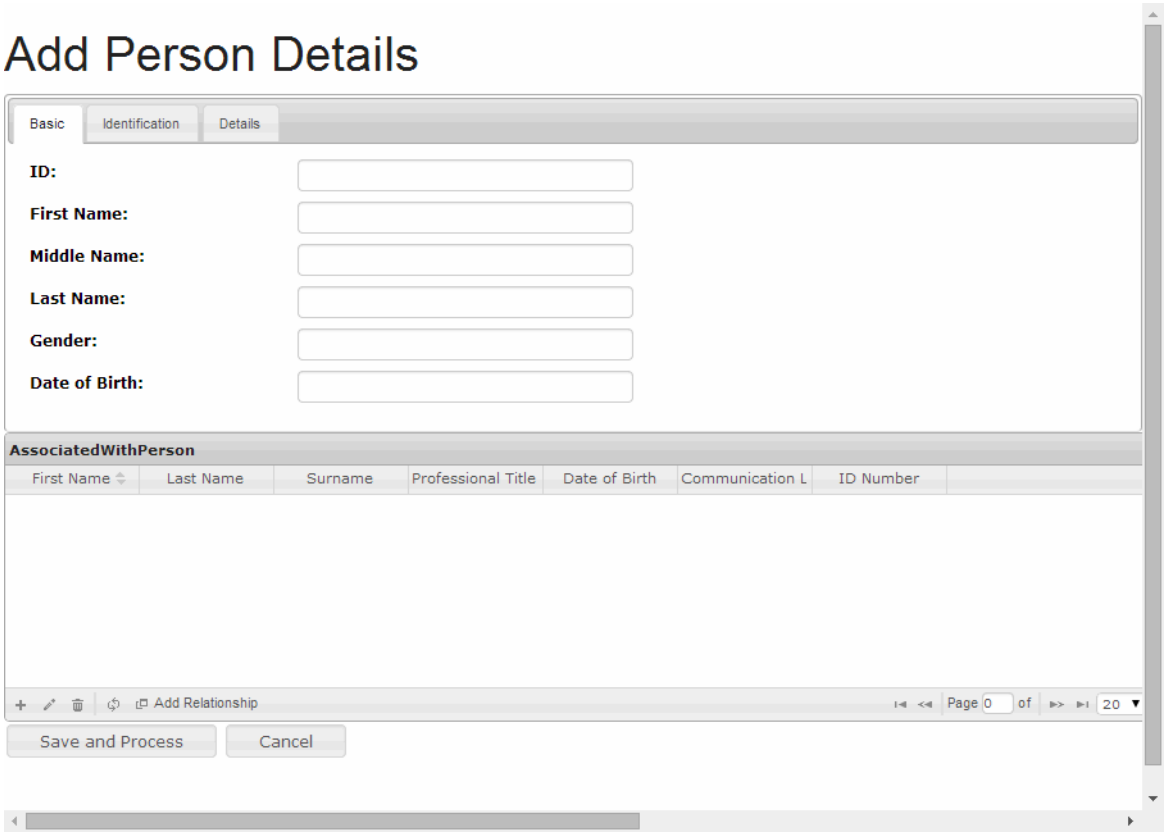
12. Select the **Datagrid** from the **Select Pane** section. Select the attributes that should be part of the datagrid and click **Finish**.



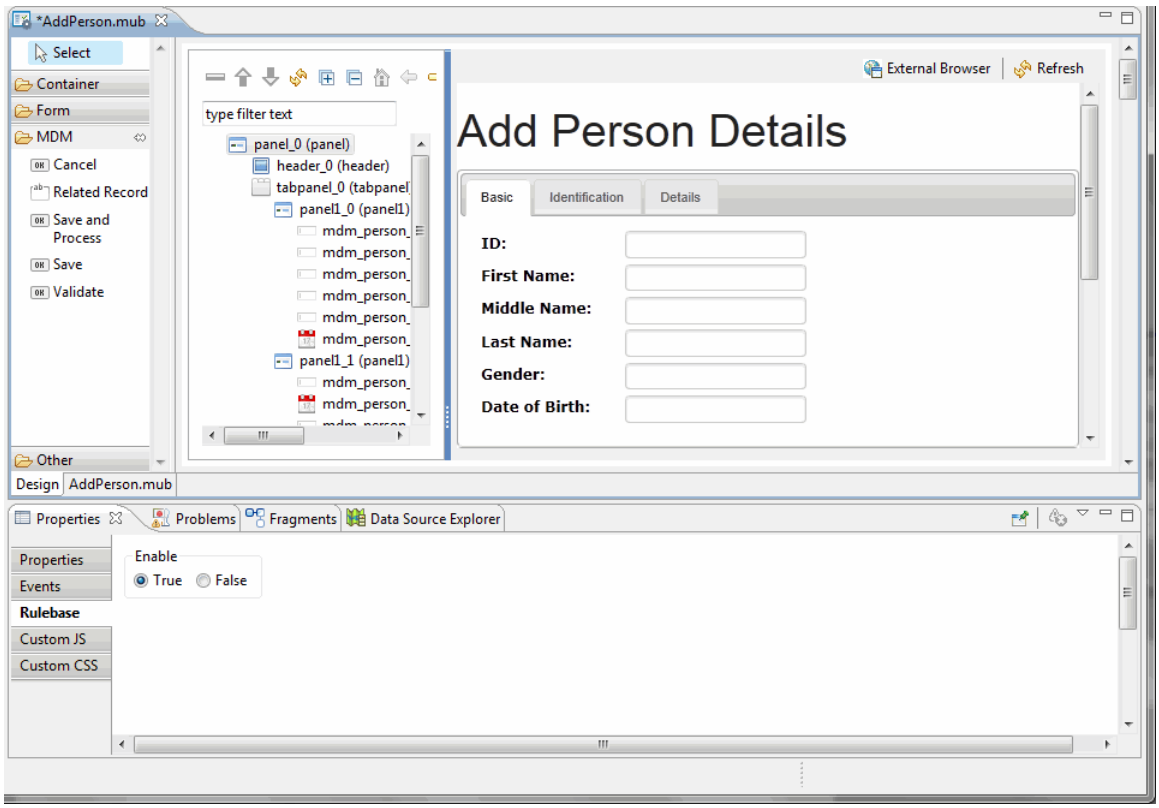
13. From the **MDM** widget, select **Save and Process** button and drop it on the panel. Similarly select **Cancel** button and drop it on the panel. The **Save and Process** and **Cancel** buttons are displayed on the canvas area.



14. Click the **External Browser** on the top right corner of the canvas to view the UI page on a browser.



15. To associate the rulebase with the UI page, select the root level panel and click on the **Rulebase** on the properties section.
16. Select **True** to enable the rulebase.



Deploy MDM UI Page

The newly created UI page must be deployed on the MDM application in order to use it. You can deploy the UI page in either of the follow ways:

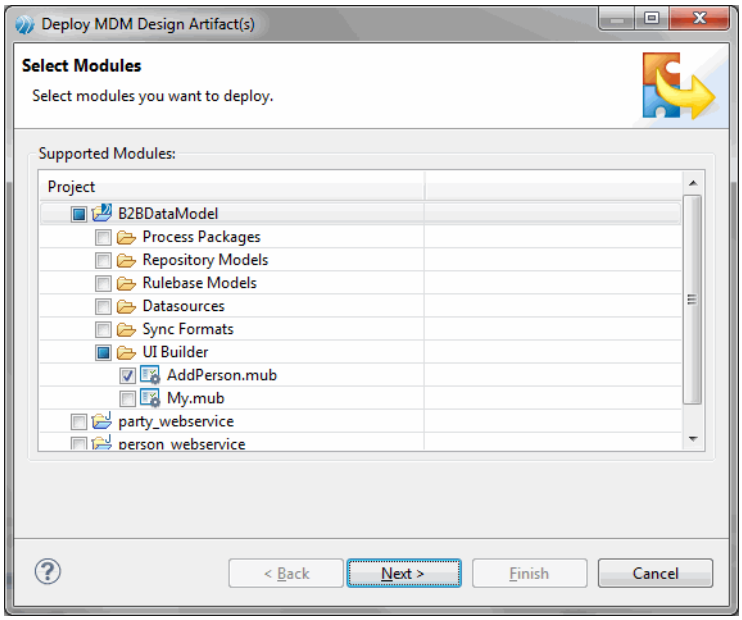
- **Add to Menu**: You can add the UI page to an existing menu or custom page menu or new menu.
- **Link to Metadata Operation**: You can link the UI page to the metadata operation like add, modify and view.

Out of box Record UI is self contained and does not point to custom pages.

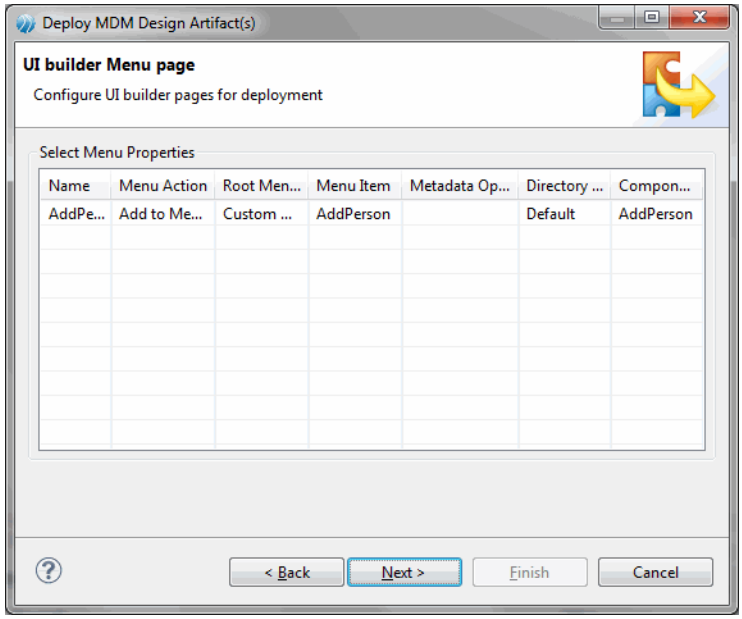
Adding to Menu (Custom Page or Existing Menu or as New Menu)

Procedure

1. On the **Deployment Servers** section, Right click the connected server and click **Deploy Module**.

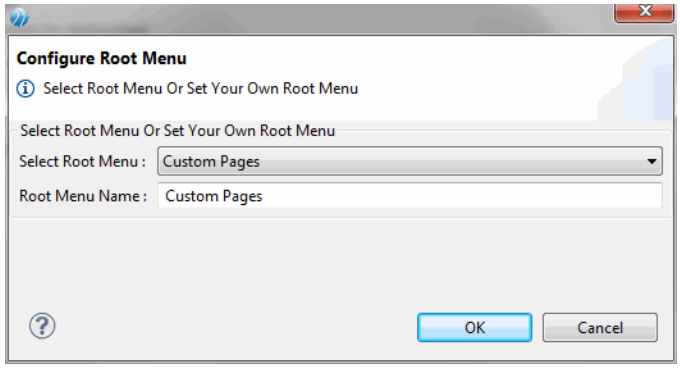


2. Select the newly created MDM UI project. For example, **AddPerson.mub** and click **Next**.
3. The **UI Builder Menu page** is displayed.



