SAP Composite Application Framework

Creating a Callable Object: Composite Application Web Dynpro Component

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

Business Scenario

The integration of CAF Core and Guided Procedures enables you quickly to develop a complete workflow dealing with and maintaining user defined data objects without writing a single line of code.

In this example the maintenance of the data for a specific employee is embedded into a single workflow step. This step could, for example, be executed as part of a workflow for new employees. Once the data for a new employee has been entered by the human resources department, a workflow could be necessary for each new employee to confirm the data entered.

About This Document

This document shows you how to integrate a *Property Editor UI Pattern* into a new *Composite Application Web Dynpro Component* type Callable Object.

General Prerequisites

- You should either have completed the How-to on “Configuring the Property Editor UI Pattern” (which you can find within the RKT CAF-Core learning map), or you should at least have another configuration for the *Property Editor UI Pattern* available on the same J2EE Engine on which your GP framework resides.

- You need to have finished the CAF Core RKT-Material; otherwise you will not be able to complete this How-to, because the Entity Service will be missing.

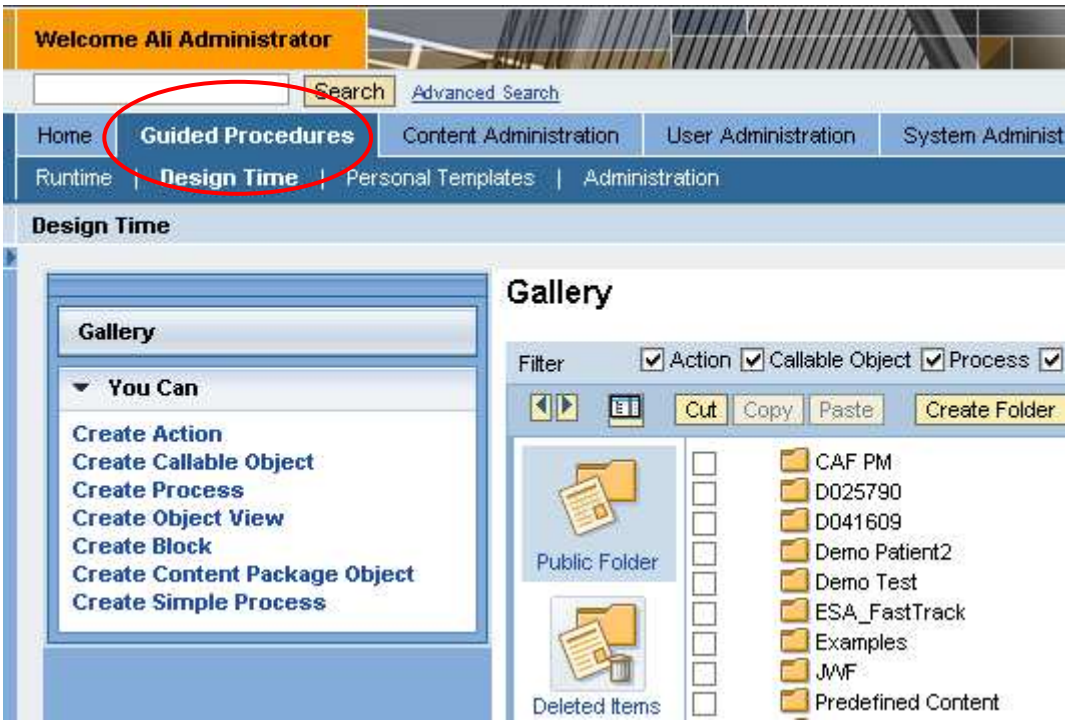
Applicable Release

This tutorial is compatible with the following release “Beginning with SAP NetWeaver ’04s SPS06”.

Disclaimer

The Step By Step Solution

To open your Enterprise Portal, enter **`http://<Server>:<Port>/irj/portal`**. Navigate to tab *Guided Procedures* and choose *Design Time*.



