Using the Property Editor UI Pattern
© Copyright 2005 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP AG
Neurottstraße 16
69190 Walldorf
Germany
T +49/18 05/34 34 24
F +49/18 05/34 34 20
www.sap.com

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies (“SAP Group”) for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

Disclaimer
Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP’s Support Services and may not be modified or altered in any way.
Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example Text</td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.</td>
</tr>
<tr>
<td>Example text</td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td>Example text</td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td>Example text</td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td>&lt;Example text&gt;</td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>

Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚠️</td>
<td>Caution</td>
</tr>
<tr>
<td>🔍</td>
<td>Example</td>
</tr>
<tr>
<td>💡</td>
<td>Note</td>
</tr>
<tr>
<td>💡💡</td>
<td>Recommendation</td>
</tr>
<tr>
<td>💡💡💡</td>
<td>Syntax</td>
</tr>
</tbody>
</table>
About This Document

The property UI pattern can be used to edit and view attributes of entity services in various customizable ways. There are different layouts and formats possible and even macros can be written to change the data representation.

Prerequisites

Before you start with this tutorial you should have installed the following Software:

- SAP Web Application Server Java 7.0
- SAP NetWeaver Developer Studio 7.0

Applicable Releases

This tutorial is compatible with the following releases:

- SAP NetWeaver ’04s
- SAP Composite Application Framework (CAF) 7.0

Disclaimer

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Creating a Development Component Project

1) Open the NetWeaver Development Studio. Switch to the CAF Perspective by selecting “Composite Application Services” as perspective from the menu Window -> Open Perspective -> Other...
   An icon will appear in the left pane.

2) Create a new Development Component Project by navigating File -> New -> Project. The New Project window will open.
3) Select *Development Component* -> *Development Components Project* and click *Next*. 
4) Select Local Development -> MyComponent and click Next.

This tutorial does not discuss development using NetWeaver NWDI. If you want to use the NWDI, please refer to http://help.sap.com for developing components using remote configurations.

5) In the next screen, enter/select the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>carpool (use only small letters)</td>
</tr>
<tr>
<td>Caption</td>
<td>Car Pool</td>
</tr>
<tr>
<td>Domain</td>
<td>SAP xApps</td>
</tr>
<tr>
<td>Type</td>
<td>Composite Application Services</td>
</tr>
</tbody>
</table>
Click on *Next* twice.

A screen will display the list of projects that will be created as a part of this Development Component Project.
Click on Finish.

Creating DDIC Types

1) Open the carpool project in NetWeaver Development Studio. Switch to the Dictionary Perspective by clicking on in the Toolbar. Alternatively choose Window/Open Perspective/Other/Dictionary.

2) In the Dictionary Explorer expand the tree carpool -> Dictionaries -> Local Dictionary -> Data Types -> Simple Types. Select Create Simple Type from the context menu of Simple Types.
3) In the New Simple Type window, enter the following.
   Simple Type Name: salutation
   Simple Type package: com.sap.carpool.customtypes

   ![New Simple Type window]

   Click on Finish button. Click Yes if prompted with the Could not check package name alert (this means that the IDE could not check if you have reserved the name space for your package already).
   The Editor for simple type salutation will be opened.

4) Choose
   Built-in type string
   Max length 30.
5) Switch to the **Representation** tab. Enter **Salutation** in the fields **Field Label**, **Column Label** and **Quick Info**.
6) In the Dictionary Explorer window, select Add to Public Part from the context menu of simple type salutation.
7) In the Public Part Editor select types_compilation and click on OK. Repeat the same steps (6 and 7) for types_assembly. This is done so that the type that you created is available in the data type selection list for New Attribute creation.

8) Save the Meta data.
We have now got a new custom data type of type “String” available. This data type is currently just a simple String. Later on we will use it as an enumeration, but this is not designed on IDE level but in the runtime configuration on the J2EE engine. We will configure that in the paragraph named “Defining Enumerations”.

Creating an Entity Service

1) Expand the project carpool in the Service Explorer. Right click on the node Entity Services and then click New.

Enter the Service Name as Employee.
Click on *Finish*. This creates an Entity Service. The *Employee* Entity Editor will be displayed with several Tab Strips.
2) Switch to the Attributes tab. In the Attribute Tree Structure, right click on the root node Employee and then click Create Attribute.
3) Enter the Attribute Name as “id” and Attribute Description as “Employee ID”. Please check the checkbox “Key”. Afterwards click on the Browse button to select the Data Type.
Select the data type `com.sap.caf.core.long` and click OK.