4. You can configure the UI builder pages. The configuration properties are as following.

- **Name**
  By default the name of UI builder page is displayed. You cannot modify the default UI builder name.
- **Menu Action**
  The menu action has **Add to Menu** and **Link to Metadata Operation** options. Select **Add to Menu** option.
- **Root Menu Name**
  By default, the root menu name is set to **Customs Pages**. You can select a different menu option in which you want the new UI page to appear or you can select an existing menu name or you can create a new menu. Click on the **Root Menu Name** column, the Configure Root Menu dialog appears.

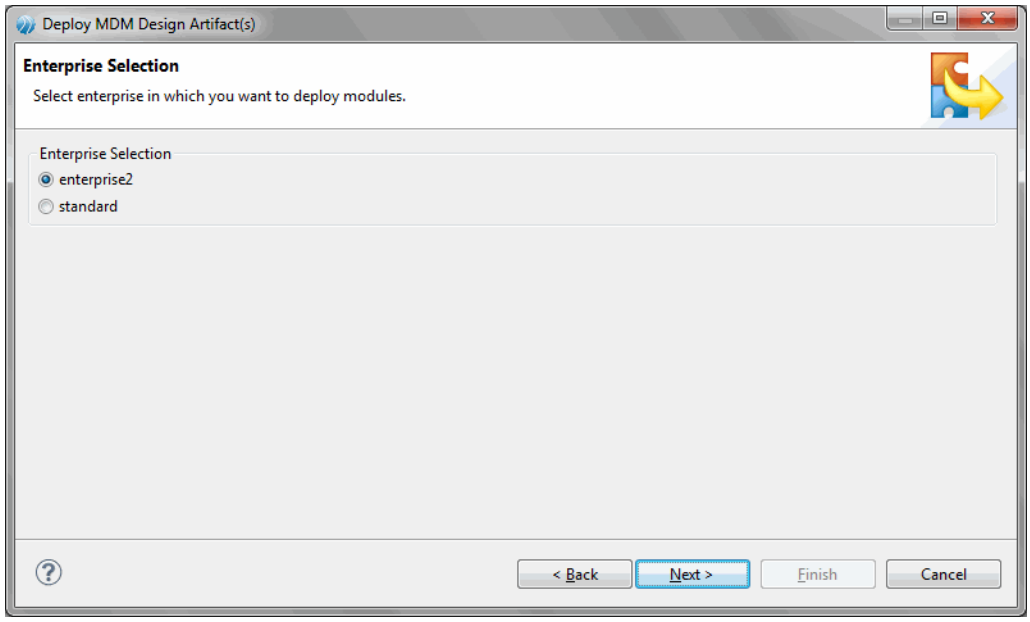


From **Select Root Menu** drop-down list, select the **Custom Pages** option if you want to add the new UI page to the Custom Pages or Select an existing menu. By default, the **Root Menu Name** field displays the root menu name as Custom Pages, you can modify the root menu and click **OK**.

- **Menu Item**
  By default the UI page name is the menu item name, you can configure it to a different name.
- **Metadata Operation**
  Since **Add to Menu** was selected in the **Menu Action** column, the **Metadata Operation** column is disabled.

- **Directory Structure**
  You can select the following directory structures:
  Default - All the artifacts of the UI page are saved in the default directory. Hence the directory structure is set to default. All the common artifacts are saved in the default directory.
  Standalone - You can change the directory structure to standalone. In standalone, the page specific artifacts are saved in the standalone directory with the UI page name.
- **Component Name**
  The default component name is same as the UI page name. You can modify the default component name. However, ensure that the component name is unique and does not contain whitespaces.

5. Click **Next**.
6. The **Enterprise Selection** page is displayed.

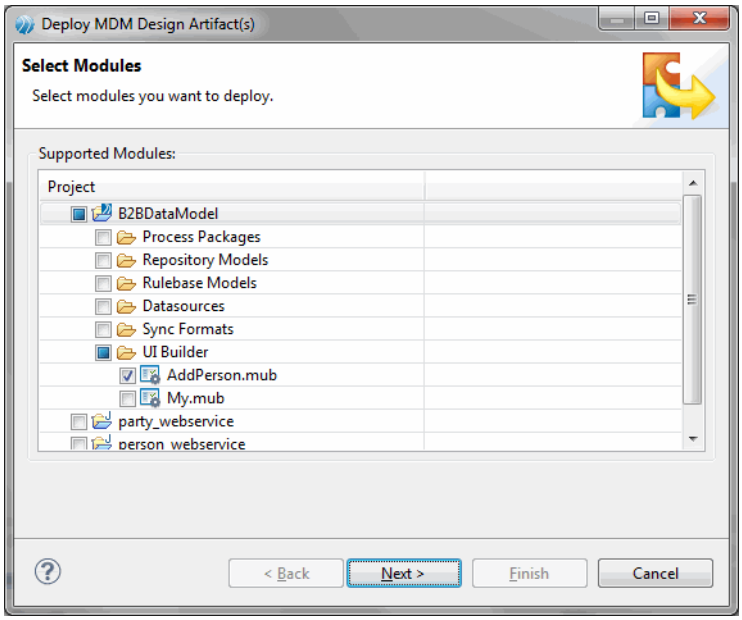


7. Select the enterprise in which you want to deploy the MDM UI page artifacts and click **Next**.
8. Click **Finish** the MDM UI Page artifacts are successfully deployed.

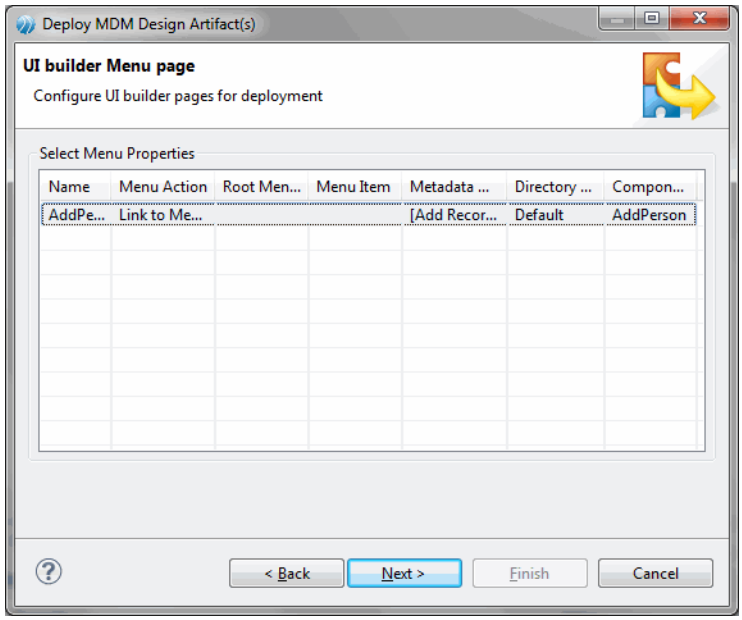
Linking to a Metadata Operation

Procedure

1. In the **Deployment Servers** section, Right click on the connected server and click **Deploy Module**.



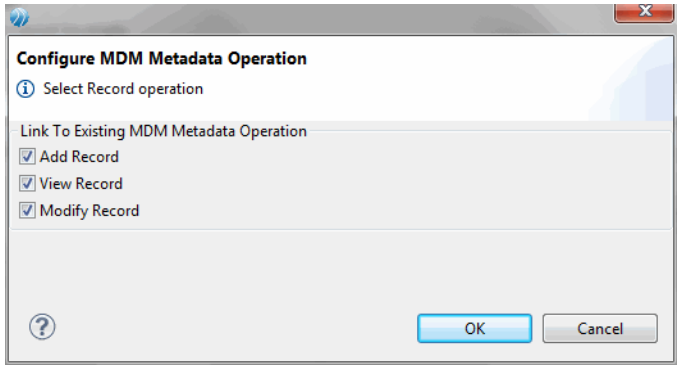
2. Select the newly created MDM UI project. For example, **AddPerson.mub** and click **Next**.
3. The **UI Builder Menu page** is displayed.



4. You can configure the UI builder pages. The configuration properties are as following.

- **Name**
  By default the name of UI builder page is displayed. You cannot modify the default UI builder name.
- **Menu Action**
  The menu action has **Add to menu** and **Link to Metadata Operation** options. Select **Link to Metadata Operation**.
- **Root Menu Name**
  Since you selected **Link to Metadata Operation**, the **Root Menu Name** column is disabled.
- **Menu Item**
  Similarly since you selected **Link to Metadata Operation**, the **Menu Item** column is also disabled.

