“Web Service -> PI -> Web Service” Scenario — A Complete Walkthrough

Applies to:
SAP NetWeaver Composition Environment 7.1, SAP NetWeaver Process Integration 7.1

Summary
In this article, I will provide a step-by-step guide on how-to implement a composite which transparently leverages a PI Web Service Proxy to mediate Web Services.

Firstly I will show you how you can define the needed PI Interfaces and mappings in the Enterprise Service Repository and how to configure the Scenario in the Integration Builder of PI 7.1. Also a Web Service has to be implemented. For that Composite Application Framework (CAF) will be used.

Second, I will show you how to implement the PI Web Service Call in CE 7.10 with the Composite Application Framework.

Author: Katharina Seiz
Company: SAP AG

Author Bio
Katharina Seiz is a Solution Designer for the SAP NetWeaver Center of Excellence, where her role and responsibility is to evaluate SAP’s cutting edge technologies and transform them into real solutions for SAP customers using the potential of enterprise SOA and realizing the full value of composite applications. She began her career with SAP in 2003 and has gathered significant experience in numerous enterprise-scaled development projects in both the ABAP and the Java world.
Table of Contents

Overview ............................................................................................................................................................. 3

Procedure ........................................................................................................................................................... 4

  Step 1: Defining Interfaces in Enterprise Service Repository ................................................................. 4
  Step 2: Implementation of Web Service which has to be called from Process Integration Inbound Interface .......................................................................................................................... 17
  Step 3: Configuration of the Scenario in Integration Builder .................................................................... 22
  Step 4: Importing of the Process Integration Web Service as external Service into Composite Application Framework .......................................................................................................................... 25
  Step 5: Creation of Application Service and exposing it as Web Service ................................................. 33
  Step 6: Testing of the Scenario with Web Service Navigator: ................................................................. 36

Related Content................................................................................................................................................ 39

Copyright........................................................................................................................................................... 40
Overview

In PI we will have an Outbound and an Inbound Interface. Via the SOAP Adapter the Outbound Interface can be called from a composite as a Web Service. The Outbound Interface is mapped to the Inbound Interface, which forwards the call to another Web Service via the SOAP Adapter. That Web Service can reside anywhere (CE, R/3, Internet ...). In my example scenario the Web Service that provides the data has been implemented in CE.

The Outbound and Inbound Interface will use the same Message and Data types.

This little Scenario was used in one of our prototypes, which should provide a little workflow, which gives a company the possibility to optimize their G/L account postings. For that a user inside the company can fill out an offline Adobe Form and send it to a central inbox. A Guide Procedure Process is started and an employee of the finance department or a shared service center receives a work item in order to approve that request form. After approval automatically a posting to the backend system is done by the process.

Here we decided to use SAP NetWeaver Process Integration as Service Bus in order to guarantee a stable and generic interface (Outbound) of the composite. The customer can now map an Inbound Interface to it which either calls an Enterprise Service, Web Service, RFC or something else, dependent on what is existing in customer landscape.
Procedure

Step 1: Defining Interfaces in Enterprise Service Repository

At first you need to define the Inbound and Outbound Interfaces in SAP NetWeaver Process Integration and also the mappings between them.

A synchronous interface consists of two messages (Request and Response) and one fault message which is used for exceptions. Each message belongs to a data type which can be deeply structured.

In order to define interfaces and mapping you need to do the following steps:

- Define data types
- Define message types
- Define outbound and inbound interface
- Define interface and message mapping

In this scenario the message types of the inbound and outbound interface are exactly the same. So the mapping between them will be very simple and the data type for the request and response needs to be created only once.

1. Data type definition for the request:

Go to your Software Component and press the right mouse button on the namespace where you would like to create the data type and click on “New”. A wizard will be displayed. Here you need to choose the “Data Type” Entry under “Interface Objects”. Enter the needed information like name and Description and click on the Button “Create”.

