Creating Data Types

SAP Composite Application Framework

Document Version 1.00 – November 2005

THE BEST-RUN BUSINESSES RUN SAP

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

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

Type Style	Represents
Example Text	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.
Example text	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	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.
`Example text`	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.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, F2 or `ENTER`.

Icons

Icon	Meaning
⚠	Caution
💬	Example
💡	Note
🧭	Recommendation
syn	Syntax

Contents

Introduction

About This Document

This tutorial is a part of a series that describes how to implement a composite application using SAP Composite Application Framework (CAF) capabilities.

The tutorial describes how to create and maintain additional data types that are required for the implementation of the composite application.

Scenario Overview

The CAF Service and UI Layer series is based on a scenario, in which a company offers educational services to employees as participants and by employees as trainers. Educational services are offered at educational events. The company has offices at different locations both in the country and abroad, and employees need to travel between these locations. For cost-saving purposes the company decides that employees traveling on the same day between the same locations will share a car. This behavior shall be encouraged by certain financial benefits for the usage of carpools. The car can be either the personal company car of an employee or a car from the car pool of the company.

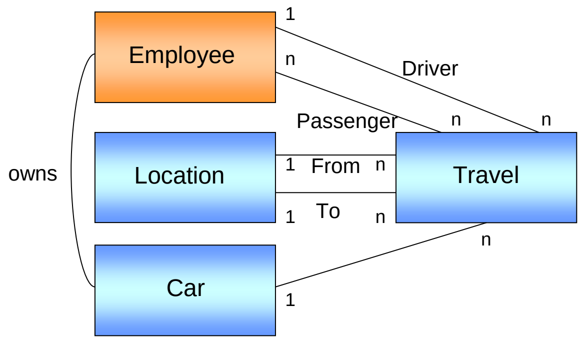
To help people find other travellers and set up car pools, the company needs a new application. This series demonstrates how to develop a prototype for such application using local data sources. The real-life productive system should work with data available in the HR (employees, travel planning, and compensations), MM (equipment management) and CRM (customer data) systems.

Data Model

The implementation of the application requires the use of the following business objects:

- Employee – the object is defined with a set of parameters, such as ID, first and last name, e-mail, and so on
- Location – this object represents travel destination; it is defined by fields, such as ID, address, city, and ZIP
- Car – the object represents either a pool car or a company car; it is assigned to an individual employee
- Travel – the object represents a trip between two locations; a travel is always one-way

The relations between them are shown in the figure below.



Data Model

An employee may create multiple travel instances. A travel always starts at a certain location and ends at a different one. Therefore a round trip to a location is maintained as two separate travel instances.

Each travel is assigned to a single car. However, a car may be assigned to multiple travels limited by the number of available seats.

Functions

The basic functions of the application are:

- Master data maintenance for the following entities:
  - Employee
  - Location
  - Car
- Search for planned travels at a certain arrival date to a certain location
- Schedule travels based on the search results or assign a new car from the pool. If a pool car is not available, a message is displayed.

Prerequisites

The following table describes the prerequisites for running this tutorial.

Software	The tutorial is compliant with: • Sneak Preview SAP NetWeaver 04 – Web Application Server 6.40 Java. You can download the sneak preview at **`http://sdn.sap.com`**
Documents	Before you start with this tutorial, see: • Configuring Your System for CAF Service and UI Layer Development • Creating a Local Entity with Maintenance UI

Recommended Readings

After you have completed this tutorial, you can familiarize yourself with the following documents:

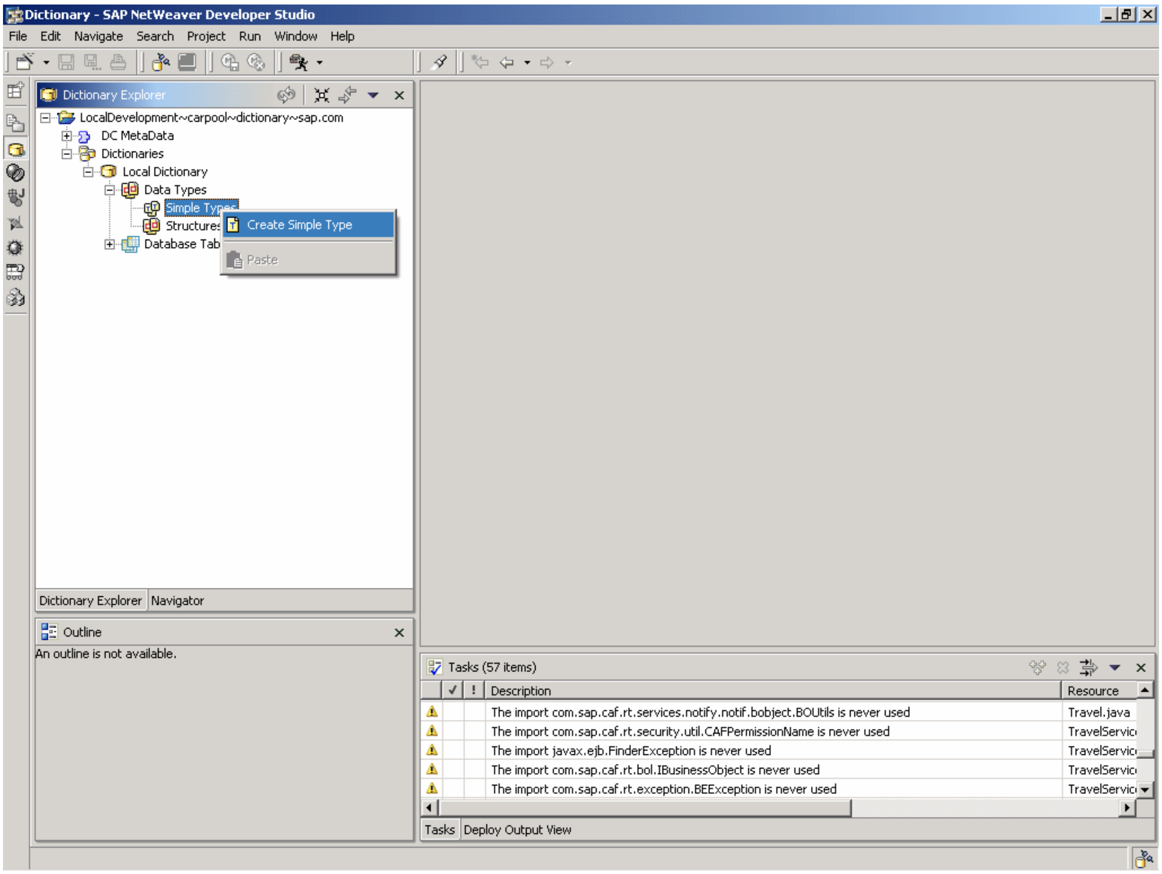
- Creating an Application Service

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.

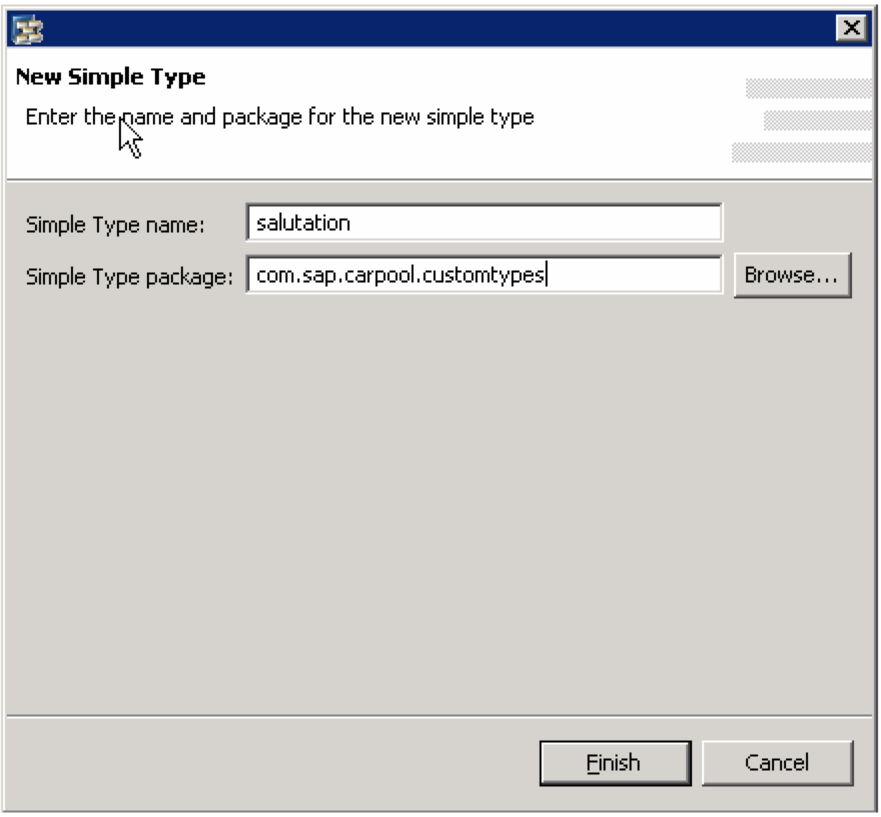
Creating DDIC Types

1. Open the carpool project in the SAP NetWeaver Development Studio. Switch to the Dictionary Perspective.
2. Browse to *carpool* → *Dictionaries* → *Local Dictionary* → *Data Types* → *Simple Types*. Choose *Create Simple Type* from the *Simple Types* context menu.



✓	!	Description	Resource
		The import com.sap.caf.rt.services.notify.notif.bobject.BOUtils is never used	Travel.java
		The import com.sap.caf.rt.security.util.CAFPermissionName is never used	TravelServic
		The import javax.ejb.FinderException is never used	TravelServic
		The import com.sap.caf.rt.bol.IBusinessObject is never used	TravelServic
		The import com.sap.caf.rt.exception.BEException is never used	TravelServic

3. In the *New Simple Type* window, enter the following data.
   - Simple Type Name: salutation
   - Simple Type package: com.sap.carpool.customtypes



