

Creating an Application Service



# SAP Composite Application Framework

Document Version 1.00 - November 2005



SAP AG  
Neurottstraße 16  
69190 Walldorf  
Germany  
T +49/18 05/34 34 24  
F +49/18 05/34 34 20  
[www.sap.com](http://www.sap.com)

© 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<sup>®</sup>, 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, 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<sup>™</sup>. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java<sup>™</sup> 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

## Icons

Type Style	Represents	Icon	Meaning
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.		Caution
	Cross-references to other documentation.		Example
<b>Example text</b>	Emphasized words or phrases in body text, graphic titles, and table titles.		Note
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.		Recommendation
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.		Syntax
<b>Example text</b>	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.		

## Contents

Introduction.....	2
About This Document.....	2
Scenario Overview .....	2
Prerequisites .....	3
Recommended Readings.....	3
Disclaimer.....	3
Creating Application Service .....	3
Configuring UI Pattern for the Application Service.....	11

## 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 an application service that uses existing entities.

### 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.

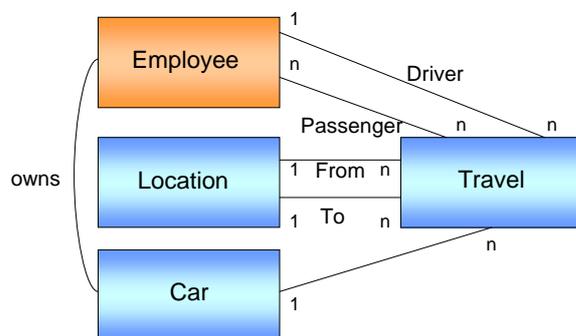
To help people find other travellers and set up car pools, the company needs a new application. This series demonstrates how to develop a prototype for such application using local data sources. The real-life productive system should work with data available in the HR (employees, travel planning, and 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: <ul style="list-style-type: none"> <li>• Sneak Preview SAP NetWeaver 04 – Web Application Server 6.40 Java. You can download the sneak preview at <a href="http://sdn.sap.com">http://sdn.sap.com</a></li> </ul>
Documents	Before you start with this tutorial, see: <ul style="list-style-type: none"> <li>• Configuring Your System for CAF Development</li> <li>• Creating a Local Entity with Maintenance UI</li> <li>• Creating Data Types</li> </ul>

## Recommended Readings

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

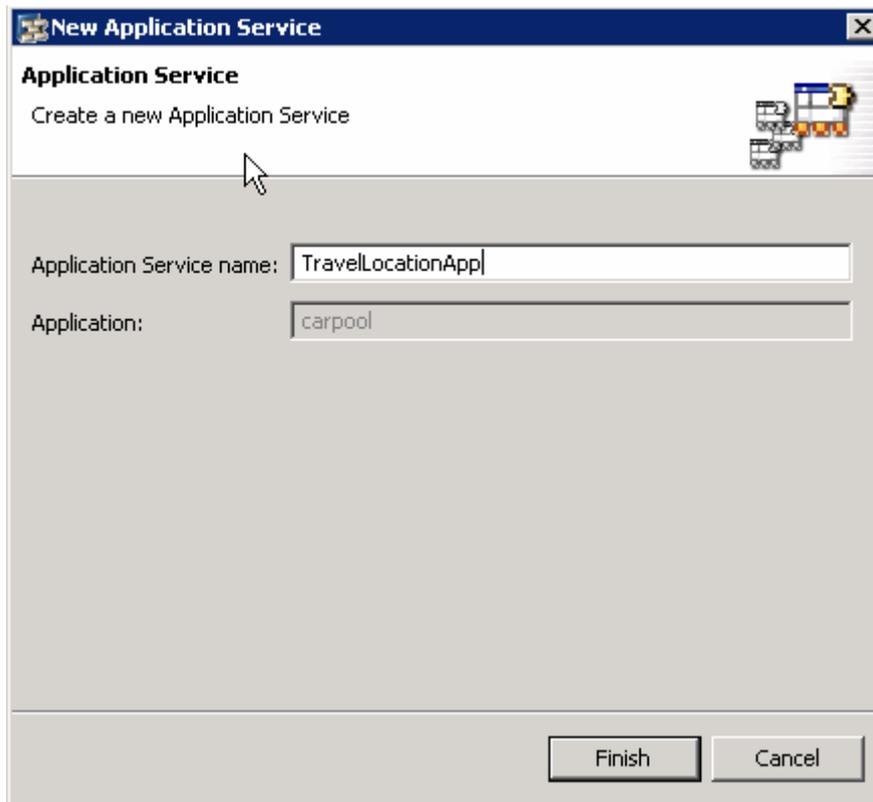
- Testing Entity and Application Services
- Implementing Permission Checks
- Creating a Web Dynpro UI for a Composite Application

