How To... Use SOA Configuration to Call RFCs and Web Services from within SAP NetWeaver BPM

Applicable Releases:
SAP NetWeaver Composition Environment 7.2

Topic Area:
Development and Composition

Capability:
Composition

Version 1.0
March 2010
## Document History

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>First official release of this guide</td>
</tr>
</tbody>
</table>
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</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><strong>Example text</strong></td>
<td>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><strong>Example text</strong></td>
<td>User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example text&gt;</code></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>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>📹</td>
<td>Caution</td>
</tr>
<tr>
<td>📒</td>
<td>Note or Important</td>
</tr>
<tr>
<td>🎨</td>
<td>Example</td>
</tr>
<tr>
<td>📈</td>
<td>Recommendation or Tip</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Introduction** .................................................................................................................. 1

2. **SAP NetWeaver BPM process example** ........................................................................ 2

3. **Prerequisites** .................................................................................................................. 3

4. **Process instantiation Web service** .................................................................................. 4
   4.1 Create a Service Group project ...................................................................................... 4
   4.2 Import the process start Web service ............................................................................. 5
   4.3 Configure the process start Web service ........................................................................ 7
   4.4 Configure the Services Registry .................................................................................... 9
   4.5 Configure the Java Provider System ............................................................................. 11
   4.6 Define Java User Account Settings ............................................................................. 13
   4.7 Business Scenario Configuration .................................................................................. 17

5. **ABAP Web service** ........................................................................................................ 21
   5.1 Publish the ABAP Web service to the Service Registry ................................................ 21
   5.2 Import the ABAP Web service into the BPM process .................................................... 23
   5.3 Configure the Automated Activity ................................................................................. 24
   5.4 Configure the ABAP Provider System .......................................................................... 25
   5.5 Define the ABAP User Account Settings ..................................................................... 26
   5.6 Business Scenario Configuration ................................................................................. 28

6. **PI Web service** ............................................................................................................. 29
   6.1 Publish the PI Web service to the Service Registry ...................................................... 29
   6.2 Import the PI Web service into the BPM process ........................................................ 32
   6.3 Configure the Automated Activity ............................................................................... 34
   6.4 Configure the PI Provider System ................................................................................. 35
   6.5 Define the PI User Account Settings ........................................................................... 37
   6.6 Business Scenario Configuration ................................................................................. 39

7. **Third Party Web service** .............................................................................................. 40
   7.1 Publish the third party Web service to the Service Registry ......................................... 40
   7.2 Import the third party Web service into the BPM process ......................................... 42
   7.3 Configure the Automated Activity ............................................................................... 44
   7.4 Configure the third party Provider System ................................................................. 46
   7.5 Define the third party User Account Settings ............................................................... 48
   7.6 Business Scenario Configuration ............................................................................... 48

8. **Integrating WebDynpro and CAF** ............................................................................... 49
   8.1 Product list CAF Web service ...................................................................................... 49
   8.2 Process instantiation using WebDynpro ................................................................. 51
8.3 Import the CAF Web service into WebDynpro ................................................. 52
8.4 Import the process start Web service into WebDynpro ................................... 53

9. **Execute the process** ....................................................................................... 55

10. **ABAP RFC** .................................................................................................... 56
    10.1 Import the RFC into the BPM process ......................................................... 56
    10.2 Define the RFC Service Group ................................................................. 57
    10.3 Perform the RFC SOA Configuration ....................................................... 59

11. **(Optional) Application Communication** .................................................... 62
    11.1 Configure the Provided Services ............................................................. 63
    11.2 Configure the Consumer Services .......................................................... 64

12. **Appendix** ...................................................................................................... 65
    12.1 PI Configuration ....................................................................................... 65
    12.2 BPM process start Web service ............................................................. 70
    12.3 Process Context ....................................................................................... 72
1. Introduction

This How-To Guide uses the new SOA Configuration approach to configure different providers and consumers and therefore does not make use the traditional Destination approach. Since there could potentially be thousands of service providers and consumers which interact with each other a more advanced configuration approach are needed to manage the services. SOA Configuration first aims to simplify the configuration steps of large amounts of services, second it provides an automatic configuration and finally allow for role separation. Developers can define the configuration options during design time and then after deployment administrators can do the final assignment.

This How-To Guide consists of several different scenarios that follow the steps outlined below. First all services are published to the Service Registry from where they will be imported into the SAP NetWeaver Developer Studio. Automated Activities represent Web services and are mapped to Service Groups. Communication Profiles enforce the method of authentication that providers stipulate and User Accounts can be used to map Service Users to specific Provider Systems. Finally a Business Scenario is used to encapsulate the consumer and provider configuration settings for the entire example scenario.
2. SAP NetWeaver BPM process example

The order process described in this How-To Guide includes 2 user interfaces, 5 Web service and one RFC (Remote function call). Each service is individually discussed and builds on each other to produce the completed process as shown below.

The initial order form makes use of a Java Web service to display a list of products to include in the order; the order information is submitted to the process start Web service whereupon the process starts. The order information will then move through various automate activities that in turn calling Web services in different back end systems. First an ABAP Web service is called to retrieve the customer zip code information so that the third party external .Net Web service can return movie listings within a the customers zip code. Simultaneously the order is created in an ABAP back end system via a PI Web service. During the last step an audit trial consisting of the order information is stored in an ABAP system via a direct RFC call. All the Web services are published to the Services Registry and they enforce at least Basic Authentication.
3. Prerequisites

The following is a list of prerequisites in order to get this scenario to work

- SAP NetWeaver Composition Environment 7.2
- SAP NetWeaver Process Integration 7.1 and higher
- SAP NetWeaver 7.0 EhP1

Ensure you have the following configured

- Enterprise Services Registry
- Functional PI Web service
- Functional ABAP Web service
- Functional ABAP Remote Function Call (RFC)
- Functional Third Party .Net Web service

Additionally knowledge of the following development tools are requested

- Java WebDynpro
- SAP NetWeaver Business Process Management (BPM)
- Composite Application Framework (CAF)
4. Process instantiation Web service

A BPM processes consist of a mandatory Web service that starts the process. During this exercise you will learn how to import a Web services Description Language (WSDL) document into the BPM process and assign it to the process start event; then perform the necessary SOA configuration settings so that the process start Web service endpoints are exposed.

4.1 Create a Service Group project

1. Create a Service Group Configuration project

   To be able to reuse Service Groups in multiple consumer applications, you have to share these Service Groups in a uniform location. You create and share Service Groups in Service Group configuration development components (DCs). Using Service Groups you ensure that service references in different client applications are grouped together and will be configured to use the same provider at runtime. Proceed to create the Service Group Development Component project called soa/sg.