Approaches to Building Up Our Model

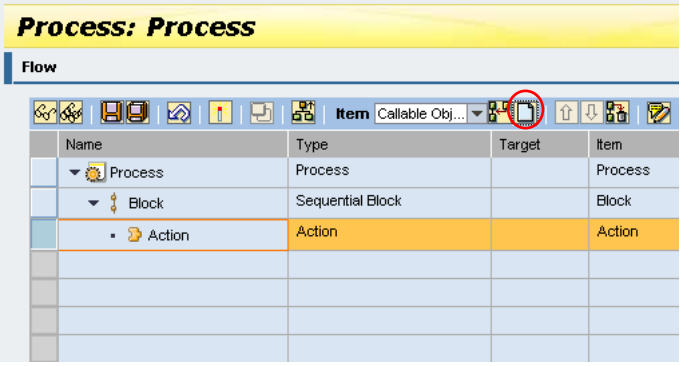
You are free to choose either the top-down approach and create the process, block and action and then insert a new callable object; or you can start with the callable object first without having any other model item.

Top-Down Approach

In this case you first create all design time objects (process, block, and action) that will make use of this callable object.

For further details on this top-down approach see **Error! Reference source not found.**.

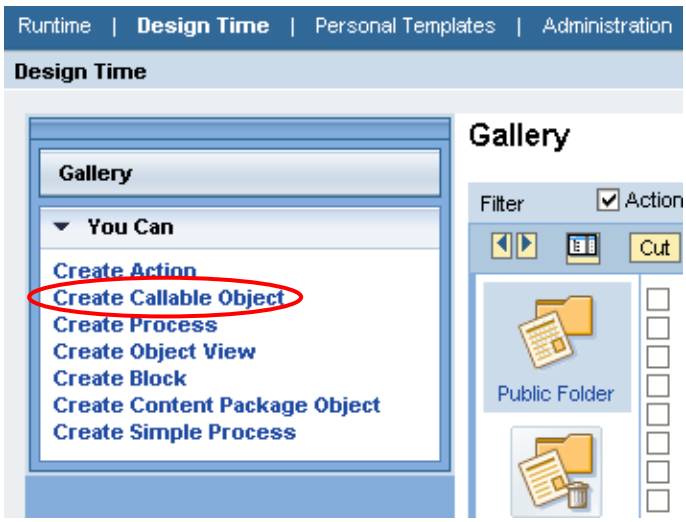
1. Choose *(Create New)* to embed a new callable object into our existing *Action.*



Bottom-Up Approach

In this case you create the callable object first and insert it in the actions and blocks you create later.

1. In the contextual panel *(You Can)*, choose *Create Callable Object* to open the Callable Object design time.



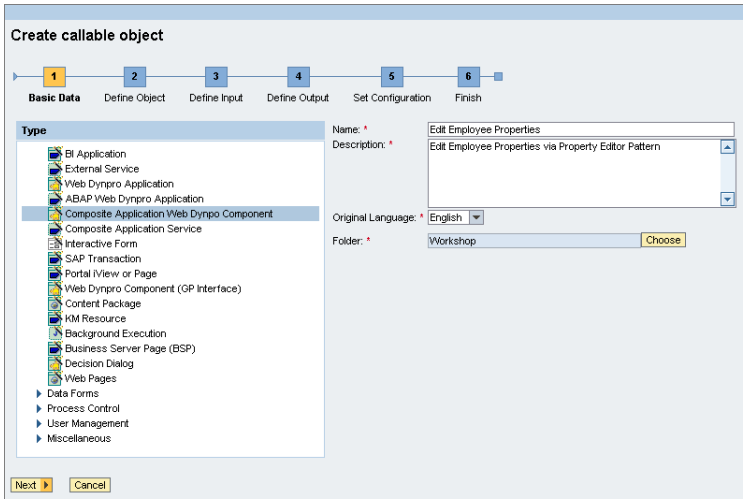
How to Create the Callable Object

2. Select *Composite Application Web Dynpro Component* from the *Type* list.

Enter the following data:

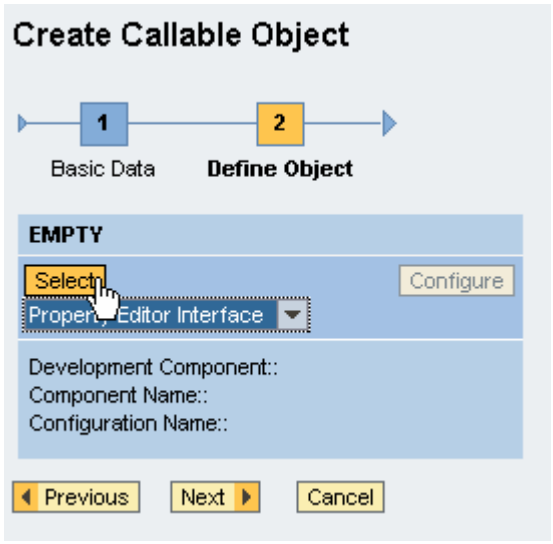
- *Name*, for example **`Edit Employee Properties`**
- *Description*
- *Language*
- Choose a location for your callable object *(Folder)*

Choose *Next*.



3. From the drop-down box on the *Define Object* screen, select *Property Editor Interface*.

Choose *Select*.



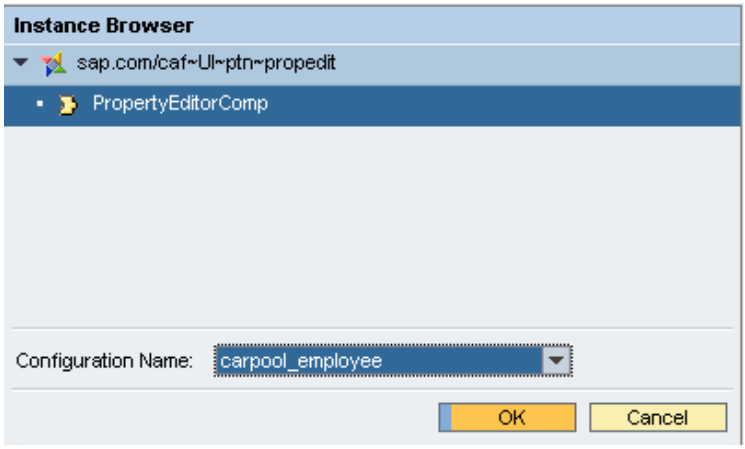
4. In the pop-up window:

a. Expand the DC node `sap.com/caf~UI~ptn~propedit`

b. Select the component name *PropertyEditorComp*.

c. Select the configuration name *carpool_employee* or any other available configuration.

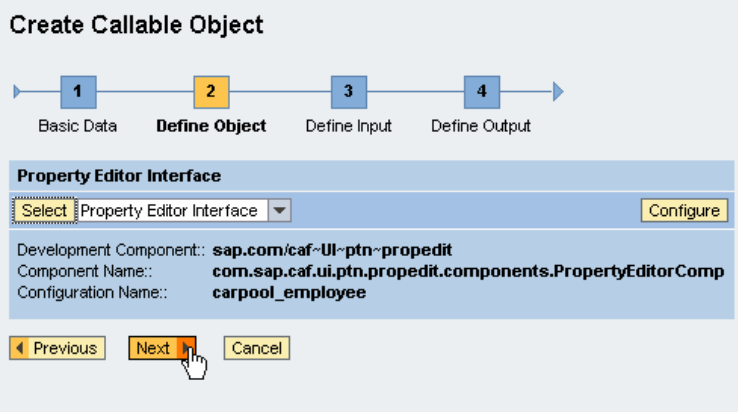
If the configuration name is missing, you need to complete



the CAF core RKT material first.

d. Choose *OK* to close the pop-up.