## Disclaimer

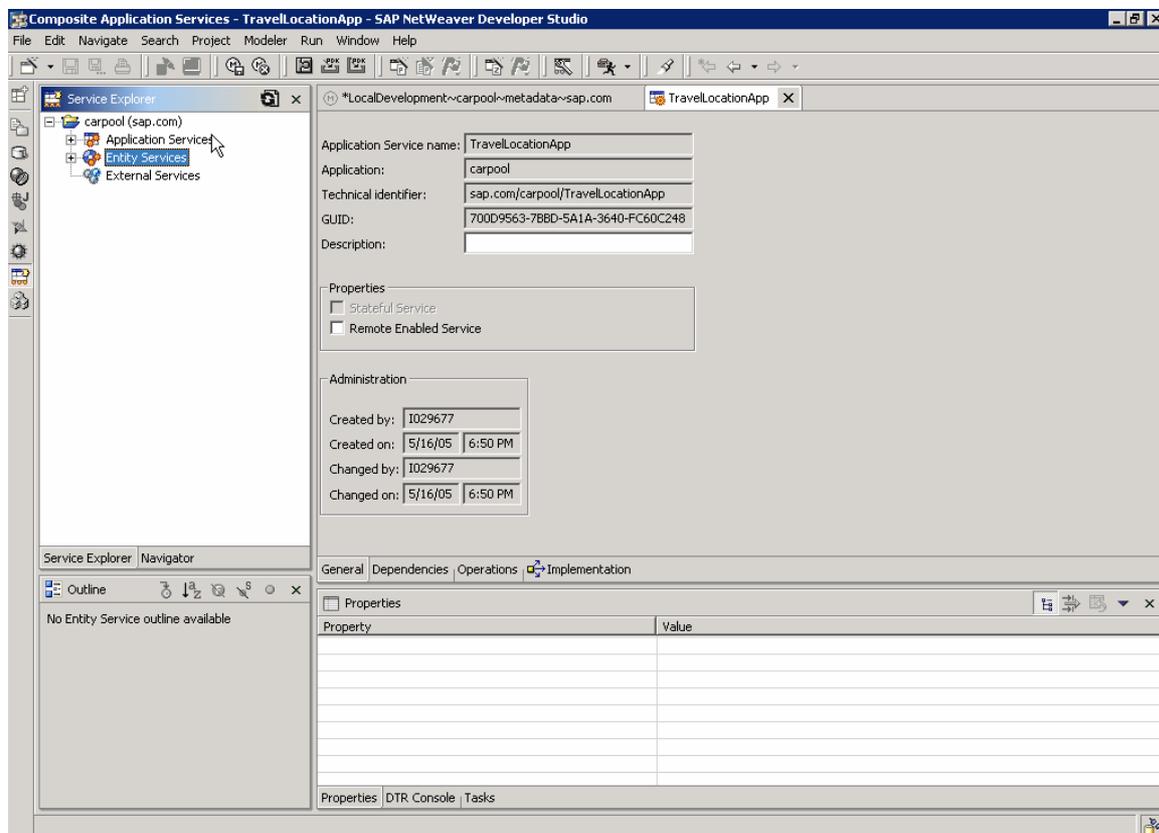
Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the 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 Application Service

1. Open the carpool project in the SAP NetWeaver Developer Studio.
2. To create a new application service, open the context menu of Application Services in the Service Explorer view and choose *New*.

