SAP Composite Application Framework

Implementing Permission Checks

Version 1.00 – July 2006

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

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
	Syntax

Contents

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 an Application Service

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

Applicable Releases

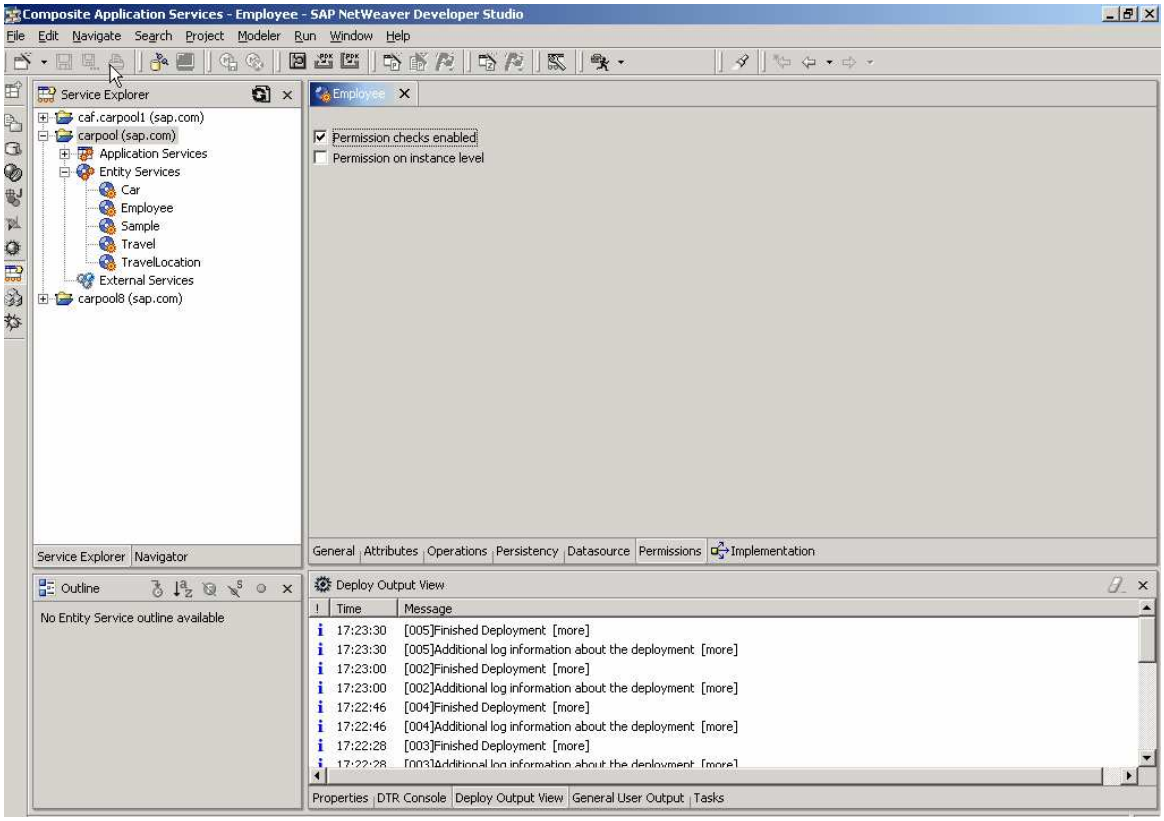
This tutorial is compatible with the following releases:

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

Disclaimer

Implementing Permission Checks for Entities

1) In the NetWeaver Development Studio, Open the Entity Service *Employee*. Switch to the tab Permissions. Select the checkbox *Permission Checks Enabled*.



Generate the Project Code, Build the component and deploy it to the J2EE engine.

