

Generic User Interface Services

Solution



Chapter: Generic UI Services

Theme: Extended Value Selector



At the end of this Exercise, you are able to:

- Define Simple Data Types in the Java Dictionary.
- Create an Extended Value Selector.
- Change the values of the Value Selector programmatically.

4 Developing, Step-by-Step

Note: In this exercise you have to type in some lines of java code. For your convenience, you can cut and paste the code from the corresponding solution project.

But keep in mind: Some of the methods are automatically generated by Web Dynpro. For this reason the methods names you have to choose can differ from the method names of the solution.

4-1 Define a simple data type called *Cities_All* in the Java Dictionary.

In the Web Dynpro Explorer:

Expand the nodes *WD01_Basics_GenericUI / Dictionaries / Local Dictionary / Data Types*.

Open the context menu of *Simple Types*.

Choose *Create Simple Type*.

Name	Cities_All
Package	com.sap.training.wd.genericui.simple types

Value	Description
EUR_WDF	Walldorf
EUR_HD	Heidelberg
AMS_MXC_CTY	Mexico City
AMS_BN_AIR	Buenos Aires
EUR_LDN	London
EUR_PRS	Paris
EUR_ROM	Rome
EUR_MDR	Madrid
AMN_NY	New York
AMN_WSHG	Washington
AMN_BSTN	Boston

In the Data Type Editor, choose the tab *Enumeration* and create some entries as displayed in the graphic.

Use the following prefixes for the values:

EUR	Europe
AMS	South America
AMN	North America
OCE	Oceania
ASA	Asia
AFR	Africa

Save the new project data by choosing *Save All Metadata* from the toolbar.

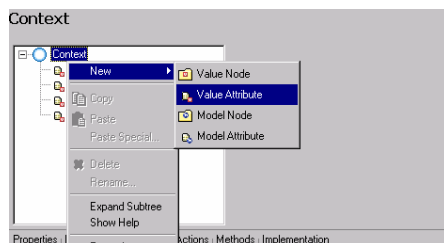
4-2 Declare the value attributes *ctx_cityFrom* and *ctx_cityTo* in the context of the view controller

In the Web Dynpro Explorer:

Expand the nodes *WD01_Basics_GenericUI / Web Dynpro / WD Components / Exc_GenUI / Views*.

Open the context menu of *StartView* and choose *Edit*.

In the View Editor, choose the tab *Context* and create a new value attribute

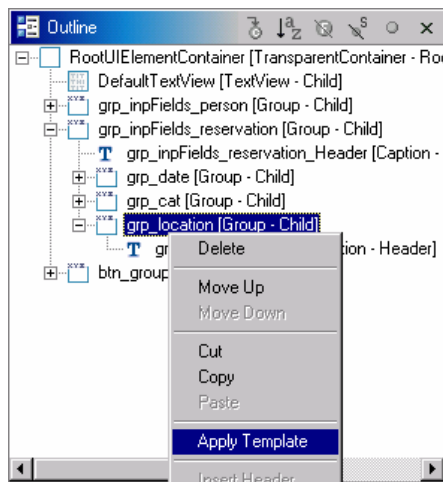


Define the following value attributes:

Name	Type
ctx_cityFrom	com.sap.training.wd. genericui.simpletypes. Cities_All
ctx_cityTo	com.sap.training.wd. henericui.simpletypes. Cities_All

Save the new project data by choosing *Save All Metadata* from the toolbar.

4-3 Define two *InputField* UI elements with corresponding *Labels* for pickup location and drop down location



In the View Editor, choose the *Layout* tab.

The Outline view will appear. Several UI elements are predefined for the view *StartView*.

Choose *RootUIElementContainer* / *grp_inpFields_reservation* / *grplocation*

From the context menu choose *Apply Template*

In the wizard that appears, choose:

Template	Form
Fields	ctx_cityFrom, ctx_cityTo

Save the new project data by choosing *Save All Metadata* from the toolbar.

4-4 Define the data binding between the input fields and the corresponding context values.

Define the following properties for the input field UI elements:

Property	Value
ctx_cityFrom.value	ctx_cityFrom
ctx_cityTo.value	ctx_cityTo

4-5 Create a Web Dynpro application

In the Web Dynpro Explorer, expand the node *WD01_Basics_GenericUI*
Expand the node *Web Dynpro* and open the context menu for *Applications*.

To open the wizard, choose *Create Application*.