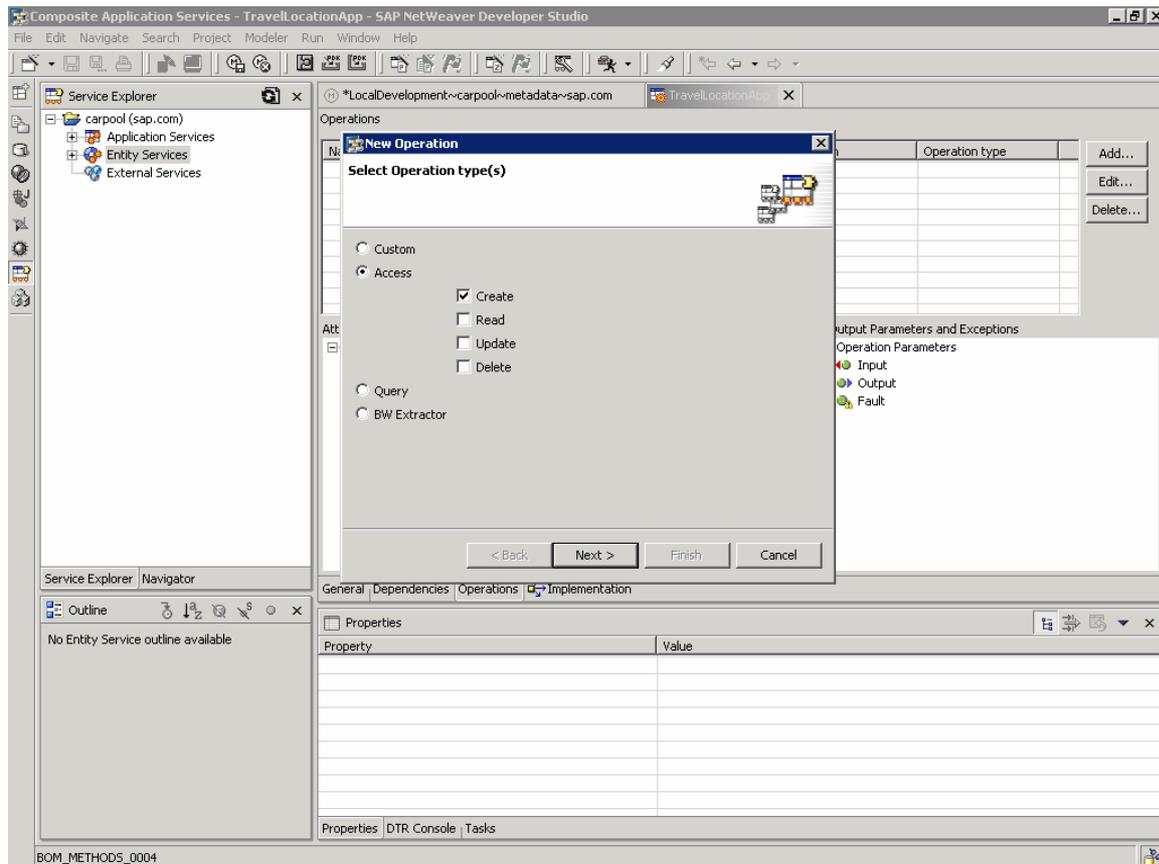


Choose *Finish*. A new application service appears in the Service Explorer view.