2. Create a dependency from the BPM project to the Service Group project

   After you have created a Process Composer project called soa/bpm proceed to add a dependency to the Service Group project created in the previous step. This will allow the BPM project to have access to the Service Groups within the Service Group project; any new Service Groups defined within the BPM project can then be stored within the Service Group project.
For optimal re-use the suggested approach is to add the Service Group development component project to its own software component.

4.2 Import the process start Web service

1. Import the WSDL file into the Business Process

This step assumes that you have access to the meta-data of an abstract asynchronous Web service. Refer to the appendix for the complete WSDL document; it consists of only input parameters that are made up of a header structure and a collection of items. To import the WSDL file expand the Service Interface folder > right click > select Import WSDL. You can import abstract WSDL documents from either the local file system, Enterprise Services Repository or the Services Registry.

For more information on how to start BPM processes, visit

2. Create a Service Group

A Service Group represents the provider from a consumer point of view. The tools of the SAP NetWeaver Developer Studio capture information about the provider systems and group the services, which can be invoked on the same provider system in the Service Groups. While running the WSDL import wizard you will be asked information about Service Groups. At this point you can either re-use or create a new Service Group. Service Groups will be mapped to specific back end systems so it makes sense to assign the Service Group an appropriate logical name. Proceed to create a new Service Group; in the example below we named the Service Group LOCALHOST_J2EE. Remember to select the previously created Service Group project, later on when you perform the SOA configuration tasks you will map this Service Group to a
specific back end provider system. In the case of the process start Web service the provider system will be the local BPM server.

Notice the Service Interface Order_Si that will be used throughout this How-To Guide and is used for the Web service that starts the process.

3. Verify the Service Group within Service References
   The Service Reference entity represents the information about services consumed in a uniform way across consumer applications developed using different tools such as a WebDynpro or Business Process Management. The system collects information about the port type (service endpoint interface) and authentication of the Web service under a single entity called service reference. For each service reference, you specify the authentication level with which the consumer application consumes the respective service. You can select one of the following authentication profiles for the appropriate authentication level (No Authentication, Technical User, Business User, Business or Technical User). Expand the Connectivity folder and double click on Service References. Verify that there is a reference to the previously created Service Group.

Optional. Notice the Authentication Profile by looking at the properties of the Service Interface. Leave the default assigned Authentication Profile as is. Choose this option (Business or Technical User) if you cannot determine the authentication required by the Web service during design time. At a later stage, the administrator can determine the authentication level and provide the concrete security mechanism.
Optional reading dealing with Service Groups

http://help.sap.com/saphelp_nwce72/helpdata/en/62/5c651a35e545e7b2dce2f352f95e07/frameset.htm

4.3 Configure the process start Web service

1. Create a Message Trigger

Take note that imported WSDL documents appear under the Service Interfaces folder. Perform the following steps to incorporate the process start Service Interface into the process. First create a Message Trigger that references the Order_Si Service Interface. Then assign the Message Trigger to the process start Event.

2. Perform the necessary Data Mapping

Since the process start Web service is asynchronous you only need to map the input parameters; however from the process point of view the data is mapped within the Output Mapping tab. Proceed to map the data from the outputs of the start event on the left side of the screen into the process context on the right side of the screen. This step assumes that you have
already imported a XML Schema Definition (XSD) document that contains the process context structure; in the next step we provided an example of the context structure used within the process.

3. Process Context

The process context acts as a storage area for the data that moves throughout the different steps within the process. The example process context consists of both a header and order structure and a collection of ordered items and movies. Right click on the Data Types folder to import the process context XSD file.

Refer to the appendix section for the complete example.

4. Compile and Deploy

Proceed to compile and deploy your BPM process to the server.
4.4 Configure the Services Registry

1. Setup the Services Registry

If you already configured the Service Registry then you can proceed to Step 2.

We will briefly go over all the configuration tasks needed to setup the Service Registry however please refer to the following Help URL for detailed instructions.


Within the SAP NetWeaver Administrator navigate to Configuration Management > Scenarios > Configuration Wizard > Start the Connection Configuration to Central Services Registry task.

Optional. Execute the Service Registry Destination Creation task if you plan to publish Web services from an older ABAP system. You do not have to execute this task if the ABAP system is at least NW 7.00 SP14 or NW 7.10 SP04.

You can verify the settings after the tasks finished by navigating first to the Single Service Administration within SOA Management and search for the ServiceRegistrySi service interface.

Notice the Web service endpoint URL that points to the Service Registry.
Optional. Perform this step only if you executed the Service Registry Destination Creation task. Navigate to the Destination Template Management within SOA Management and verify that both the UDDI_DESTINATION and the CLASSIFICATION_DESTINATION are created.

2. Publication to Service Registry

Verify that the Publication Rules are Active. This will ensure that the Java Web services deployed on the BPM server will automatically be published to the Services Registry. Within SOA Management navigate to Application and Scenario Communication and click on Publication Rules. Ensure that the SAP_DEFAULT_PUBLISH_ALL publication rule is active.
4.5 Configure the Java Provider System

1. Create a Communication Profile

   SOA Configuration is performed within the SAP NetWeaver Administrator on the SAP NetWeaver Composition Environment where the BPM process is deployed. Since the BPM server acts as the provider system the first step in setting up such a provider is to create a Communication Profile. Communication Profiles contain policies with which the communication between clients calling the BPM process and the process itself has to comply. In this scenario we will use Basic Authentication; therefore the start process Web service must offer endpoints that comply with the policies defined in this Communication Profile.

   Within SOA Manager click on System Connections and create a new Communication Profile (BASIC_AUTH_LOCALHOST) and define the Authentication Method as User Name/Password. Later on you will assign this Communication Profile to the process start Web service in order to enforce Basic Authentication when starting the process.

   ![Communication Profile](http://help.sap.com/saphelp_nwce72/helpdata/en/96/a75742b6081053e10000000a155106/frameset.htm)

   If Single Sign-On (SSO) between the systems are required then you need to create a Communication Profile that supports SAP Logon Tickets or SAML Assertion instead of User Name/Password authentication. Refer to the following URL for more information.


2. Create the Provider System

   To establish a connection from a consumer to a provider system, you have to specify several configuration settings. The system connection contains specific information about the provider system in the landscape, such as host name and port number, as well as information about the Communication Profile defining the policies for communication with that system, the mechanism for the discovery of Web services on it, and the concrete authentication credentials to access the WSDL or WSI sources. Note that you can only create one connection for every provider system in the landscape. Since the Services Registry acts as a single point to discover services we will use the Service Registry as the discovery mechanism. All Web services used in this How-To Guide is published in the Service Registry. To setup the BPM server as a provider system you need to create a new Provider System of Type Java. You can use the lookup...
window to select the Java system. The value help displays all systems, whose Web services are published to the Services Registry.