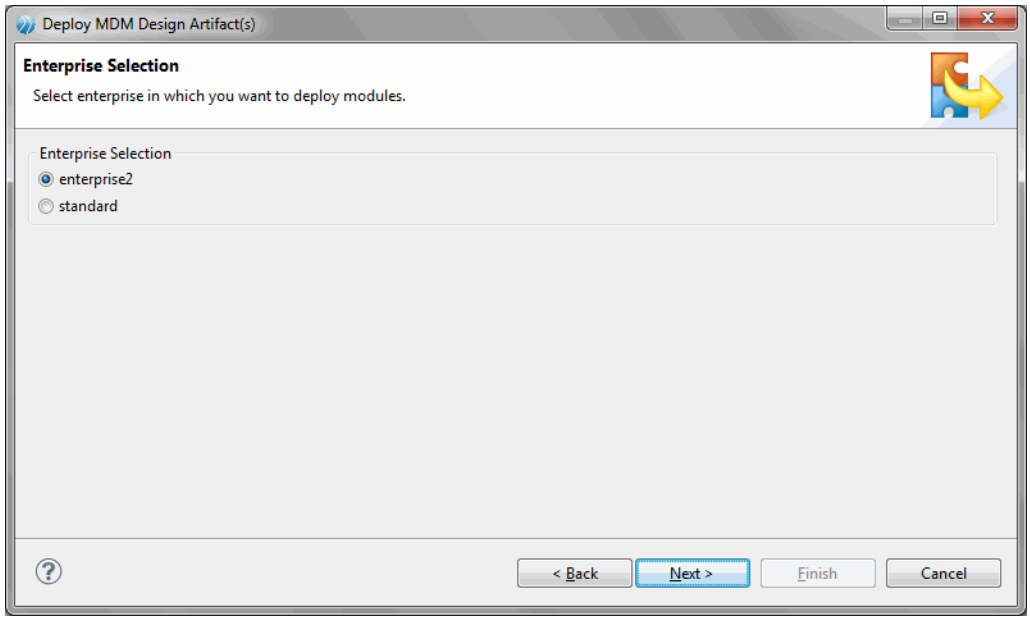
- **Metadata Operation**
  On selecting **Link to metadata Operation** option the **Metadata Operation** column is enabled. Click the **Metadata Operation** column. The metadata operation provides the **Add Record**, **View Record**, and **Modify Record** options.



From **Link To Existing MDM Metadata Operation** section, select the appropriate option. The available options are **Add Record**, **View Record**, and **Modify Record**. Click **OK**.

- **Directory Structure**
  You can select the following directory structures:
  Default - All the artifacts of the UI page are saved in the default directory. Hence the directory structure is set to default. All the common artifacts are saved in the default directory.
  Standalone - You can change the directory structure to standalone. In standalone, the page specific artifacts are saved in the standalone directory with the UI page name.
- **Component Name**
  The default component name is same as the UI page name. You can modify the default component name. However, ensure that the component name is unique and does not contain whitespaces.

5. Click **Next**.
6. The **Enterprise Selection** page is displayed.



7. Select the enterprise in which you want to deploy the MDM UI page artifacts and click **Next**.
8. Click **Finish** the MDM UI Page artifacts are successfully deployed.

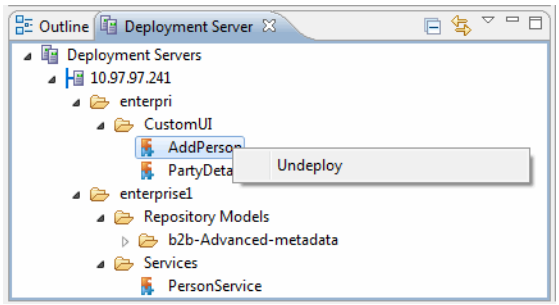
Undeploying the MDM UI Builder Page

You can undepoy the UI builder page from the server by following these steps.

Procedure

1. In the Deployment Server view, expand Deployment Server.
2. Then expand <MDM Server><EnterpriseName>**<Custom UI > <Component name>**. You will find the listings of the UI Builder forms. The forms are listed based on their component names. For example, AddPerson .
3. Right click on the deployed **Component Name (AddPerson)** and select **Undeploy**.
4. You will get a message to confirm undeployment. Click **Yes** to undeploy.

Result



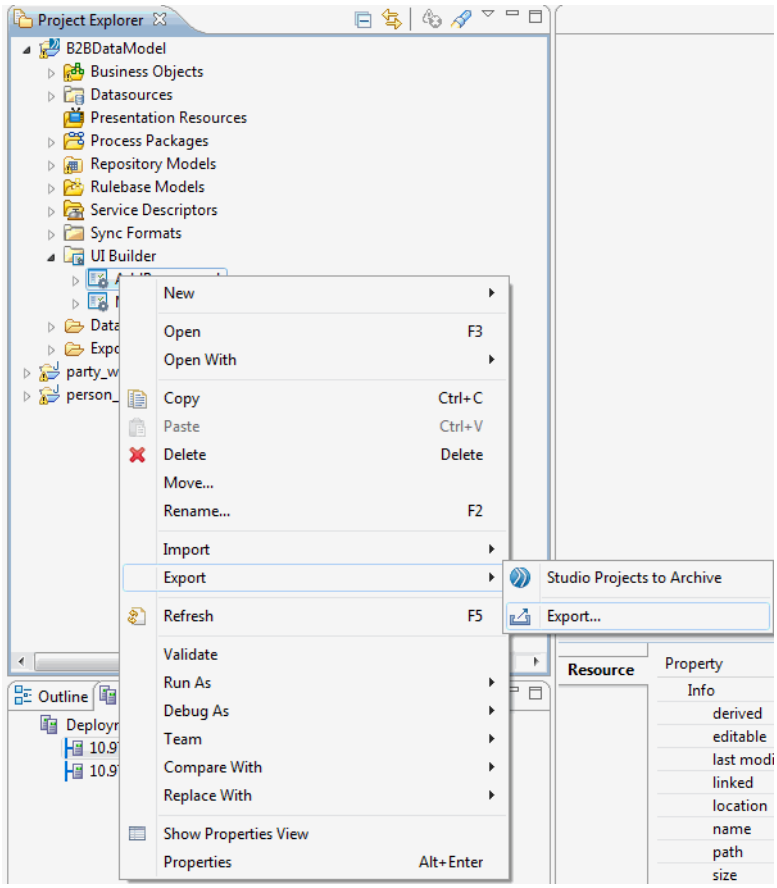
This undeploys the selected component, and a backup of it is internally renamed and stored.

Exporting the UI Page Artifacts

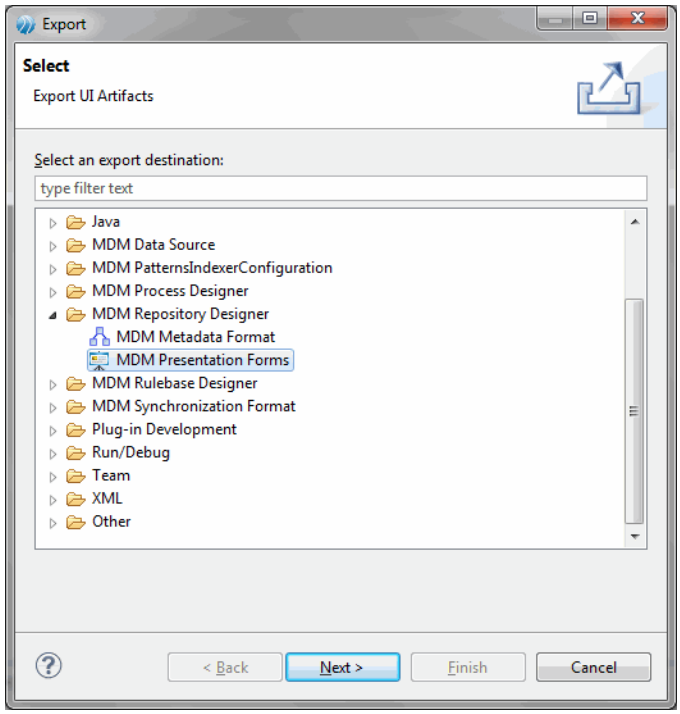
Once the UI page design is complete, you can export the UI page artifacts if you want to deploy the UI page manually or take backup of the designed artifacts. The artifact contains HTML page, JavaScript libraries and generated code, CSS files, and images.

Procedure

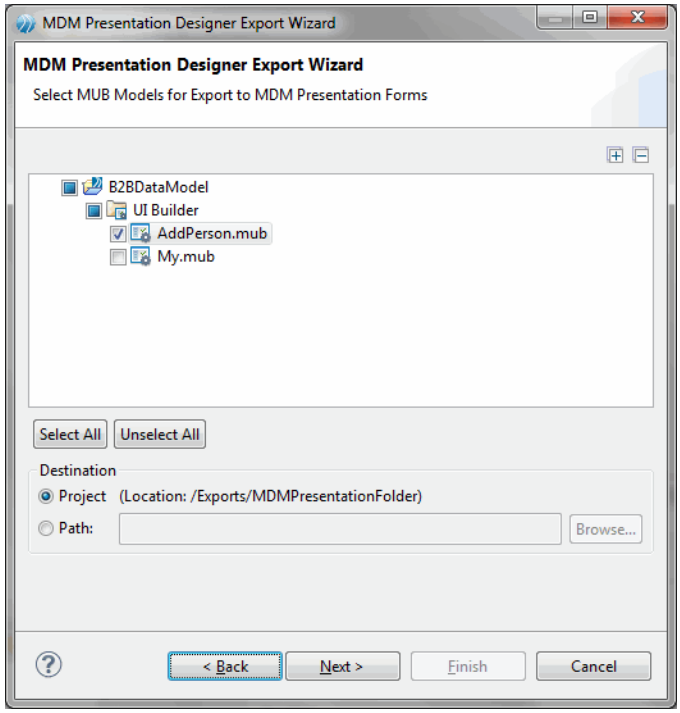
1. To export the artifacts, right click on **.mub**, and select **Export** > **Export**. For example, **AddPerson.mub**.



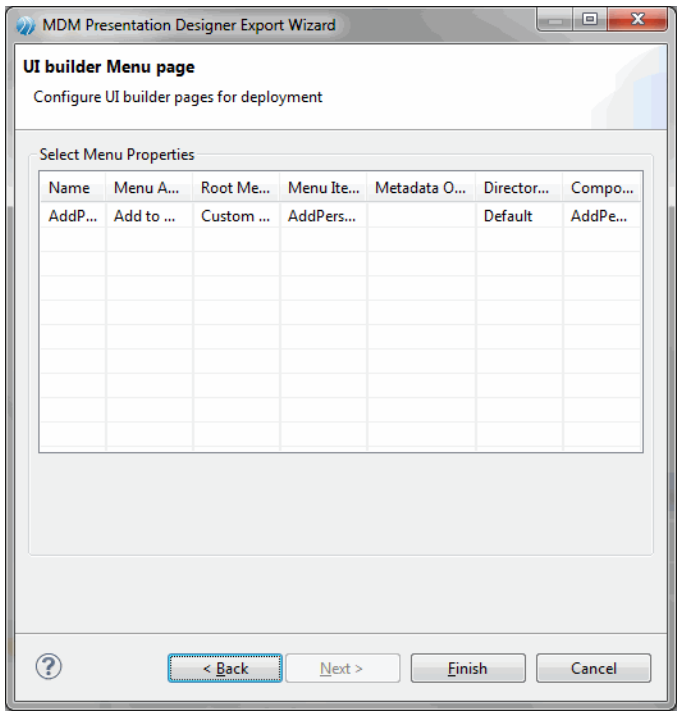
2. The **Export UI Artifacts** wizard displays the export destination page.



