Service Composition with SAP Composite Application Framework Capabilities In SAP NetWeaver CE 7.1

Applies to:
SAP NetWeaver Composition Environment 7.1, SR3 – also the SDN downloadable SAP NetWeaver CE 7.1 trial version can be used for working this exercise.

Summary
This tutorial explains the basics of SAP NetWeaver CE’s composite application framework (CAF). It explains how business objects are created, permission handling is executed and application services are modeled with CAF. Additionally you may learn to work with web services and get to know the debugging tool for CAF services.

Authors: Roland Münster, Abdul Razack & Bogdan Vatkov
Company: SAP AG
Created on: October 2007

Author Bio
Roland Muenster
is a member of the Regional Implementation Group (RIG) EMEA team of SAP NetWeaver. He is responsible for Java, Java EE, and related topics. Furthermore, he is a contact for technical questions in the area of SAP NW CE. Prior to joining the SAP NW RIG team, Roland was technical consultant and developer in different SAP areas for seven years. He also was project lead for SAP's Java training curriculum.

Abdul Razack
is a member of SAP NetWeaver Regional Implementation Group US responsible for implementation activities, knowledge transfer, and feedback to development. His career spans 14 years in various areas of Information Technology including product development, integration, solution definition, solution architect, and pre-sales activities.

Bogdan Vatkov
is a development architect with the SAP NetWeaver Composition Tools. He is focused in the areas of Model-Driven Architecture, Service-Oriented Architecture, Service Adaptation and Composition. He joined SAP as a software developer in the SAP NetWeaver Composite Application Framework area in 2004.
# Table of Contents

Composite Application Development With SAP NetWeaver ................................................................. 2

Introduction to the exercise .......................................................................................................................... 2

Prerequisites .................................................................................................................................................. 2

Exercise 1: Model Business Objects ........................................................................................................ 3

Building the Eclipse Project .......................................................................................................................... 3

Building Business Object “Project” .............................................................................................................. 7

Testing the Business Object .......................................................................................................................... 11

Building Business Object “ProjectTask”. Defining relationship of type Composition .................................. 13

Testing the Business Objects ..................................................................................................................... 15

Exercise 2: Model and test Permission checks .......................................................................................... 17

Model Permission Checks ............................................................................................................................ 17

Testing the Business Objects ..................................................................................................................... 18

Configuring access rights with the Authorization Tool ............................................................................. 19

Exercise 3: Model an Application Service .................................................................................................. 21

Defining the Application Service (Usage of AS as façade) ....................................................................... 22

Testing the Application Service .................................................................................................................. 27

Updating the Application Service (Use AS for service composition) ....................................................... 28

Enhance the Application Service .............................................................................................................. 32

Configuring the external service call ......................................................................................................... 40

Exercise 4: Web Service Enablement for an Application Service ............................................................. 42

Expose Application Service as Web Service ............................................................................................... 42

Test the newly created Web Service ............................................................................................................ 43

Exercise 5: Debugging CAF Services ........................................................................................................ 45

Configure J2EE Engine ............................................................................................................................... 45

Configure NetWeaver Studio ........................................................................................................................ 46

Launch Debug Session ............................................................................................................................... 48

Appendix ...................................................................................................................................................... 52

Setting up the environment .......................................................................................................................... 52

Install sample application’s “Education Data Model” ................................................................................. 54

Modeling remote persistency for Business Object “Project” ...................................................................... 55

Copyright ..................................................................................................................................................... 62
Composite Application Development With SAP NetWeaver

Composite applications empower companies to drive innovative business processes, leveraging existing IT investments. SAP NetWeaver Composition Environment provides a methodology and toolset to efficiently develop and manage composite applications, following SAP’s Enterprise SOA principle. SAP NetWeaver Composition Environment combines infrastructure components such as SAP Web Application Server, Web Dynpro-based user interfaces, business process management for workflow support, and an infrastructure for service and business object definition.

Introduction to the exercise

This document describes the basics of composite application framework.

- We will start with the creation of a business object, here “Project” and “Project Task”, the recording Model a Business Object with CAF (19 min) shows this exercise in movie format.
- In the second part you’ll learn how these business objects can be handled with permissions. See screencam Design authority concept for Business Objects (5 min).
- The third part explains what the purpose of CAF application services are, how they are modeled and how they can be applied. Recording Create service with own business logic (19 min) explains this feature more detailed. The screencam Configure and test the modeled services (7 min) shows the configuration and test of CAF application services.
- Within CAF, it’s very easy to consume and create Web Services: chapter four will explain this valuable feature. (No recording available.)
- And finally the tutorial depicts the debug features of CAF (no recording).