Assign WS for the Connectivity Type; leave the default SAP_DEFAULTPROFILE. Note that the system connection is established over the highest active version of the profile. Finally, you can choose a specific search mechanism for services on the provider system (WSDL or WSIL documents, or Services Registry). Make sure to select the Services Registry.

3. Metadata User credentials

When you define the credentials for the Metadata user note that this information is not used during runtime; we will define the runtime user information later. The Metadata information is used during design and configuration to retrieve WSDL metadata from the local Java system.
4. Verify and Test Connectivity

You can use the built-in feature to simulate connectivity and to verify that the provider system is setup correct. Simply click on the Ping System button; if everything is configured correctly you should see a list of Java Web services deployed to the Java system and that is published to the Services Registry.

4.6 Define Java User Account Settings

During this step you will define the User Account information that will be used during runtime to start the BPM process. This User Account will be mapped to a specific BPM provider system.

1. Create a new User Account

If the Web services that you want to consume require authentication, you can provide this authentication in entities called User Accounts. When you create a User Account, you specify different authentication methods in the same entity. At a later stage, you can reuse and assign this account to one or more than one Web service on a provider system. When you assign the User Account to a Web service, the system uses the authentication credentials in the account to invoke the Web service at runtime. Navigate to SOA Management > Application and Scenario Communication > and click on User Account Management. Create a new User Account and supply the service user credentials. Remember this user needs to have the necessary rights to start a BPM process. Proceed to create a new User Account of type WS and enter the authentication credentials used during runtime.
Refer to this Help document for more details regarding BPM user roles.


If Single Sign-On (SSO) between the systems are required then you need to create a Communication Profile that supports SAP Logon Tickets or SAML Assertion instead of a Service User Name/Password authentication and a Service User Account is not needed.

Refer to the following URL for more information dealing with SSO.


2. Create a new User Account Assignment

To consume Web services which require authentication and you want to consume them using a dedicated service user, you have to assign a User Account to them. You can assign User Accounts to one or more than one Web service running on a provider system. In addition, you can assign the same User Account to one or more than one Service Group that contains a reference to the Web service. The consumer application invokes the business logic on the provider side using the authentication methods which you provide in the User Account. The User Account assignments which you create are also used by the system when you configure Service Groups. Use the lookup window to select the previously created User Account. Click Next to continue.

As before use the lookup window to select the provider system that the User Account has access to. Since this User Account will be used to start the BPM process makes sure to select the local BPM server.
Next assign the concrete process start Web service. After you have deployed the BPM process the start Web service will automatically be deployed to the server. Use the lookup window to select the process start Web service that you initially imported into the BPM process.

In this step use the Service Group that you created earlier when you imported the process start Web service into the process composer. Proceed to map the Service Group to this User Account.
The final result should look as follows. Notice the service name and Service Group assignments.

Keep in mind you do not have to assign concrete values for the service name or for the Service Group. If you want to re-use User Accounts when calling other Web services then refrain from specifying a specific service name. The same applies for the Service Group, if more than one Service Group needs to use a single User Account then refrain from specifying specific Service Group.
4.7 Business Scenario Configuration

A business scenario is a logical group of service definitions; it can contain many different service groups and provider systems however a given service group can only be assigned to one provider system that in turn can only use a single communication profile. You can think of a business scenario as a work list containing the service definitions necessary to complete the business process. Here you assign the Communication Profiles to one or more service definitions grouped in a business scenario. By assigning a Communication Profile to service definitions, you trigger the configuration of the respective service definitions. This means that you trigger the creation of a service endpoint for every service definition to which you assign a Communication Profile.

1. Create a new Configuration Scenario

Navigate to SOA Management > Application and Scenario Communication > Business Scenario Communication.

In this example we named the business scenario SOA_ORDER_PROCESS. As you continue through the wizard you will be asked to configure both the consumer and the provider side. Click on the Add button to first assign the LOCALHOST_J2EE Service Group; then click on the Assign Provider System button to assign the local BPM provider system. Click Next.

![Business Scenario Configuration Diagram](image-url)
To configure the provider side click on the Add button and use the lookup window to select the process start Web service; then Click on the Assign Profile button and select the basic authentication profile that you created earlier. This will assign an endpoint that will enforce basic authentication. Click Finish.

After you finished the configuration ensure that the State is processed and that the green icon is visible. The Service Group configuration is now complete and maps to the local BPM provider system; from this point forward any consequent consume applications can simply re-use this Service Group.

Scroll down to the bottom of screen and click on the Security tab. Notice the Service User Information that is automatically assigned due to the existence of the User Account Assignment created earlier.
Let’s turn our attention over to the Provided Services section. Click on the Provided Service tab and ensure again that the State is Processed; a green icon confirms that the process start web service is complete and enforces basic authentication.

2. Single Service Administration

After the business scenario is saved, the system performs a background process. During the background process, the framework examines the connections to the provider systems which you assigned to the Service Groups, and creates configuration entities for the consumer proxies – logical ports. The system applies the settings from the provider system’s connections to the logical ports. The Processing State column of the Service Group table shows the state of the configuration. You can check the created service endpoints in SOA Management > Application and Scenario Communication > Single Service Administration and verify that the BPM process start Web service endpoint were assigned correct. When you search for the Order_Si service interface you should see the newly added endpoint, the service name is made up of both the Communication Profile and the Business Scenario.

Notice the existance of a second Service Endpoint; this was automatically created when the process were first deployed; you may delete this endpoint. Consequent deployment will not add this endpoint again.
3. Service Registry

Within the Service Registry search for the BPM process start Web service, you can do that by filtering by a particular system. Notice the Web service (Order_Si) with the endpoint that enforces basic authentication.

Optional: Click on the Test button at the bottom the screen to start the process.
5. ABAP Web service

In this scenario you will learn how to call an ABAP Web service from within an Automated Activity. You will first import the Web service into the process; then perform all the modeling activities and as in the previous scenario once you have deployed the process perform the SOA configuration steps.

5.1 Publish the ABAP Web service to the Service Registry

1. Transaction SOAManager

This step assumes that you have already completed the necessary setup steps into order to publish Web services from transaction SOA Manager to the Services Registry.

Refer to the following SAP help document explaining the steps to configure connectivity to the Services Registry and how to publish from an ABAP system.

http://help.sap.com/saphelp_nw70ehp1/helpdata/en/ca/aeed3629c54f13ba642d82d8c28e51/frameset.htm

In short locate the Service Registry consumer proxy found within Single Service Administration; then create a Logical Port and assign the endpoint location of the Service Registry. The screenshot below shows and example how to define the URL including user account information.
Note the recommended consumer proxy `CO_SERVICES_REGISTRY_FLAT_SI` to configure the connection to the Services Registry. This is the interface between the ABAP back-end system and the Services Registry system, and contains the appropriate authentication information. If the ABAP system version is not on at least NW 7.00 SP14 or NW 7.10 SP04 then a different consumer proxy is used. Please refer to the aforementioned help document for more details.