![Create Object Dialog](image)
Now you can define the new data type:

2. Data type definition for the response:
Here the same steps are needed as for the request data type.

3. Message type definition for the request:
Go to your Software Component and press the right mouse button on the namespace where you would like to create the message type and click on “New”. A wizard will be displayed. Here you need to choose the “Message Type” Entry under “Interface Objects”. Enter the needed information like name and Description and click on the Button “Create”.
4. Message type definition for the response:

Here the same steps are needed as for the request message type. For “Data type Used” choose now the response data type created in step 2.
5. Outbound Interface definition:

Go to your Software Component and press the right mouse button on the namespace where you would like to create the outbound interface and click on “New”. A wizard will be displayed. Here you need to choose the “Service Interface” Entry under “Interface Objects”. Enter the needed information like name and Description and click on the Button “Create”.

![Create Object interface screenshot](image)

[Image: Create Object interface screenshot]
Choose “Outbound” as Category and “Synchronous” as Mode and select the message types created in step 3 and 4 for the Request and Response Message.

<table>
<thead>
<tr>
<th>Role</th>
<th>Type</th>
<th>Name</th>
<th>Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>Message Type</td>
<td>GLAccountPostingRequest</td>
<td><a href="http://sap.com/co">http://sap.com/co</a>*</td>
</tr>
<tr>
<td>Response</td>
<td>Message Type</td>
<td>GLAccountPostingResponse</td>
<td><a href="http://sap.com/cc">http://sap.com/cc</a></td>
</tr>
<tr>
<td>Fault</td>
<td>Fault Message Type</td>
<td>FaultMessage</td>
<td><a href="http://sap.com/co/haspoc">http://sap.com/co/haspoc</a></td>
</tr>
</tbody>
</table>
6. Inbound Interface definition:

Here the same steps are needed as for the Outbound Interface. Now Choose "Inbound" as Category and "Synchronous" as Mode and select the message types created in step 3 und 4 again for the Request and Response Message.
7. Interface Mapping:

Go to your Software Component and press the right mouse button on the namespace where you would like to create the Interface Mapping and click on “New”. A wizard will be displayed. Here you need to choose the “Operation Mapping” Entry under “Mapping Objects”. Enter the needed information like name and Description and click on the Button “Create”.

Now enter the Source Operation (= Outbound Interface) and the Target Operation (= Inbound Interface).

For each Message (Request, Response and Fault Message) a Mapping Program is needed. So as last step you need to create a Message Mapping for all three Message Types.
8. Message Mapping:

Go to your Software Component and press the right mouse button on the namespace where you would like to create the Message Mapping and click on “New”. A wizard will be displayed. Here you need to choose the “Message Mapping” Entry under “Mapping Objects”. Enter the needed information like name and Description and click on the Button “Create”.

Now you have to enter the source and the target message via the existing value help.
In our Scenario source and target message types are the same. After entering the right message types the structures are displayed and per Drag and Drop each field of the source can be mapped to the right field to of the target.

That has to be done for the Request, Response and Fault Message. Now you can assign these Mapping Programs to the Interface Mapping created in Step 7.
At the end you created the following Objects:

- Operation Mappings
  - IM_GLAccountPostingMapping
- Message Mappings
  - MM_GLAccountPosting_FaultMessage
  - MM_GLAccountPosting_Request
  - MM_GLAccountPosting_Response
- Service Interfaces
  - GLAccountPostingSync_In
  - GLAccountPostingSync_Out
- Message Types
  - GLAccountPostingRequest
  - GLAccountPostingResponse
- Fault Message Types
  - FaultMessage
- Data Types
  - ExchangeFaultData
  - ExchangeLogData
  - GL_Account_Posting
  - GL_Account_Posting_Item
  - GLAccountPostingResponse

Use of already existing Web Service Definitions to create an Interface:

The steps for defining the data types and message types can be skipped if a description of the interface already exists.

For example:

In the described scenario at runtime the inbound interface should forward the call to a Web Service which can reside anywhere. So in the most cases a description of such a Web Service already exists. This Web Service description (WSDL) can be imported into the Enterprise Service Repository as external definition and the request and response message can be used directly in the inbound interface which you still have to create.

