GIS Integration Cookbook for Web Services

Integration of Geographical Services into SAP SCM 7.0 Applications Using Web Services

Marcus Zahn

Project Architect SCM Transportation Network

Stefan Merkel

SCM Business Processes Consultant

Target Audience

- Technical Consultants
- Master Data / System Administrators

SAP AG
Dietmar-Hopp-Allee 16
69190 Walldorf
Germany
T +49/18 05/34 34 34
F +49/18 05/34 34 20
www.sap.com

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.

Disclaimer
Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP’s Support Services and may not be modified or altered in any way.

Typographic Conventions

Type Style	Represents
Example Text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, titles of graphics and tables
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, source code as well as names of installation, upgrade and database tools.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.
`EXAMPLE TEXT`	Keys on the keyboard, for example, function keys (such as `F2`) or the **`ENTER`** key.

Icons

Icon	Meaning
⚠	Caution
❖	Example
💡	Note
⬆	Recommendation
⬆	Syntax

Contents

1 Getting Started

1.1 About this Document

Purpose

This cookbook provides a central starting point for technically integrating SAP external GIS functions into SAP SCM applications using Web services.

Relevant Applications

This cookbook is relevant for all SCM 7.0 applications such as APO Transportation Planning and Vehicle Scheduling (TPVS), Extended Warehouse Management (EWM), and Transportation Management (TM), which use GIS services in a business process. Additional information regarding GIS services in SAP SCM applications can be found here.

⚠ Constraints

This document does not:

- Describe the GIS concept, usage, or configuration in SAP SCM applications (see White Paper for GIS Integration)
- Describe a technical connection to a specific external Geo Service provider or product
- Recommend using a specific Geo Service provider

1.2 History of Changes

The following table provides an overview of the most important changes that have been made in the latest versions.

Document Version	Release Date	Important Changes
1.00	2010/12/21	Initial version
1.10	2012/02/16	Release version for Note 1685381

1.3 GIS Document Structure

The following GIS documents are available:

White Paper for GIS Integration

The white paper *Using Geographical Services with SAP SCM 7.0* explains the GIS concept of SCM applications, generally required settings, and integration strategies (including available data structures).

Integration Cookbook for GIS Web Services

The cookbook *GIS Integration Cookbook for Web Services* explains how external GIS services can be integrated into SCM applications using Web services. This is a general, technical document based on the GIS concepts described in the *White Paper for GIS Integration*.

GIS Vendor Integration Cookbooks

Each GIS Vendor Integration Cookbook explains the implementation process for a specific GIS product. They are based on the general *Integration Cookbook for GIS Web Services*.

Note that the availability of these GIS Vendor Integration Cookbooks depends on different prerequisites and may be provided by different sources (GIS Vendor, SAP consulting, SAP standard development).

2 Web Service Overview

Web services encapsulate functionality and enable distributed applications to be established over the Internet. Web services can be accessed using standard protocols such as HTTP and SOAP.

In the context of GIS, Web services can offer a range of spatial algorithms for different purposes such as the following:

- Geo-Coding
- Geo-Routing (including Distance and Duration Determination)
- Geo-Map display

Accessing Web Services with a Client Proxy

Web service operations are called and the results returned using XML messages. If a machine-readable description of the service written in the Web Service Description Language (WSDL) is available, the client-side code for message processing can be generated automatically. The resulting consumer proxy provides access to the Web services using simple method calls in ABAP. The proxy handles sending and receiving of SOAP messages internally.

Authentication

Authentication may be required at the server that is accessed with the Web service. To submit this authentication, the GIS vendor must provide the required data along with the contract. The ID, for example, can be included in every request header.

Access

Web services enable you to request information in different ways. The following operations are available:

- **Synchronous:** A set of requests is sent to the Web service and the operation returns the results after the calculation has finished.
- **Asynchronous:** For long running calculations, a set of requests is sent to the Web service and the results are retrieved in a subsequent call.

2.1 Web Service Interface

You must check whether the Web service that you want to implement meets the GIS requirements. You must also check how the calls are to be realized and whether the Web services are synchronous or asynchronous.

For more information, see the Web service documentation provided by your selected GIS vendor.

3 Web Service Integration Guide

This section contains the information that you require to integrate Web services into your application system.

⚠ The configuration instructions in this section apply to SAP NetWeaver Release 7.02.