2. Services Registry

After publication is complete verify that the Web service is published to the Services Registry by filtering by the particular ABAP system that you published from. Pay special attention to the information on both the Endpoints and System Details tabs; notice the System Type is assigned ABAP.
5.2 Import the ABAP Web service into the BPM process

During this step you will import the ABAP Web service from the Services Registry into the process. Within the Process Composer expand the Process Modeling and Service Interface folders. Right click and select Import WSDL to start the import wizard and make sure to select the Services Registry radio button.

1. Create a Service Group

Once you have connected to the Services Registry select the desired Service Interface and click Next. You will then be asked to provide information dealing with the Service Group. Create a new Service Group by supplying the ABAP server logical name. Do not forget to select the Service Group Project created earlier. As a result the ABAP Service Group will be added to the Service Group Project and can therefore be re-used in different projects.

2. Service References

Verify that a Service Reference to the newly added ABAP Service Group is included in the list of Service References. Expand the Connectivity folder and double click on Service References. Also note that the imported ABAP Web service is visible within the Service Interfaces folder.
5.3 Configure the Automated Activity

1. Add an Automated Activity to the BPM Process

In this step you will add an Automated activity that references the previously imported ABAP Web service; you can simply drag and drop the Service Interface into the diagram. Then click on Automated Activity and select the Interface tab. If not already defined select the Service Interface, including the specific operation; ensure the newly created Service Group is selected.

Continue to add the required input and output mappings to the automated activity. Since the ABAP Web service is synchronous you need to map both the input and output parameters accordingly. The customer ID will be the input and the zip code will be the output.

2. Proceed to compile and deploy your BPM process to the server.
5.4 Configure the ABAP Provider System

1. Create a Communication Profile for ABAP

The first step in setting up the ABAP provider system is to create a Communication Profile. In this scenario we will use Basic Authentication; therefore the ABAP Web service must offer endpoints that comply with the policies defined in this Communication Profile.

Within SAP NetWeaver Administration navigate to SOA Manager > System Connections and create a new Communication Profile called BASIC_AUTH_ABAP and define the Authentication Method as User Name/Password.

2. Setting up the Provider System

In order to setup ABAP as a provider proceed to create a new Provider System of Type ABAP. Since the ABAP Web service is published to the Services Registry you can use the lookup window to select the specific ABAP Server.

Assign WS as the Connection Type and select the ABAP Communication Profile created in the previous step. Since the ABAP Web service is published to the Services Registry remember to set the Services Source radio button selection to the Services Registry.
3. Metadata User credentials

When you define the credentials for the Metadata user note that this information is not used during runtime; this will be defined at a later stage. The Metadata information is used during design and configuration to retrieve WSDL metadata from the corresponding system. The Metadata user needs to have access to the ABAP system in order to read the Web service Metadata.

4. Verify and Test Connectivity using the Ping button; you should see the published ABAP Web service in the list.

5.5 Define the ABAP User Account Settings

During this step you will define the User Account that will be used during runtime to call the ABAP Web service. You will then map this User Account to the specific ABAP provider system.

1. Create a new User Account

Proceed to SOA Management > Application and Scenario Communication > and click on User Account Management. Create a new User Account and supply the Service User credentials. Remember this user needs to be defined within the ABAP system and have the necessary rights to execute the Web service.

It’s always a good idea to test the Web service using the WS Navigator first to make sure that the security roles have been assigned in the ABAP system.
If Single Sign-On (SSO) between the systems are required then you need to create a Communication Profile that supports SAP Logon Tickets or SAML Assertion instead of a Service User Name/Password authentication and a Service User Account is not needed.

Refer to the following URL for more information dealing with SSO.


2. Create a new User Account Assignment

Click on the Assignment tab and reference the previously created User Account; then map that to the ABAP Provider system accordingly. Also note the Service Name and Service Group that were created earlier.

Since we showed you the detailed screen shots in the previous scenario it is assumed you know how to complete this step. Refer to the previous scenario for more details how to create an Assignment.
5.6 Business Scenario Configuration

1. Add the ABAP Service Group

Since we already created a Configuration Scenario earlier we can simply add an additional Service Group to the existing scenario. Proceed to SOA Management > Application and Scenario Communication > Business Scenario Configuration. Within the Service Group tab click the Add button to assign the ABAP Service Group and then assign the ABAP provider system accordingly by clicking on the Assign Provider System button.

![Business Scenario Communication: Configuration](image)

Ensure that the State is Processed and that the icon is green meaning that the configuration for the ABAP scenario is complete.

💡 You can click the refresh Icon located in the right center of the screen until the assignments have been processed.
6. PI Web service

In this scenario you will learn how to call a PI Web service from an automated activity within a BPM process and configure the connectivity using SOA Configuration. Previously we showed how to call and configure both a Java and ABAP Web service. This section will primarily focus on the steps needed to call a synchronous Web service hosted on PI. As before you will first import the Web service into your BPM process; then perform all the modeling activities and deploy the process to the SAP NetWeaver Composition Environment. Finally you will perform the necessary runtime SOA configuration settings.

6.1 Publish the PI Web service to the Service Registry

1. Configure PI to publish to the Service Registry

   We will assume that you have already completed the necessary setup steps into order to publish Web services from the Integration Builder to the Services Registry. Refer to the following SAP help document explaining the steps to configure the Services Registry.

   http://help.sap.com/saphelp_nwpi711/helpdata/en/47/d391d7b8fc3c83e10000000a42189c/frameset.htm

   Within the SAP NetWeaver Administrator on the PI system navigate to SOA Management > Application and Scenario Communication > Single Service Administration. On the Consumer Proxies tab, search for ServicesRegistrySi Service Interface and complete the configuration as stated in the aforementioned Help document.

   Notice the location of the the Service Registry endpoint URL.
2. Integration Builder

Right click on the specific *Sender Agreement* and select *Publish to SR*. This will publish the Web service with its corresponding endpoints including the server information to the Services Registry.

Throughout this scenario we will be referring to the *SENDER_SI* PI Service Interface.

This step assumes that the PI Web service enforces at least Basic Authentication.

Refer to the Appendix for complete PI configuration details.
3. Service Registry

After publication is complete verify that the Web service is published to the Services Registry. Within the Service Registry filter by the particular PI system that you published from. Pay special attention to the information on both the Endpoints and System Details tabs. Notice the System Type is assigned IS (Integration Server).

Be aware there could be more than one PI System entry published to the Services Registry depending on if publication were performed from the PI Integration Server or from the PI ABAP stack. This scenario focuses on publication from the Integration Server (IS) therefore ensure that you select the correct Integration Server (contains the SAP client suffix) from the system drop down list.