3. Select **MDM Presentation Forms** under **MDM Repository Designer** and click **Next**.
4. The **MDM Presentation Designer Export** wizard displays the **MUB** models.

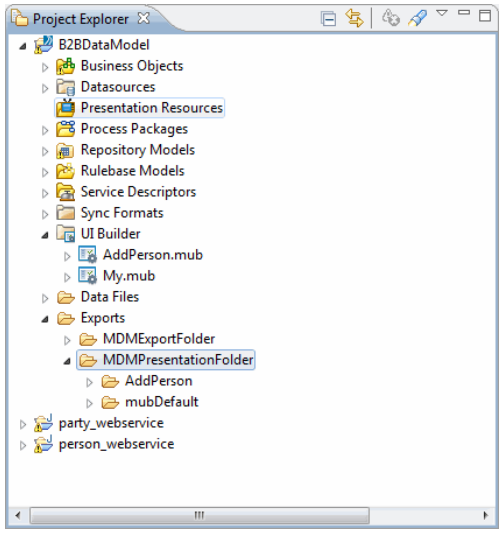


5. Select the appropriate **MUB** model, for example **AddPerson.mub**. Select the project location and click **Next**.
6. The UI Builder menu page is displayed.



7. You can configure the UI builder menu page. For more information on the configuring menu pages refer to Deploy MDM UI Page.
8. Click **Finish**. The project explorer show the MDM Presentation folder.

Result



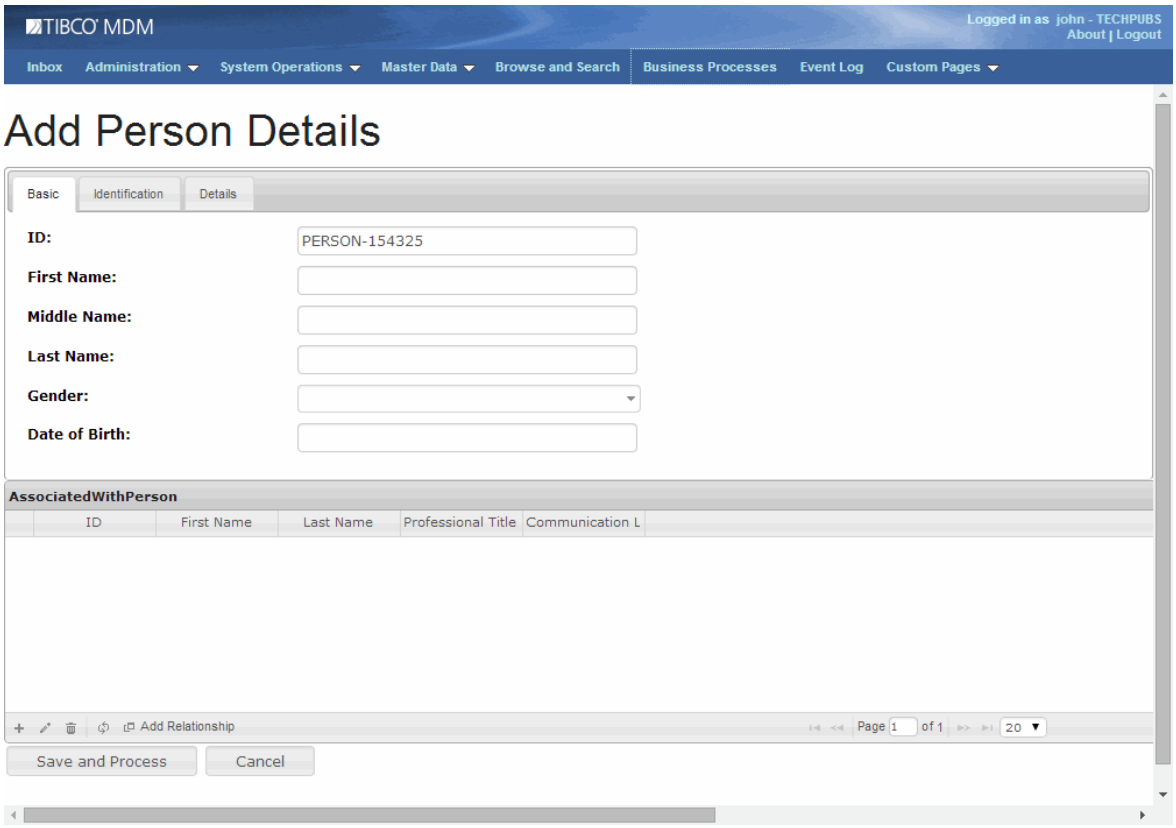
Accessing the UI Page on MDM Application

To view the newly created UI page, log into MDM. If you have deployed the new UI page by selecting the add menu option and the root menu as Custom Pages, then the newly created page appears in the Custom Pages menu.

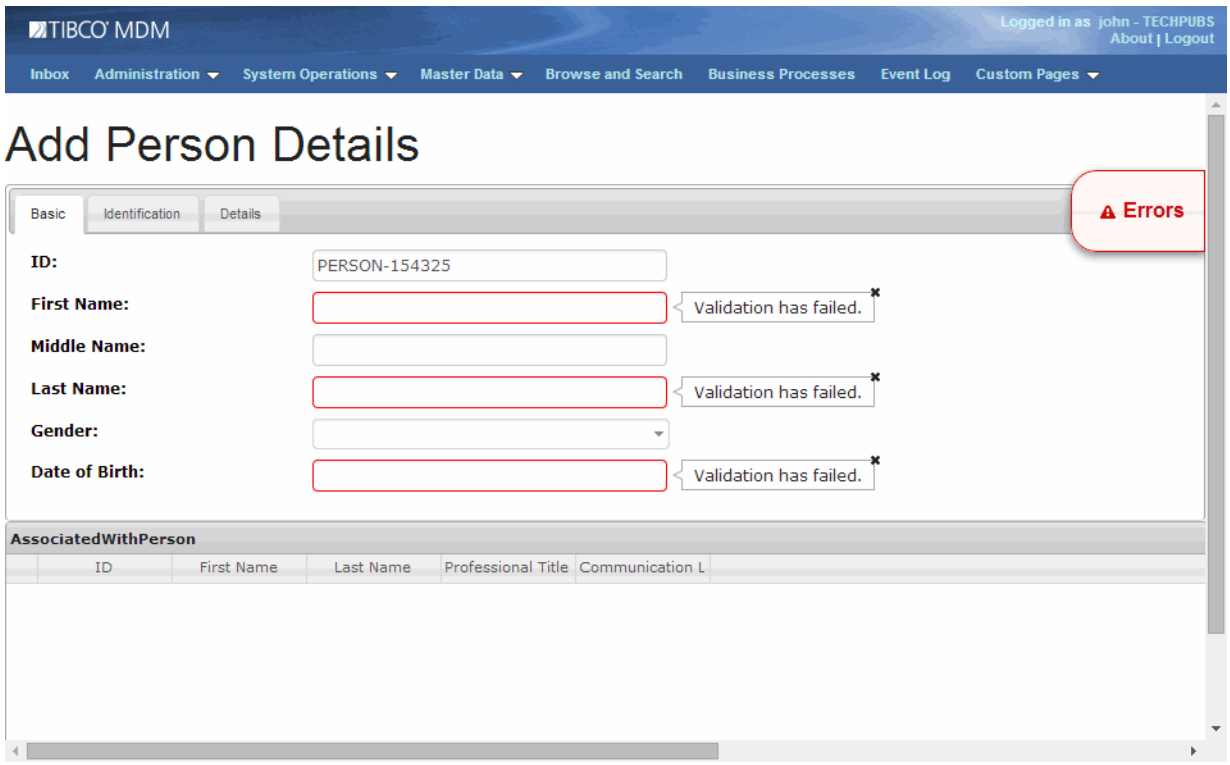
Similarly if you have selected a different root name then you will see that menu name instead of Custom Pages. In addition, if you have linked the UI page to a metadata operations Create Record, Modify Record and View Record then you must use the Browse and Search option and select the repository on which the new UI page is created. Then access the metadata operations.

Procedure

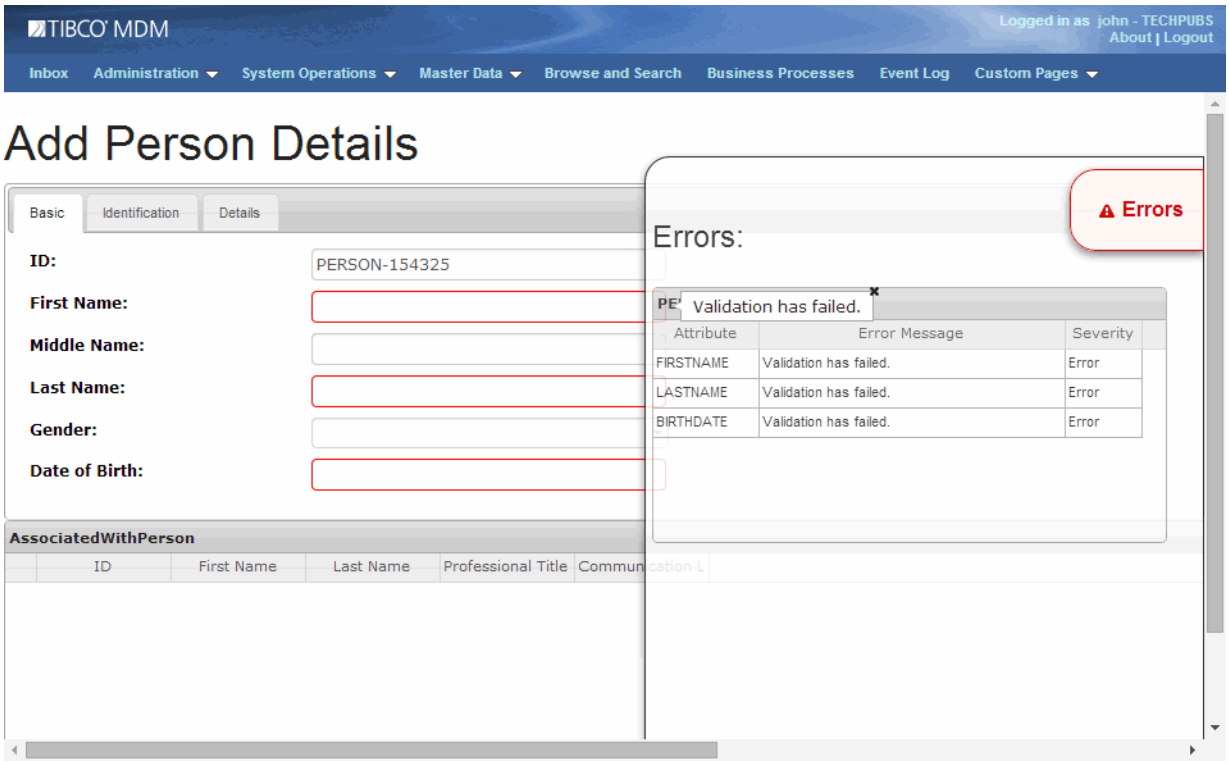
1. On the **Customs Pages** menu, click the new MDM UI page. For example, **AddPerson**.
2. The **Add Person Details** page is displayed.