3.1 Web Service Integration

This section explains how you can integrate Web service calls into SAP NetWeaver 7.0 applications using the process controller framework. This essentially involves bundling the calls into a process controller method and including this method in a process controller strategy (for more information, see the link to the *Process Controller* documentation in the References section).

Process Controller Method Structure for Web Services

1. Web service calls require an ABAP class, which can be integrated into the framework. The request instances are the basis for the determinations inside the process controller method. In principle, the method consists of the following blocks:
2. Preparation
    a. Instantiation of the proxy
    b. Cast of the /SCTM/CL_REQUEST instances into required format of to-be-executed service
    c. Preparation of the requests for the Web Service
3. Determination logic
    a. Call of the Web service
    b. Logic for retrieving the result
4. Update of the *Request Object*
    a. Add results to the result attribute of the Request Object
    b. Delete the requests for which a result is available from the request attribute of the Request Object

This process is illustrated in Figure 3-1 below.

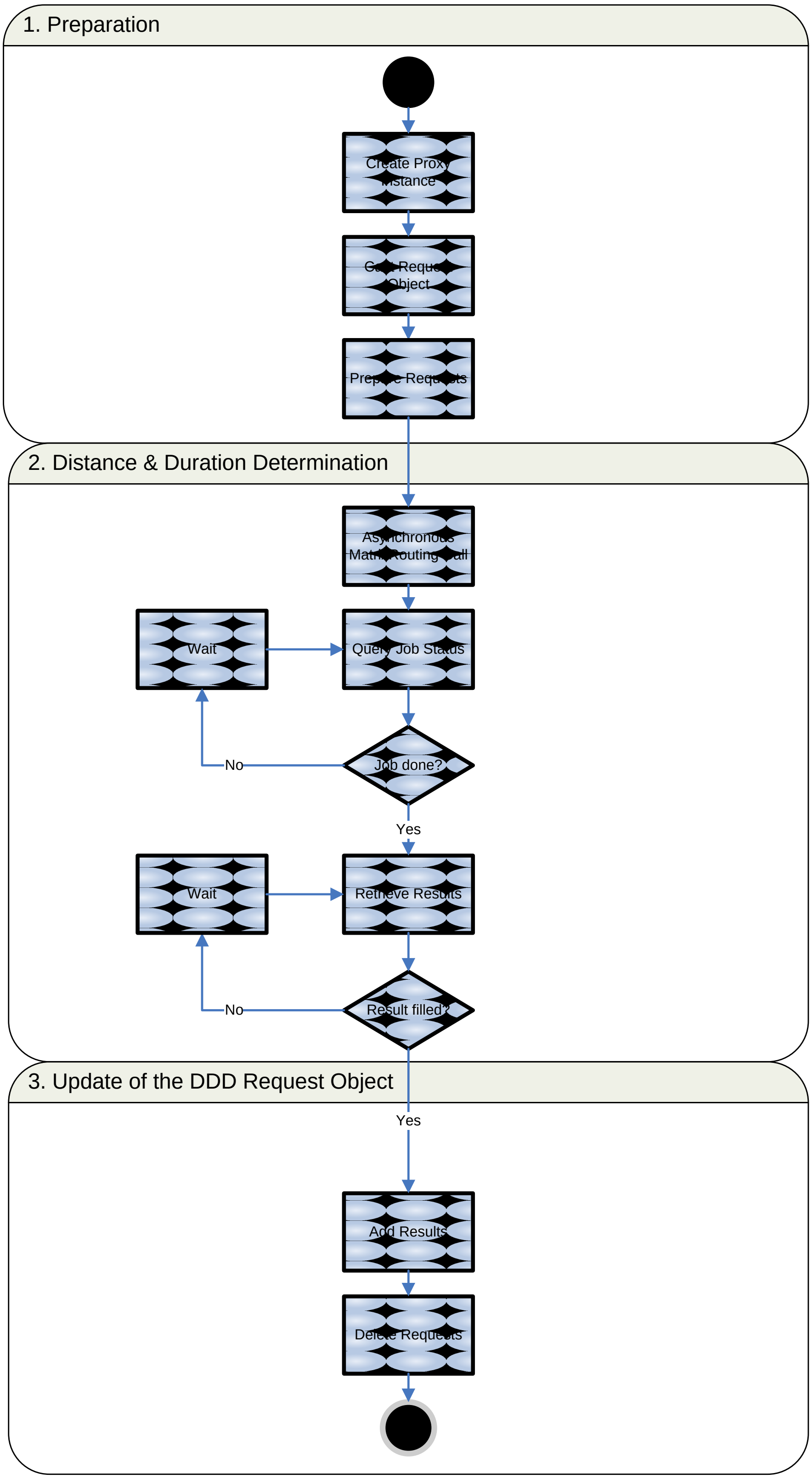


Figure 3-1: Example Activity Diagram for the Process Controller Method

3.2 Web Service Configuration

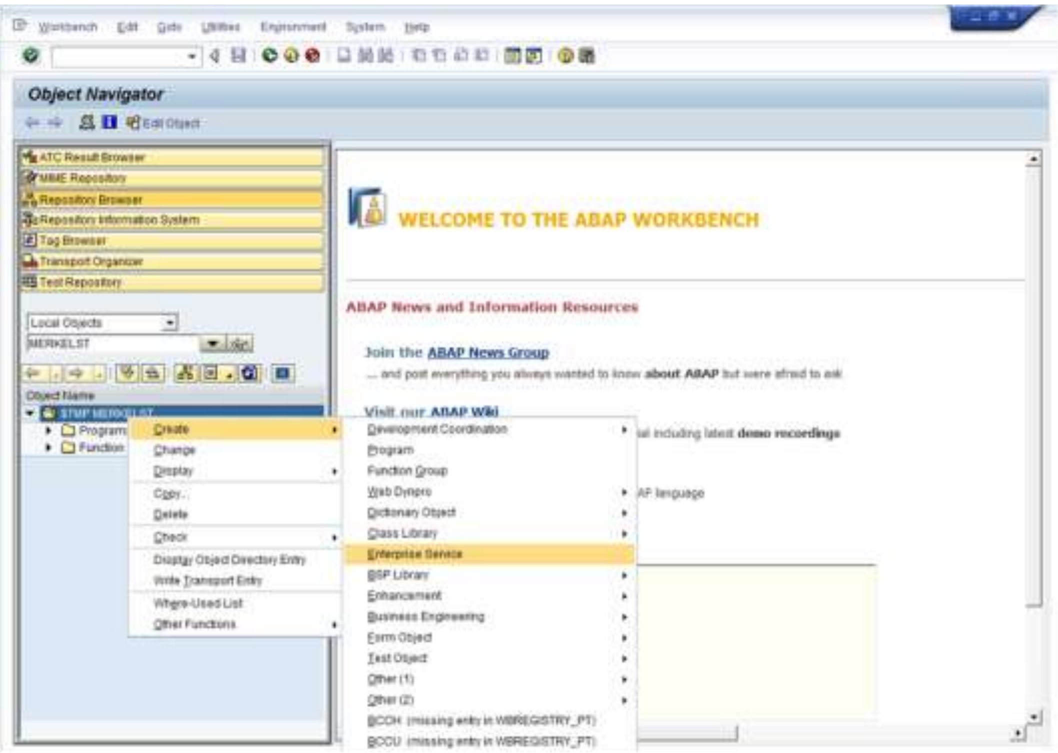
This section describes how to set up the connection to the Web service in your application system. The procedure involves creating a client proxy together with a logical port.

A client proxy is based on a WSDL document and can be used to send or receive messages. It allows the application developer to focus on business functions, while technical aspects such as creating a valid SOAP message are carried out automatically by the proxy implementation. The logical port contains technical, service call-specific data for a Web service client proxy.