New Simple Type

Enter the name and package for the new simple type

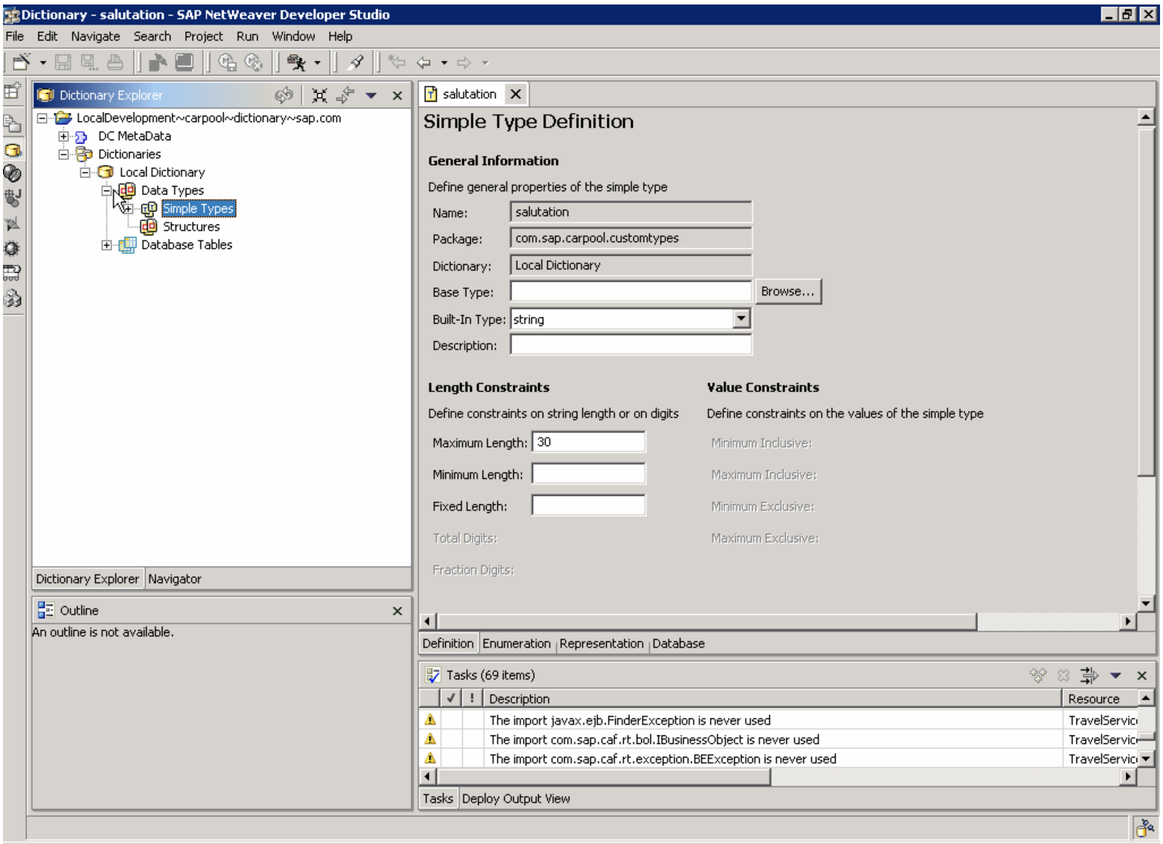
Simple Type name: salutation

Simple Type package: com.sap.carpool.customtypes

Choose *Finish*. Choose *Yes* if prompted with a *Could not check package name* alert.

The simple type editor opens.

4. Select String for the built-in type, and enter 30 for Max length.



Simple Type Definition

General Information

Define general properties of the simple type

Name: salutation

Package: com.sap.carpool.customtypes

Dictionary: Local Dictionary

Base Type: Browse...

Built-In Type: string

Description:

Length Constraints

Define constraints on string length or on digits

Maximum Length: 30

Minimum Length:

Fixed Length:

Total Digits:

Fraction Digits:

Value Constraints

Define constraints on the values of the simple type

Minimum Inclusive:

Maximum Inclusive:

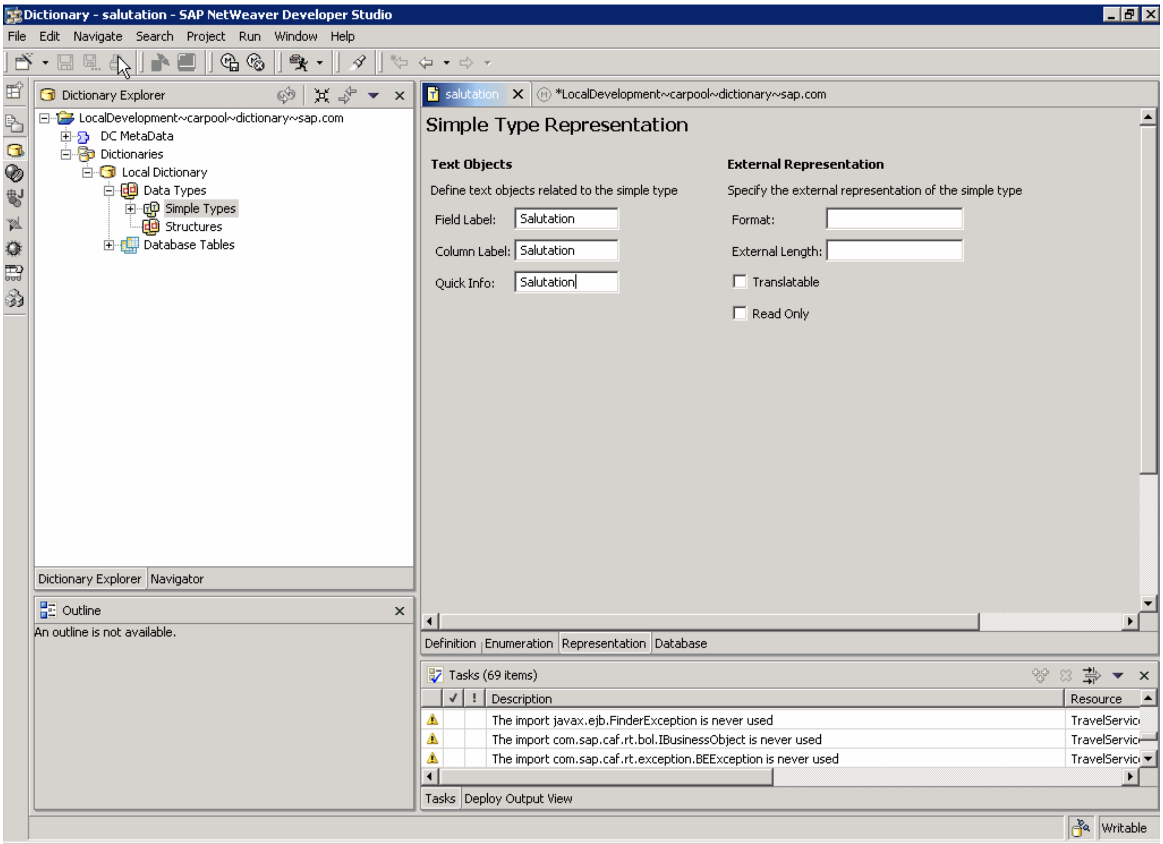
Minimum Exclusive:

Maximum Exclusive:

Tasks (69 items)

✓	!	Description	Resource
		The import javax.ejb.FinderException is never used	TravelServic
		The import com.sap.caf.rt.bol.IBusinessObject is never used	TravelServic
		The import com.sap.caf.rt.exception.BEException is never used	TravelServic

5. Switch to the Representation tab page. Enter Salutation in the fields Field Label, Column Label and Quick Info.



An outline is not available.

Simple Type Representation

Text Objects

Define text objects related to the simple type

Field Label: Salutation

Column Label: Salutation

Quick Info: Salutation

External Representation

Specify the external representation of the simple type

Format:

External Length:

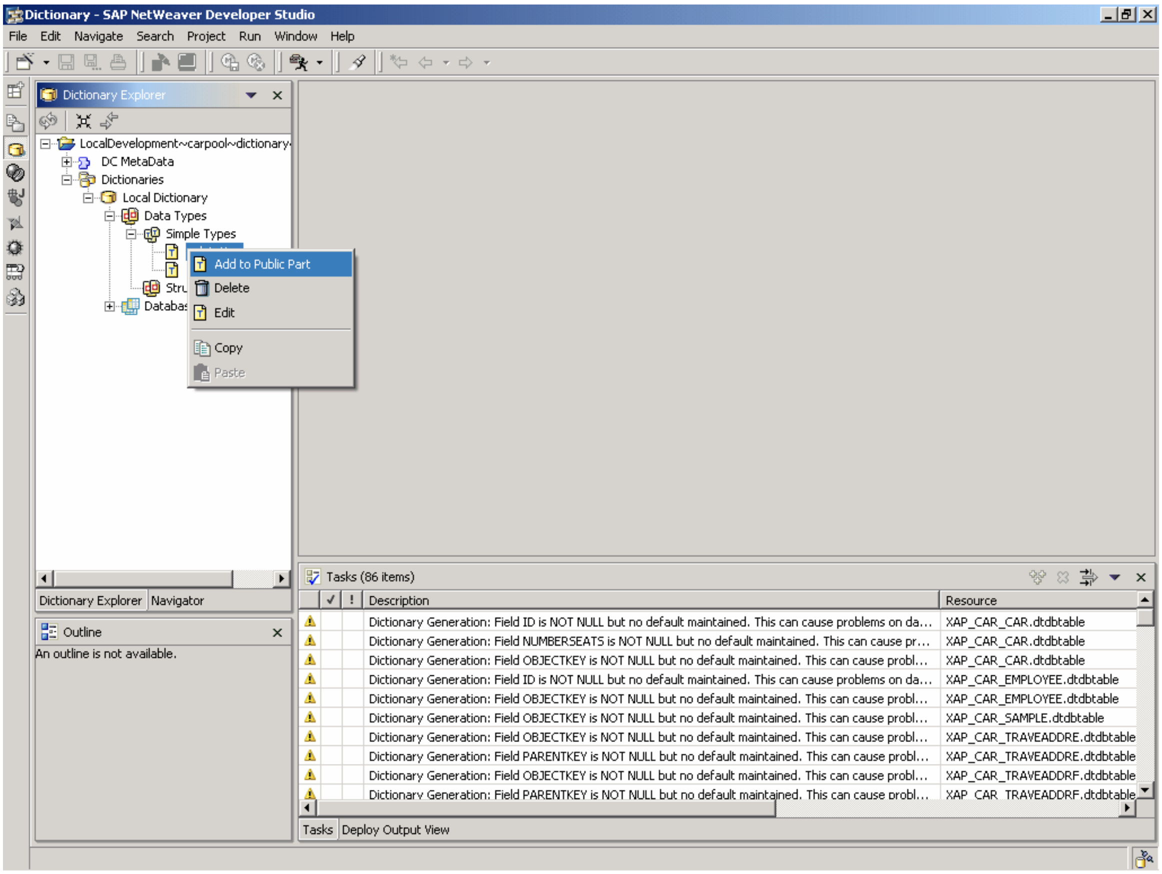
□ Translatable

□ Read Only

Tasks (69 items)