Click on *Finish*. A new attribute *id* will be added to the Attributes tree in the right pane.
If you select the attribute *id*, the properties of the attribute will be displayed in the *Properties* tab at the bottom of the screen.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>True</td>
</tr>
<tr>
<td>Object Type</td>
<td>Field attribute</td>
</tr>
<tr>
<td>Cardinality</td>
<td>1</td>
</tr>
<tr>
<td>Data Type</td>
<td>com.sap.zst.ze.long</td>
</tr>
<tr>
<td>DB Field</td>
<td>ID</td>
</tr>
<tr>
<td>Description</td>
<td>Employee ID</td>
</tr>
<tr>
<td>Language Dependent</td>
<td>False</td>
</tr>
<tr>
<td>Mandatory</td>
<td>True</td>
</tr>
<tr>
<td>Name</td>
<td>id</td>
</tr>
</tbody>
</table>

Among these some of the properties are editable. E.g. *Cardinality, DB Field, Description, Language Dependent, Mandatory* and *Name*. 
4) Similarly add more attributes to Employee of the data types mentioned in the following list.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Key</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>com.sap.caf.core.long</td>
<td>ID</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>firstName</td>
<td>com.sap.caf.core.shortText</td>
<td>First Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastName</td>
<td>com.sap.caf.core.shortText</td>
<td>Last Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>com.sap.caf.core.longText</td>
<td>Email</td>
<td></td>
<td></td>
</tr>
<tr>
<td>salutation</td>
<td>com.sap.carpool.customtypes.salutation</td>
<td>Salutation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The newly created custom data types will appear in the data type window.

5) Switch to the Operations tab.
Click on the Add button.

Enter the following.
Operation Name:  findByLastName
Description:  Find by Last Name
Select lastName in the attribute list and click Finish.
The new operation will appear in the Operations list.
6) Please add also another Findby method called “getAll” without any search parameters! This method is very convenient at least during testing.

7) Switch to the Permissions tab and uncheck both the check boxes.

8) Save the Meta Data by selecting Save All in the File menu.

This completes the Creation of Entity Service for Employee. Generate all project code, build everything and deploy it to the J2EE engine.

Defining Enumerations

This section explains the procedure for creating enumerations for the attributes. Enumeration is a list of permissible values for the attribute. This appears as a drop down list of values in the UI pattern. Enumeration can be defined only for Custom Data Types. In this example we will define enumeration for the Custom Data Type salutation.
1) Launch the CAF Runtime Configuration page
http://<was_host>:<was_port>/
webdynpro/dispatcher/sap.com/caf~UI~configbrowser/Config

Navigate through Administrative Tools -> Custom Enumeration Type Editor
Alternatively, you can launch the URL http://<was_host>:<was_port>/
webdynpro/dispatcher/sap.com/caf~UI~typeeditor/TypeEditor
2) Enter the Custom data type name `com.sap.carpool.customtypes.salutation` in the input field beside Type drop down, and click on the Add button. Now click on the Add button at the bottom of the screen to add the following values:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr.</td>
<td>Mr.</td>
</tr>
<tr>
<td>Mrs.</td>
<td>Mrs.</td>
</tr>
</tbody>
</table>

3) Click on the Save button to save the enumeration.
4) Test the creation of Employee entities with the service browser.

**Introduction to the Property Editor Pattern Demo**

In section 8.1 a simple Property Editor Pattern is configured that is embedded into an Object Editor Pattern in section 8.2.

Finally, in section 8.3, the Property Editor Wrapper Pattern is explained.

You can also use a Property Editor Wrapper Pattern instead of a Property Editor Pattern right from the beginning. The only disadvantage of this approach would be that you would not be able to embed this pattern into the Object Editor Pattern in section 8.2.

Furthermore the missing Property Editor Pattern could not be reused in other How-tos.
The Step By Step Solution

Configuring the Property Editor Pattern


2. Choose “Property Editor”

3. Click on “New Configuration”

4. Enter the configuration Name “carpool_employee”.

5. Choose “Data Source Obtained by Query”

6. Click on “Select” to select the query.
7. Search in the Service Modules list for the service called “sap.com/carpool/EmployeeService”

8. Select the Query “findByLastName”

9. Select the Aspect “Employee”

10. Click on OK.

11. Next we create an external parameter. Enter the name “last_name” in the new parameter field.

12. Press “Create”.

13. Bind this parameter by choosing it in the “Parameter Binding” table for the Parameter “lastName” in the column “Binding Path”.

This now means that an additional parameter has been added to the external interface of the UI pattern. If
this interface were
called via a
Freestyle
WebDynro UI, this
parameter would
now be offered and
could be filled as
an input for the
query. (See also
the How-to on
creating a
FreeStyle
WebDynpro UI for
a composite
application).

14. Switch to the
“Logical Fields”
Tab. Here we can
create additional
fields that can be
computed out of
(several) attributes.