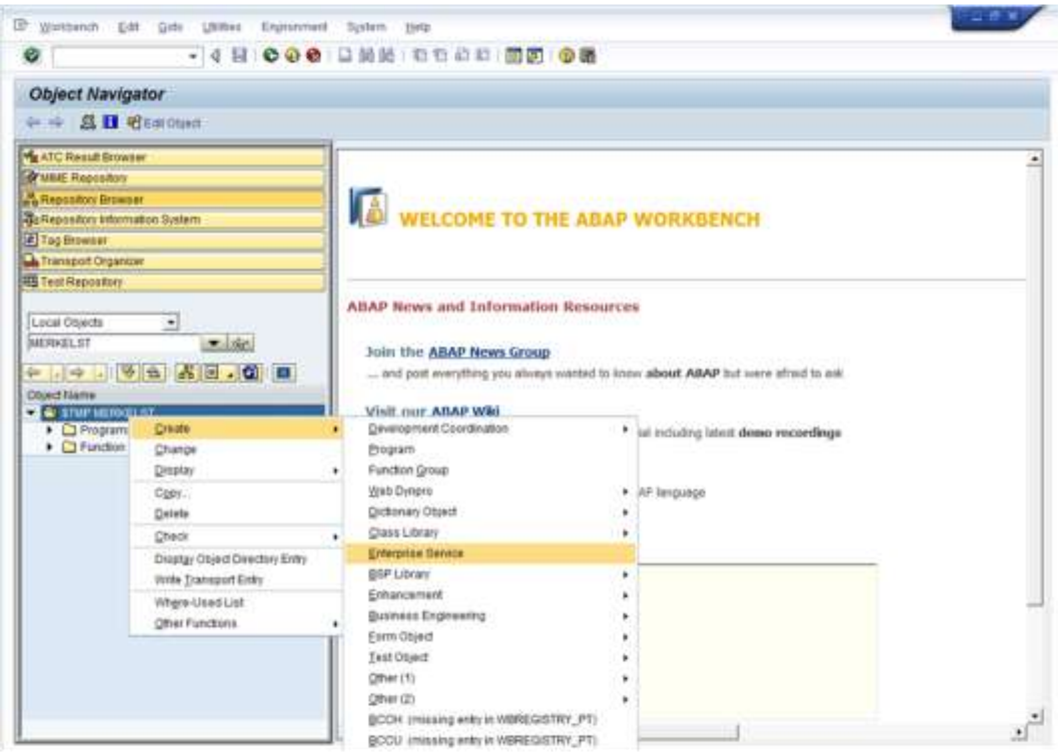
3.2.1 Client Proxy

You can create the client proxy for the Web service in the ABAP. To do so, proceed as follows:

1. In the *Repository Browser* (transaction `SE80`), open the context menu at package level and choose ▶ *Create* →*Enterprise Service* ◀. A dialog box appears.



2. Select *Service Consumer* and choose *Continue*.



WELCOME TO THE ABAP WORKBENCH

ABAP News and Information Resources

Join the ABAP News Group

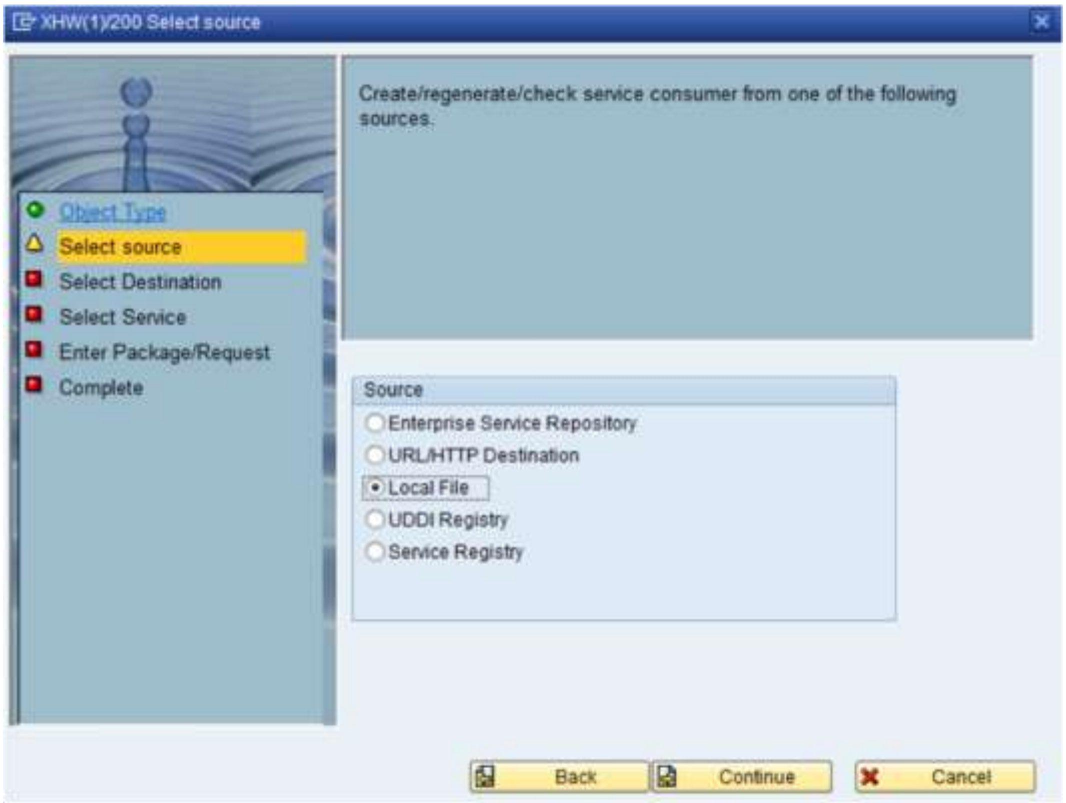
... and post everything you always wanted to know about ABAP but were afraid to ask

