Creating a Callable Object: Composite Application Service
## 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>EXAMPLE TEXT</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>Example text</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>Example text</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>&lt;Example text&gt;</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>EXAMPLE TEXT</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

Scenario ........................................................................................................................................ 2
Prerequisites .................................................................................................................................. 2
    Approaches to building up our model ...................................................................................... 2
    Disclaimer ................................................................................................................................. 3
    Applicable Releases ................................................................................................................. 3
The Step By Step Solution ............................................................................................................. 4
    Check or create Endpoint Alias .............................................................................................. 4
    Create Callable Object in Guided Procedures Design Time.................................................... 6
Scenario

Guided Procedures (GP) is the business process layer for the Composite Applications Framework (CAF) in SAP NetWeaver. As such, GP enables you to expose CAF applications as callable objects, and use them in processes.

CAF Core Application Services can be implemented rapidly and offer extensive possibilities for accessing or manipulating data of entities modeled in the Composite Application Framework. These capabilities mean that they are a perfect source for the backend access of composite application data in a Guided Procedure.

This tutorial describes how easily a callable object can be implemented that accesses a Composite Application Service.

Prerequisites

This guide is based on the assumption that you already have an implemented Composite Application Service with at least one method available, and that it is deployed on the J2EE Engine.


- The Composite Application Service with at least one method available that is to be integrated into a Process must be deployed on the used J2EE Engine
- It is helpful if an Endpoint Alias for EJB remote calls is configured for the J2EE Engine where this CAF resource is deployed. If it has not yet been done, this tutorial will give details of the necessary steps.

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 create all the Design Time objects (Process, Block, and Action) that will make use of this callable object beforehand.
  For further details on this top-down approach see [2]

- Bottom-up approach
  In this case you create the callable object first and insert it into the embedding Design Time objects later.
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.

Applicable Releases

This tutorial is compatible with the following release” Beginning with SAP NetWeaver ’04s SPS6”.
The Step By Step Solution

In the first optional part of the solution we will check, and if necessary configure an Endpoint Alias as this is necessary for communicating with the Composite Application Service. (An Endpoint Alias is a communication endpoint between your J2EE Engine and the connected SAP Systems).

You can skip this section if you already have an endpoint configured.

The second part will guide you through the steps to implement the callable object that wraps the Composite Application Service thus making it capable for integration.

Check or create Endpoint Alias

First you have to ensure that an Endpoint Alias exists for your server where the CAF Core service that you want to call is available.


   Navigate to the Guided Procedures tab and the Administration tab.

2. Click on the Configure Endpoints link under SAP System.

   Configure Adapters
   Maintain Queues
   Manage Impersonalized Forms

   Configure Endpoint Connection Properties
3. On this page choose the “Endpoint Alias for EJB Remote Call” from the drop-down list as the Endpoint Alias Type and press Go.

4. In the resulting table you see a list of the configured Endpoint Aliases. Here you can Add new aliases, Edit existing aliases, or even Remove them.

Check that the J2EE server on which you deployed the Composite Application Service is listed here.

5. If there is no Endpoint Alias defined for your J2EE server then click on the Add... button.

6. We assume that the Service we want to call is running on the same machine (locally).

7. Enter the Endpoint Alias Name, in our example it is “LocalhostEndpoint”

8. Choose the Endpoint Alias Type: “Endpoint Alias for EJB Remote call”

9. The Remote Provider URL consists of two parts: the server-name (in our case it is “localhost”) and the port, which is derived from the http port of
your server by replacing the last digit with a “4”. Therefore this value is 
<host>:<p4_port> - such as "localhost:53004" for example.

10. Enter the user name for the Security Principal.

11. Enter the password for the Security Credentials.

Finally, click **Save**

12. You should see the “LocalhostEndpoint” Endpoint Alias in the result table

<table>
<thead>
<tr>
<th>Endpoint Alias</th>
<th>Endpoint Alias Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastTrack_FT2</td>
<td>Endpoint Alias for EJB Remote call</td>
</tr>
<tr>
<td>LocalhostEndpoint</td>
<td>Endpoint Alias for EJB Remote call</td>
</tr>
<tr>
<td>localhost</td>
<td>Endpoint Alias for EJB Remote call</td>
</tr>
</tbody>
</table>

**Create Callable Object in Guided Procedures Design Time**

Navigate to tab **Guided Procedures** and to tab **Design Time**.
1. In the left section of the window named ‘You Can’, choose **Create Callable Object** to open the Callable Object Design Time.
2. In the first screen select the type of the callable object: Composite Application Service

3. Enter the basic data for the callable object:
   - Name, e.g. "CO_CompAppService"
   - Description, e.g. “Callable Object for Composite Application Service”
   - Language, e.g. “English”
   - Folder: (if blank or does not contain the correct value, select the storage location by using the Choose button)
4. Click **Next**.

5. On the first create object step screen, choose an Endpoint Alias by clicking on **Choose** next to the relevant field.
6. In the drop-down box for the “Endpoint Alias Type” choose “Endpoint Alias for EJB Remote call”

Click on the Go button to carry out a search.

7. From the result list select the Endpoint Alias (e.g. the “LocalhostEndpoint” created previously).

Click on Choose

8. From the “Deployed Application Services” table select the name of the application, the name of the service and finally the name of the method you would like to call (in columns Application Name, Service Name, Method Name).

Click on Next
9. In this “Define Input” step you can see all the input parameters of the method you have previously chosen. Click on Next without modifying anything.

10. In the “Define Output” screen you find the output parameters for the chosen method. Click on Next in this step as well.

11. In the last step click on Finish and Open.
12. Now you can test your new callable object by selecting the **Test** tab at the bottom of the page.

13. Enter the input values for the method of the Component Application Service exposed by this callable object.

   Click on **Execute**
14. You should see the result status and the output value(s) of the exposed method.

15. You can activate your callable object by clicking on the **Activate** icon at the top of the page or you can do it implicitly by activating the Process in which this service method is used.