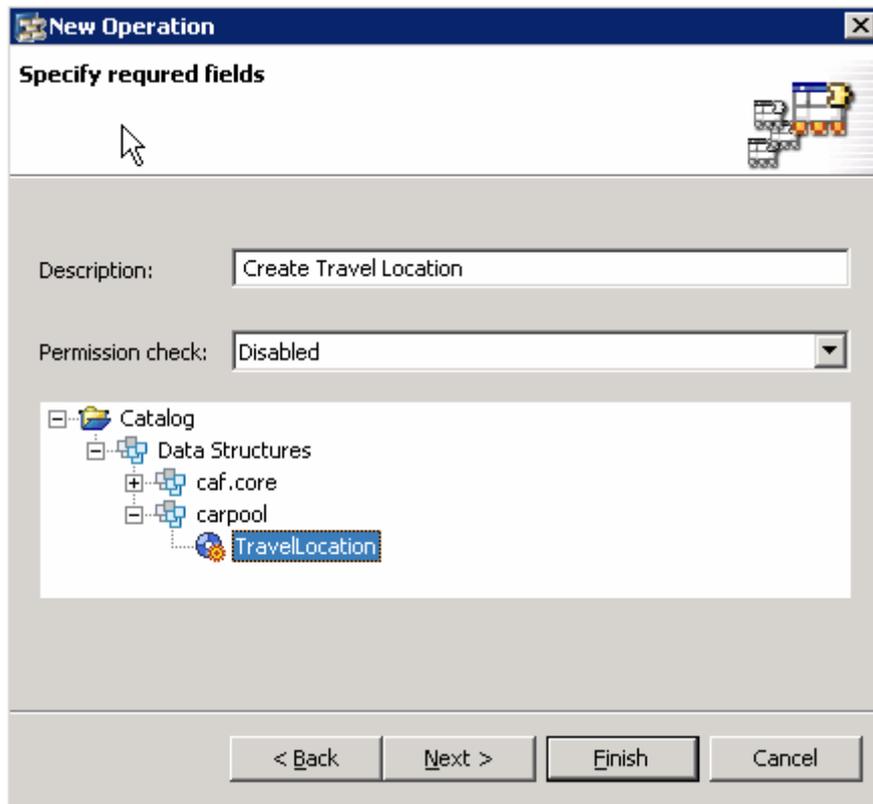


3. Open the editor of the *TravelLocationApp* application service. Switch to the *Dependencies* tab page.

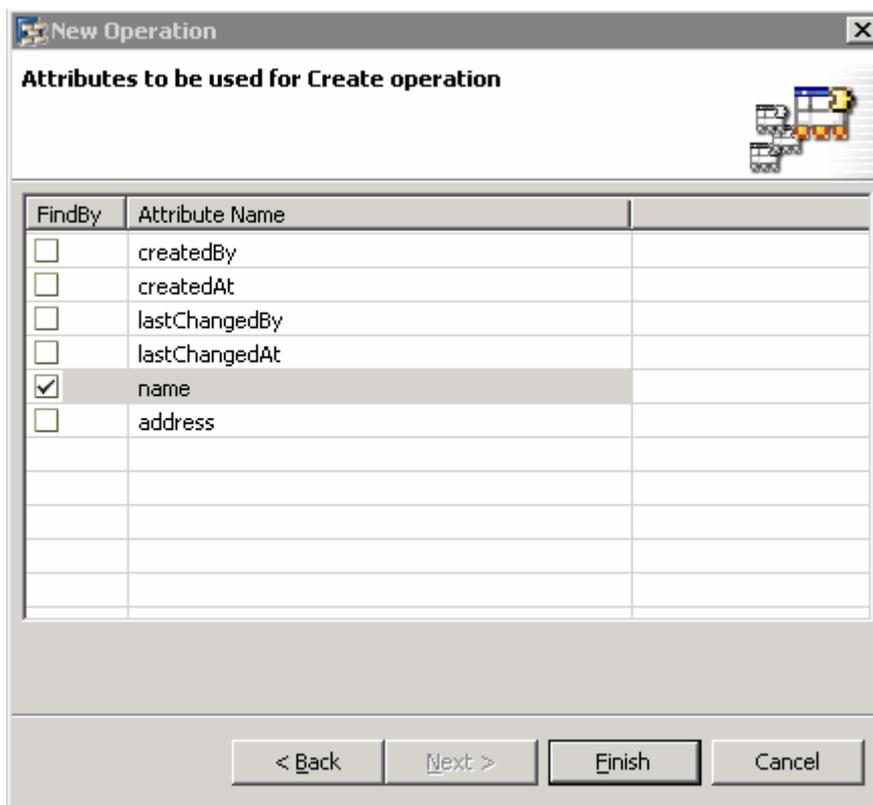




- b. Choose *Next*.
- c. Enter the following data in the *New Operation* screen:
  - Description: Create Travel Location
  - Permission Check: Disabled
- d. Select *TravelLocation* from the tree.



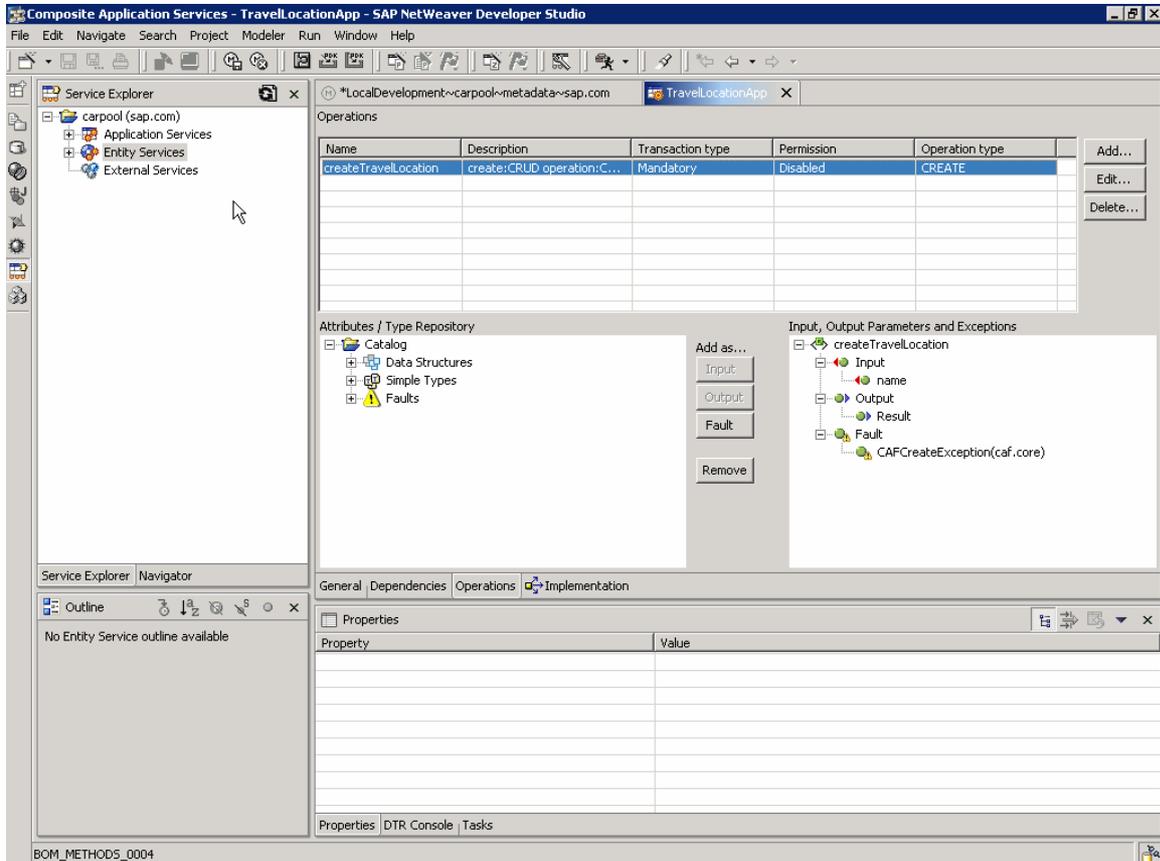
- e. Choose *Next*.
- f. Select the `name` attribute. This is the input parameter for the method.



- g. Choose *Finish*. A new operation `createTravelLocation` is added to the *Operations* list.