Visit our ABAP Wiki

...al including latest demo recordings

...AP language

3. Select the source of the service definition (either *URL/HTTP Destination* or *Local File*) and choose *Continue*.



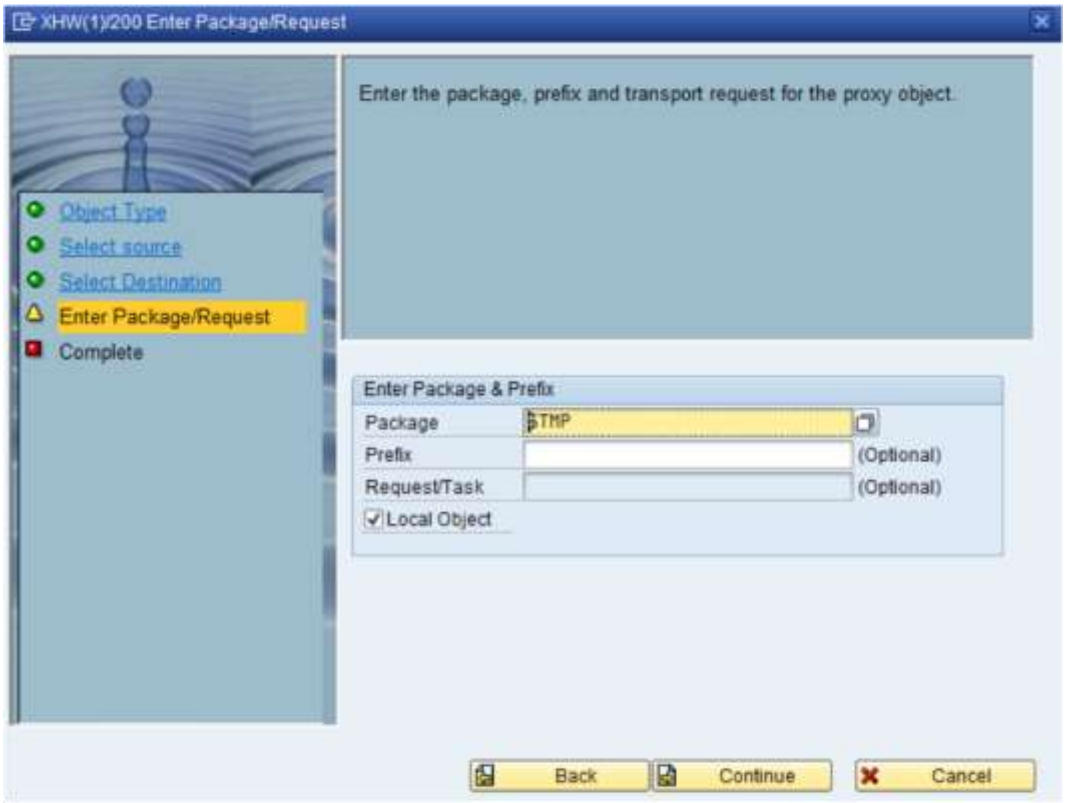
XHW(1)/200 Select source

Create/regenerate/check service consumer from one of the following sources.

Source

○ Enterprise Service Repository
○ URL/HTTP Destination
● Local File
○ UDDI Registry
○ Service Registry

4. Depending on your previous selection, enter the URL or the path to the WSDL document and choose *Continue*.
5. Enter at least the name of the target package and then complete the procedure on the following screen.