Optional. Only if the publication process assigned the System Type of the PI Integration Server to ABAP proceed to change the System Type to IS (Integration Server). Click on the Systems Tab > select the PI Integration Server > Edit > change the System Type to IS.
6.2 Import the PI Web service into the BPM process

Continue within SAP NetWeaver Developer Studio. During this step you will import the PI Web service from the Services Registry into the BPM process. Within the Process Composer expand both the Process Modeling and Service Interface folders. Right click and select Import. Make sure to select the Services Registry radio button and continue through the Wizard.

1. Import the Web Service

Once you have connected to the Services Registry select the SENDER_SI Service Interface and click Next. You will then be asked to provide information dealing with the Service Group. In Java, a Service Group can be local for a Development Component (DC) or shared among multiple development components. If it is local, it can contain only Service References for services imported into the same DC. That is why it is recommended for bigger applications to use shared Service Groups in order to decrease the number of Service Groups an administrator would have to deal with.

When searching for Web services published from the PI Integration Server remember to filter by the PI Integration Server and not by the PI ABAP System.
2. Create a Service Group

Create a new Service Group by supplying the PI server name. Do not forget to select the Service Group Project created in the first scenario. As a result the PI Service Group will be added to the Service Group project and can therefore be re-used in different projects.

3. Service References

As before verify that the newly added Service Group is included in the List of Service References. The Service Reference entity represents the information about services consumed in a uniform way across consumer applications developed using different tools such WD and BPM. Expand the Connectivity folder and double click on Service References. Also verify that the PI Web service is visible within the Service Interfaces folder.
6.3 Configure the Automated Activity

1. Add an Automated Activity to the process

In this step you will add an Automated activity that references the previously imported PI Web service; you can simply drag and drop the Service Interface into the diagram. Then right click on PI Web Service in the diagram and select the Interface tab. If not already defined select the SENDER_SI Service Interface, including the SEND operation and make sure that the newly created Service Group is selected.

![Diagram showing Automated Activity]

2. Assign the Data Mapping

Next add the required mapping to the Automated Activity. Since the PI Web service is synchronous you need to map both the input and output parameters accordingly. Input parameters or the data that you send into the PI Web service are translated to input mappings and output mappings translate to Web service response messages. The PI Web service accepts order information and once the order is created the PI Web service will in turn respond with an Order ID.

To start the mapping exercise click on the Input Mapping tab and map the process context order fields from the left hand side of the screen to the input parameters of the Web service on the right hand side of the screen. Similarly, click on the Output Mapping tab and map the Order ID from the Web service response into the Process Context Order ID field.
3. Proceed to compile and deploy your BPM process to the server.

6.4 Configure the PI Provider System

1. Create a Communication Profile for PI

In this PI scenario SOA Configuration is done within the SAP NetWeaver Administrator on the SAP NetWeaver Composition Environment where the BPM process is deployed. The BPM process will act as the consumer and PI will act as the provider. The first step in setting up the PI provider system is to create a Communication Profile. In this scenario we will use Basic Authentication; therefore the PI Web service must offer endpoints that comply with the policies defined in this Communication Profile.

Within the SAP NetWeaver Administrator (on the BPM system) > SOA Manager > System Connections create a new Communication Profile (BASIC_AUTH_PI) and define the Authentication Method as User Name/Password.

2. Setting up the Provider System

In order to setup PI as a provider proceed to create a new Provider System of System Type Integration Server. You can use the lookup window to select the specific Integration Server from within Services Registry.

Web services published from the Integration Builder within PI will have a system type of Integration Server.
3. **Metadata User credentials**

When you define the credentials for the Metadata user note that this information is not used during runtime; this will be defined at a later stage. The Metadata information is used during design to retrieve WSDL metadata from the corresponding PI system.
4. Verify and Test Connectivity

Click on the *Ping System* button; if everything is configured correct you should see a list of Web services published from the PI Integration Server to the Services Registry.

![System Connections: Provider Systems](image)

6.5 Define the PI User Account Settings

During this step you will define the User Account information that will be used during runtime to call the PI Web service. You will then map this User Account to the specific PI provider system.

1. Create a User Account

Proceed to *SOA Management > Application and Scenario Communication > User Account Management*. Create a new User Account and supply the Service User Credentials. Remember this user needs to be defined within PI and have the necessary rights to execute the Web service.

Test the PI Web service first using the WS Navigator in order to make sure that the Web service works and that the appropriate security roles are assigned in the back end PI system.
2. Create a new User Account Assignment

Within the Assignment tab create a new Assignment and reference the previously created User Account information. Note the assigned Service Name and Service Group that were created earlier.
6.6 Business Scenario Configuration

As mentioned before the notion of a Business Scenario groups multiple services in a particular business process enabling you to encapsulate all the settings for different Web service providers and consumers and allowing you to do it once.

1. Add the PI Service Group

Assign the PI Service Group to the PI provider system. Proceed to SOA Management > Application and Scenario Communication > Business Scenario Configuration. Select the previously created Business Scenario. Within the Service Group tab click the Add button and assign the PI Service Group; then assign the PI Provider System accordingly.

Ensure that the State is Processed and that the icon is green meaning that the configuration for the PI scenario is complete.
7. Third Party Web service

In this scenario you will learn how to call a third party Web service from an Automated Activity within a BPM process and configure the connectivity using the new SOA Configuration approach. The third party Web service accepts a zip code and returns a list of movies. The same approach will be used as before to explain how to call a synchronous third party Web service from SAP NetWeaver BPM. The Services Registry will again be used as the discovery mechanism. Finally you will perform the necessary runtime SOA configuration settings so that the BPM process can call the third party Web service.

7.1 Publish the third party Web service to the Service Registry

By entering the endpoint WSDL of a service definition you can publish third-party service definitions to the Services Registry

1. In the Services Registry, choose the Service Definition tab and select Publish.
2. In the endpoint WSDL field, enter the endpoint WSDL URL and choose Next


Optional. Verify the HTTP Proxy settings if you are unable to connect to the external service from within your network. Within SAP NetWeaver Administrator > SOA Management > SOA Middleware Global Settings > HTTP Proxy
3. Select *New System* and enter details about the system that provides the service. Enter the system name, the host name and the system type and choose Next.

4. Leave default service state as *Configured* and choose *Finish*. The third party Web service has been published. Verify the service by filtering by the System within the Service Definition tab.
7.2 Import the third party Web service into the BPM process

During this step you will import the third party Web service from the Services Registry into the BPM process

1. In the SAP NetWeaver Developer Studio open the Process Composer perspective

2. Expand the Process Modeling and Service Interface folders. Select the WSDL folder and from the context menu select Import WSDL to start the Import Wizard. Make sure to select the Service Registry radio button and click the Next button.