5. In the *Attributes/Type Repository* list, select *Faults* → *caf.core* → *ServiceException* and choose *Fault*.

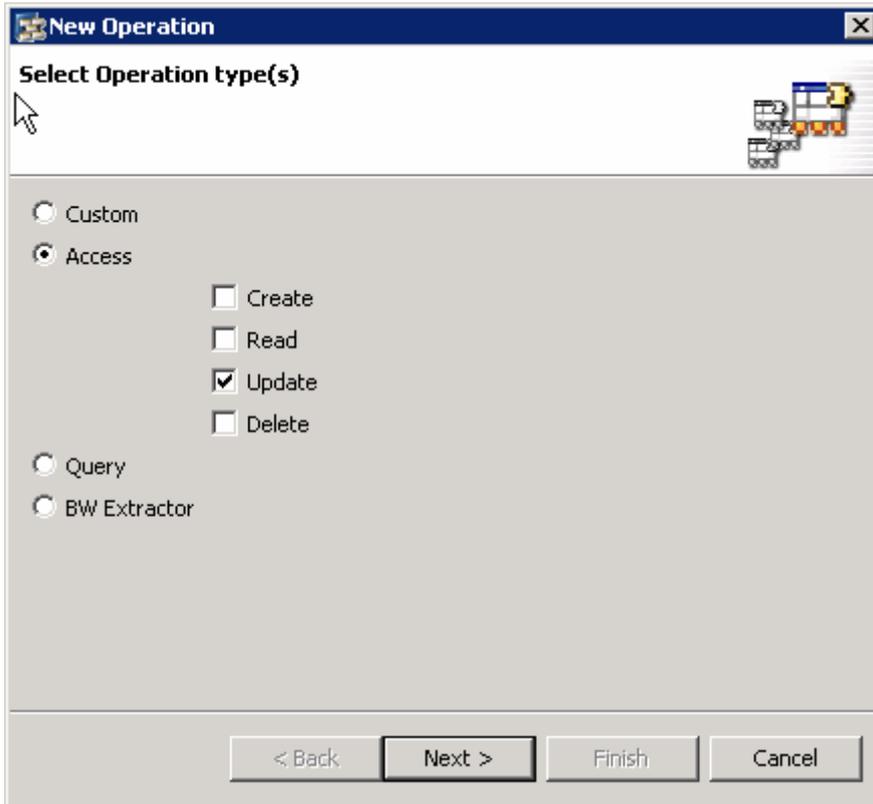
The *ServiceException* will appear in the right pane.



6. Repeat the procedure to create an update operation with the following parameters for the entity service *TravellLocation*:

- Description: Update Travel Location
- Permission Check: *disabled*

Select *TravellLocation* from the tree.



Add a new `ServiceException` as explained in the step 5.

7. Switch to the *Implementation* tab page. We have to write the code for Create and Update methods we just created.

We will create a new method for generating unique IDs for newly created travel locations.

- a. Add the following code between the *Internal Coding* tags:

```
public static synchronized long getUniqueId() {  
    return System.currentTimeMillis();  
}
```

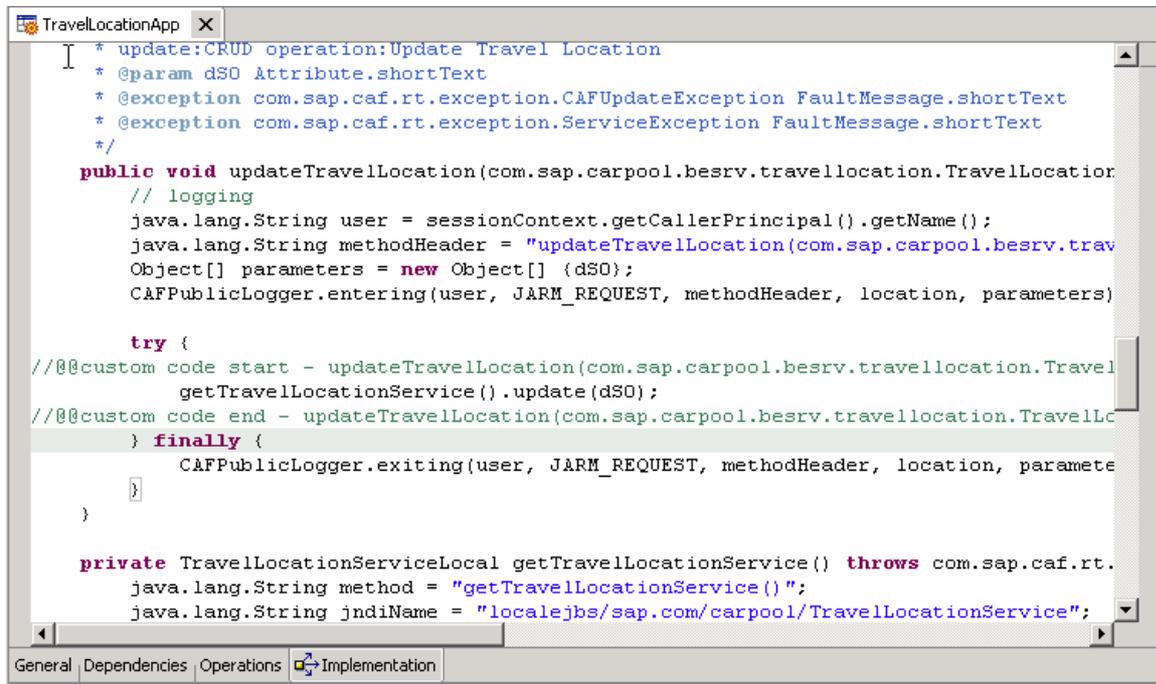


This is a simple approach to create a unique ID. In real life a more complicated technique should be used.