Prerequisites

You need an installed and configured SAP NetWeaver CE 7.1 SR3 system. You can get it via download from SDN as a 90 days trial version. Additionally you need to install the SAP NetWeaver Developer Studio. This piece of software can also be gotten from SDN download page (i.e. it’s part of the SAP NetWeaver CE 7.1 installation). If you need to work with RFC/BAPI function modules, a SAP ERP backend system is needed – however this is not coercively necessary.

Some exercises make use of service from the so-called sample application EDM (education data model). Please find the explanation how these services can be made ready to be used in the appendix.

You can get the solution for this exercise here:
Exercise 1: Model Business Objects

In this exercise we will build the Business Objects (or Entities) Project and ProjectTask which will have a relationship of type Composition to each other. This means that once the Project gets deleted the ProjectTasks will be deleted as well. The Entities are built using the SAP NetWeaver Developer Studio and the Composite Application Service perspective. After finishing this exercise you know how to model relationships between business objects and test them.

See also recording Model a Business Object with CAF (19 min).

Building the Eclipse Project
1. Create a new project by choosing File->New->Project from the menu bar.

2. Choose Development Infrastructure/Development Component for the type of project, then click Next.
3. In the next screen choose Composite Application as type for Development Component.

4. The development component should be defined in the Local Development branch since we are not using NWDI. Expand the Local Development node and choose My Components. Click Next.
5. Choose and enter an appropriate name for the new component, click Next and click Finish.
Building Business Object “Project”

6. In the Service Explorer, right click on node modeled and choose New Business Object from the context menu.

7. Enter the name of the Business Object: “Project”

8. In the radio button group choose Create new structure and click Finish

9. After the Business Object Node metadata editor opens on the right hand pane, navigate to tab Structure and press button Edit Main Structure to add some additional data fields to the Business Object Node.
10. In the column Existing Types, expand the type tree caf.core/primitive.

   Add the following attributes by highlighting the STRING type and then clicking the Add attribute icon in the middle column:

   Fieldname      Type
   title          String
   description    String
   schedule       String
   firstname      String
   lastname       String
   comment        String

   In the right column change the names of the attributes as shown on the picture.

11. Save the application and close the Project structure editor
12. In the BO metadata editor choose tab Operations. Highlight method create and have a look at the assigned input-, output-, and fault parameters.

As you can see the create method does not offer input parameters so far.

13. To add required input parameters, navigate back to tab Structure, press button Edit Main Structure again, and change the cardinalities of all attributes you added in the prior step to 1..1 (Mandatory) by double-clicking into the appropriate column.

14. Save the application and close the Project structure editor.
15. In the BO metadata editor choose tab Operations again.

**Add a new Operation to your Business Object Node.**
Press button Add on the right hand side. In the wizard that appears enter the operation name “findByKey”.

Make sure to switch on flag Select of parameter key (see screenshot).

**Hint:** In the background a new (finder-)method is generated that allows searching for Project Business Objects with a given key-parameter.

Press button Finish afterwards.

16. Switch to the Permissions tab and make sure that the checkbox Permission checks enabled is not checked:

![Permission check disabled]

**Use Local Persistency**

17. Navigate to tab Persistency. Make sure that the dropdown field Backend is set to Local.

18. Save the application.

**Use Remote Persistency**

In the persistency tab page you can select if the Business Object is persisted locally or if you want to use remote persistency. Refer to the appropriate chapter in the Appendix to see, how you can model Remote Persistency for your Business Object.
Generate, Build, and Deploy application

19. In the Composite Application Explorer highlight the project node with the project name (here: “ce253ex”) and choose (in sequence):
   - Generate all project code
   - Build application
   - Deploy application

20. For Deployment, you may need to login with administrator credentials.

Testing the Business Object

21. In the SAP NetWeaver developer studio right click on Business Object Node Project and select Test Service (you may need to login with administrator credentials)
22. The service browser comes up. On the left side drill down to node Project and select it.

![Service Composition With SAP Composite Application Framework Capabilities In SAP NetWeaver CE 7.1](image1)

23. In the right hand pane click on New. The first row of the table underneath will be highlighted. The row is ready for data entry now.

![Data Component](image2)

24. Enter some test data for the order. You only have to fill in the fields “title”, “description”, “schedule”, “firstname”, “lastname”, and “comment”. Press button Save afterwards. The remaining fields will be filled automatically.

25. Enter some additional data as shown in the prior step. Select one of the rows, highlight the key field, and copy it to your clipboard.

26. Select operation findByKey on the left hand pane, paste the key value from the clipboard into the appropriate entry field, and press Execute Query button. The result should be displayed in the result table.
Building Business Object “ProjectTask”. Defining relationship of type Composition.

1. In order to create relationship of type composition in CAF you have to create a new Business Object Node within an existing Business Object Node.

   In SAP NetWeaver Developer Studio open node Project and choose New BO Node from the context menu.