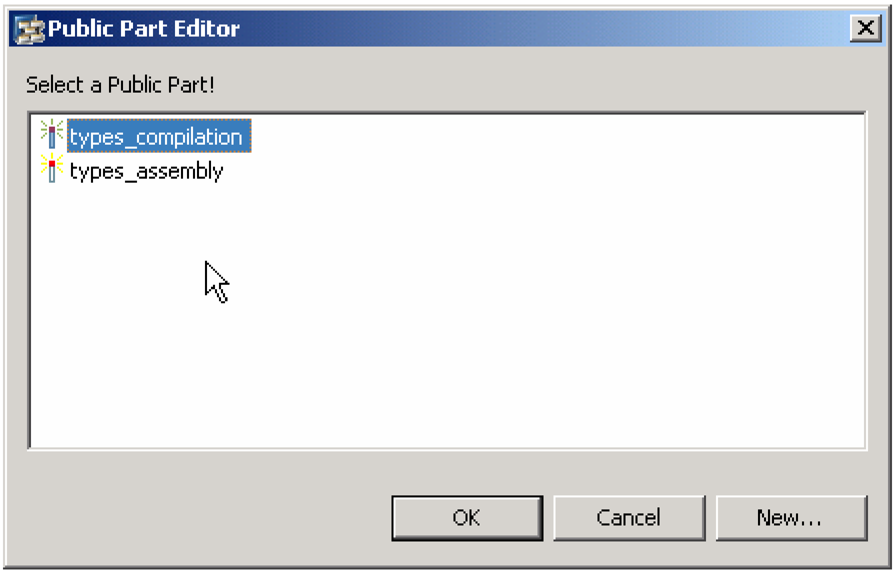
✓	!	Description	Resource
⚠		The import javax.ejb.FinderException is never used	TravelServic
⚠		The import com.sap.caf.rt.bol.IBusinessObject is never used	TravelServic
⚠		The import com.sap.caf.rt.exception.BEException is never used	TravelServic

6. In the Dictionary Explorer view, choose Add to Public Part from the context menu of the `salutation` simple type.



	✓	!	Description	Resource
⚠			Dictionary Generation: Field ID is NOT NULL but no default maintained. This can cause problems on da…	XAP_CAR_CAR.dtdbtable
⚠			Dictionary Generation: Field NUMBERSEATS is NOT NULL but no default maintained. This can cause pr…	XAP_CAR_CAR.dtdbtable
⚠			Dictionary Generation: Field OBJECTKEY is NOT NULL but no default maintained. This can cause probl…	XAP_CAR_CAR.dtdbtable
⚠			Dictionary Generation: Field ID is NOT NULL but no default maintained. This can cause problems on da…	XAP_CAR_EMPLOYEE.dtdbtable
⚠			Dictionary Generation: Field OBJECTKEY is NOT NULL but no default maintained. This can cause probl…	XAP_CAR_EMPLOYEE.dtdbtable
⚠			Dictionary Generation: Field OBJECTKEY is NOT NULL but no default maintained. This can cause probl…	XAP_CAR_SAMPLE.dtdbtable
⚠			Dictionary Generation: Field OBJECTKEY is NOT NULL but no default maintained. This can cause probl…	XAP_CAR_TRAVEADDRE.dtdbtable
⚠			Dictionary Generation: Field PARENTKEY is NOT NULL but no default maintained. This can cause probl…	XAP_CAR_TRAVEADDRE.dtdbtable
⚠			Dictionary Generation: Field OBJECTKEY is NOT NULL but no default maintained. This can cause probl…	XAP_CAR_TRAVEADDRF.dtdbtable
⚠			Dictionary Generation: Field PARENTKEY is NOT NULL but no default maintained. This can cause probl…	XAP_CAR_TRAVEADDRF.dtdbtable

7. In the public part editor, choose `types_compilation` and click on OK. Repeat the same for `types_assembly`. You do this, so that the type you created be available in the data type list when you create new attributes.



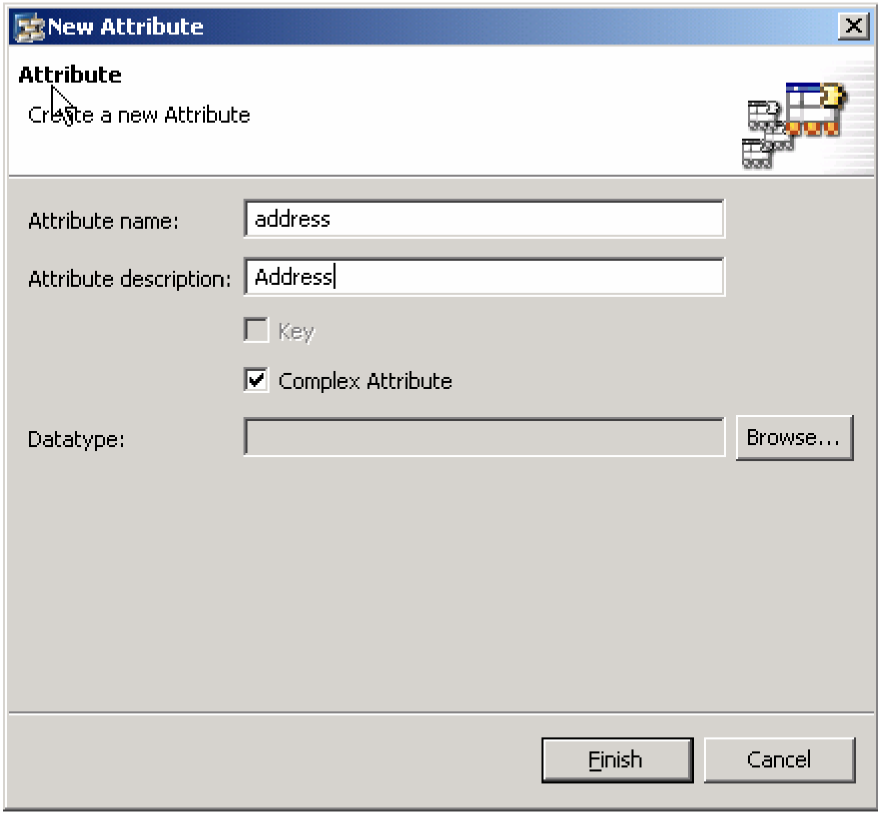
8. Save the metadata.

