Creating Complex Data Types and Java Data Dictionary Types
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example Text</strong></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><strong>Example text</strong></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></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><strong>Example text</strong></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><strong>Example text</strong></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><strong>&lt;Example text&gt;</strong></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><strong>EXAMPLE TEXT</strong></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>
## Contents

Requirements and Dependencies .................................................................................. 2  
Applicable Releases ..................................................................................................... 2  
Disclaimer .................................................................................................................... 2  
Creating DDIC Types .................................................................................................... 2  
Creating Complex Attributes ...................................................................................... 8  
Defining Enumerations ................................................................................................ 12
Requirements and Dependencies

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

This tutorial is based on the following How-to Guides:

- Create a local entity service with maintenance UI

Additionally you should have configured CAF as described in the How-to Guide “Installation and Configuration Guide”.

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 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.

![Select a Public Part](image1.png)

![Select a Public Part](image2.png)

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 paragraph 4 “Defining Enumerations”.
Creating Complex Attributes

1) Switch to *Composite application Services* (CAS) Perspective. Open the Entity Service *TravelLocation*.

Create a new Attribute with the following details.

*Attribute name:* address

*Attribute Description:* Address

Select *Complex Attribute* and click on *Finish*.

**Note:** You do not specify any data type here.

The attribute address will appear in the attribute tree with a special icon to denote that it is a complex attribute.
2) Select *Create Sub Attribute* from the context menu of the *address* attribute.
3) Create sub attributes with the following specifications

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>Address</td>
<td>com.sap.caf.core.longText</td>
<td></td>
</tr>
<tr>
<td>city</td>
<td>City</td>
<td>com.sap.caf.core.shortText</td>
<td></td>
</tr>
<tr>
<td>zip</td>
<td>Zip Code</td>
<td>com.sap.caf.core.shortText</td>
<td></td>
</tr>
<tr>
<td>country</td>
<td>Country</td>
<td>com.sap.caf.core.shortText</td>
<td></td>
</tr>
</tbody>
</table>

4) Save the Metadata.
5) Now also add a new attribute of the following specification to the *Employee* entity.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>salutation</td>
<td>Salutation</td>
<td>com.sap.carpool.customtypes.salutation</td>
<td></td>
</tr>
</tbody>
</table>

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

6) Save the Metadata, Generate the project code, Build and deploy the project.

7) Add the newly added attributes to the Object Editor UI pattern and test them. Please find the description of the necessary steps in chapter “Maintenance UI configuration using Pattern UI” in the How-to document “Create local entity with Maintenance UI”
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.

Now test the entity service Employee in the Object Editor pattern. The field salutation will appear as a drop down with two values – Mr. and Mrs.
www.sap.com/netweaver