An association models relationships between objects of same-level classes

An aggregation is a special case of association. It is used to represent ownership or a whole/part relationship

A composition is a strong kind of aggregation in which if the aggregate is destroyed, then the parts are destroyed as well (In aggregation, this is not necessarily true)
2. In the wizard that appears, enter “Task” as name, select Create new structure, and press button Finish.

3. After the Business Object Node metadata editor opens on the right hand pane, navigate to tab Structure and press button Edit Main Structure to add some additional data fields to the Business Object Node.

<table>
<thead>
<tr>
<th>Fieldname</th>
<th>Type</th>
<th>Cardinal</th>
</tr>
</thead>
<tbody>
<tr>
<td>taskname</td>
<td>String</td>
<td>1:1</td>
</tr>
<tr>
<td>taskowner</td>
<td>String</td>
<td>1:1</td>
</tr>
</tbody>
</table>

Save the application and close the Project structure editor.

4. Switch to tab Permission and remove permission checks for the BO Task by switching off the Permission Checks enabled checkbox.
5. In the Composite Application Explorer highlight node Project and navigate to tab Associations.

Make sure that BO Project contains an entry for BO Task.

6. Change the cardinality to NONE_TO_MANY.

Generate, Build, and Deploy application (See steps 19-20)

Testing the Business Objects

7. In the SAP NetWeaver developer studio right click on Business Object Node Project and select Test Service

8. The service browser comes up. On the left side drill down to node Project and select it.

Add some data for Business Object Node project as shown in prior steps 24 and 25

9. Underneath the Project table you will find more input fields

Press button Create Related Row. The first row of the according table is now highlighted and ready for data input

10. Enter some test data for
the Project Task. You only have to fill the fields shown right.

11. Add a second line item as depicted right.

12. Click on the Save button to save the Project together with its Tasks. Automatically keys for the Project as well as for the Tasks will be generated and displayed.

13. Open node Tasks and choose findAll. In the appropriate table you will find all Tasks you entered in the prior step.

14. Now open node Project and choose findAll.

Delete one of the Projects by pressing the Delete button and Save button in sequence.

This will also delete the associated Tasks Business Objects.

You can check this by open node Tasks again and choosing findAll. In the appropriate table you will see that the appropriate Task BO’s have been deleted due to the Composition relationship.
Exercise 2: Model and test Permission checks

In this exercise you will learn how to model and test permission checks for your entities. See screencam Design authority concept for Business Objects (5 min).

Model Permission Checks

1. In the SAP NetWeaver Developer Studio select the Project entity of the last exercise (double-click).

2. Click on the Permissions tab at the bottom of the middle pane.

3. Switch on the Permissions checks enabled checkbox and save the metadata.

Generate, Build, and Deploy application (See steps 19-20)
Testing the Business Objects

4. In the SAP NetWeaver developer studio right click on Business Object Node Project and select Test Service.

5. The service browser comes up. On the left side drill down to node Project and select it.

6. In the right hand pane click on New. The first row of the table underneath will be highlighted. The row is ready for data entry now.

7. Enter some test data for the order.

8. Click on the Save button to save the Project.

9. An error pops up due to permission conflicts!
Configuring access rights with the Authorization Tool

10. Open Internet Explorer and enter the following URL:
    http://<hostname>:<port>/caf
    The CAF Runtime Configurator comes up.

11. Click on Administrative Tools, then on Authorization Assignment.
12. Within the authorization assignment tool select node x253ce modeled Project.

13. In the right hand pane click on the button New Rule.

14. Automatically a business rule will be generated with some default values set, e.g. the current user will automatically be assigned to the Access Control List of the current entity with permissions set to fullcontrol (see screenshot).

15. As the default values fit our requirements for this exercise, simply click on Save Business Rule.

16. With the service browser create once again a new Project. This time the Save operation will work and a key for the new Project created.
Exercise 3: Model an Application Service

In this exercise it will be shown how you can create an Application Service. Application Services implement the business logic of a CAF application and expose the functionality of the application to other systems or applications. An Application Service can be used in two directions:

On the one side an AS acts as a façade for a given Business Object providing several business methods to add business logic and manipulate the underlying BO’s.

On the other side you can use Application Services for service composition. In that case the Application Service consumes one or more predefined (Web) Service (Service Orchestration), enhances their functionality by adding additional code, and exposes this new functionality externally as a new operation.

See recordings Create service with own business logic (19 min) and Configure and test the modeled services (7 min).

- **General**
  Defines general properties for the Application Service.

- **Dependency**
  Defines the Entities and External Services that are used by the Application Service.

- **Operations**
  Defines the Interfaces of Operations implemented by the Application Service including life-cycle and custom methods

- **Datasource**

- **Implementation**
  Defines the Implementation of the Application Service. Java coding is entered here to implement each operation defined in the Operations tab.
Defining the Application Service (Usage of AS as façade).