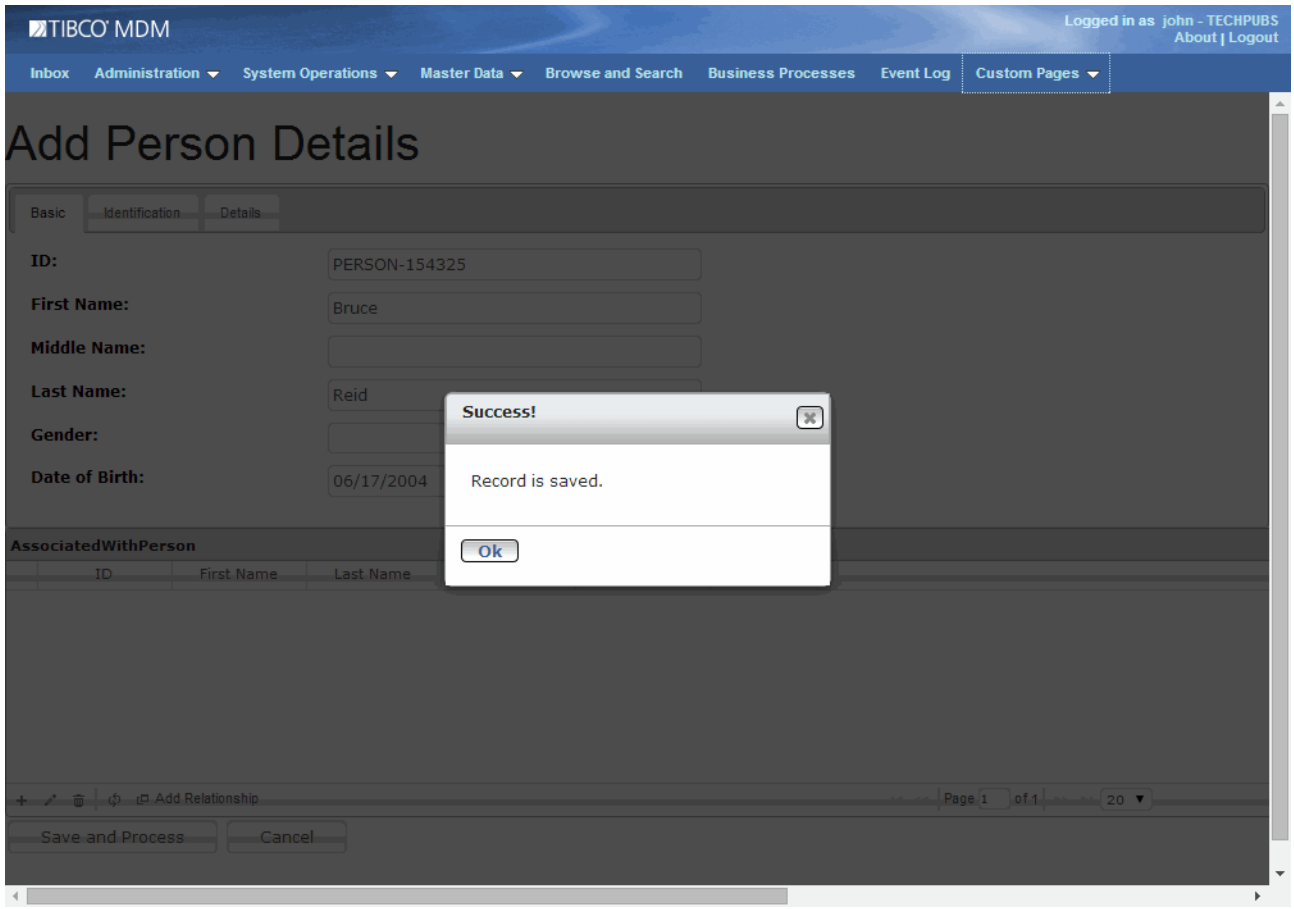
3. If you try to save the page without entering the required details, rulebase validations are invoked as rulebase was associated during the page creation.



4. The validation error is shown on the right hand side. To view the error click **Errors**. The validation details are displayed.



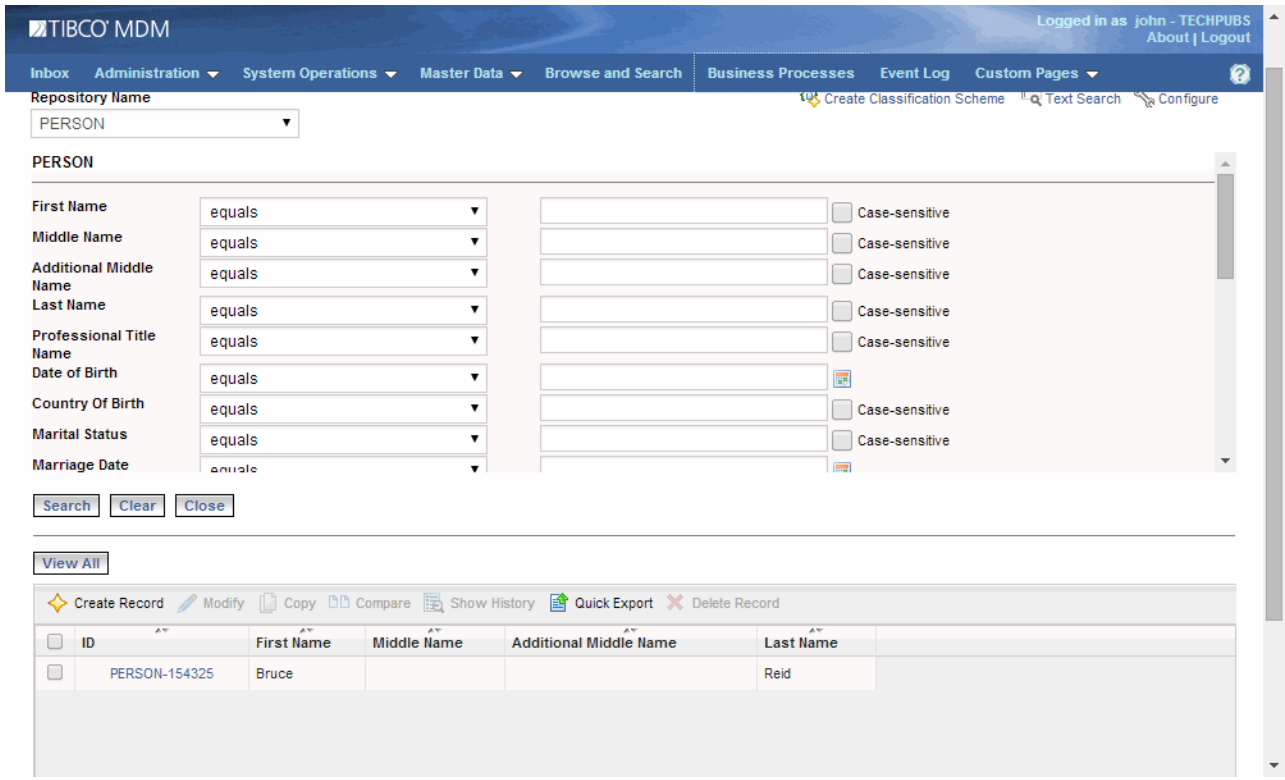
5. Enter the required details in the respective fields and click **Save and Process**.



6. The record save success message is displayed. Click **OK**.

7. The newly added person details in displayed in the **Browse and Search** page. For example, PERSON-154325 was newly added using the new MDM UI page.

Result



TIBCO MDM Analytics View Creation

This chapter introduces you on how to create views on database and on repository model in MDM Studio and deploy them on server side.

With business rapidly growing, the information about business needs are saved in a large database. The information is very important and must be analyzed more frequently to understand business trends.

The statistical analysis is done by extracting the data from multiple tables in the database and the results of their analysis are plotted in the form of charts and graphs. The analysis is available to various stakeholders in the organization. The stakeholders review and use it for their business growth. However, the analyzed data is very sensitive and is available to only few users in the organization. These users might hail from different user groups and user roles.

The analyzed data are processed, configured, and viewed using different types of analytics software. TIBCO Spotfire is one such software that is integrated with TIBCO MDM and seamlessly takes care of the statistical analysis.

A new wizard is built in MDM Studio to create views on database and on repository model for analysis. The views are deployed on the TIBCO MDM and Spotfire servers. The statistical analytics are accessed to their full set of capabilities by the analytics tools such as Spotfire Web Player.

Creating a View on Database for MDM Analytics

You can create a view on a database.

Prerequisites

Before creating views, create the connection profiles.