Optional. If you are unable to connect to the external host make sure to change your proxy settings within Window > Preferences > General > Network Connections.
3. Create a new Service Group for the configuration of the imported Web service called DOTNET_IGNYTE. select the Service Group DC created in the first scenario (soa/sg) and push the Finish button.

As mentioned previously, the Service Group will be added to the Service Group DC and can therefore be shared from multiple consumer application DCs.

4. As a result a new Service Reference to the newly added Service Group is included in the List of Service References of the Process Composer DC. Expand the Connectivity folder and double click on Service References.

If multiple consumers refer the same Service Group, administrators can configure them in a single step. By configuring a Service Group, the administrators can then configure all references belonging to that group.
5. (Optional) Open the Service Group DC by opening up the Java EE perspective. You should see the name of the Service Group in the list

7.3 Configure the Automated Activity

In this step you will add an automated activity that references the previously imported third party Web service

1. Add the Automated Activity to the process

   Drag and drop the GetTheatersAndMovies operation into the diagram. Highlight the GetTheatersAndMovies activity, go to the Property tab and select the Interface tab. If not already defined, select the MovieInformationSoap Service Interface, including the GetTheatersAndMovies operation and make sure that the newly created Service Group is selected.

2. Assign the Data Mapping

   Given a zip code and radius, the third party Web service will return a list of theaters and movies currently playing. To map the input and output parameters highlight the third party Automated Activity, open the Properties tab and select the Input Mapping tab. Map the zip code from the process context (left) to the input structure of the Activity (right) as shown in the image below.
Since the third party Web service is synchronous you need to map both the input and output parameters accordingly. Input parameters or the data that you send into the third party Web service are translated to input mappings and output mappings translate to Web service response messages.

Double click on the radius field and enter 10 as default value.

Similarly click on the Output Mapping tab and map the Movie node from the Web service response (left) into the Process Context Movie node (right), then map the Rating and the Name nodes as shown in the image below.

3. Save, compile and deploy your BPM process to the server.
7.4 Configure the third party Provider System

1. Create a Communication Profile for the third party provider.

   In this scenario SOA Configuration is done within the SAP NetWeaver Administrator on the SAP NetWeaver Composition Environment where the particular BPM process is deployed. The BPM process will act as the consumer and the third party Website will act as the provider. The first step in setting up the provider system (Third Party Website) is to create a Communication Profile. In this scenario we won’t need Basic Authentication since the third party Web service does not require any authentication.

   Navigate to SOA Management > Technical Configuration > System Connections and select the Communication Profiles tab and click the New button to start the creation wizard. Make sure you enter the following information:

   a. Profile Name: **NO_AUTH_EXTERNAL_SYSTEM**
   b. Authentication: **None**

2. Setting up the Provider System

   The Provider System captures specific information about the third party including the information about the Communication Profile created earlier. Since we published the third party Web service earlier to the Services Registry we already captured all the information needed to complete the Provider System configuration.

   Navigate to **SOA Management > Technical Configuration > System Connections** and select the System Provider tab and click the **New** button to start the creation wizard. Use the lookup window to select the third party provider system that is published to the Services Registry.
3. Metadata User credentials
   Since the third party Web service does not require any authentication, you do not need to enter the credentials in this step.

4. Third Party Provide System Details
   Assign the Communication Profile that you created earlier to the third party provider. Simply make the selection in the Profile drop down list. Remember to select WS for the Connectivity Type.
5. Service Search Settings

Since this web service is published to the Service Registry assign the Service Source accordingly.

6. Verify and Test Connectivity

As before use the built in feature to simulate connectivity and to verify that the third party provider system is setup correct. Simply click on the Ping System button; if everything is defined correct you should see a list of Web services published from the third party provider system to the Services Registry.

7.5 Define the third party User Account Settings

During this step you can define the User Account information that will be used during runtime to call the third party Web service. However, since the access third party Web service is unrestricted, you don’t need to define a User Account.

7.6 Business Scenario Configuration

As before assign the third party provider system to the third party Service Group.

Navigate to SOA Management > Application and Scenario Communication > Business Scenario Communication. Select the SOA_ORDER_PROCESS business scenario and click the Edit button.

1. Within the Service Group Tab click the Add button to assign the third party Service Group (DOTNET_IGNYTE) and assign the external provider system. As a result you should see a Green Status Icon if the configuration were accepted.
8. Integrating WebDynpro and CAF

Since the main part of the process is complete you can turn your attention over to the User Interface that will start the process. This example includes a rudimentary example where the end user selects products from a catalog and submits the order for processing. The order form will be developed using Java WebDynpro and a CAF (Composite Application Framework) Web service is used to retrieve the product catalog.

8.1 Product list CAF Web service

1. This step assumes that you have some familiarity with creating a CAF Web service. Once you have completed the Web service proceed to deploy the project to the server.

2. Navigate to the Business Scenario Communication and click to the Provided Services tab, since this Web service will act as a provider of the products proceed to add the CAF Web service to the list. Click on the Add button and search for the Product_Si Web service; then assign the BASIC_AUTH_LOCALHOST Communication Profile. This will enforce Basic Authentication for clients that call this Web service.
3. Verify that the configuration added the appropriate endpoint by navigating to Single Service Administration within SOA Management. The new endpoint Service Name consists of both the Profile and Business Scenario name.

Notice the second endpoint in the list. This was automatically created when you deployed the CAF application. You may delete that endpoint.
4. Logon to the Services Registry and verify that the Product Web service is published and that the correct end points are visible.

**8.2 Process instantiation using WebDynpro**

Create a WebDynpro project and create a dependency to the Service Group project.
8.3 Import the CAF Web service into WebDynpro

1. Adaptive Web Service Model

To import Web services into WebDynpro use the Adaptive Web Service Model functionality and continue through the import wizard. Since the CAF Web service is published to the Services Registry remember to select the Services Registry radio button during the wizard.

2. Import the Web service

Once you have connected to the Services Registry search for the Product_Si Service Interface and click Next. You may filter by the specific provider system.

3. Service Group

As before you will be asked to provide information dealing with the Service Group. Since you already created a dependency to the Service Group project you already have access to the local Java server Service Group where the product Web service is deployed. Choose the existing local host Service Group.
Refrain from creating a new Service Group. Select the Choose existing radio button and select the local host Service Group from the drop down list.

### 8.4 Import the process start Web service into WebDynpro

1. Adaptive Web Service Model

As before use the Adaptive Web Service Model functionality and continue through the import wizard. Since the process start Web service is published to the Services Registry remember to select the Services Registry radio button during the wizard.
2. Import the Web Service

Once you have connected to the Services Registry search for the Order_Si Service Interface and click Next. As before you will be asked to provide information dealing with the Service Group. Since you already created a dependency to the Service Group project you have access to the local Java server Service Group where the product Web service is deployed.