Assume you want

- to hide the complexity of your business model to your clients
- to add business logic on top of your business model, e.g. you want to display all Tasks objects to a given Project:

```
ProjectService

Project
          1
  ----> Task
       n
```

This is where you can define an Application Service that acts as a facade to your business model

- Define BO Project and Task (described in former chapter)
- Define Application Service ProjectService. Add dependencies to BO Nodes Project and Task (tab dependency)
- Within AS, define a method “findTasksByProjectID” (tab operations)
- Within AS, implement method (tab implementation)

1. In the NetWeaver Developer Studio right-click on the modeled node and select New Application Service from the context menu.
2. Enter “ProjectService” as the name for the application service.

3. In the Metadata editor on the right hand pane navigate to tab Dependency.

   In the left column open the tree that displays the existing BO’s and highlight BO Project.

4. Click on the Add button in the middle.

5. Repeat prior steps for BO Task.

Add a new Operation “createProject”

6. Navigate to tab Operations to add an additional operation.

   Press the Add button in the right column.

   Enter “createProject” as name for the operation and press Finish.
7. Highlight the new entry in the list on the top and create some Input Parameters in the panel below.

8. Add input parameters. In the left side of the Operations Panel, navigate to [LocalDevelopment]caf.core->primitive->Data types-> Simple Types. Highlight the data type STRING and press the Add to Input button. Change the name of the parameter to “title” afterwards.

9. Repeat the prior step to add the following parameters

<table>
<thead>
<tr>
<th>Fieldname</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>task</td>
<td>String</td>
</tr>
<tr>
<td>owner</td>
<td>String</td>
</tr>
</tbody>
</table>

**Hint:** For simplicity reason we add only some of the needed parameters to create the Business Object Nodes.

10. Add an output parameter of type Id. You will find it in the path services->Data Types-> Simple Types->Id.

11. Finally ad a Fault type CAFServiceException. You will find it in the path faults->Data Types->Faults.

12. Save the application
Add a new Operation “AddTaskToProject”

13. In tab Operations press the Add button in the right column again.

Enter “addTaskToProject” as name for the operation and press Finish.

14. Highlight the new entry in the list on the top and create some Input Parameters in the panel below.

15. Add an input parameter of type Id. You will find it in the path services->Data Types-> Simple Types->Id.
Change the name of the parameter to “projectID” afterwards.

Repeat the prior step to add the following input parameters

<table>
<thead>
<tr>
<th>Fieldname</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>task</td>
<td>String</td>
</tr>
<tr>
<td>owner</td>
<td>String</td>
</tr>
</tbody>
</table>

16. Add an output parameter of type Id. You will find it in the path services->Data Types-> Simple Types->Id.

17. Finally add a Fault type CAFServiceException. You will find it in the path “faults > Data Types > Faults”.

Hint: For simplicity reason you will only add some of the needed parameters to create the Business Object Nodes.
18. Save the application

Implement the Operations

19. Before you can implement the methods, you must save the project and generate code to ensure that the model is in synch with the Java implementation.

Switch to Implementation tab and click on the Java implementation code (“ProjectServiceBeanImpl.java”)

Hint: The following code snippets are already provided for you in your local workspace. If you do not want to type all the necessary code, you can open the File Explorer, navigate to your local workspace, open the appropriate text file and copy and paste the code snippets.

Navigate to method “addTaskToProject” and insert the following code:

```java
@com.sap.caf.dt.CAFOperation(name = “addTaskToProject”)
public java.lang.String addTaskToProject(
    java.lang.String projectID, java.lang.String task, java.lang.String owner)
    throws com.sap.caf.rt.exception.CAFServiceException {

    TaskServiceLocal taskService = this.getTaskService();
    Task newTask = taskService.create(task, owner);
    // persist new task BO
    taskService.update(newTask);
    // establish relationship of new Task entry (child)
    // to project (parent)
    this.getProjectService().addTask(projectID, newTask.getKey());
    return newTask.getKey();
}
```

If there is an error displayed because of not imported data types correct this by organizing imports. Right click in the source window and choose Source->Organize Import” (Keystroke combination: Ctrl+Shift+o). You may need to do this several times while entering the code.
Navigate to method “createProject” and insert the following code:

```java
@com.sap.caf.dt.CAFOperation(name = "createProject")
public java.lang.String createProject(java.lang.String title, java.lang.String task,
java.lang.String owner) throws com.sap.caf.rt.exception.CAFServiceException {

    ProjectServiceLocal projectService = this.getProjectService();
    Project project =
        projectService.create(title, "description", "tbd", "tbd", "tbd", title);

    // persist new project BO
    projectService.update(project);

    // add task to project
    this.addTaskToProject(project.getKey(), task, owner);

    return project.getKey();
}
```

Organize imports to get rid of any errors.