Go to your Software Component and press the right mouse button on the namespace where you would like to import the Web Service Definition and click on “New”:
A wizard will be displayed where you can enter a name and a description. Enter the needed information and click on the button “Create”.

<table>
<thead>
<tr>
<th>External Definition</th>
<th>ImportedWebService</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td></td>
</tr>
<tr>
<td>Software Component Version</td>
<td>SAP CUSTOM BOC Basis 1.00</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Folder</td>
<td></td>
</tr>
</tbody>
</table>
In the next screen click on the Import Icon which opens a File Browser, where you can now choose a WSDL File, which describes your Web Service.
On the tab “Messages” you will now find the 3 message types which are used for Request, Response and Fault Message in the imported Web Service.

In the Interface Definition these Message Types can now also be used for Request, Response and Fault Message.
Step 2: Implementation of Web Service which has to be called from Process Integration Inbound Interface

Now we need the Web Service which has to be called from the SAP NetWeaver Process Integration via the Inbound Interface.

In this scenario we will implement a very simple Web Service in CE with CAF, but in general this could also be an Enterprise Service in an ABAP Backend System or something else.

The Web Service needs to have the same structure as the Inbound Interface defined in the Process Integration System.

Go to the Enterprise Service Repository and display the Inbound Interface. Go to the Tab “WSDL” and press the Button “Export WSDL to file” and save the WSDL file locally on your machine.

In NetWeaver Developer Studio show the Composite Application Framework Perspective and create a new CAF project. On the folder “external” use right mouse click and choose “Create Web Service Provider”.

  <soap:FaultMessages>
    <soap:complexType>
      <soap:sequence>
        <soap:element name="FaultMessage" type="soap.com/td"/>
      </soap:sequence>
    </soap:complexType>
  </soap:FaultMessages>
</wss>
In the shown wizard select the radio button “Remote Location / File System” and press the button “Next”.

Browse for your saved WSDL file and press the button “Finish”.

© 2008 SAP AG
An application service is created which is exposed as a Web Service.

Now you can implement the new service. Display the application service and go to tab “Implementation”.

Click on the link to the Java Source and implement the service.

You can test this Web Service in the Web Service Navigator. Go to the Web Service Navigator via URL “http://<host>:<port>/wsnavigator”.

---

**Web Service -> PI -> Web Service** Scenario — A Complete Walkthrough

[Composite Application Explorer] [Enterprise Service Browser]

**Generated Files**

- GLAccountPostingSync_In
  - EJB Abstract Class
  - GLAccountPostingSync_InServiceLocal.java
  - GLAccountPostingSync_InServiceLocalImpl.java

**General**

- Implementation
You will find the application service in the list of local services. Select this service.

Select the operation:
Enter some data and press the button “Execute”

You will get the result:

Please copy the WSDL URL which is displayed under “Service Information”. This is needed for the configuration of the scenario in SAP NetWeaver Process Integration.
Step 3: Configuration of the Scenario in Integration Builder

The configuration of this scenario has to be done in the Integration Builder of SAP NetWeaver Process Integration.

You need to do the following steps:

- Create Communication Component for sender and receiver
- Create Communication Channels for sender and receiver using the SOAP Adapter
- Define Sender and Receiver Agreement
- Define Receiver and Interface Determination

Communication Component:
Communication Channel:

Communication Channel for the receiver:
In the field “Target URL” the WSDL URL, copied in step before, has to be pasted.
Sender and Receiver Agreement:

Receiver and Interface Determination:

At the end the following Objects are created:
Step 4: Importing of the Process Integration Web Service as external Service into Composite Application Framework

In order to use the Web Service which calls the Outbound Interface in SAP NetWeaver Process Integration you need a WSDL URL. This you will get if you display the Sender Agreement in the Integration Builder again. You will find the entry: “Sender Agreement” → “Display WSDL” in the menu. This will display the WSDL and also the URL to this WSDL.
The WSDL with endpoint is displayed and also the WSDL URL is shown.

This URL can be used now in Composite Application Framework to import it as external service.

In the SAP NetWeaver Developer Studio show the Composite Application Framework Perspective and create a new Project. On the folder “external” use right mouse click and choose “import Web Service”.