Creating Complex Attributes

1. Open the Composite Application Services (CAS) perspective. Open the *TravelLocation* entity.

1. Create a new attribute with the following parameters.

- Attribute name: address
- Attribute Description: Address
- Complex Attribute: *enabled*



New Attribute

Attribute

Create a new Attribute

Attribute name: address

Attribute description: Address

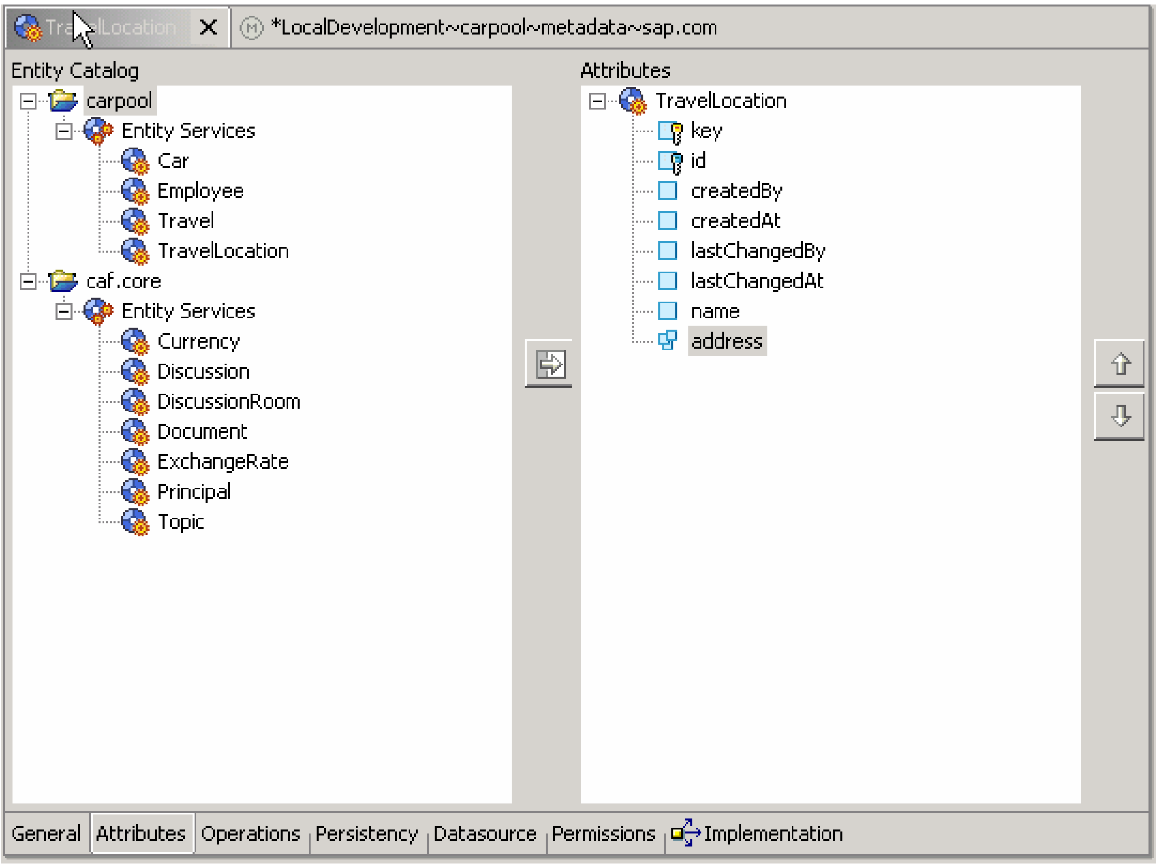
☐ Key

☑ Complex Attribute

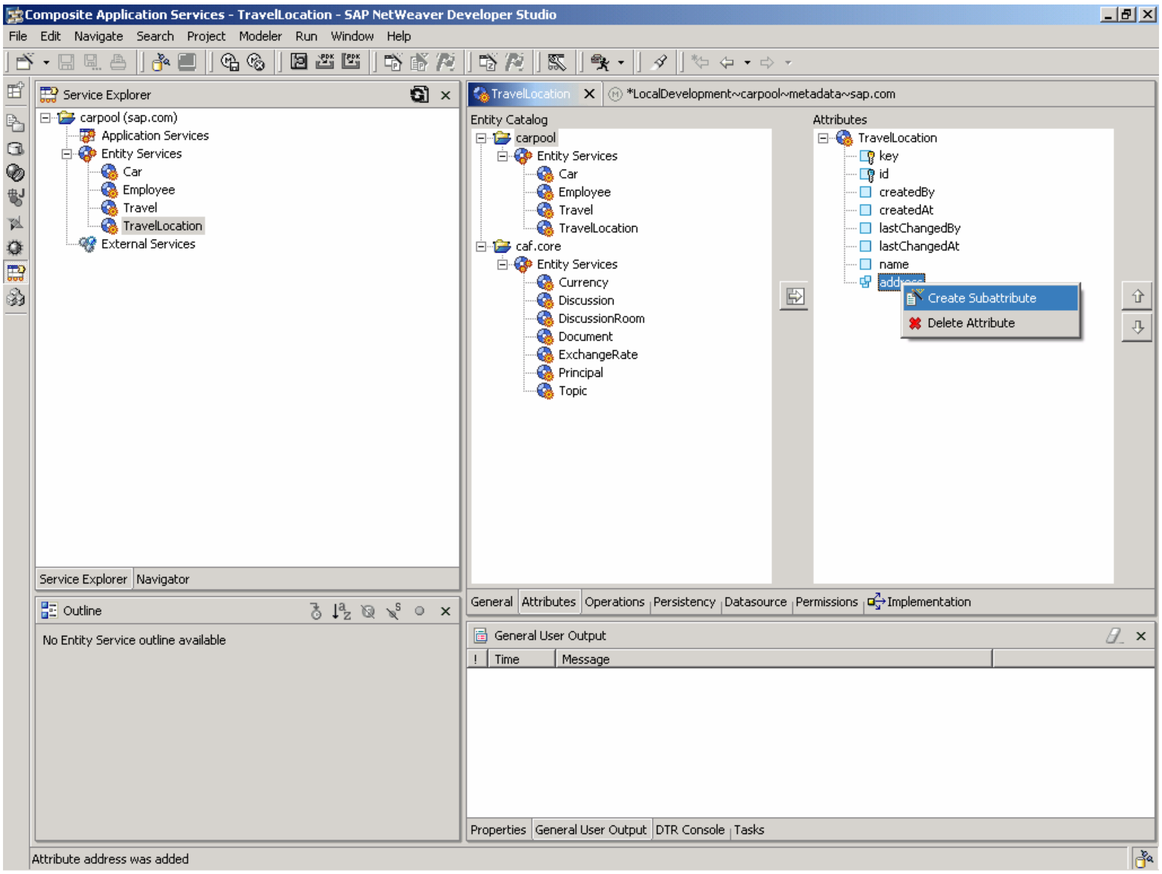
Datatype:

You do not specify any data type here.

The `address` attribute appears 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.