Generate, Build, and Deploy application (See steps 19-20)

Testing the Application Service

1. In the SAP NetWeaver developer studio right click on Application Service ProjectService and select Test Service

2. The service browser comes up. On the left side drill down to node Result_createProject / createProject and select it.

3. Enter some input data for the new Project/Task and press buttons Execute Query and Save.
4. Navigate to node Project / Find all and select it.
   The new Project and the assigned Tasks should appear in the list.

Updating the Application Service (Use AS for service composition).

In this exercise we show you how you can use Application Services for service composition. This option gives you the chance to enhance the functionality of the underlying WS. For example you can add some additional business logic before you call the WS or after you received return parameters from it and/or you can call several WS and orchestrate them into a new operation.

- Provide an means to access existing business functionality in systems already deployed in the landscape

- Two types of External Services are currently supported:
  - Web Services
    - Import Web Services using WSDL
    - Allows import via HTTP (URL), File, or UDDI
  - RFC / BAPI
    - Provides direct access to SAP R/3 & ERP systems utilizing the RFC protocol

- Are imported via the External Service Modeler
1. In this exercise you will use a predefined web service that has already been deployed to your J2EE server. (Hint: The Web Service is part of SAP’s reference application “Education Data Model”. See Appendix for more information).

Before you can use it you have to check if the web Service is available on your system.

Open a browser window and navigate to the NetWeaver Administrator.

http://<hostname>:<port>/nwa

2. Navigate to Operation Management->Systems-> Start & Stop

3. Choose Java EE Applications on the left hand panel.

   In the Application List enter “jee” in the entry field for filter criteria and press ENTER key.

   An entry should appear as displayed on the picture on the right.

   Select “xapps~ra.edm.prmgmt.jee.app” and make sure that the status in the table below is set to “STARTED”

**Hint:** If you do not find an entry in the Application List you have to download and configure the EDM model before you can proceed with the following exercise. Please find the description how to configure the EDM in the APPENDIX of this document.
If the EDM Application runs properly you can test the Web Service.

Open a browser window and navigate to the Web Service Navigator:
http://<hostname>:<port>/wsnavigator

In the entry field for filter criteria enter “jee” and press ENTER key. An entry “…ProjectDataService Bean” should appear as displayed on the picture on the right.

Click on the entry to navigate to the available methods of the Web Service.

In the next screen click on entry getAllEmployees.

NOTE: Copy the URL of the WSDL to your clipboard (You need this URL in a later step)

Press the Execute button to test the Web Service.

The result will be displayed on the next screen.
Now that you successfully tested the Web Service you can proceed with enhancing the Application Service
Enhance the Application Service

1. **Import External Service**

2. **Choose Type**

3. **Enter URL for WSDL file**

As you have seen in prior step the underlying Web Service `getAllEmployees` returns all employees that are defined in the Education Data Model. Assume you want to offer a similar operation in your Application Service that displays employees of a certain department. Here is what you have to do:

- Import the Web Service you want to re-use
- Generate an appropriate operation in the Application Service
- Add an additional method in the Application Service where you add some business logic before you call the Web Service
1. In the NetWeaver Developer Studio right-click on the external node and select Import Web Service from the context menu.

![Composite Application Explorer](image)

2. In the wizard that appears choose Remote location/File System and click on Next.

![WSDL URL access](image)

3. In the next screen paste the URL of the Web Service (see Step 85) from the clipboard to the input field.

Note: In the URL change the server name to “localhost”.

Click on Finish.
4. Successful result: The imported operations should now be visible in the Service Explorer.

5. Generate methods for CAF Application Service.

   You now need to access one of these imported web services from your application service. In order to do this, you need to map the methods into the proper format.

   Right click on the imported external service and choose Default mappings from the context menu.
6. Select method `getAllEmployees`, choose the Create methods in existing Application Service, navigate to the Application Service `ProjectService`, and then click OK

The mapped method should now be available in your application service (See picture on the right)

7. Open the Application service by double clicking on the `ProjectService` node. Switch to the Operations tab.

Notice that flag Implemented for method `getAllEmployees` is switched off.

This means that the method is mapped to an external service. This was done automatically when you generated the mappings.
8. Switch to tab Datasource. Notice that the mapping was done automatically for you.

9. Switch back to tab Operations. Add a new method by clicking on the Add button on the right.

   Enter “getEmployeeByDept” as name and click on Finish.
10. Highlight the new entry in the list on the top and create some Input Parameters in the panel below.

11. Add an input parameter of type Id. You will find it in the path services->Data Types->Simple Types->Id.

   Change the name of the parameter to “departmentID” afterwards.

12. Add an output parameter of type GetAllEmployeesResponse.

   The path to find this type is displayed on the picture on the right.

13. Finally add a Fault type CAFServiceException. You will find it in the path faults->Data Types->Faults.

   The structure for the Input-, Output-, and Fault-parameters should look like as displayed on the picture on the right.