Refrain from creating a new Service Group. Select the Choose existing radio button and select the local host Service Group in the drop down list.

3. Service References

Verify that the referenced Service Group is included in the List of Service References. Expand the Connectivity folder and double click on Service References. Each Adaptive Web Service model uses the corresponding Web Service.
9. Execute the process

This step assumes that you have some familiarity with both WebDynpro development and mapping within BPM. After you imported both the product search Web service including the Web service to start the process proceed to develop the Order Form. Ensure that the Order button calls the process start Web service passing in the order information. Continue to develop the Confirmation user interface as shown below, the data displayed on the Confirmation form can be obtained from the process context by mapping the data into the user interface.

![Confirmation Form Image]
10. ABAP RFC

So far we have only worked with Web services as the main connectivity medium to call provider systems; you will now learn how to call RFCs directly from within your process. The last step in the process involves storing the order information in an ABAP table.

10.1 Import the RFC into the BPM process

1. To access an ABAP system during design time you first need to add an R/3 Configuration destination within SAP NetWeaver Developer Studio. From the Window menu select Preferences and enter the system information.

2. Open the Search Console view and import the ABAP RFC into the BPM process. From the Window menu > Show view > other and search for the Search Console. Select the particular R/3 destination that you created in the previous step and filter by the specific RFC. You can simply drag the RFC into the BPM process.
10.2 Define the RFC Service Group

1. Ensure that RFC Service Reference is assigned to an ABAP Service Group. Select the properties of the Automated Activity and click on the *Interface* tab. If no Service Reference is defined proceed to select *New* from the drop down list.
2. Since we already defined a Service Group for this particular ABAP system choose the existing ABAP Service Group.

![Image of New Service Reference window]

3. Verify that there are two different Service References under the ABAP Service Group, one reference is for the Web service and the other reference is for the RFC.

![Image of Service References window]

- CRM_ADAP
- urn:sap.com:document:apifunctions
- CUSTOMERADDRESS
- PLIS
- LOCALHOST_2ERF
- DOTNET_IGNITE
10.3 Perform the RFC SOA Configuration

1. Earlier you created a Communication Profile of type Web service for this particular ABAP provider, proceed to create a second Communication Profile but this time of type RFC.

![Image showing System Connections: Communication Profiles]

2. Assign the RFC Communication Profile to the ABAP provider system.

   Since the ABAP provider system already exists you can simply add this additional RFC profile under the *Connectivity* tab. Remember to select RFC and assign the Profile created in the previous step.

![Image showing System Connections: Provider Systems]
3. Within the Connection section enter the ABAP server information. The Repository Connection credentials are only used during the configuration time and not during runtime. You will assign the runtime User Account information in the next step.

4. Create a new User Account of type RFC. In this example we are using a Service User to connect to the ABAP back end system during runtime.

5. As before create a new Assignment and reference the previously created RFC User Account. Assign the desired ABAP provider system including the RFC service name and Service Group.
6. There is nothing to do in this step simply verify that the RFC Interface was assigned automatically within the Business Scenario.

7. This completes the SOA Configuration tasks, proceed to execute and run your application.
11. (Optional) Application Communication

Throughout this document we neatly bundled all the SOA Configuration settings related to the entire scenario within a single Business Scenario. This section will demonstrate an alternative approach to Business Scenarios referred to as Application Communication.

When dealing with many different providers and consumers within a given scenario it’s suggested to add all the configurations to a single Business Scenario as were shown in this document; this allows you to do the configuration in one single contained area. When dealing with a smaller number of Web service providers and Web service consumers you can instead perform the SOA configuration using Application Communication rather than creating a Business Scenario.

Since there is no wizard guiding you through the steps you have to do the configuration for each Java project separately. What makes Application Communication a bit trickier is that you need to know the names of the development components including if they are a provider or a consumer of services. The scenario outlined in this document consists of 3 development components. Both the CAF and the BPM development components are providers of services; the CAF project is the provider of the product Web service and the BPM project is the provider of the BPM process. The third Service Group project is synonymous for the consumer.

The easiest way to understand Application Communication is …

- To configure a consumer assign a Service Group to a Provider System
- To configure a provider assign a Communication Profile to the Provided Web service

Within SAP NetWeaver Administrator navigate to SOA Configuration > Application Communication. Notice the 3 projects including the Consumed Services and Provided Services tabs.
11.1 Configure the Provided Services

1. CAF Product Web service

Search for the CAF development component (DC) and select the Provided Services tab. Click Edit > Assign Profile. Select the BASIC_AUTH_LOCALHOST profile. This will enforce basic authentication when consumer applications call this Web service.

2. BPM start Web service

As before search for the BPM development component (DC) and select the Provided Services tab. Click Edit > Assign Profile. Select the BASIC_AUTH_LOCALHOST profile. This will enforce basic authentication for consumers trying to instantiate the BPM process.
11.2 Configure the Consumer Services

1. Service Group Project

Search for the Service Group development component (DC) and select the Consumer Services tab. Click Edit > Assign the corresponding Provider System to each Service Group. This will enable the corresponding service user to be used when accessing the specific Provider System.

2. Final result

Save the configuration. The Green Icon is an indication the configuration is complete. Proceed to execute and run your application as before.
12. Appendix

12.1 PI Configuration

Integrating SAP Process Integration is important when connecting SAP BPM to other systems in a heterogeneous landscape. Such a use case would be if your legacy systems are not web service enabled and you need to communicate using a proprietary protocol. When integrating SAP PI in such a case this is known as a mediated scenario. The following diagram illustrates the connectivity options available when SAP BPM communicates with SAP PI in a mediated scenario.

In the diagram above, the arrows are pointing either one way or bi-directionally. This illustrates either an asynchronous or synchronous communication pattern.

In our example we are utilizing a synchronous web service call and will be leveraging the WS adapter found in SAP PI. For communications going into SAP PI you must use the WS, SOAP or RFC adapters. For communications going out of SAP PI into your legacy systems you can leverage any supported adapter.