5. Choose *Next*.

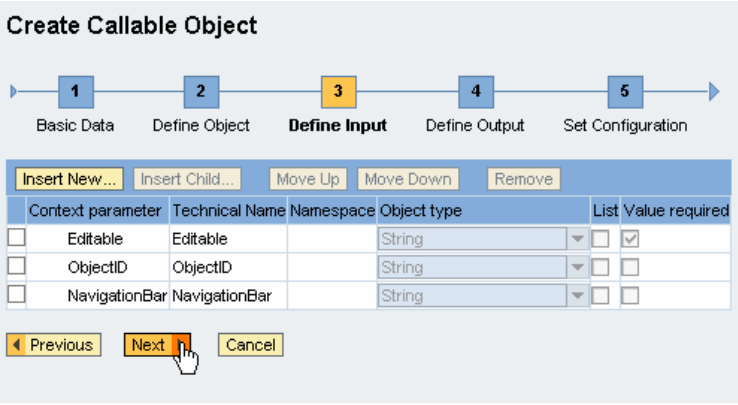


6. In the next steps you do not need to change anything.

All the standard input parameters for the *Property Editor UI Pattern* are listed.

Here we could add further parameters, but as the *Property Editor UI Pattern* would not use them anyway we continue.

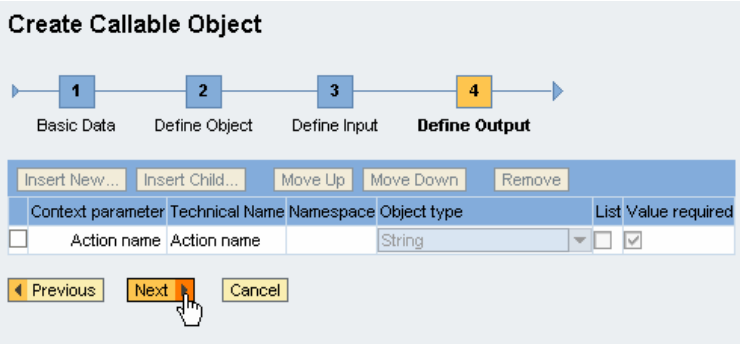
Leave all the input and output parameters for *the Property Editor Pattern* available and choose *Next*.



7. In the next step the output parameters are listed.

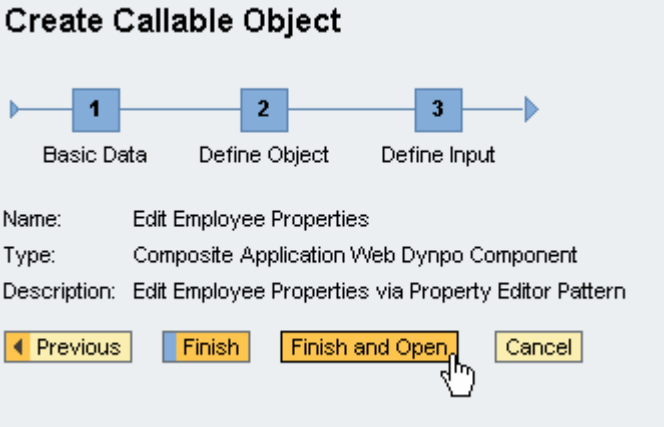
Choose *Next.*

To skip the following step that is not required, choose *Next* again.