In the wizard, enter:

Name	WD01_Basics_GenericUI
Package	com.sap.training.wd.genericui

Accept the other suggested values and choose *Next*.

Select *Use existing component* and choose *Next*.

Select

Web Dynpro Component	Exc_GenUI
Interface View	Exc_GenUI_WindowInterfaceView
Startup Plug	Default

and choose *Finish*.

5 Building, Deploying, and Running, Step-by-Step

5-1 Deploy and run the Web Dynpro application.

In the Web Dynpro Explorer:

Expand the nodes *WD01_Basics_GenericUI / Web Dynpro / Applications*.

Open the context menu for *WD01_Basics_GenericUI*

To deploy and run the application, choose *Deploy new Archive and Run*

Successful result:

In this exercise, you have developed an extended value selector and filled it with static data (City names for pickup and drop down locations). The static data has been defined in the Java Dictionary at design time.

6 Optional:
Fill the extended value selector dynamically at runtime.

6-1 Define two new data types in Java Dictionary.

6-1-1 Define a data type called *Cities_Reg*.

Name	Cities_Reg
Package	com.sap.training.wd.genericui.simple types

Leave the enumeration blank. This type will be filled dynamically and contains the cities of the selected region.

6-1-2 Define a data type called *Region*.

This type is needed for the `RadioButtonGroup` to select a region.

Name	Region
Package	com.sap.training.wd.genericui.simple types

In the Data Type Editor, choose the *Enumeration* tab and create some entries:

Value	Description
OCE	Oceania
ASA	Asia
AFR	Africa
EUR	Europe
AMB	North America
AMS	South America

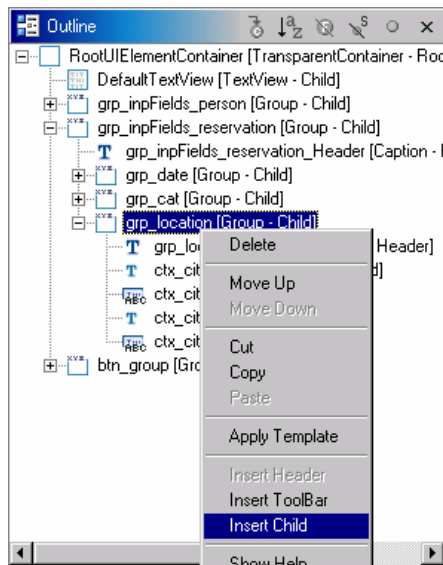
Note: Use the same abbreviations as key as you have used as prefix for the cities data type (see exercise 4-1).

Save the new project data by choosing *Save All Metadata* from the toolbar.

6-2 Define a new value attribute *ctx_region* in the context of the view controller (type *Region*)

Take a look at exercise 4-2.

6-3 Define a new radio button group UI element *RadioButtonGroupByKey* in the pre-defined group UI element *grp_location*.



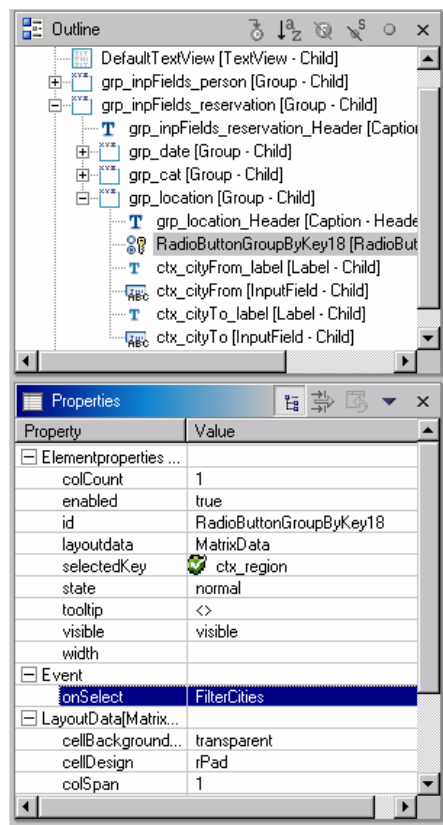
In the View Editor, choose the *Layout* tab.

The Outline view will appear.
Several UI elements are predefined for *StartView*.

Choose *RootUIElementContainer* / *grp_inpFields_reservation* / *grp_location*.

From the context menu choose *Insert Child*

Choose *RadioButtonGroupByKey*



Change the properties of the UI element:

colCount	2
selectedKey	ctx_region

Save the new project data by choosing *Save All Metadata* from the toolbar.