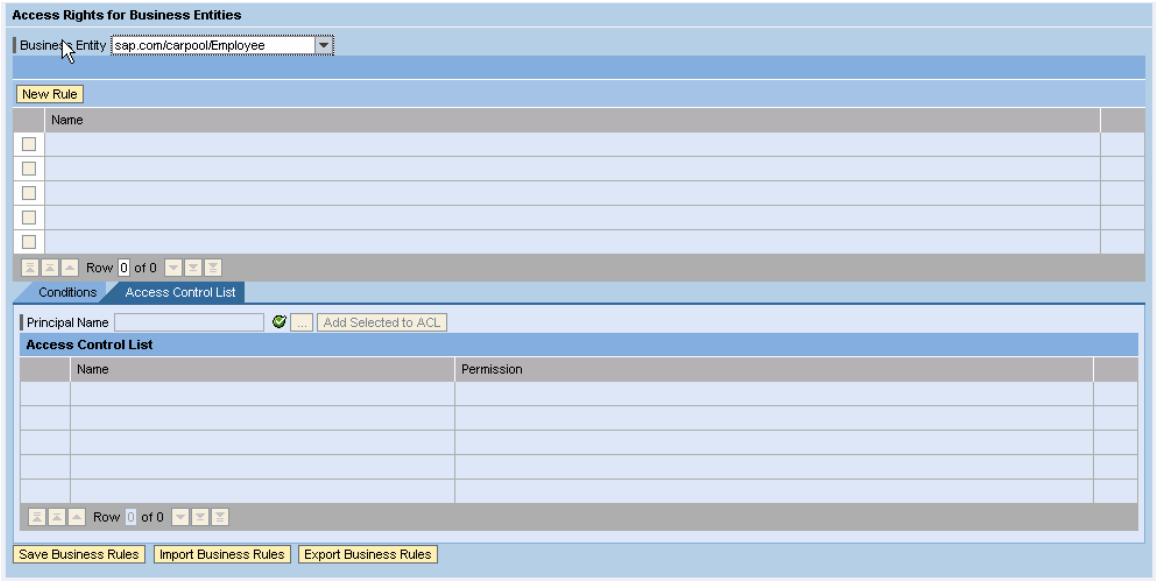
2) Now try to create an instance of Employee using the URL http://<was_host>:<was_port>/webdynpro/dispatcher/sap.com/caf~UI~ptn~objectedit or/ObjectEditor?app.configName=createEmployee (see How-to “Create local entity with maintenance UI”, chapter “Maintenance UI configuration using Pattern UI”).

You will receive an error message such as “User <username> has no create permissions for the Object Id sap.com/carpool/Employee”

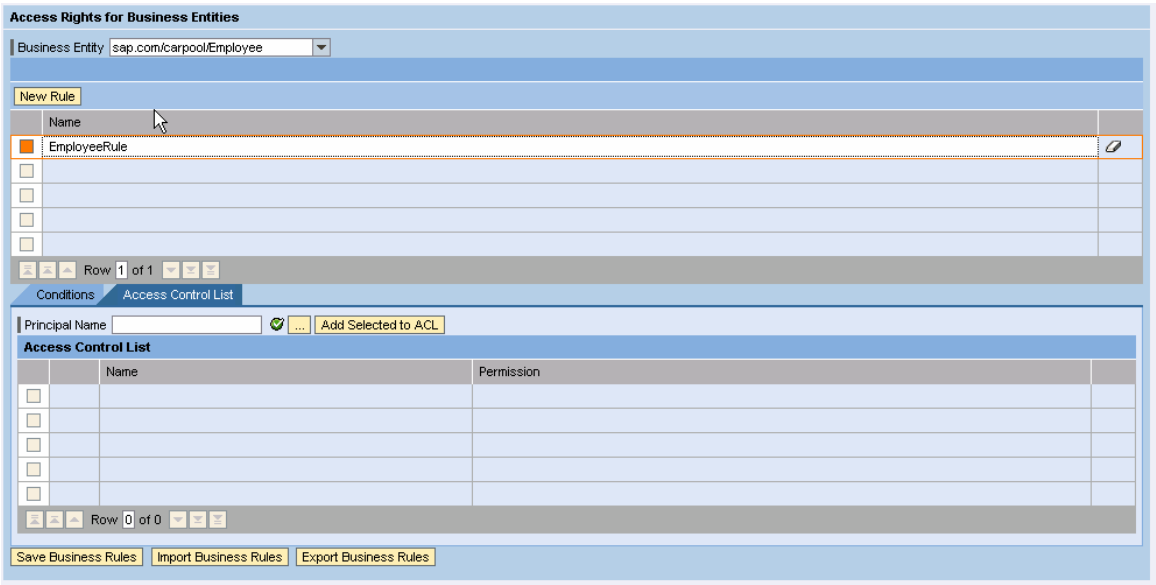
This is caused by the new Permission settings you have made in the Employee service: permission checks are done at runtime.

The following section explains the procedure to setup the permissions for different users.

3) To set the Authorization for the Entity Service launch the URL http://<was_host>:<was_port>/webdynpro/dispatcher/sap.com/caf~UI~ptn~authorizati on/Authorization

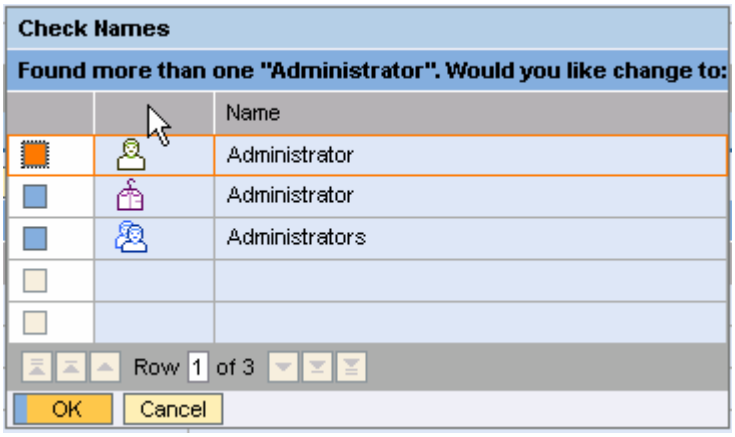


4) Select *sap.com/carpool/Employee* from the *Business Entity* dropdown and click the button *New Rule*. Enter the rule name *EmployeeRule*.



