

Generic User Interface Services Solution



Chapter: Generic UI Services

Theme: Simple Value Selector



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

- Define Simple Data Types in the Java Dictionary.
- Create a Simple Value Selector.

4 Developing, Step-by-Step

4-1 Define a Simple Type data type called *VehicleType* in the Java Dictionary.

In the Web Dynpro Explorer:

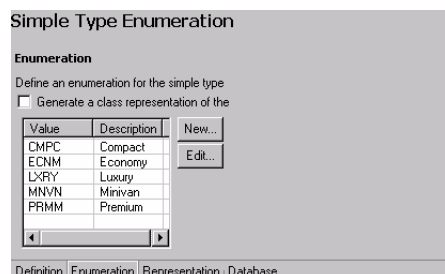
Expand the nodes *WD01_Basics_GenericUI / Dictionaries / Local Dictionary / Data Types*.

Open the context menu of *Simple Types*.

Choose *Create Simple Type*.

Enter:

Name	<i>VehicleType</i>
Package	<i>com.sap.training.wd.genericui.simple types</i>



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

Value	Description
CMPC	Compact
ECNM	Economy
LXRY	Luxury
MNVN	Minivan
PRMM	Premium

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

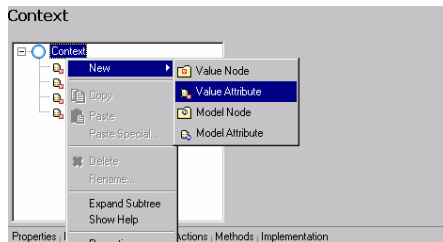
4-2 Declare a value attribute *ctx_vhclType* of the type *VehicleType* in the context of the view controller.

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

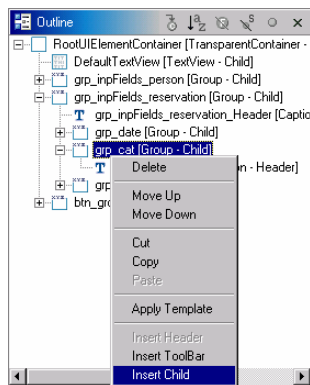


Name	<i>ctx_vhclType</i>
Type	<i>com.sap.training.wd.genericui.simpletypes.VehicleType</i>

Note: By default, the context types are of Type string. Choose the context attribute and update the data type in the *Properties* view.

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

4-3 Define a *DropDownByKey* UI element with a corresponding *Label*.



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 / grpVhclType*.

From the context menu choose *Insert Child*

Insert a *DropDownByKey* UI element:

Name	<i>DropDownByKey_vhclType</i>
selectedKeyType	<i>ctx_vhclType</i>

Insert a *Label* UI element:

Name	<i>lbl_vhclType</i>
lableFor	<i>DropDownByKey_vhclType</i>
Text	<i>Vehicle Type</i>

Move the Label in front of the DropDownByKey UI element.

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

4-4 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*.

Successful result:

The generated Web Dynpro application object completes your project structure.
You can now deploy your project and start your Web Dynpro application.
WD01_Basics_GenericUI enables you to address the Web application as a whole, when you launch this complete application in the next step

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:

WD01_Basics_Sol09 - SAP Web Dynpro - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address http://p110825:50000/webdynpro/dispatcher/local/WD01

Welcome to Web Dynpro!

Person data

Firstname *
Lastname *

Reservation data

Date begin 15.07.2004 Date end 15.07.2004

Category Premium

Save Res

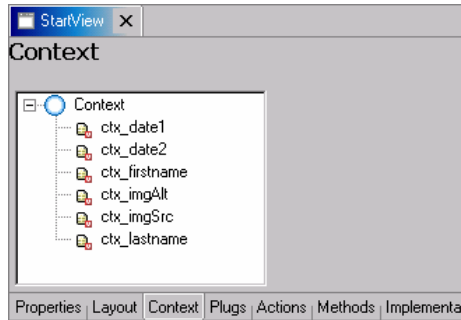
Compact
Economy
Luxury
Minivan
Premium

Done Local intranet

As a result of this exercise, the view contains a dropdown list box, which itself contains the constants for car categories and uses the simple value selector for displaying the texts. The set of valid values in the input field is restricted to key entries of the predefined set of constants.

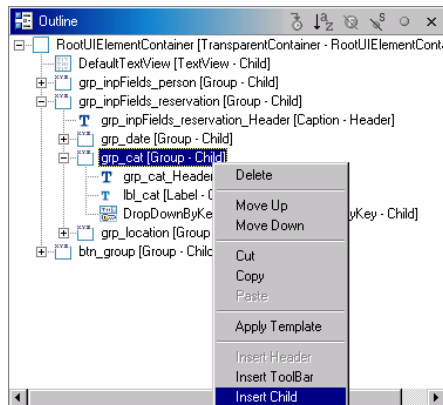
6 Optional 1:
Add an image and change its visibility depending on the selected value of the input field *category*.