- c. In the updateTravelLocation method add the following code in the try block.

```
getTravelLocationService().update(dSO);
```



```

TravelLocationApp X
I
 * update:CRUD operation:Update Travel Location
 * @param dSO Attribute.shortText
 * @exception com.sap.caf.rt.exception.CAFUpdateException FaultMessage.shortText
 * @exception com.sap.caf.rt.exception.ServiceException FaultMessage.shortText
 */
public void updateTravelLocation(com.sap.carpool.besrv.travellocation.TravelLocation
// logging
java.lang.String user = sessionContext.getCallerPrincipal().getName();
java.lang.String methodHeader = "updateTravelLocation(com.sap.carpool.besrv.trav
Object[] parameters = new Object[] {dSO};
CAFPublicLogger.entering(user, JARM_REQUEST, methodHeader, location, parameters)

try {
/**@custom code start - updateTravelLocation(com.sap.carpool.besrv.travellocation.Travel
getTravelLocationService().update(dSO);
/**@custom code end - updateTravelLocation(com.sap.carpool.besrv.travellocation.TravelL
} finally {
CAFPublicLogger.exiting(user, JARM_REQUEST, methodHeader, location, paramete
}
}

private TravelLocationServiceLocal getTravelLocationService() throws com.sap.caf.rt.
java.lang.String method = "getTravelLocationService()";
java.lang.String jndiName = "localejbs/sap.com/carpool/TravelLocationService";
General Dependencies Operations Implementation

```

8. You have created the application service. Save the metadata, generate the project code, build the DCs and deploy the project.

## Configuring UI Pattern for the Application Service

We will use two UI patterns for testing the Application Service TravelLocationApp.

- Object Editor – to create TravelLocation
  - Relation Tab – to create the address for the Travel Location
9. Launch the CAF Ui configuration Browser using the URL [http://<was\\_host>:<was-port>/webdynpro/dispatcher/sap.com/caf~UI~configbrowser/ConfigBrowser](http://<was_host>:<was-port>/webdynpro/dispatcher/sap.com/caf~UI~configbrowser/ConfigBrowser)
  1. Select the Relation Tab from the left-hand pane.



2. Choose *New Configuration*. In the *Relation Tab Configurator* screen, enter the following data:

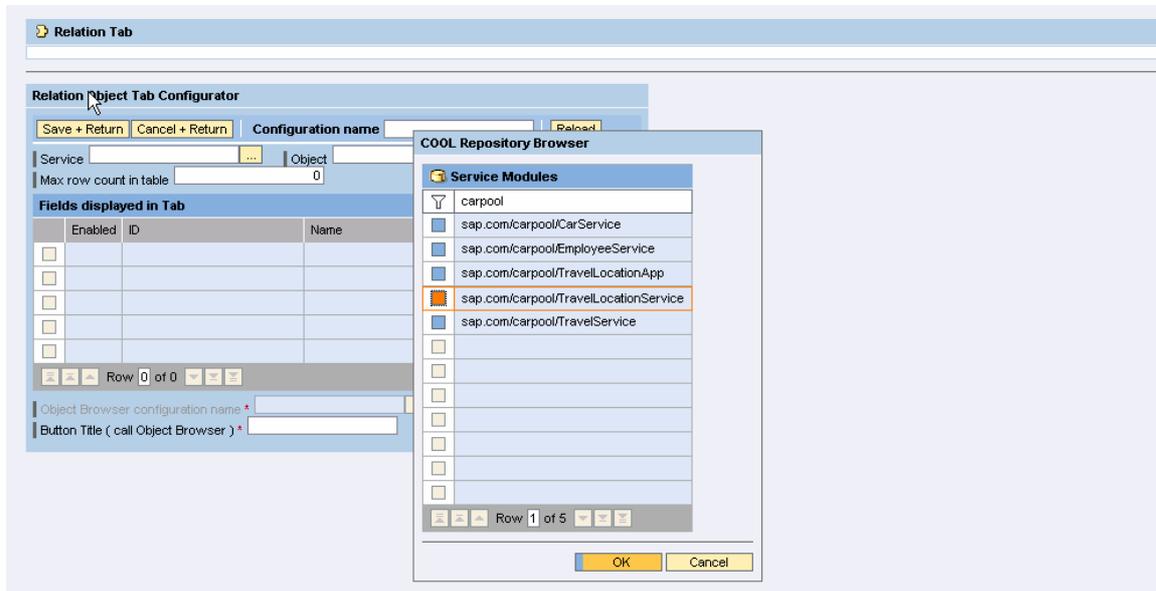
- Configuration Name: carpool\_TravelLocation
- Service: sap.com/carpool/TravelLocationService
- Object: sap.com/carpool/TravelLocation\_address
- Max row count in table: 1
- Button Title: Edit Address