5) In the Principal Name input field, enter the User Name (e.g. Administrator) to whom you want to give access to Employee entity. Then click on the button *Add Selected to ACL*.

If the name entered has multiple matches, a popup will be displayed with all the options. Select one and click on the OK button.



The User will be added to the Access Control List (ACL). Select full control as the permission for the user. This allows the user to read, create, update and delete Employee.

6) Select the required Authorization in the Permission field. The following options can be used.

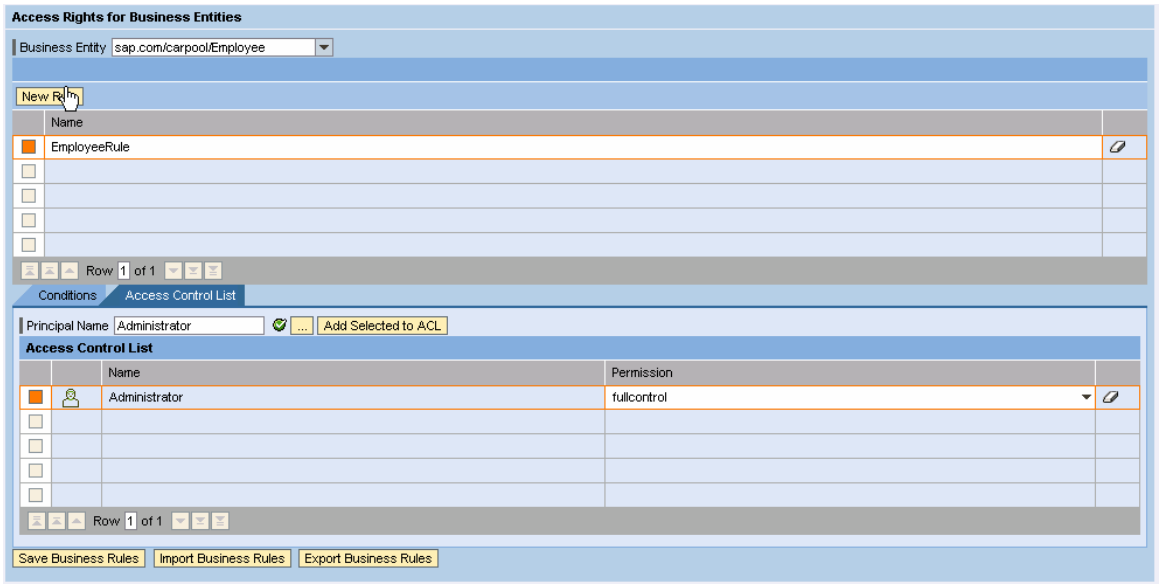
Read - The user has display rights to all instances of Employee Service.

Create - The user can create an instance of Employee service.

Update - The user can modify an existing instance of Employee Service

Delete - The user can delete an instance of Employee.

Full Control - The user has all of the above rights.



Click on the *Save Business Rules* button to save the access settings.

This completes the permission settings for the user. To verify, create an instance of Employee Service through the Object Editor. It should be successful now.

Implementing Permission Propagation for Entities

Entities might have related entities. For example, in our business case, entity Travel is related to other entities Car, Employee and TravelLocation. CAF provides options to propagate the permissions (authorizations) from parent entity to the related entities.

The access rights to a certain object could depend on the context in which the entity is invoked. To reduce complexity and maintenance effort these permissions must not be stored redundantly at each single instance of the entities that belong to the relationship network but need to be determined dynamically from the relationships of every entity.