![Image of a composite application framework with a project and import options](image-url)
A wizard will be shown. Choose the radio button “Remote Location/File System” and press “Next”.

In the input field “URL” paste the WSDL URL which you got from the Sender Agreement in the Integration Builder and press “Finish”.
After that you will find the imported service and all needed data types underneath the folder "external".

In order to use the right connection to the SAP NetWeaver Process Integration system, a logical destination is needed, which has to be mapped to this external service.
Go to the NetWeaver Administrator and follow the path “SOA Management” → “Technical Configuration” → “Destination Template Management”.

Here you have to create a new logical destination. The destination type has to be “WSDL” and in field “URL” you have to paste again the WSDL URL from the Sender Agreement.
On tab “Security” you have to choose “HTTP Authentication” in the drop down box and to select the radio button “User ID / Password (Basic)”. After that press the button “Details”.

Here you have to enter user id and password for the Process Integration System.

Now the new destination has to be mapped to the external service.

Go to the “CAF – Runtime Configuration” (Url: “http://<host>:<port>/caf”) and choose “Administrative tools”.

The administrative tools provide administrators with a set of applications they need to manage authorizations, KM and other configurations of the CAF runtime.

The test tools provide developers with a set of applications they need to test CAF services.
Then choose “External Service Configuration”.

Now Choose “Service Registry”: 
In the list “External Services” you have to select your imported service.

In the list “Destinations” you have to select the created logical destination. Then press the button “Map” and then “Save”.

Step 5: Creation of Application Service and exposing it as Web Service

In order to use that imported external service in the composite we need to create an application service within Composite Application Service, which is mapped to the external service.

Go to your Composite Application Framework project and to the folder “external” and use the right mouse click on the imported external service and choose “Default mappings”.

In the PopUp choose the external service and press button “OK”.

Target Application Service:
- Create methods in new Application Service
- Create methods in existing Application Service

© 2008 SAP AG
The new application service which is mapped to the external service is created:

- GLAccountPostingSync_Out
- Operations

At the end the new application service has to be exposed as Web Service.

Use the right mouse button and click on the new application service. In the context menu choose “Expose service as Web Service”.

In the PopUp enter a name for the new Web Service and select the application service. Press the button “OK”.

- GLAccountPostingSync_Out
The Web Service is created now.

Generate, build and deploy the project.
Step 6: Testing of the Scenario with Web Service Navigator:

Now you can test the scenario within the Web Service Navigator. Go to the Web Service Navigator (URL: http://<host>:<port>/wsnavigator).

You will find the application service in the list of local services. Select this service.
Select the operation:

![SAP NetWeaver WS Navigator](image)

- Select Service
- Select Interface
- Select Operation
- Enter Input Parameters
- Result

Service Information


Operations

- GLAccountPostingSync_Out

Export Test Data

Enter some data and press the button “Execute”.

![SAP NetWeaver WS Navigator](image)

Service Information


Configuration

- Timeout (in seconds): 60
- Endpoint in WSDL: GLPostingProcessingPort

Execute

Parameters

- GLAccountPostingRequest:
  - DocumentID: 123456
  - DocumentDate: 10/21/2008
  - PostingDate: 10/21/2008
  - DocumentType: 
  - CompanyCode: G063
  - Reference: 
  - DocumentHeaderText: [text]
  - Recurring: True
  - OneOF: True
  - Currency: 
  - Description: 
  - GLAccountPostingItem: [ ]
Now the whole scenario is configured and implemented. The application service created in Step 5 can now be used in composites or other applications.
Related Content

SAP NetWeaver Process Integration - Demo Example Configuration
RFC -> XI -> Web Service - A Complete Walkthrough (Part 1)
RFC -> XI -> Web Service - A Complete Walkthrough (Part 2)
Invoke Web Services using SAP XI
Consuming XI Web Services using Web Dynpro – Part I
Consuming XI Web Services using Web Dynpro – Part II
Consuming XI Web Services using Web Dynpro – Part III
When to use which CAF service?
Copyright

© 2008 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, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated 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®, 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.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

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.