14. Save the application
15. Before you can implement the methods, you must save the project and generate code to ensure that the model is in synch with the Java implementation.

Switch to Implementation tab and click on the Java implementation code ("ProjectServiceBeanImpl.java")
Navigate to method "getEmployeeByDept" and insert the following code:

```java
@com.sap.caf.dt.CAFOperation(name = "getEmployeeByDept")
public com.sap.nwce.ra.edm.ws.services.GetAllEmployeesResponseType
   getEmployeeByDept(java.lang.String departmentID)
   throws com.sap.caf.rt.exception.CAFServiceException {

   // create response object
   GetAllEmployeesResponseType response = new GetAllEmployeesResponseType();

   // call Web Service
   GetAllEmployeesResponseType allEmployeesOut =
      this.getAllEmployees(new GetAllEmployeesType());

   // iterate over the result and filter the list
   List<EmployeeBeanType> allEmployees = allEmployeesOut.getReturn();
   for (EmployeeBeanType type : allEmployees) {
      if(type.getEmployeeDepartment().equals(departmentID)) {
         response.getReturn().add(type);
      }
   }

   return response;
}
```

If there is an error displayed because of not imported data types correct this by organizing imports. Right click in the source window and choose Source->Organize Imports (Keystroke combination: Ctrl+Shift+o). You may need to do this several times while entering the code.

**Save, Generate, Build, and Deploy the application**

**Hint:** You will proceed with configuring the application. Testing will be done after the next exercise.
Configuring the external service call

Defining the Web Service Destination for the EDM web service

1. In a web browser, navigate to http://<server>:<port>/nwa. You may need to login with administrator credentials.

   Navigate to Configuration Management->Infrastructure ->Web Services Configuration (lower right corner)

2. Click on WS Destinations, then Create Destination
3. Select WSDL for Destination Type and choose an appropriate name (here: “CE253_EDM_ProjectData”) as Destination Name. The URL should be entered as the URL to the EDM web service (use the WSNavigator to discover the URL, see Step 81)

Make sure to use “localhost” as server name.

… then click Save

Mapping the CAF Web Service to the correct WS Destination using the External Service Configuration tool

4. In a web browser, navigate to http://<server>:<port>/caf. You may need to login with administrator credentials.

5. Click on Administrative Tools, then click on External Service Configuration

6. Click on Service Registry

Expand the correct destination (here: “demo.sap.com / ce253ex / Web Services”) in the left column.

Click on ProjectDataServiceBean.

Then choose the correct web service destination (here: CE253_EDM_ProjectData) in the right column.

Click Map, and then
Save to save the mapping.

**Hint:** Before can test the Application you will learn how to publish an application service’s method as Web Service in the next exercise.

**Exercise 4: Web Service Enablement for an Application Service**

In this exercise you will learn how to publish an application service’s method as Web Service.

**Expose Application Service as Web Service.**

1. In the NetWeaver Developer Studie right-click on the ProjectService node and select Expose Service as Web Service from the context menu.

2. In the wizard that appears enter an appropriate name (here: “ce253exProjectService”) for the WS, choose getAllEmployees, and getAllEmployeeByDept and press OK button.

Save, Generate, Build, and Deploy application
Test the newly created Web Service

1. In a web browser, navigate to http://<server>:<port>/ws navigator. You may need to login with administrator credentials.

2. Enter a part of the name (here: “ce253” in the input field for filter criteria and press ENTER key.

3. Click on the appearing entry “…_ProjectService”

4. In the next screen choose method getEmployeeByDept
5. In the next screen enter a departmentID in the input field (see table on the right for some valid values) and press Execute.

<table>
<thead>
<tr>
<th>DEPARTMENT_ID</th>
<th>MANAGE...</th>
<th>NAME</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINADM</td>
<td>9</td>
<td>Finance ...</td>
<td>1</td>
</tr>
<tr>
<td>GLOBALIT</td>
<td>10</td>
<td>Global IT</td>
<td>1</td>
</tr>
<tr>
<td>HARDWPROD</td>
<td>10</td>
<td>Hardware...</td>
<td>1</td>
</tr>
<tr>
<td>HR</td>
<td>8</td>
<td>Human ...</td>
<td>1</td>
</tr>
<tr>
<td>MARKETING</td>
<td>5</td>
<td>Marketing</td>
<td>1</td>
</tr>
<tr>
<td>SALESOP</td>
<td>15</td>
<td>Sales ...</td>
<td>1</td>
</tr>
<tr>
<td>STRATDEV</td>
<td>4</td>
<td>Strategic...</td>
<td>1</td>
</tr>
<tr>
<td>STRATSOURCE</td>
<td>9</td>
<td>Strategic...</td>
<td>1</td>
</tr>
<tr>
<td>TECHSUPP</td>
<td>8</td>
<td>Technical...</td>
<td>1</td>
</tr>
</tbody>
</table>

6. Successful result: In the next screen all Employees of a certain department are displayed.
Exercise 5: Debugging CAF Services

In this exercise you will learn how to configure the J2EE Engine and the NetWeaver Studio for debugging CAF services.