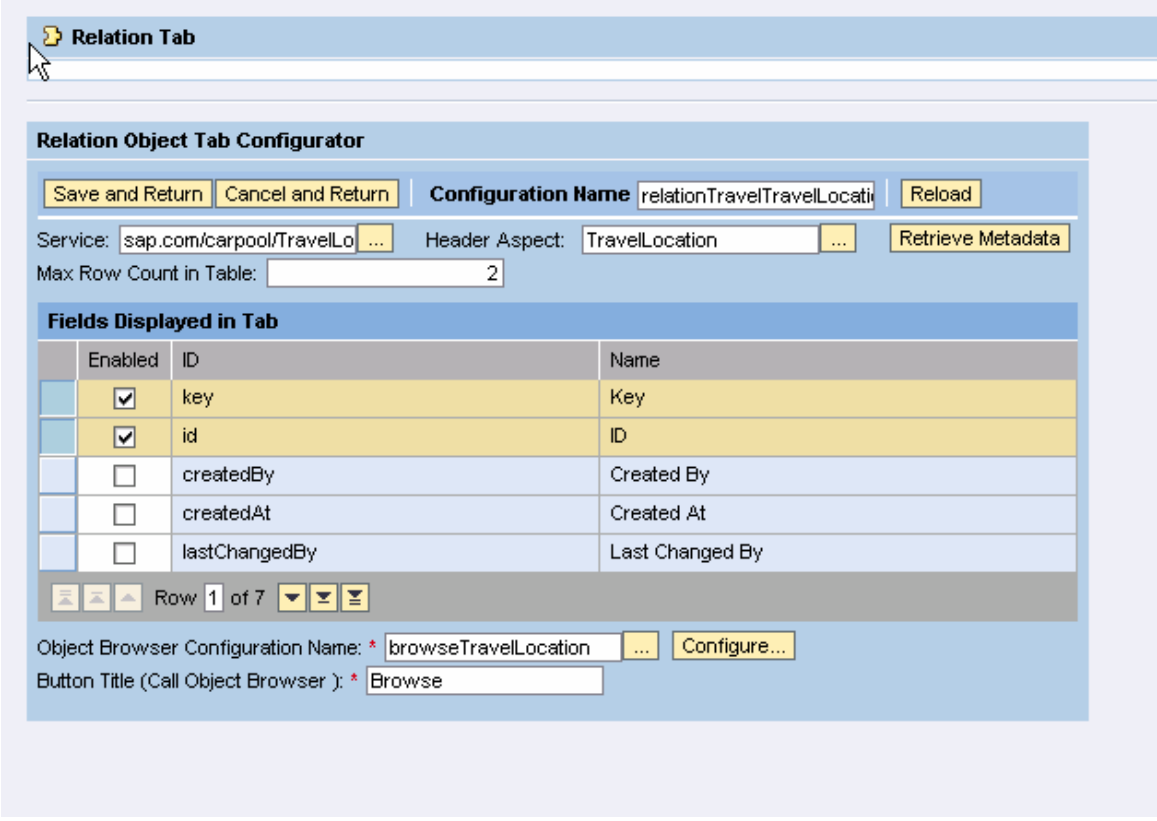
In the following section we will discuss how to propagate permissions for Travel Entity to the TravelLocation entity. This can be extended to other related entities.

1) At the beginning you will set up an UI pattern to test the Permission Propagation scenario.

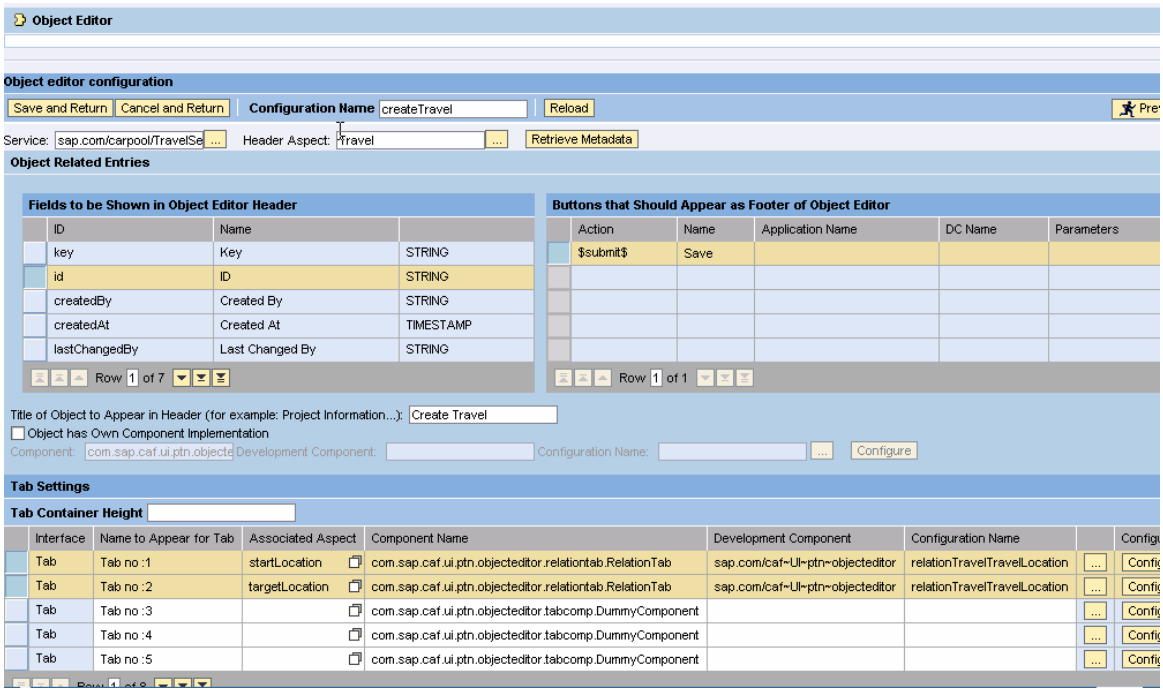
Open the *carpool* project in your IDE and enable permission checks for the entities *Travel* and *Travel Location*. Select only the first checkbox in the Permission tab. Save, Build and Deploy the DC.