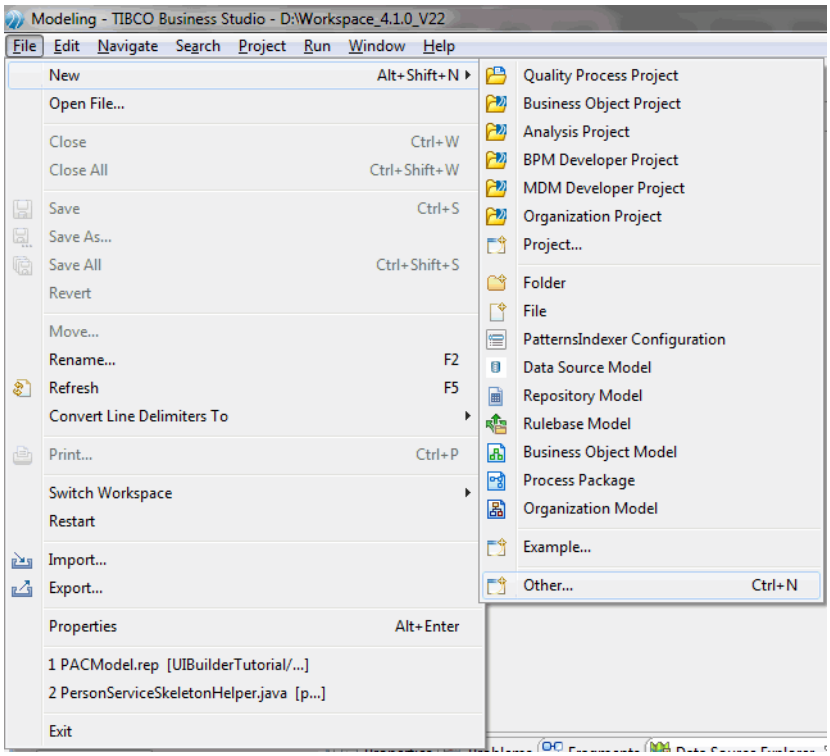
For more information on creating connection profiles, refer to *Appendix A - Data Source Explorer* in *Repository Designer User Guide*.

To create views, you must have administrator privileges on the database.

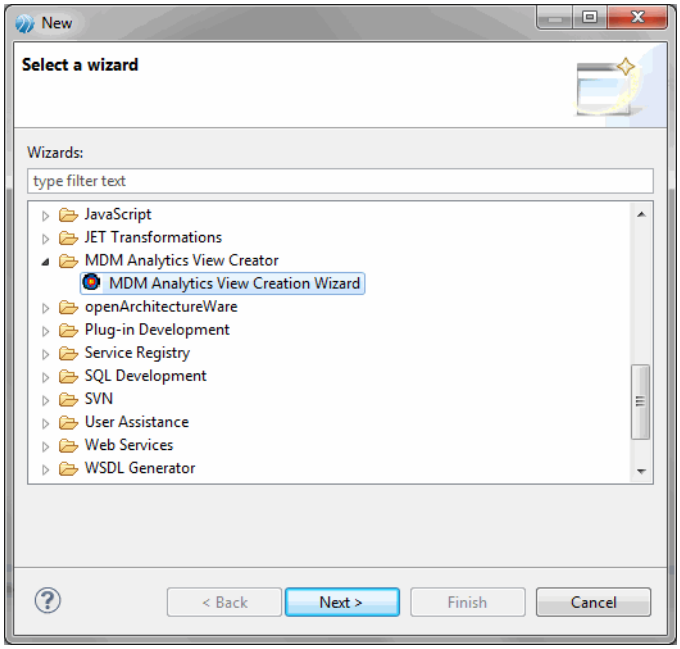
View creation is supported only for Oracle and SQLServer database. PostgreSQL is not supported.

Procedure

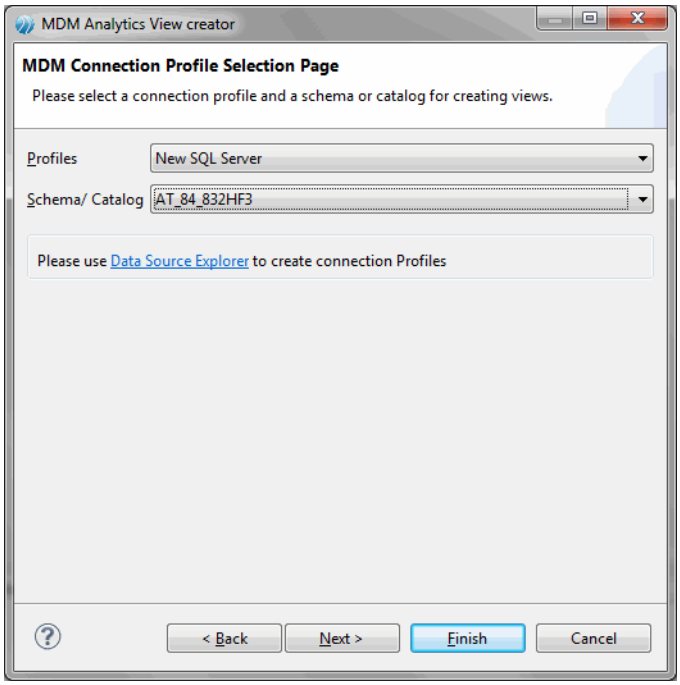
1. Navigate to **File** > **New** > **Other** .



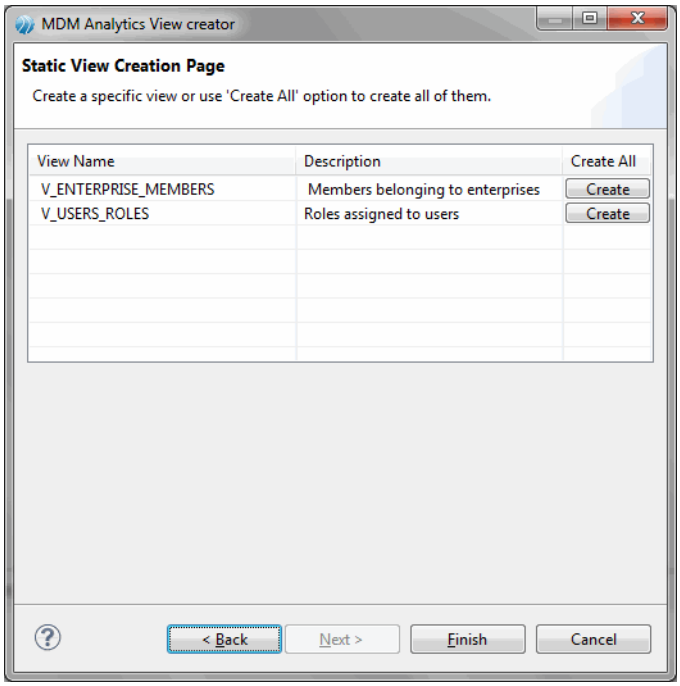
2. The **Select a wizard** screen is displayed.



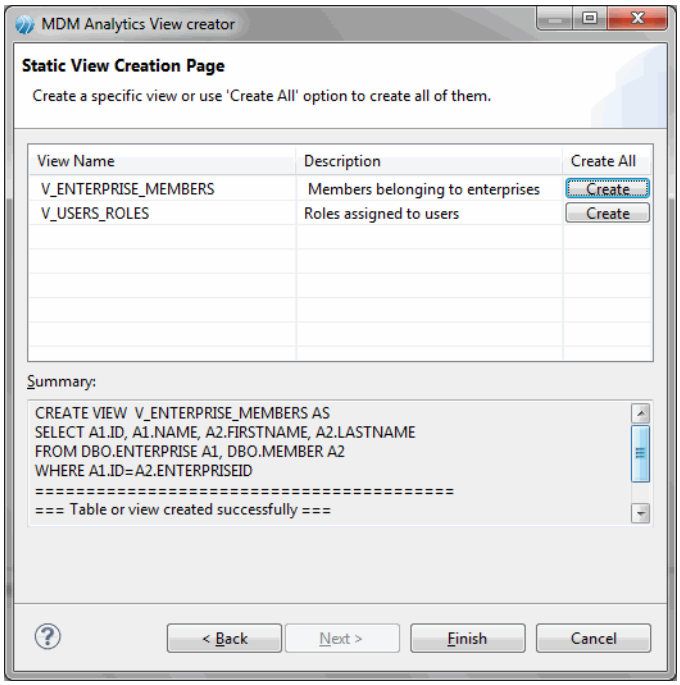
3. Select **MDM Analytics View Creation Wizard** from the **MDM Analytics View Creator** tree node.
4. Click **Next**. The wizard displays the **MDM Connection Profile Selection Page** screen.



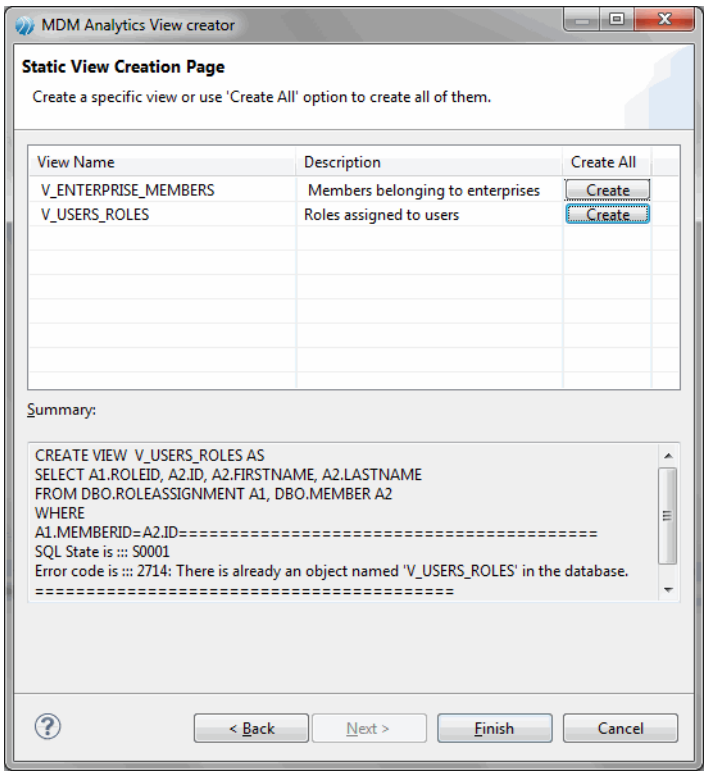
5. Select the connection profile from the **Profile** field and the schema or catalog from the **Schema/ Catalog** field. If you have not created the connection profile, click the **Data Source Explorer** link to create a new connection pool.
6. Click **Next**. The wizard displays the **Static View Creation Page** screen.