6-1 Declare two new value attributes *ctx_imgAlt* and *ctx_imgSrc* in the context of the view controller of *StartView*.



For detailed information about the necessary steps take a look at exercise 4-2.

6-2 Add an Image UI element to the category group *grpVhclType* and bind the context attributes *ctx_imgAlt* and *ctx_imgSrc* to it.



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* / *grpVhclType*.

From the context menu choose *Insert Child*

Insert a *Image* UI element with the following properties:

id	<i>img_vhcl</i>
alt	<i>ctx_imgAlt</i>
source	<i>ctx_imgSrc</i>
visible	<i>none</i>
colspan	2

6-3 Define a new variable *image* to refer to the image UI element and to change its visibility programmatically

In the view Editor of *StartView*, choose the Implementation tab.
Navigate to the end of the class definition, where you can define additional attributes :

```
/*
 * The following code section can be used for any Java code that
 * is not to be visible to other controllers/views or that
 * contains constructs currently not supported directly by Web
 * Dynpro (such as inner classes or member variables etc.). </p>
 *
 * Note: The content of this section is in no way
 * managed/controlled by the Web Dynpro Designtime or the Web
 * Dynpro Runtime.
 */
//@@begin others
static IWDImage image;
//@@end
```

Define a static variable of type *IWDImage* as shown above. Let the SAP NetWeaver Developer Studio import the class definition.

6-4 Initialize the *image* variable in method *wdDoModifyView()*.

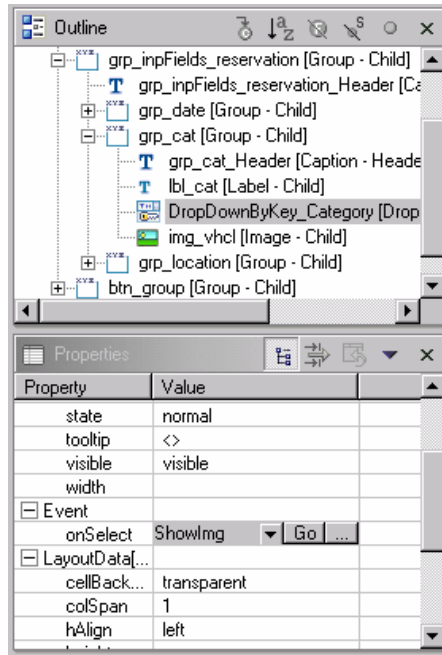
```
public static void wdDoModifyView(
    IPrivateStartView wdThis,
    IPrivateStartView.IContextNode wdContext,
    com.sap.tc.webdynpro.progmodel.api.IWDView view,
    boolean firstTime)
{
    //@@begin wdDoModifyView
    if(firstTime)
    {
        image = (IWDImage)view.getElement("img_vhcl");
    }
    //@@end
}
```

6-5 Define a new action *ShowImg* and assign this action to the *DropDownByKey* UI element.

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

Define the action:

name	ShowImg()
event handler	use default
fire plug	None



In the View Editor, choose the tab *Layout*

The Outline view appears.

Choose *RootUIElementContainer* / *grp_inpFields_reservation* / *grpVhclType*.

Choose the *DropDownByKey* UI element.

In the Properties view choose:

onSelect	ShowImg
----------	---------

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

6-6 Assign the action to the *DropDownByKey* UI element

In the view Editor of the view *StartView*, choose the tab *Implementation*.

Choose the generated method *onActionShowImg()*

```
public void onActionShowImg(...) {
    //@@begin onActionShowImg(ServerEvent)
    String vehicleType =
        wdThis.wdContext.currentContextElement().getCtx_vhclType();

    wdThis.wdGetContext().currentContextElement().setCtx_imgSrc(
        vehicleType+".gif");
    wdThis.wdGetContext().currentContextElement().setCtx_imgAlt(
        vehicleType+".gif");

    image.setVisible(WDVisibility.VISIBLE);
    //@@end
}
```

The first JAVA statement returns the key of the selected vehicle type from the selector field.

The second JAVA statement sets the name of the image to be displayed as value of the context attribute.

The third JAVA statement sets the name of the image to be displayed as the property *alt* of the context attribute.

Finally, the last JAVA statement changes the visibility of the image.

Note: The images must have the same names as the vehicle type keys of the selector field. Make sure, that the images are stored in the corresponding exercise mimes directory.

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.

Successful result:



As a result of this exercise, an image will be visible depending on the selected value of the simple value selector.