Configure J2EE Engine.

The first step is to start the engine in the Debug mode.

1. Open SAP Management Console.

   Now, open the nodes below your system node (in the picture right the system is CE2).

   Below the system node you will find at least three more nodes, one for the database, and two Java Instance nodes.

   Select the Java Instance node where the server process is running in the left panel, and click on node AS Java Process Table as shown in the picture.

2. In the right panel highlight the server node (usually server0) and choose All Tasks->Enable Debugging from the context menu.
Configure NetWeaver Studio.

Now you have to configure the NetWeaver Developer Studio.

3. Open the IDE. Switch to the SAP AS Java Cluster overview. Choose Window->Show View ->Other->Server->SAP AS Java Cluster Overview. Click OK. The SAP Java AS Cluster View will be displayed. Maximize it to view the complete server tree.

4. Now, select the server node (usually server0). Check the state of the engine from the right pane.

5. Note the Debug Port Number (in the picture 50326)

The next step is to set the breakpoints in the code.

6. In this exercise you will set breakpoint in the code of Application Service ProjectService, so that when you test the service from the Web Service Navigator execution will stop at the breakpoint.

7. Switch to the Implementation tab of the ProjectService service, and click on ProjectServiceBeanImpl.java

8. Place a breakpoint at the beginning of method getEmployeeByDept
The next step is to launch the application in the debug mode. For this you need to create a Debug configuration.

9. From the menu select Run -> Debug.

10. Select Remote Java Application and click on New button

A new configuration will be created.

11. Enter the connection properties, host and port. Enter the debug port number in the Port field. Click on Apply and then Debug.

This will create the Debug configuration. The Debug views will be displayed on the screen.

(If they don’t appear you can simply select Window -> Open Perspective -> Other-> Debug)
Launch Debug Session.

The next step is to launch the application.

12. Launch the Web Service Navigator by entering the URL http://<server>:<port>/ws navigator. You may need to login with administrator credentials.

13. Enter a part of the name (here: “ce253”) in the input field for filter criteria and press ENTER key.

14. Click on the appearing entry “…_ProjectService”

15. In the next screen choose method getEmployeeByDept
16. In the next screen increase the Timeout, enter a valid input value for departmentID, for example “FINADM”, and press Execute.
17. Switch over to the IDE. The control would have come to the break point.

You can use the following keys to step through the code

<table>
<thead>
<tr>
<th>Key</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>F5:</td>
<td>Step Into Jumps to Next Statement</td>
</tr>
<tr>
<td>F6:</td>
<td>Step Over The next command is executed without jumping to the current statement.</td>
</tr>
<tr>
<td>F7:</td>
<td>Step Return If you previously choose F5, then you can cancel the debugging on the current command by choosing F7</td>
</tr>
<tr>
<td>F8:</td>
<td>Resume The application exits the debug mode and continues with execution</td>
</tr>
</tbody>
</table>

18. At any point of time during debugging, if you want to see the values of the variables, select the variable from the code and click Watch from the context menu.

The value of the variable will appear in the Expressions window.

19. To change the values of the variables at any time, switch to the Variables window of the Debug perspective and choose change value from the context menu.

20. Change the value and click OK.
21. To terminate the debugging, in the Debug view, go to the Context menu of the top node. (e.g. ProjectServiceBeanImpl [Remote Java Application]).

22. Select Terminate. This will terminate the debug session.

23. Again select Remove All Terminated from the context menu of the top node.

This ends the debug session completely.
Appendix

Setting up the environment

Start Developer Studio by double clicking the icon on your desktop.

Developer Studio starts.

Before you start developing with SAP NetWeaver Developer Studio you have to make sure that the system settings (e.g. message server port) are set properly.

Navigate to the Welcome page of your local J2EE engine:

Start->Programs->SAP NetWeaver Composition Environment 7.1 -> Application Server ->Welcome Page

Click on icon System Information.
Be aware of the Message Server Port (for example “3602” in the screenshot on the right).

In SAP NetWeaver Developer Studio, navigate to Window/Preferences and highlight node SAP AS Java.

Make sure the message server host and port are set properly.
Install sample application’s “Education Data Model”

**Step 1: Import the sample application’s sources**

Start NetWeaver Developer Studio.

Open the Welcome Page and press button Project Management and Employee Service.

**Step 2: Importing UME scripts**

To make sample users, groups and roles available, you must import the UME scripts provided with the dictionary project. This must be done once for a server cluster before running the sample application. It cannot be done automatically for security reasons.