- 6-4 Define a new action `FilterCities`. Bind this action to the property *onSelect* of the `RadioButtonGroup`.

In the view Editor of `StartView`, choose the tab *Actions*.

Define the action:

name	FilterCities
event handler	use default
fire plug	none

Change the properties of the *RadioButtonGroupByKey* UI element:

onSelect	FilterCities
----------	--------------

Save the new project data by choosing *Save All Metadata* from the toolbar

- 6-5 Declare a helper method *fillValueSet* to fill the value selector.

6-5-1 Declare the helper method *fillValueSet* (Parameter Boolean initialize).

In the view Editor of the view *StartView*, choose the tab *Methods* and add the method *public void fillValueSet(Boolean initialize)*.

6-5-2 Implement the Method *onActionFilterCities*.

In the view Editor of *StartView*, choose the tab *Implementation*.
Choose the generated method *onActionFilterCities()* and add the following source code:

```
public void onActionFilterCities(  
    com.sap.tc.webdynpro.progmodel.api.IWDCustomEvent wdEvent )  
{  
    //@@begin onActionFilterCities(ServerEvent)  
  
    fillValueSet( false );  
  
    //@@end  
}
```

6-5-3 Define a new value attribute *ctx_cityAll* in the context of the view controller (type *Region*).

Take a look at exercise 4-2.

6-5-4 Implement the Method *fillValueSet*.

Choose the tab *Implementation*, navigate to the generated method and implement the method body.

Note: You can copy and paste the code from the solution project.

```
public void fillValueSet( boolean initialize )
{
    //@@begin fillValueSet()

    // the next code lines show, how you get the reference to the
    // value sets of the context fields
    IModifiableSimpleValueSet cityAllValueSet =
        wdContext.getNodeInfo()
            .getAttribute("ctx_cityAll")
            .getModifiableSimpleType()
            .getSVServices()
            .getModifiableSimpleValueSet();
    IModifiableSimpleValueSet cityFromValueSet =
        wdContext.getNodeInfo()
            .getAttribute("ctx_cityFrom")
            .getModifiableSimpleType()
            .getSVServices()
            .getModifiableSimpleValueSet();
    IModifiableSimpleValueSet cityToValueSet =
        wdContext.getNodeInfo()
            .getAttribute("ctx_cityTo")
            .getModifiableSimpleType()
            .getSVServices()
            .getModifiableSimpleValueSet();

    cityFromValueSet.clear();
    cityToValueSet.clear();

    // now you get the size of the cities_all enumeration and
    // you get the selected region from the RadioButtonUIGroup:
    int size = cityAllValueSet.size();
    String region =
        wdContext.currentContextElement().getCtx_region();

    // in this section you iterate through the cities_all
    // enumeration and extract the key and the value for each
    // entry.
    for (int idx = 0; idx < size; idx++)
    {
        Object key = cityAllValueSet.getKey(idx);
        String keyStr = key.toString();
        String text = cityAllValueSet.getText(idx);
        if (initialize)
        {
            // at start of application (initialize==true) you store all
            // entries from the cities_all enumeration to the
            // cities_From / cities_To enumeration without selection.
            cityFromValueSet.put(keyStr,text);
            cityToValueSet.put(keyStr,text);
        }
        else
    }
```



```

        {
            if(keyStr.startsWith(region)) {
                // this section is processed at runtime, when the user
                // selected an entry from the RadioButtonGroup.
                // If region key and city-prefix match, the city is stored
                // in the cities_From / cities_To enumeration
                cityFromValueSet.put(keyStr,text);
                cityToValueSet.put(keyStr,text);
            }
        }
    }
}

// the next lines fill the input fields with default values
try
{
    wdContext.currentContextElement().setCtx_cityFrom(
        cityFromValueSet.getKey(0).toString());
    wdContext.currentContextElement().setCtx_cityTo(
        cityToValueSet.getKey(0).toString());
}
catch (Exception exc) {
    wdContext.currentContextElement().setCtx_cityFrom( "");
    wdContext.currentContextElement().setCtx_cityTo( "");
}
}
//@@end
}

```

Save the new project data by choosing *Save All Metadata* from the toolbar.

6-6 Update the method *wdDoInit()*

```

public void wdDoInit() {
    //@@begin wdDoInit()
    initialize();
    fillValueSet( true );
    //@@end
}

```

Save the new project data by choosing *Save All Metadata* from the toolbar.

6-7 Build, deploy and run the application

Take a look at exercise 5-1.