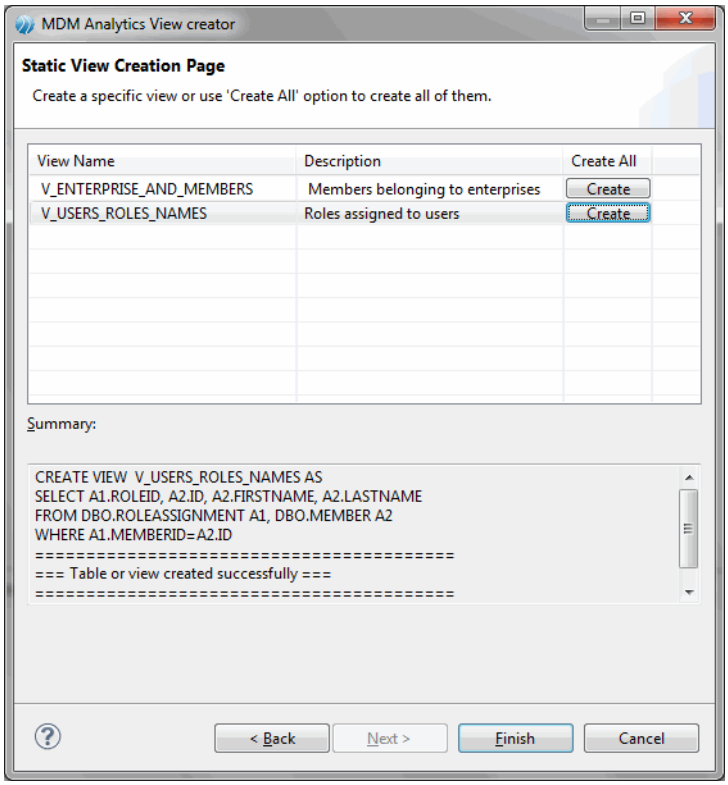
7. The pre created views for the select connection profile and schema is displayed. Select the view name and click **Create**.
8. The newly created view is displayed in the **Summary:** field. The summary field displays the SQL statements, which is executed for creating the view. On successful execution of the view, a success message is displayed at the end of the SQL statements.
9. To create another view from the available view names, select the view and click **Create**.



10. If the view is already created for the selected view name, an error is displayed in the **Summary:** field.



11. Double-click the view name and modify the view name. For example, `V_USERS_ROLES_NAMES`.
12. Click **Create**. The view is created with the modified view name.



13. If there are many views, click **Create All**. Views are simultaneously created for all the available view names.
14. Click **Finish**.
15. Click **Cancel** if you do not want to create views.

Analysis and Deploying in Spotfire

After creating views, log on to the Spotfire server and create the analysis for the views and deploy it on the Spotfire server.

The newly created analysis can be viewed using the TIBCO Spotfire Web Player.

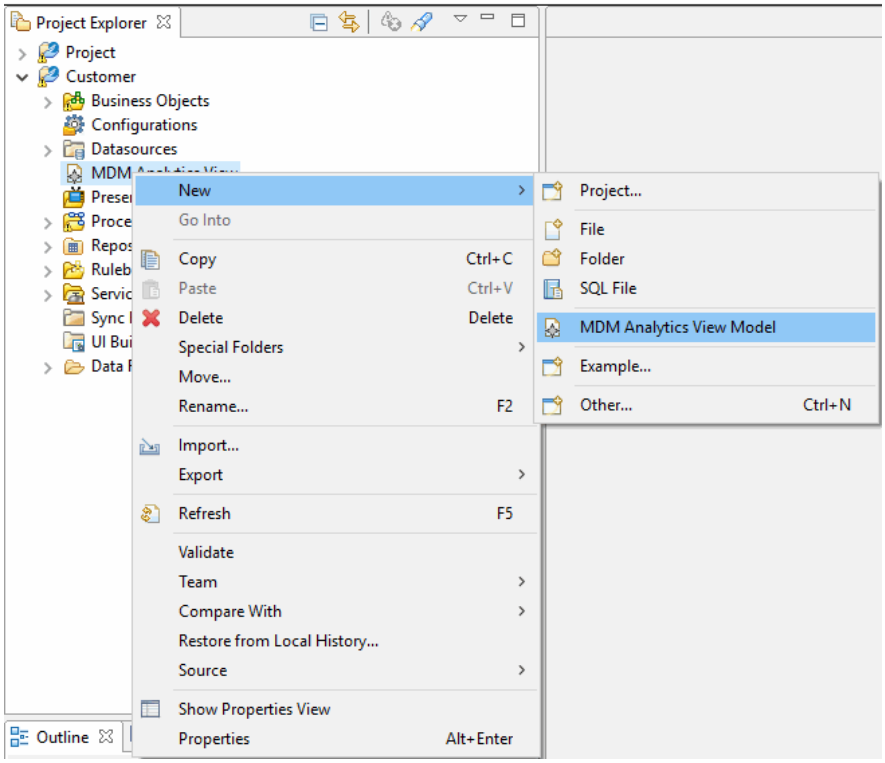
For more information on creating analysis and deploying on Spotfire server, refer to the *TIBCO Spotfire* documentation.

Creating a View on Repository Model for MDM Analytics

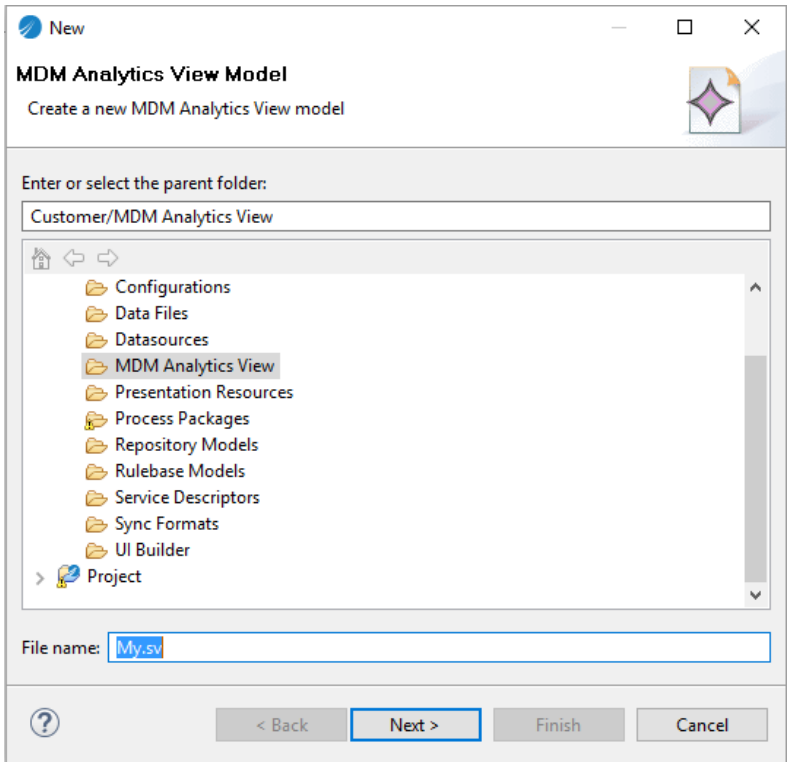
You can create a view on a repository based on the repository model selection.

Procedure

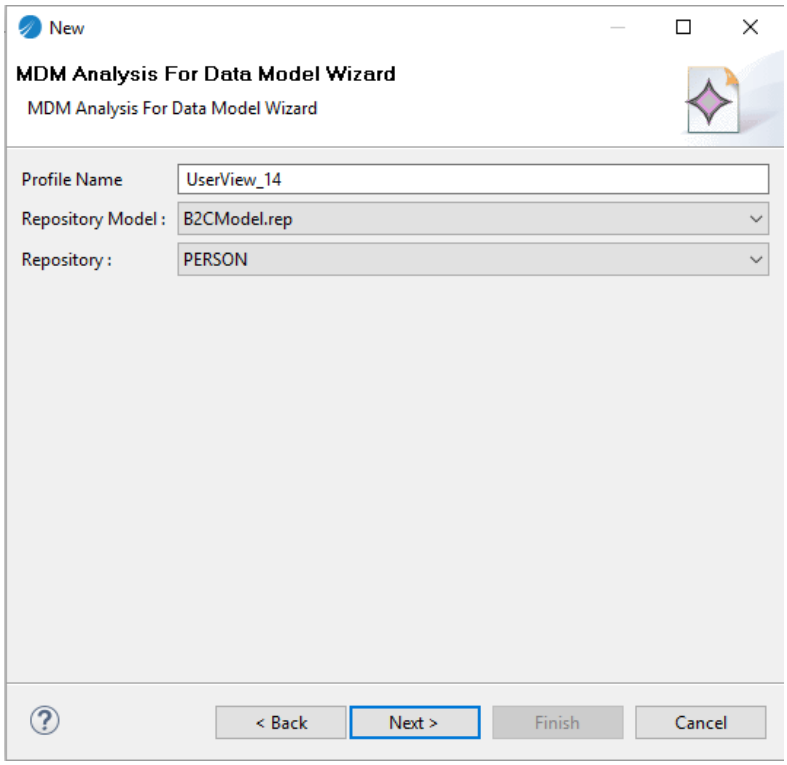
1. Right click on **MDM Analytics View** > **New** > **MDM Analytics View Model**.



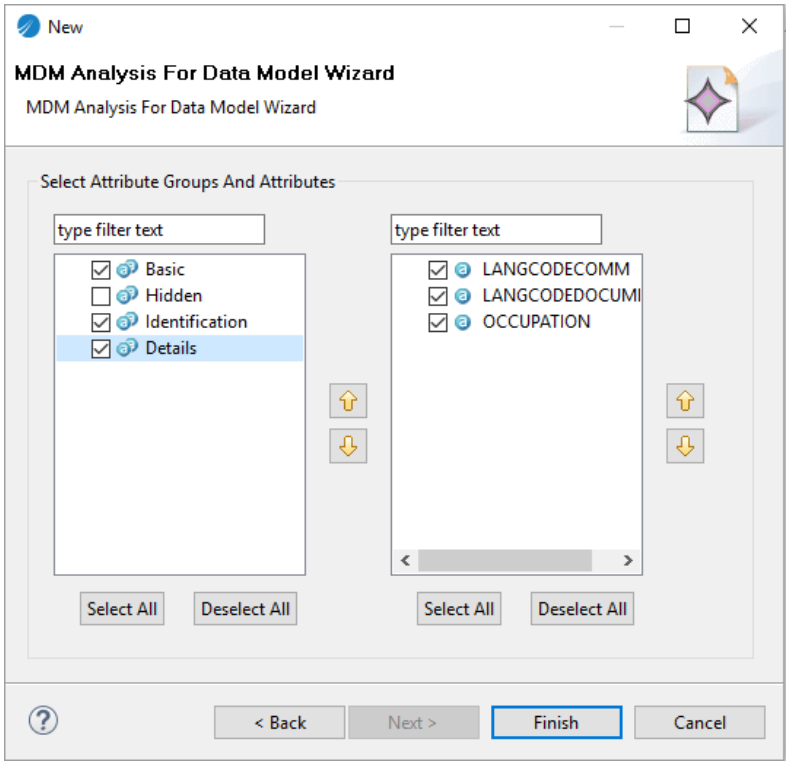
2. The **MDM Analytics View Model** wizard screen is displayed.