- To make sure that the UME scripts are available in the imported projects, open the Navigator view for the dictionary project ra_edm_prjmgmt_dict.
- Expand the resource/ume folder and check whether the three script files user_employee_itelo.dat, group_employee_itelo.dat and role_employee_itelo.dat are located there.
- Then, start your browser with the server start page http://<host>:<port> (“Administrator”, “abc123”)
- Start the User Management application.
- Choose the Batch Import function using the corresponding tab of the application’s window.
- Switch off flag “Overwrite Existing Data”.
- Upload the UME scripts from the dictionary project folder in following order:
  1. user_employee_itelo.dat
  2. group_employee_itelo.dat
  3. role_employee_itelo.dat

**Step 3: Check Preferences of your NetWeaver Developer Studio**

See prior chapter in this Appendix (“Setting up the environment”).

**Step 4: Deploying the Dictionary Tables**

Open the Dictionary perspective

In the Dictionary Explorer, select the project node ra_edm_prjmgmt_dict.

Create an archive for this project and deploy it by choosing a corresponding function from the context menu of the project node.
**Step 5: Deploying the Ear**

Open the J2EE perspective.

Open the Servers view and choose Add and Remove Projects from the context menu of the servers node.

Add the project xapps~ra.edm.prjmgmt.jee.app and start deployment with Finish.

Enter the user and password for deployment - if required.

**Step 6: Running the Application**

Enter the corresponding URL in your browser.

For login you may use any user and the initial password from the user_employee_itelo.dat file mentioned in step 2.

**Modeling remote persistency for Business Object “Project”**

In this exercise you will learn how to connect the Business Object “Project” to a backend system and save the data remotely.

**Preparation: Import external service**

Before you can use remote persistency for you Business Object you have to import an external service (either RFC or Web Service)

**Using RFC**

From the context menu of node external choose Import RFC.

In the wizard that appears specify the SAP Server you want to connect to and enter the logon information
On the next screen select the RFC Modules that need to be a part of the Proxy class. You can either search for the RFC Module or browse the Application Component Hierarchy.

Press Finish.

Using Web Service

From the context menu of node external choose Import Web Service.

In the wizard that appears choose Remote Location/File System.

On the next screen select or enter a wsdl URL and press Finish.

Successful result: After you've imported the external services (RFC or Web Service) the structure will be displayed below the external node.

The picture on the right displays the structure of an external web service “ProjectBean”.
(Re-)Create Business Object

**Hint:** In the following you will create a new Business Object "ProjectR". You can of course re-use the existing Business Object "Project" and skip the next step (147)

Create a new Business Object "ProjectR" containing the same attributes and operations as Business Object "Project" (see step 6 – step 16)

**Hint:** The example in the following slides makes use of an external Web Service that is based on a Java EE 5 application.

**Use Remote Persistency**

Navigate to tab Persistency. Make sure that the dropdown field Backend is set to Remote.

Save the application.

Navigate to tab Datasource.

Select one of the operations of the Business Object Node in the column Target Operations on the left side.

Press button Create Mapping.

In the wizard that appears drill down to the external service, choose the appropriate method, and press OK.
Now you can map the parameters from the external service (on the right) to the parameters of the target operation (on the left) by simply dragging and dropping from left to right in the middle pane.

Repeat the prior steps to map remaining methods.

Save the application.

Save, Generate, Build, and Deploy the application

Hint: You will proceed with configuring the application. Testing will be done after the next exercise.

Configuring the external service call (Web Service)
If you use a Web Service as external service you have to

- configure the Web Service Destination using NWA (see step 103 to step 105).
- map the CAF Service to the correct WS Destination using the External Service Configuration tool (see step 107 to step 108)
Service Composition With SAP Composite Application Framework Capabilities In SAP NetWeaver CE 7.1

Configuring the external service call (RFC)
If you use a RFC as external service you have to

- configure the RFC Destination using NWA (step 156 to step 164).
- map the CAF Service to the correct RFC Destination using the External Service Configuration tool (see step 107 to step 108)

In a web browser, navigate to http://<server>[:<port>/nwa. You may need to login with administrator credentials.

Navigate to Configuration Management ->Identity Management

Choose Destinations.

Press Create Button
In the next screen, enter some general data for the hosting system, e.g.

Hosting System: “Local J2EE System”
Destination name: “T34M800”
Type: “RFC”

Press Next button

Enter data for Connection and transport (Hint: See Sap logon for the message server name)

Press Next button

Configure authentication data

Press Finish button
Testing the Business Object

In the SAP NetWeaver developer studio right click on Business Object Node ProjectR and select Test Service.

The service browser comes up. On the left side drill down to node ProjectR and select it.

Choose findAll

Successful result: All Projects will be displayed in the result table on the right.