15. Add the Name of a
new attribute to the
field “Name”. Call it
“createdAtAndBy”. It
will combine and
format the
information on who
created this entity
service instance
and when.

16. Press the “Add”
button.

17. Next you create a
macro for the new
field. The Result
Type of the macro
will be of type
“String”. Click on “Edit
Macros”.

18. Add the following string to the Macro Editor:

```
"@format("Created at: {0,date, medium}, by: {1}", [createdAt], [createdBy])"
```

19. Press the button "Check Macro Syntax". No syntax errors should be found.

20. Please also check the tabs below:
The tab "Functions" gives you an overview on the available functions and how they can be used.
The tab "Fields" lists the fields of the entity service used that you can use in your macro.
Furthermore there are the tab "UI Constants" that may be useful if you deal with UI elements in your macro and the tab "Types" containing Types you can cast to.

21. Press OK.
22. Repeat the preceding steps for another new additional field called “lastChangedAtAnd By”. For this field please use the following macro coding: 
"@format("Last changed at: \{0,date, medium\}, by: \{1\}", [lastChangedAt], [lastChangedBy])"

23. Please switch to the tab “Physical Fields and Design”. Here we will now configure the general window layout.

24. Enter a header name “Employee Data”. Please leave the other window properties as they are.

25. In the field group “Aspect Metadata” enter the first field called “Employee Id”

26. Press “Add”

27. Choose in the column “Mapping” the attribute “id[long]” as referred attribute of the entity service.

28. In the field “Field Editor” please
choose “InputField”

29. Select in the field group “Editor Properties” for the field “Alignment” the value “FORCEDLEFT”.

30. Please leave the remaining fields as they are.

31. Please add the following fields in the same way:

<table>
<thead>
<tr>
<th>Label Name</th>
<th>Mapping</th>
<th>Field Editor</th>
<th>Label Properties</th>
<th>Editor Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Key</td>
<td>key [id]</td>
<td>TextView</td>
<td>unchanged</td>
<td>unchanged</td>
</tr>
<tr>
<td>Created</td>
<td>createdAtAndBy [string]</td>
<td>TextView</td>
<td>Visibility: BLANK</td>
<td>unchanged</td>
</tr>
<tr>
<td>Last Changed</td>
<td>lastChangedAtAndBy [string]</td>
<td>TextView</td>
<td>Visibility: BLANK</td>
<td>unchanged</td>
</tr>
<tr>
<td>Salutation</td>
<td>salutation[..]</td>
<td>DropDownByKey</td>
<td>unchanged</td>
<td>unchanged</td>
</tr>
<tr>
<td>First Name</td>
<td>firstName [shortText]</td>
<td>InputField</td>
<td>unchanged</td>
<td>unchanged</td>
</tr>
<tr>
<td>Last Name</td>
<td>lastName [shortText]</td>
<td>InputField</td>
<td>unchanged</td>
<td>unchanged</td>
</tr>
</tbody>
</table>
32. If you now click on the “Preview” button you should be shown something similar to this:

33. Afterwards close this preview window and click “save and return” in the configuration.
Optional: Embedding Property Editor in Object Editor

34. In the preview of the property editor you can only check the configuration in read-only mode. If you would like to check your configuration in edit mode you could for example embed it in an Object Editor Configuration. To do this please enter the ConfigBrowser and choose the pattern “ObjectEditor”. Click on **New Configuration**.

35. Choose the service: "sap.com/carpool/EmployeeService".

36. Choose the header aspect “Employee”

37. Click on **Retrieve Metadata** and provide a name for the $submit$ action, e.g. *Save*.

38. Select the flag “Object has Own Component Implementation”

39. Click on **...** next to the field “Configuration Name”.
40. Choose in the instance browser in the tree the component type “sap.com/caf~UI~ptn~propedit” - “PropertyEditorComp”

41. Choose the configuration name “carpool_employee”

42. Press “OK”

43. Click on the “Preview” button. Now you should be able to create a new instance for your entity service

Optional: Configuring a Property Editor Wrapper Pattern

A Property Editor Wrapper Pattern is very similar to a Property Editor Pattern. The only difference is that it has an additional tab called “Actions”. Here you can add a “save” button.

If you want to experiment with the Property Editor Wrapper Pattern then please proceed with this pattern as you did in section 8.1.

Afterwards you can carry out the following additional steps.
44. In the configuration of the Property Editor Wrapper pattern please switch to the panel “Actions”.

45. Enter a new Button called “Save” and press “Add”.

46. Enter the tooltip “Save Employee Data”

47. Choose the Context Action “Save”

48. Enter the Command ID “SAVE”.

49. If you now click on “Preview” you can also change the data.
www.sap.com/netweaver