3. Choose *Retrieve Metadata*, and select the fields address, city, zip, and country.

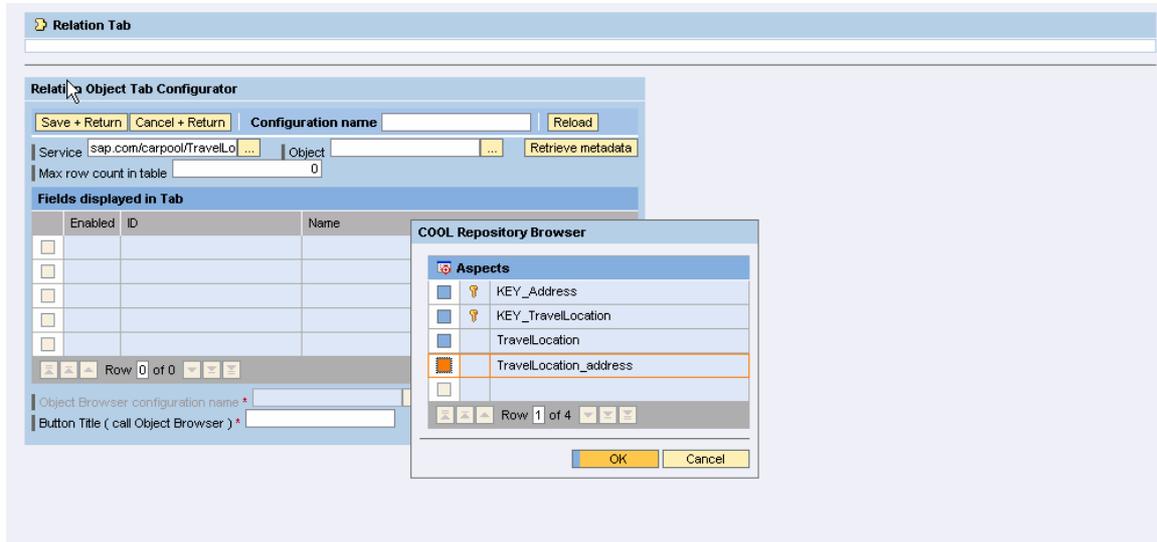


Check the Enabled checkbox for these fields.

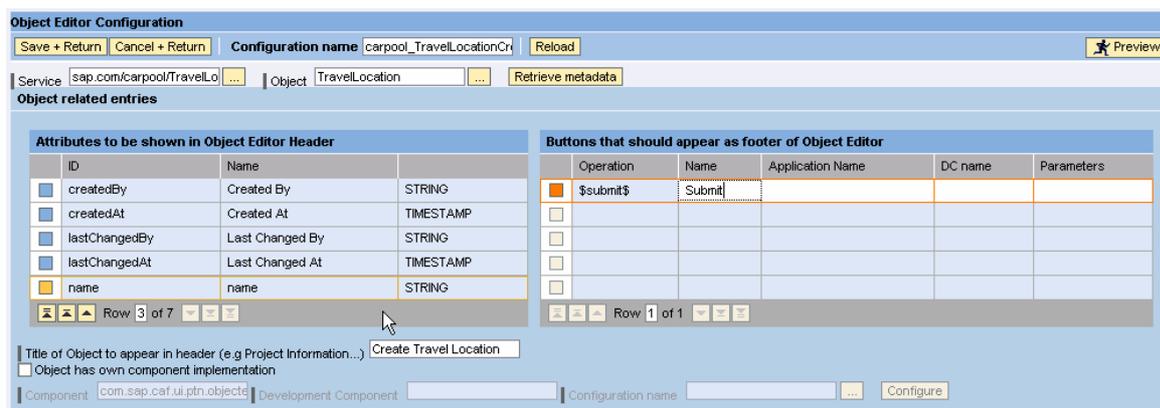
### Service Selection



## Object Selection



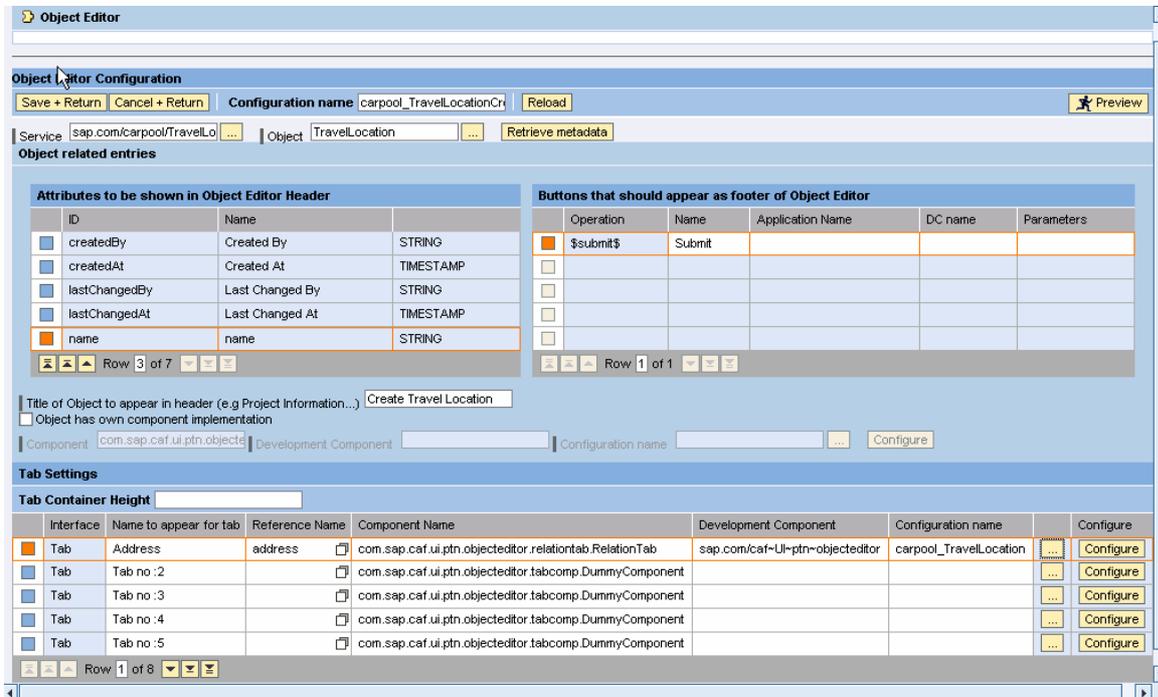
4. Choose *Save + Return*.
5. Choose *Object Editor* from the *Configuration Browser*. Create a new Configuration with the following parameters:
  - Configuration Name: carpool\_TravelLocationCreate
  - Service: sap.com/carpool/TravelLocationApp
  - Object: TravelLocation
  - Attributes to be shown in:
    - Editor Header: name
    - Button Name: Submit
    - Title of Object: Create Travel Location