XHW(1)/200 Enter Package/Request

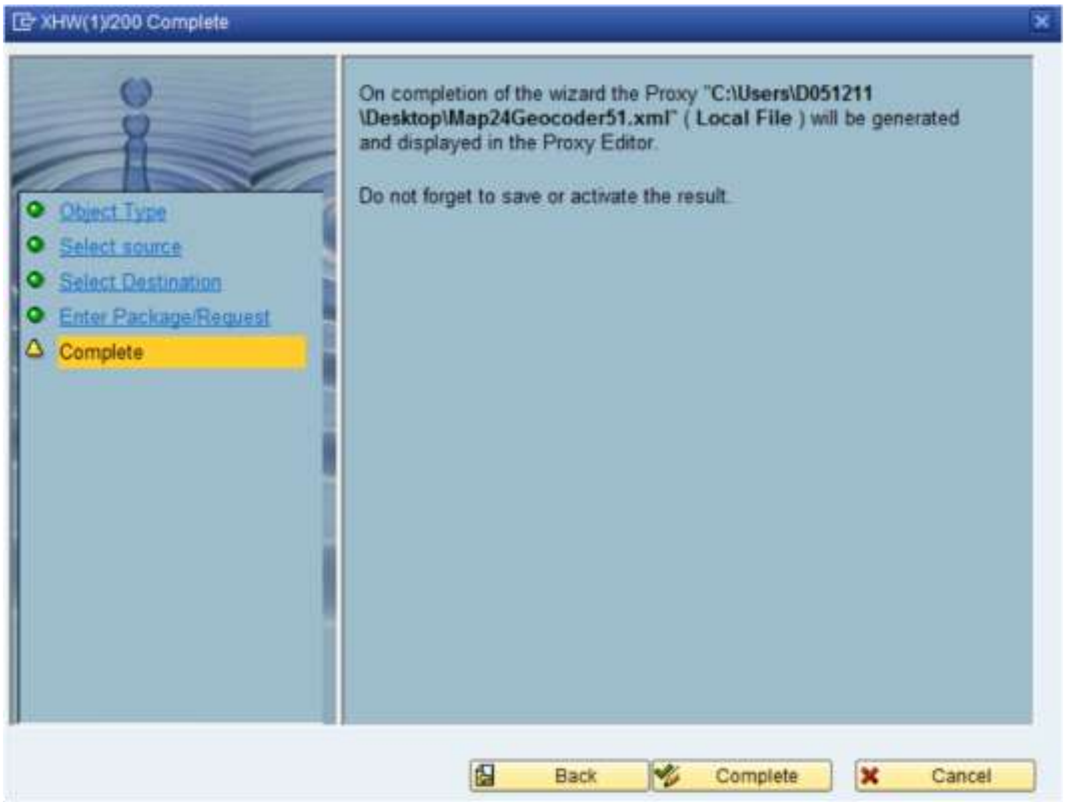
Enter the package, prefix and transport request for the proxy object.

Enter Package & Prefix

Package	$TMP	
Prefix		(Optional)
Request/Task		(Optional)

☑ Local Object

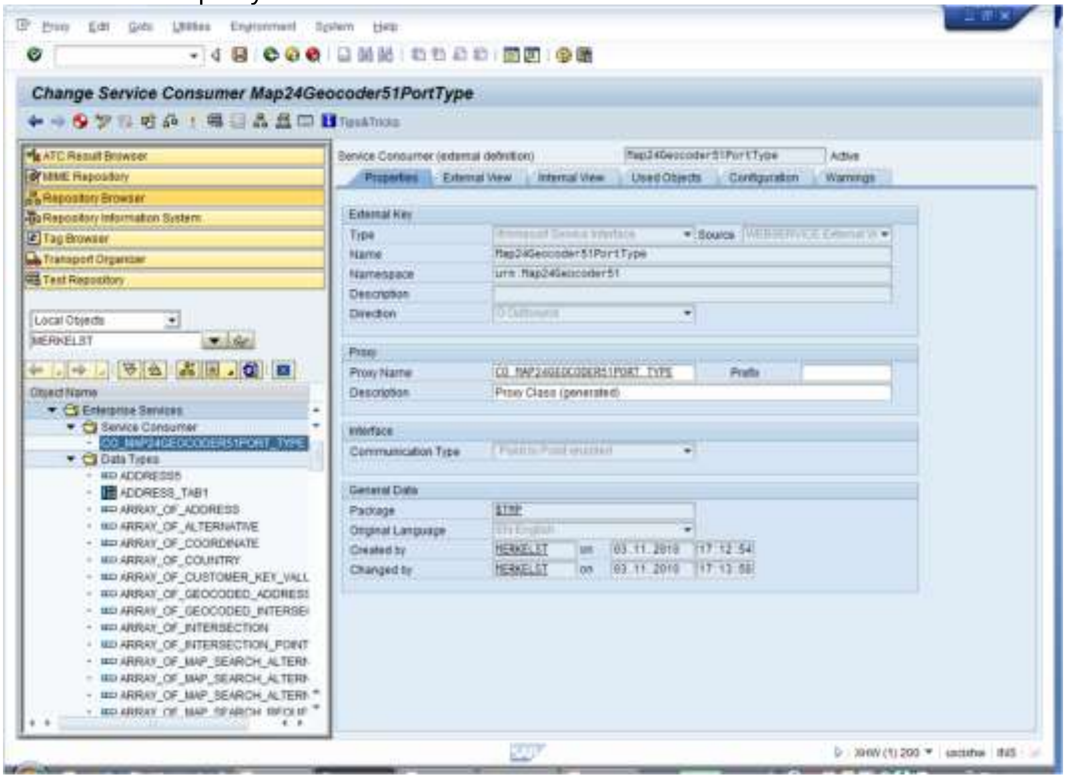
6. Complete the process of creating the client proxy.