1. Enterprise Service Repository

The Enterprise Service Repository (ESR) is a SOA artifact library holding definitions of interfaces which are being provisioned or consumed in your service enable landscape. In our example we have created our own Service Interfaces and underlying data type definitions to create a simple scenario to accept the web service call from SAP BPM. We will concentrate on the sender objects representing the synchronous message coming from SAP BPM as this is what will be bound to the BPM process. Describing the implementation of the back end order create function is outside the scope of this document. The following diagram illustrates the required objects and relationships.
The following table identifies the approach taken to create the above Sender ESR objects.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Create a Software Component in the SLD. Import into the ESR (optional).</td>
</tr>
<tr>
<td>2. Locate or create a namespace for holding your development objects.</td>
</tr>
<tr>
<td>3. Create the Data Types &quot;ORDER_DT&quot;, &quot;ORDER_ID_DT&quot;, &quot;ORDERHEADER_DT&quot; and &quot;ORDERITEM_DT&quot; as depicted in the screen shots below.</td>
</tr>
<tr>
<td>4. Create the Message Type &quot;ORDER_OUT_MT&quot; and assign the Data Type &quot;ORDER_DT&quot;.</td>
</tr>
<tr>
<td>5. Create the Message Type &quot;ORDER_IN_MT&quot; and assign the Data Type &quot;ORDER_ID_DT&quot;.</td>
</tr>
<tr>
<td>6. Create a Service Interface &quot;SENDER_SI&quot; with an outbound stateless communication pattern.</td>
</tr>
<tr>
<td>7. Create a Service Operation &quot;SEND&quot; with a synchronous communication pattern. Notice the request and response message type entries are required.</td>
</tr>
<tr>
<td>8. Assign the Message Type &quot;ORDER_OUT_MT&quot; to the request and Message Type &quot;ORDER_IN_MT&quot; to the response.</td>
</tr>
<tr>
<td>9. Save and activate.</td>
</tr>
</tbody>
</table>

Please create the receiver objects as per the table above specific to your scenario. This may be an existing Enterprise Service.

2. Service Interface Definition

The following diagram illustrates the Service Interface definition and underlying Service Operation definition. Note the highlighted sections which emphasize the synchronous communication pattern and the Message Types used.
3. Message Type and Data Type Definitions

The following diagram illustrates the Message Type definitions and underlying Data Type definitions. Note the highlighted sections which emphasize the use of multiple Data Types. This demonstrates re-use of SOA artifacts which is a key value offering of the ESR. You can create your nested Data Type definition however please keep the element types identical and pay attention to the appropriate cardinality of the header and item structures.

[Diagram showing Message Type and Data Type definitions]

4. Integration Directory

The Integration Directory is a configuration tool where artifacts defined in the ESR are combined to create an integration scenario specifying communication protocols, routing rules and interface determinations. In our example we will create a standard synchronous integration scenario. The following diagram illustrates the objects created.

[Diagram showing Integration Directory objects]
The following table details the approach taken to configure the integration scenario in the Integration Directory.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Create Business Component, Business System or Party representing CE BPM.</td>
</tr>
<tr>
<td>2. Create the sender communication channel representing CE BPM. Choose adapter type &quot;WS&quot; and configure as per screen shot below.</td>
</tr>
<tr>
<td>3. Create the receiver Communication Channel representing the system you would like to communicate with. Make sure an adapter appropriate for a synchronous scenario is chosen. E.g. RFC, SOAP, XI. Asynchronous scenarios are supported however this example is synchronous and an immediate response is expected.</td>
</tr>
<tr>
<td>4. Create a Sender Agreement and assign the sender Communication Channel.</td>
</tr>
<tr>
<td>5. Create a Receiver Determination.</td>
</tr>
<tr>
<td>6. Create an Interface Determination.</td>
</tr>
<tr>
<td>7. Create a Receiver Agreement and assign the receiver Communication Channel.</td>
</tr>
<tr>
<td>8. Save and activate.</td>
</tr>
</tbody>
</table>

The configuration steps detailed above represent a normal integration scenario. You are able to utilize other variations such as ccBPM and asynchronous communications. In such scenarios the required configuration steps may vary.

5. Configuration Overview

In the example below you can see we have created an integration scenario that calls a subsequent RFC in an ERP system. As mentioned previously this could be a proxy call or any other synchronous communication. Using PI allows you to implement complex mapping and protocol translation capabilities. Asynchronous scenarios are also supported however are outside the scope of this document.
6. Sender Communication Channel Configuration

The sender communication channel is very simple in our scenario. The only parameters required are the host and port of the PI application server. Additional security options are available.
12.2  BPM process start Web service

```xml
<wSDL:definitions xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/
xmlns:p2="http://soa.com/es/context"
name="Order_Si" targetNamespace="http://soa.com/es/java">
  <wSDL:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://soa.com/es/context"
targetNamespace="http://soa.com/es/context">
      <xsd:element name="OrderRequest_Mt" type="OrderRequest_Dt" />
      <xsd:complexType name="Header_Dt">
        <xsd:sequence>
          <xsd:element name="customerID" type="xsd:string" />
          <xsd:element name="total" type="xsd:string" />
          <xsd:element name="zip" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="Item_Dt">
        <xsd:sequence>
          <xsd:element name="id" type="xsd:string" />
          <xsd:element name="name" type="xsd:string" />
          <xsd:element name="price" type="xsd:string" />
          <xsd:element name="quantity" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="OrderRequest_Dt">
        <xsd:sequence>
          <xsd:element name="header" type="Header_Dt" />
          <xsd:element name="item" type="Item_Dt" minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  </wSDL:types>
</wSDL:definitions>
```
<wsdl:part name="OrderRequest_Mt" element="p2:OrderRequest_Mt" />
</wsdl:message>
<wsdl:portType name="Order_Si">
  <wsdl:operation name="processOrder">
    <wsdl:input message="p1:p2.OrderRequest_Mt" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="Order_SiBinding" type="p1:Order_Si">
  <soap:binding xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="processOrder">
    <soap:operation xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
      soapAction="http://sap.com/xi/WebService/soap1.1" />
    <soap:input>
      <soap:body xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        use="literal" />
    </soap:input>
  </wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
12.3 Process Context

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://demo.com/context"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://demo.com/context">
    <xsd:element name="Order_Context" type="Context"/>
    <xsd:complexType name="Ctx_Header_Dt">
        <xsd:sequence>
            <xsd:element name="customerID" type="xsd:string"/>
            <xsd:element name="total" type="xsd:string"/>
            <xsd:element name="zip" type="xsd:string"/>
            <xsd:element name="temp" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Ctx_Credit_Dt">
        <xsd:sequence>
            <xsd:element name="approve" type="xsd:boolean"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Ctx_Order_Dt">
        <xsd:sequence>
            <xsd:element name="id" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Context">
        <xsd:sequence>
            <xsd:element name="Header" type="Ctx_Header_Dt"/>
            <xsd:element name="Order" type="Ctx_Order_Dt"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" name="Item" type="Ctx_Item_Dt"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" name="Movie" type="Ctx_Movie_Dt"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
<xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ctx_Item_Dt">
  <xsd:sequence>
    <xsd:element name="id" type="xsd:string"/>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="price" type="xsd:string"/>
    <xsd:element name="quantity" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ctx_Movie_Dt">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="rating" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>