Launch *http://<was_host>:<was_port>/webdynpro/dispatcher/sap.com/caf~UI~ptn~authorization/Authorization*, and assign *fullcontrol* rights for the user for the Entity *Travel*.

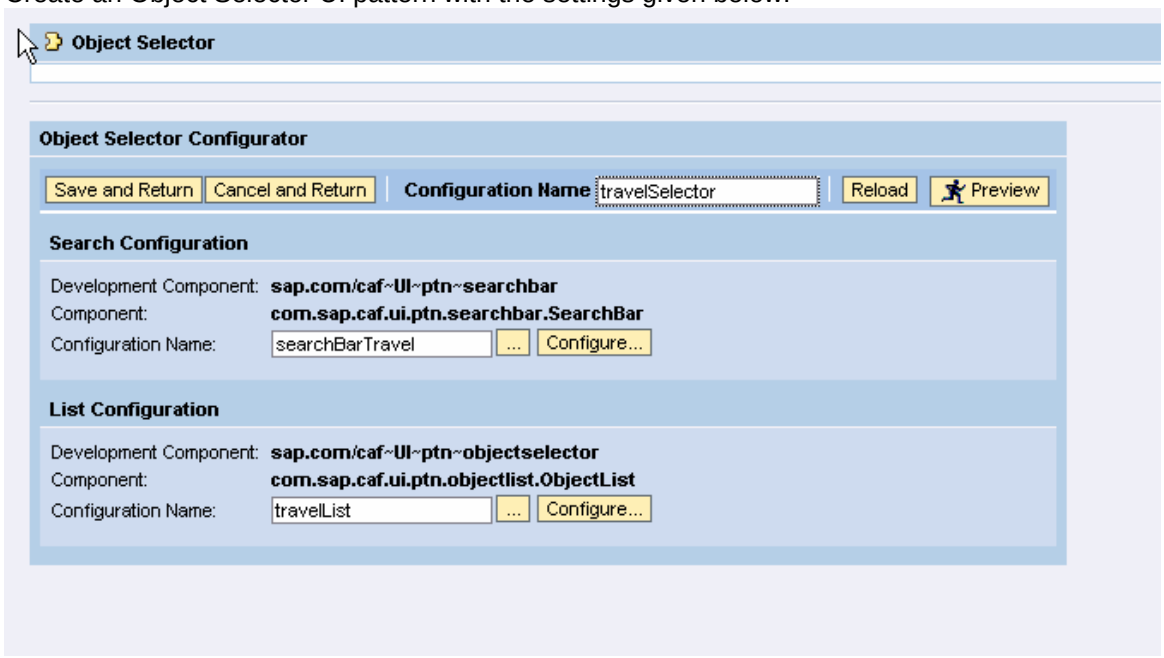
Create a *Relation Tab* pattern – *relationTravelTravelLocation*. Please not to select *id* and *name* in the list of fields.