6. Now we need to add the *Relation* tab to the Object Editor.
  - a. In the *Tab Settings* section, select the first row and enter **Address** in the column *Name to appear for tab*.
  - b. Select the *address* for the reference name.

- c. Choose  (Browse) and select `sap.com/caf~UI~ptn~objecteditor` → *RelationTab*. Select the configuration name `carpool_TravelLocation` from the dropdown list. Choose OK.

The Object Editor screen looks as follows.



**Object Editor Configuration**

Save + Return Cancel + Return Configuration name: carpool\_TravelLocationCr Reload Preview

Service: sap.com/carpool/TravelLo ... Object: TravelLocation ... Retrieve metadata

**Object related entries**

Attributes to be shown in Object Editor Header			Buttons that should appear as footer of Object Editor				
ID	Name		Operation	Name	Application Name	DC name	Parameters
<input type="checkbox"/>	createdBy	Created By		<input checked="" type="checkbox"/>	\$submit\$	Submit	
<input type="checkbox"/>	createdAt	Created At		<input type="checkbox"/>			
<input type="checkbox"/>	lastChangedBy	Last Changed By		<input type="checkbox"/>			
<input type="checkbox"/>	lastChangedAt	Last Changed At		<input type="checkbox"/>			
<input checked="" type="checkbox"/>	name	name		<input type="checkbox"/>			

Row 3 of 7

Row 1 of 1

Title of Object to appear in header (e.g Project Information...): Create Travel Location

Component: com.sap.caf.ui.ptn.objecteditor Development Component: Configuration name: Configure

**Tab Settings**

Interface	Name to appear for tab	Reference Name	Component Name	Development Component	Configuration name	Configure
<input checked="" type="checkbox"/>	Tab	Address	address	com.sap.caf.ui.ptn.objecteditor.relationtab.RelationTab	sap.com/caf~UI~ptn~objecteditor	carpool_TravelLocation ... Configure
<input type="checkbox"/>	Tab	Tab no :2		com.sap.caf.ui.ptn.objecteditor.tabcomp.DummyComponent		... Configure
<input type="checkbox"/>	Tab	Tab no :3		com.sap.caf.ui.ptn.objecteditor.tabcomp.DummyComponent		... Configure
<input type="checkbox"/>	Tab	Tab no :4		com.sap.caf.ui.ptn.objecteditor.tabcomp.DummyComponent		... Configure
<input type="checkbox"/>	Tab	Tab no :5		com.sap.caf.ui.ptn.objecteditor.tabcomp.DummyComponent		... Configure

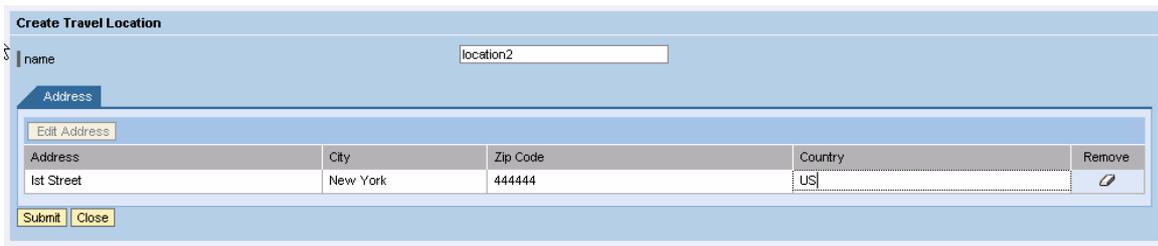
Row 1 of 8

7. To save the Object Editor configuration, choose Save + Return.

8. Test the UI by launching the following URL:

`http://<host>:<port>/webdynpro/dispatcher/sap.com/caf~UI~ptn~objecteditor/ObjectEditor?app.configName=carpool_TravelLocationCreate`

9. To enter a new address, choose *Edit Address*, and enter test data.



**Create Travel Location**

name: location2

**Address**

Edit Address

Address	City	Zip Code	Country	Remove
1st Street	New York	44444	US	

Submit Close

10. You have configured and tested the UI patterns for the application service.