On completion of the wizard the Proxy "C:\Users\D051211\Desktop\Map24Geocoder51.xml" (Local File) will be generated and displayed in the Proxy Editor.

Do not forget to save or activate the result.

7. Activate the proxy.



Change Service Consumer Map24Geocoder51PortType

Service Consumer (external definition) Map24Geocoder51PortType Active

External Key

Type	[illegible] Source WEBSERVICE External V
Name	Map24Geocoder51PortType
Namespace	urn:Map24Geocoder51
Description	
Direction	O Outbound

Proxy

Proxy Name	CO_MAP24GEOCODER51PORT_TYPE Prefix
Description	Proxy Class (generated)

Interface

Communication Type	Point-to-Point enabled

General Data

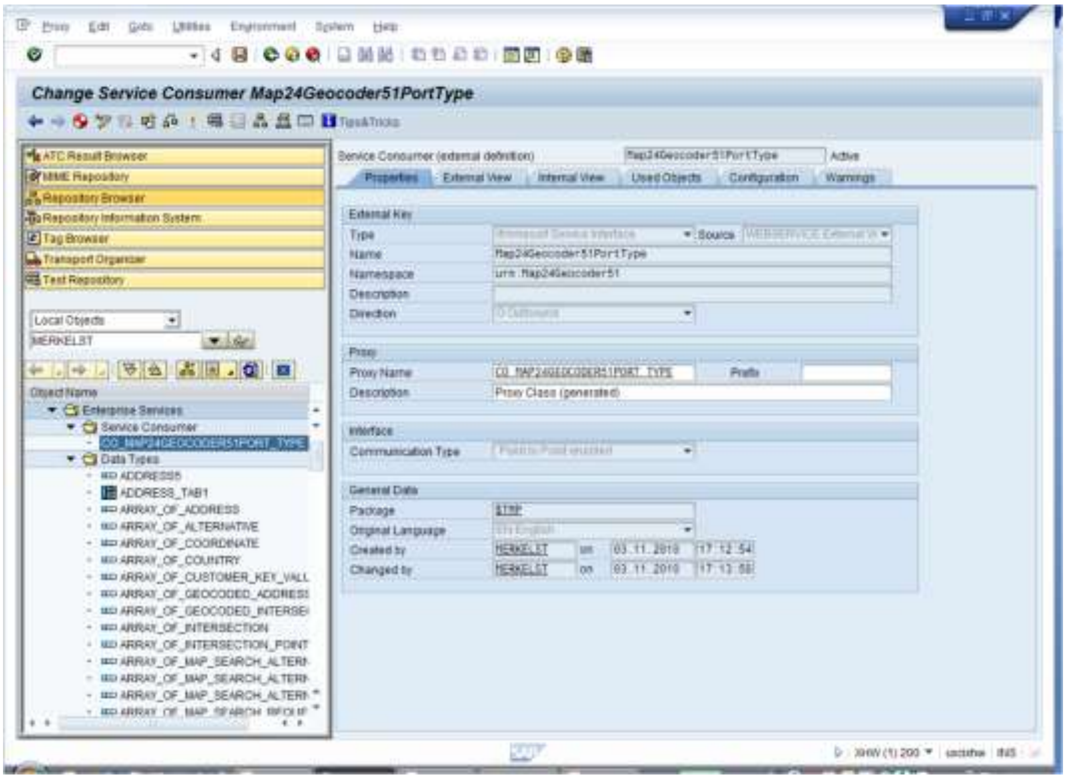
Package	$TMP
Original Language	EN English
Created by	MERKELST on 03.11.2010 17:12:54
Changed by	MERKELST on 03.11.2010 17:13:58

The client proxy is available in the target package under ▶ *Enterprise Services* →*Service Consumer* ◀.