Attribute address was added

3. Create subattributes with the following parameters:

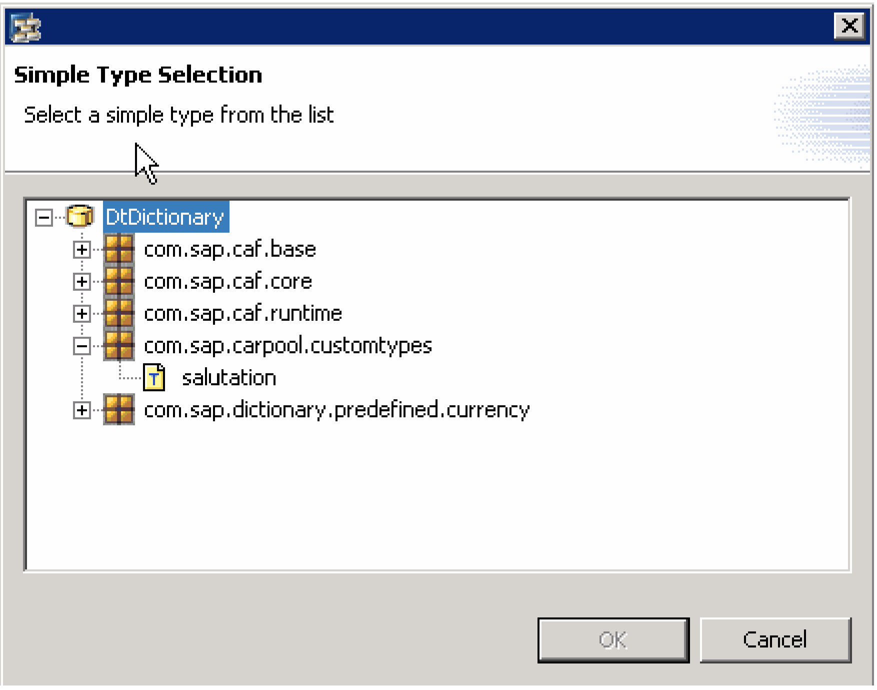
Attribute Name	Description	Data Type	Mandatory
address	Address	com.sap.caf.core .longText	
city	City	com.sap.caf.core .shortText	
zip	Zip Code	com.sap.caf.core .shortText	
country	Country	com.sap.caf.core .shortText	

2. Save the metadata.

4. Repeat step 4 to add a new attribute with the following parameters to the *Employee* entity:

Attribute Name	Description	Data Type	Mandatory
salutation	Salutation	com.sap.carpool.cust omtypes.salutation	