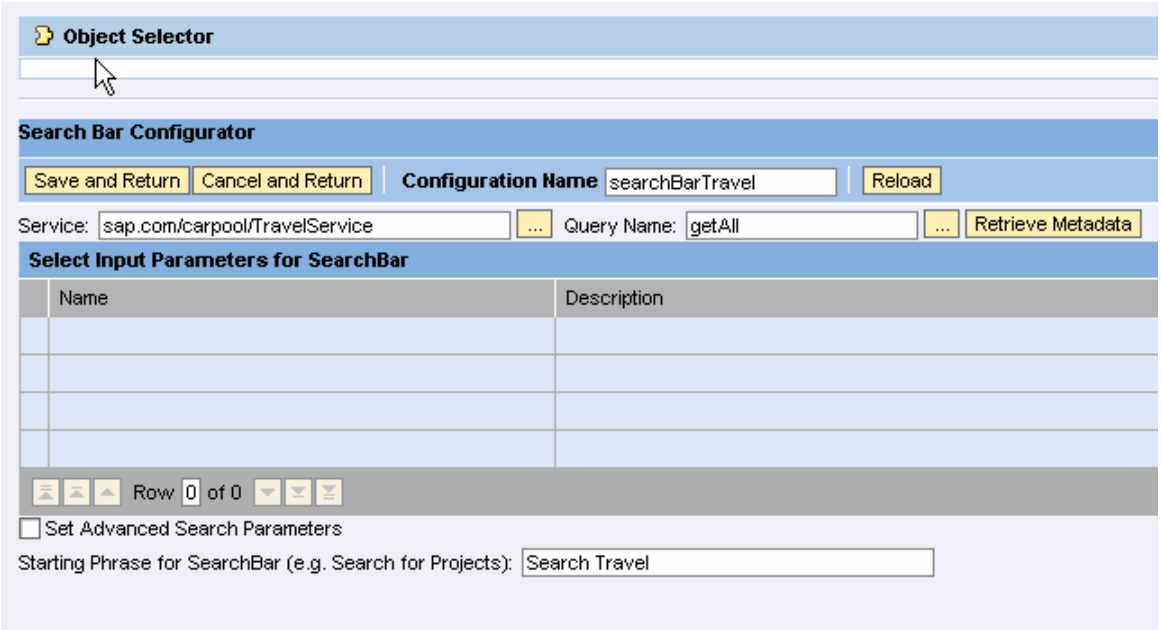
Next Create the *Object Editor* pattern *createTravel*. Select the *id* and *startDate* from the fields list. Also add the Relation Tab *relationTravelTravelLocation* in the Tabs for the aspects *startLocation* and *targetLocation*.



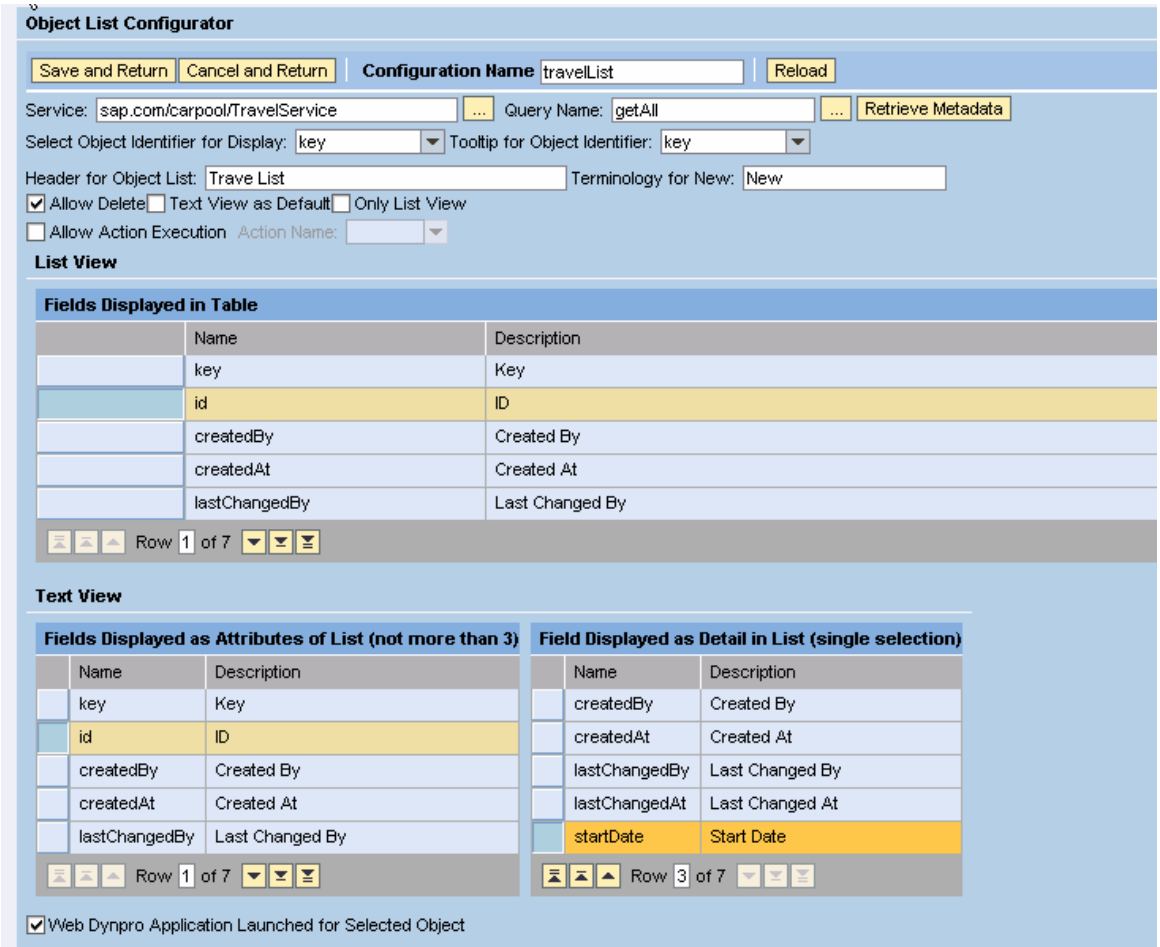
Create an Object Selector UI pattern with the settings given below.