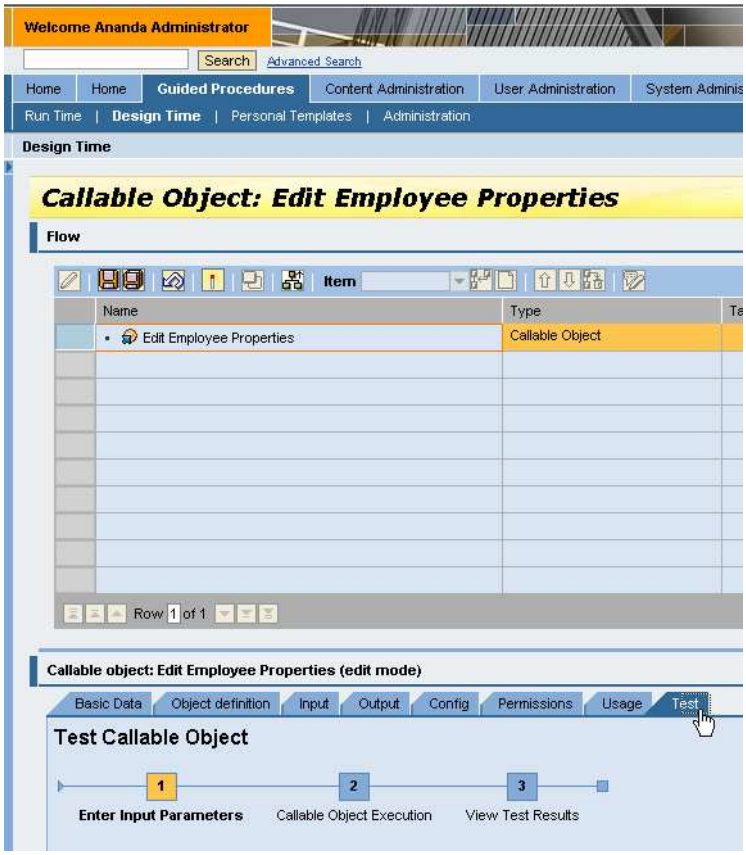
8. Finally choose *Finish and Open*.

If you started the example using the top-down approach, choose *Finish*.



How to Test Your Callable Object (Optional)

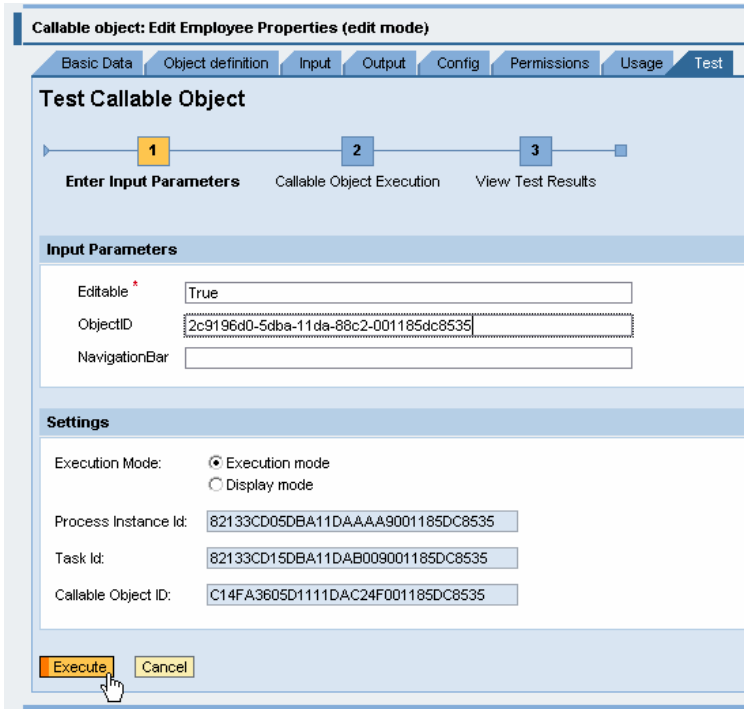
1. You have created a callable object to configure the *Property Editor Pattern*.
   You can now test the callable object from the bottom of the *Test* tab screen.



2. The only input parameter that is mandatory is "*Editable*". Here you can define whether the data shown by the *Property Editor UI Pattern* should be editable. Enter **`True`**.

   If you know the internal key of an entity service that is shown by the *Property Editor Pattern* (in our example the internal key for an employee), then you can enter this key in the *ObjectID* field. The object ID ensures that only one specific entity will be shown. Otherwise you will be able to browse all entities.

3. Choose *Execute* to execute your callable object for testing.



www.sap.com/netweaver