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



Simple Type Selection

Select a simple type from the list

3. Save the metadata, generate the project code, build and deploy the project.

5. Add the newly added attributes to the Object Editor UI pattern and test them.

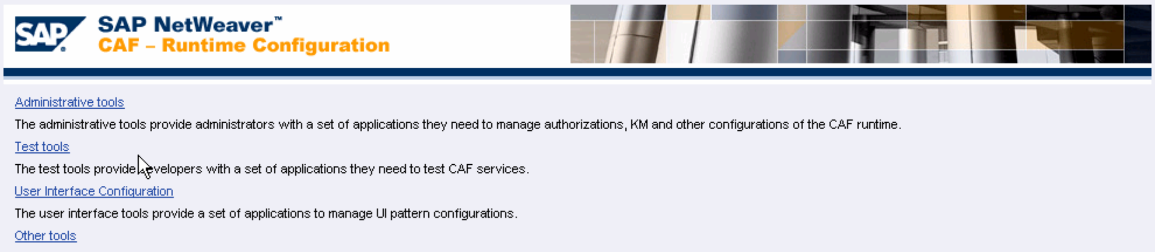
Defining Enumerations

This section explains how to create enumerations for the attributes. An enumeration is a list of permissible attribute values. It appears as a dropdown menu list in the UI pattern. Enumerations can be defined only for custom data types.

In this example we define enumeration for the custom data type `salutation`.

4. Launch the CAF Runtime Configuration page:

```
http://<host>:<port>/
webdynpro/dispatcher/sap.com/caf~UI~configbrowser/Config
```



SAP NetWeaver™
CAF – Runtime Configuration

Administrative tools

The administrative tools provide administrators with a set of applications they need to manage authorizations, KM and other configurations of the CAF runtime.

Test tools

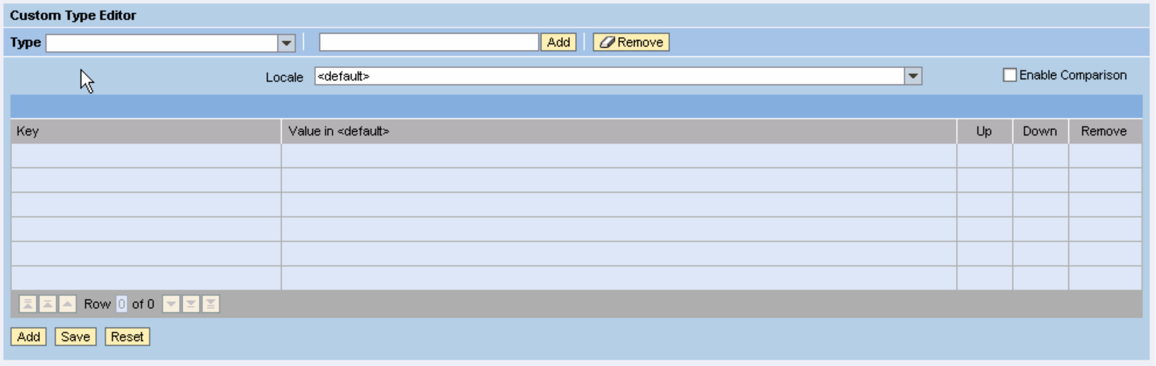
The test tools provide developers with a set of applications they need to test CAF services.

User Interface Configuration

The user interface tools provide a set of applications to manage UI pattern configurations.

Other tools

5. Navigate to *Administrative Tools* → *Custom Enumeration Type Editor*



Locale: <default>

Key	Value in <default>	Up	Down	Remove

Row 0 of 0

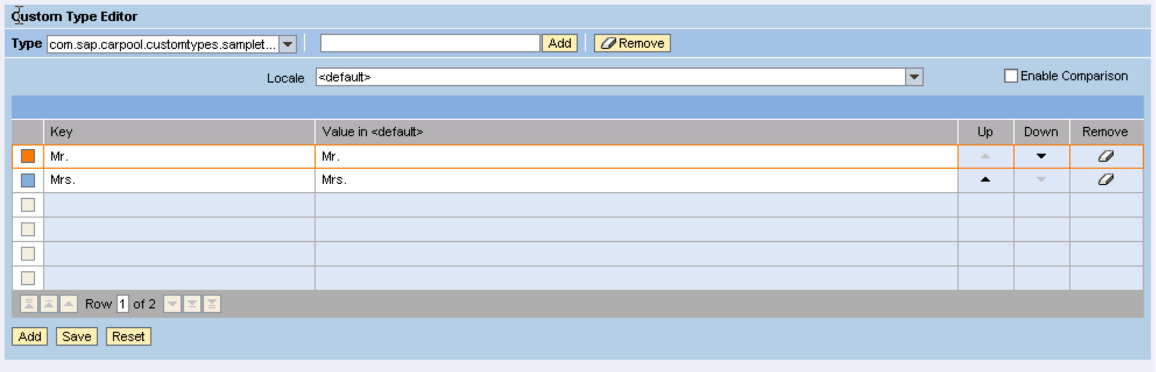
Alternatively, launch the following URL to access the enumeration type editor directly:

```
http://<host>:<port>/
webdynpro/dispatcher/sap.com/caf~UI~typeeditor/TypeEditor
```

6. Enter the custom data type name **`com.sap.carpool.customtypes.salutation`** in the input field next to the *Type* dropdown menu, and choose *Add*.

1. Now choose Add at the bottom of the screen and enter the following values:

Key	**Value**
Mr.	Mr.
Mrs.	Mrs.



Type: com.sap.carpool.customtypes.samplet...

Locale: <default>

Key	Value in <default>	Up	Down	Remove
Mr.	Mr.		▼	✎
Mrs.	Mrs.	▲		✎

Row 1 of 2

7. Choose *Save*.

2. Now test the entity service Employee in the Object Editor pattern. The field *salutation* appears as a dropdown list with two values – Mr. and Mrs.