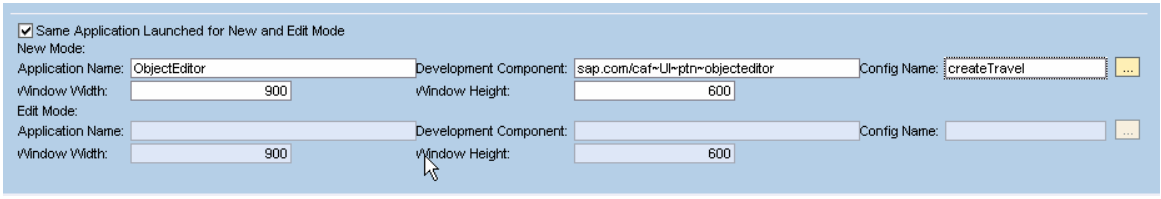


Configure the Search Bar as given below.



Configure the Object List as shown below. Note that the Object Editor pattern createTravel is used for New/Edit mode.



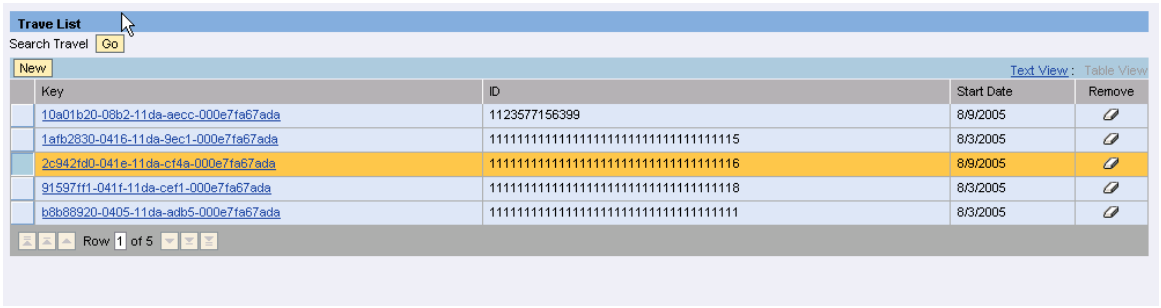


Save the Object Selector pattern and preview.

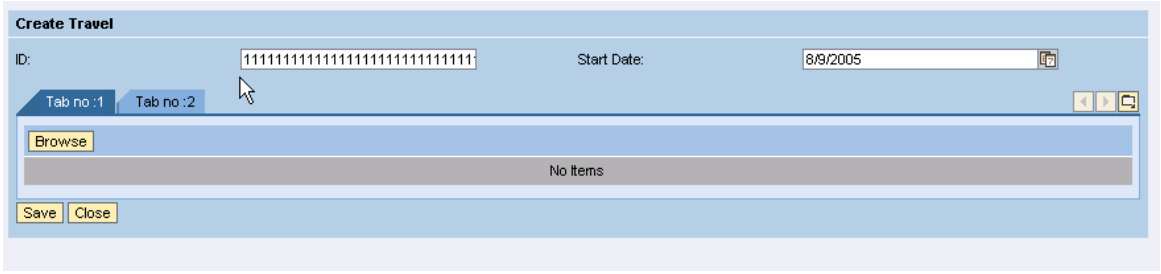
2) Launch the Authorization tool using the URL http://<was_host>:<was_port>/webdynpro/dispatcher/sap.com/caf~UI~ptn~authorization/Authorization

Create a new Business Rule to assign *fullcontrol* rights for the User for the entity *Travel*

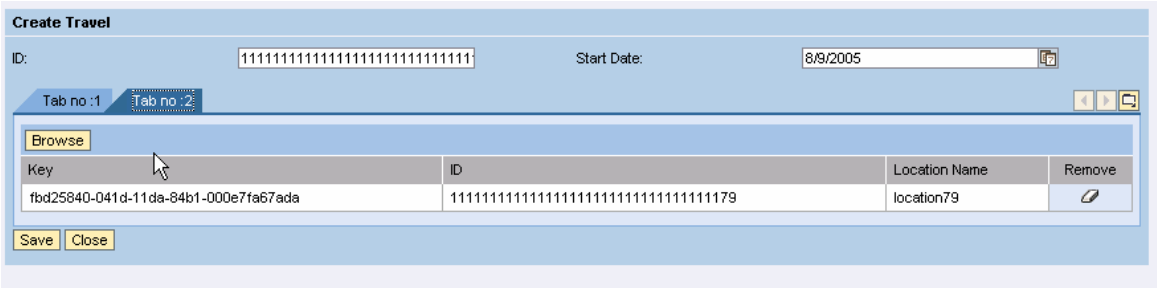
3) Preview the ObjectSelector pattern travelSelector. Click on the Go button. The existing Travels will be listed. (You should have created some Travels already).