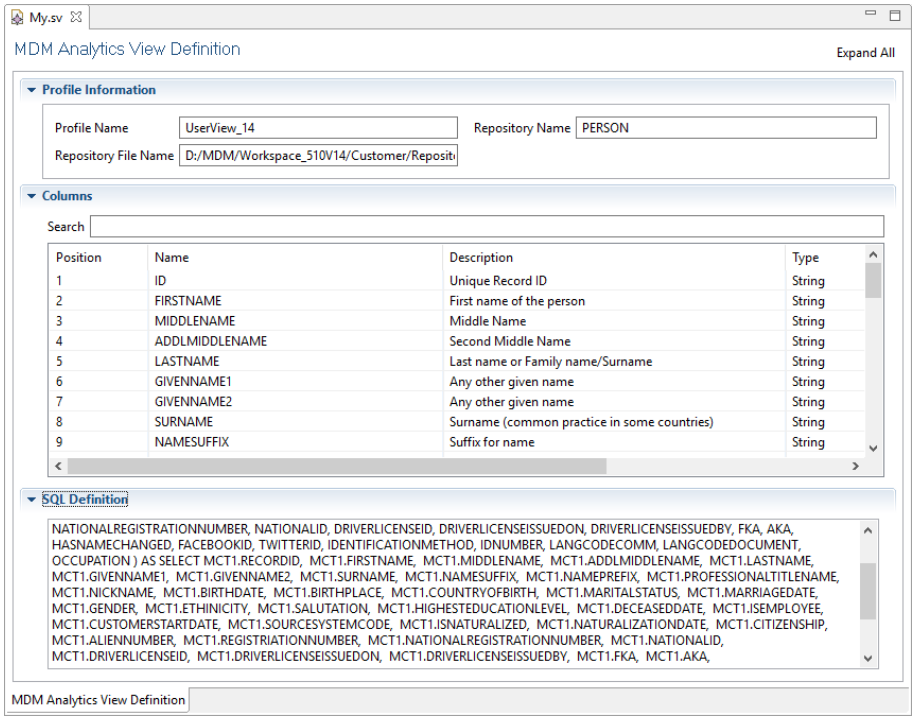
3. Select **MDM Analytics View** from the tree node.
4. Click **Next**. The wizard displays the **MDM Analysis For Data Model Wizard** screen.



5. Specify the profile name in the **Profile Name** field.
6. Select the repository model from the **Repository Model** drop-down list.
7. All the repositories for the selected repository is displayed. Select the appropriate repository from the **Repository** drop-down list.
8. Click **Next**. The wizard displays the **MDM Analysis For Data Model Wizard** screen.



9. All the attributes groups and attributes for each group is displayed. Select the attribute group and attributes which you want to in the views.
10. Click **Finish**.



The MDM Analysis View Definition displays the Profile Information, attributes as Columns, and SQL Defnition for the view.

Multivalue attributes, CSA attributes, Relationship and Relationship attributes are not supported for view creation. After a view is created, you can modify only the profile name and not the view definition. However, you can re-create the view with modification and with another view name or with same view name. If you are creating a view with same view name, ensure that you have undeployed the existing view and then re-deploy the modified view.

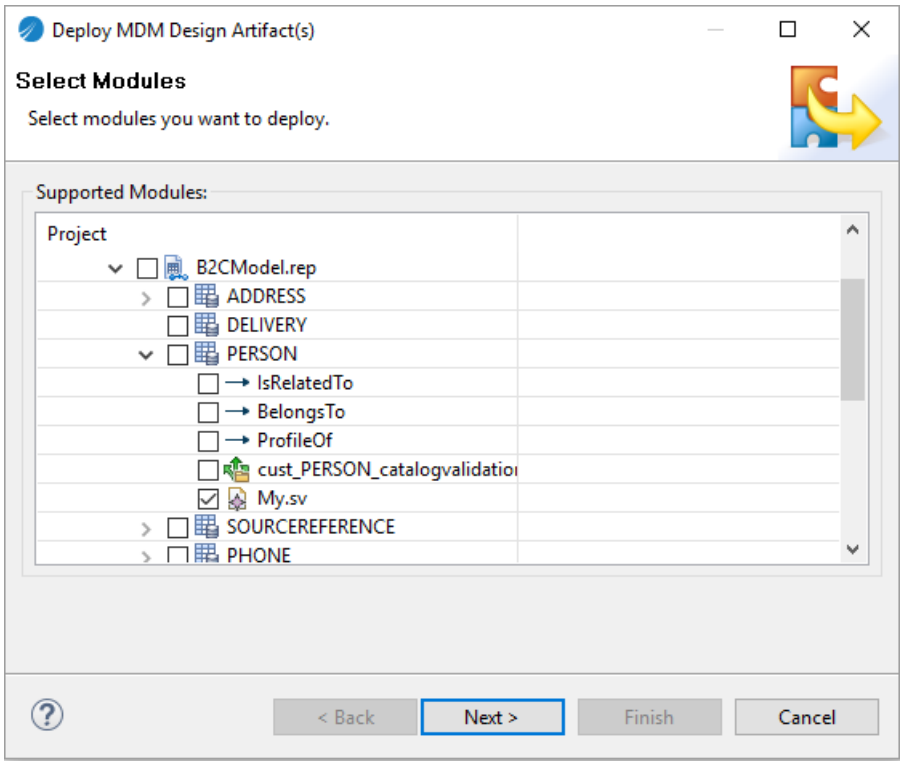
Deploying a View

You can deploy the newly created view similar to other artificats.

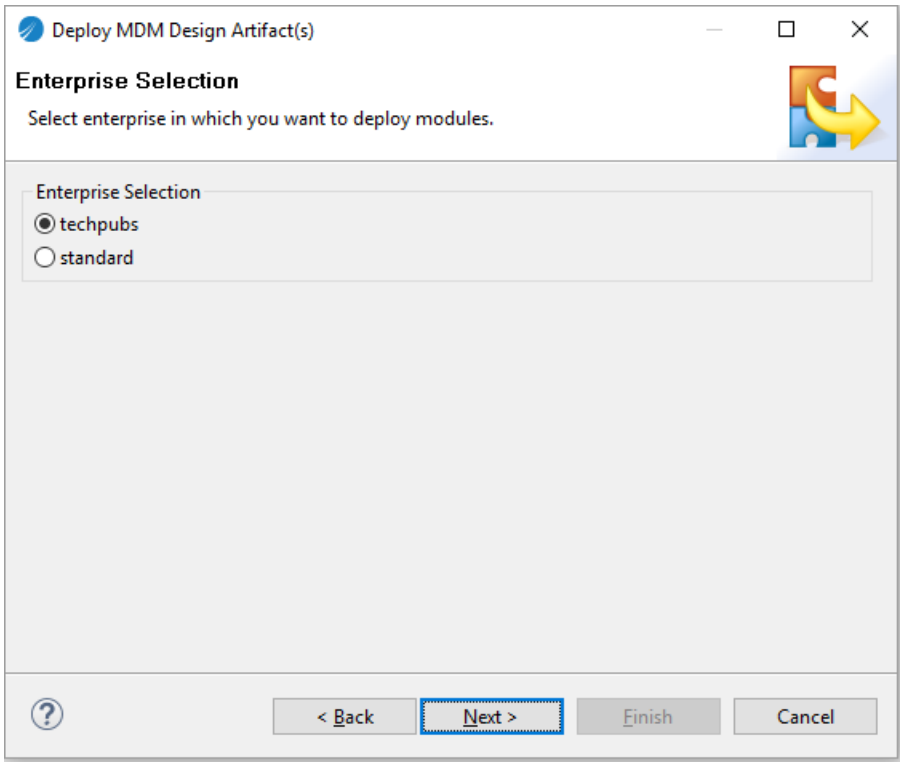
When a repository model (.rep file) is deployed, all the corresponding views are also deployed.

Procedure

1. In the Deployment pane, right click the <**MDM Server**> and select **Deploy Module.**
2. Select the checkbox corresponding to the MDM Analytics View which you want to deploy and click **Next**.

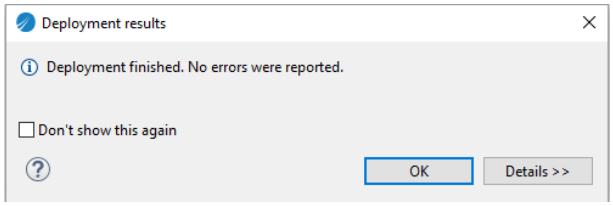


3. Select the enterprise to deploy the modules to (either the current enterprise or standard).



4. Click **Next**.
5. The validation result page is displayed with the validation messages. The errors and information messages are shown in separate sections. If there are validation errors, the **Finish** button is disabled. Click **Next**.
6. Click **Finish**.

7. The view is successfully deployed.



8. The views gets created in the database as **V_%ProfileName%**.

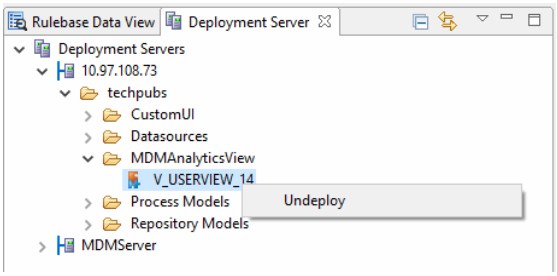
Undeploying the View

You can undepoy the MDM Analytics View from the server by following these steps.

Procedure

1. In the Deployment Server view, expand Deployment Server.
2. Then expand <MDM Server><EnterpriseName>**<MDMAnalyticsView > <ViewName>**. List of views are displayed with prefix V. For example, V_USERVIEW_14 .
3. Right click on the deployed **View Name** and select **Undeploy**.
4. You will get a message to confirm undeployment. Click **Yes** to undeploy.

Result



This undeploys the selected view.