Click on any link. The Object Editor for the selected Travel will be launched.

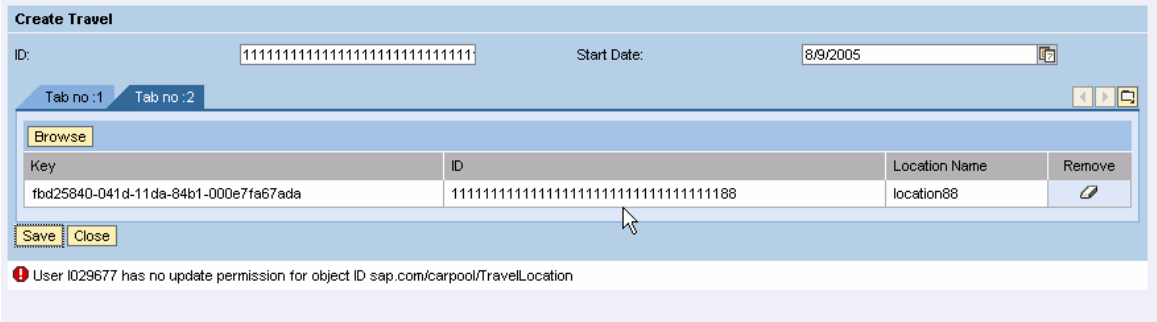


In the relations tab, browse and select a Travel Location.



Now edit the ID and the Location Name and click on the Save button.

You will get an error message saying the user does not have update permission.

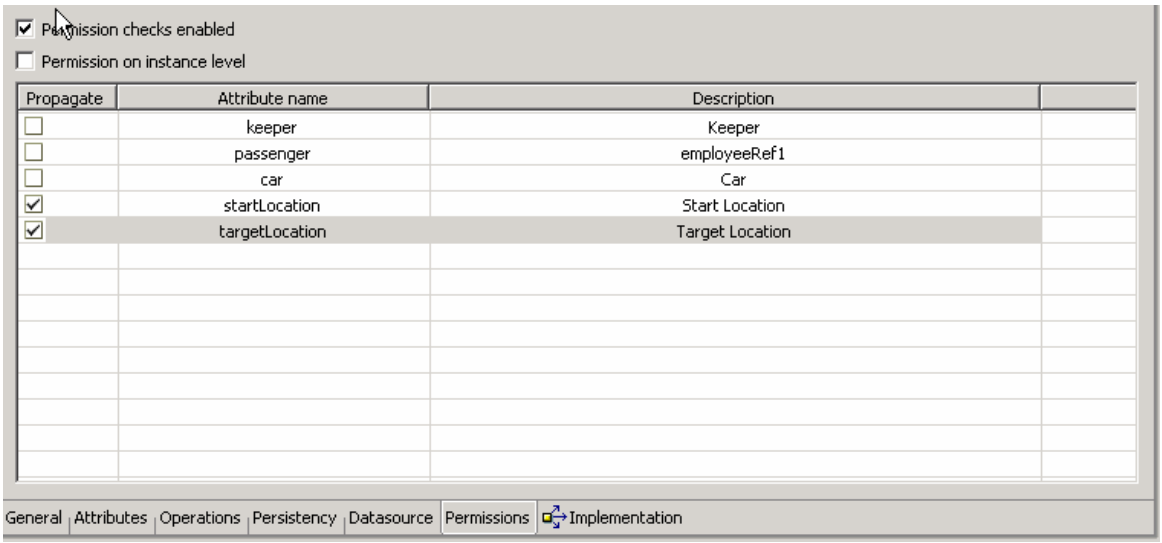


This is because you have given full control rights only to the Travel entity.

4) Now propagate the rights from *Travel* to *Travel Location*.

Open the *Travel* entity in the IDE. Switch to the Permissions tab. Select the checkbox *Permission checks enabled*. You will see a list of entities related to the *Travel* entity

Select the entries startLocation and targetLocation from the list.



5) Save, Generate Code, Build and Deploy the project.

6) Now you have to test the implications. Repeat step 3 and test Object Selector pattern.

On click of Save, you will receive a success message.

Both the entities –*Travel* and *Travel Location* will be updated. Though you have given authorization only for the *Travel* entity, you'll still be able to update the *Travel Location* entity. This is because you have defined propagation of permissions from *Travel* to its aspects startLocation and travelLocation.

www.sap.com/netweaver