3.2.2 Logical Port

In the *SOA Manager*, you can define a logical port for the client proxy. To do so, proceed as follows:

1. In the SOA Manager, choose the *Service Administration* tab page.
2. Open the *Single Service Configuration*.



Change Service Consumer Map24Geocoder51PortType

Service Consumer (external definition): Map24Geocoder51PortType Active

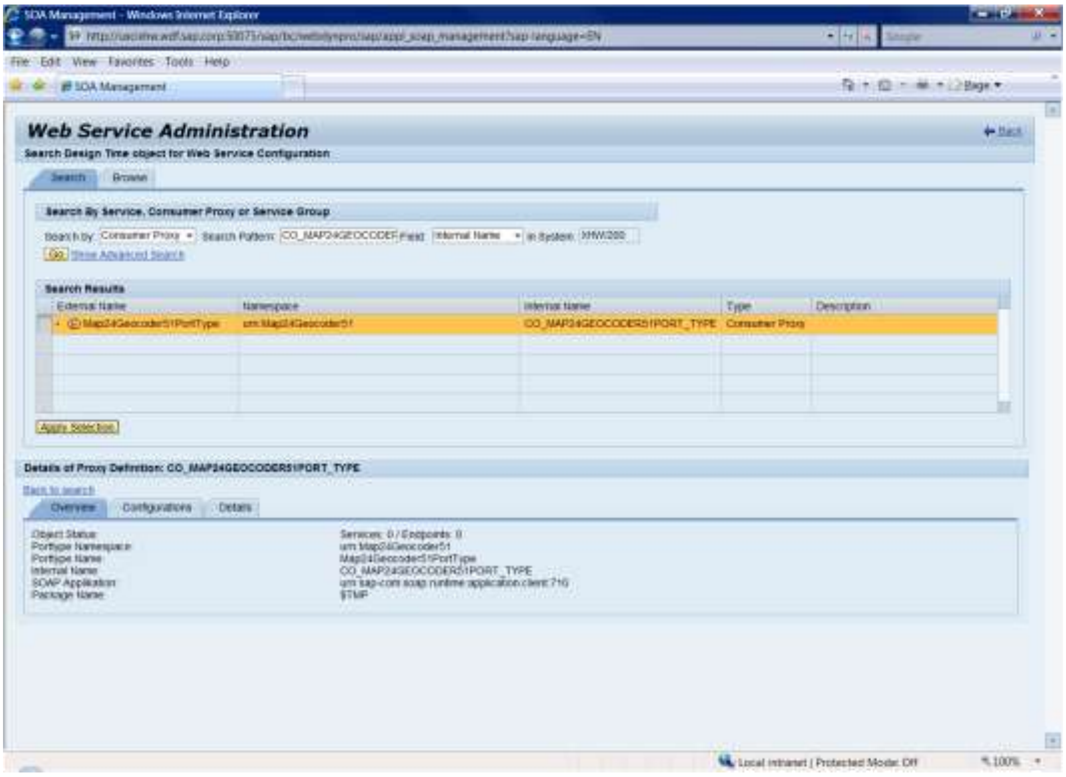
External Key
Type: Enterprise Service Interface Source: WEBSERVICE External Vi
Name: Map24Geocoder51PortType
Namespace: urn:Map24Geocoder51
Description:
Direction: O Outbound

Proxy
Proxy Name: CO_MAP24GEOCODER51PORT_TYPE Prefix:
Description: Proxy Class (generated)

Interface
Communication Type: Point-to-Point enabled

General Data
Package: $TMP
Original Language: EN English
Created by: MERKELST on 03.11.2010 17:12:54
Changed by: MERKELST on 03.11.2010 17:13:58

3. On the *Search* tab page, search by *Consumer Proxy* and enter the name of the client proxy in the *Internal Name* field.



Web Service Administration

Search Design Time object for Web Service Configuration

Search By Service, Consumer Proxy or Service Group

Search by: Consumer Proxy Search Pattern: CO_MAP24GEOCODER Field: Internal Name in System: XHW/200

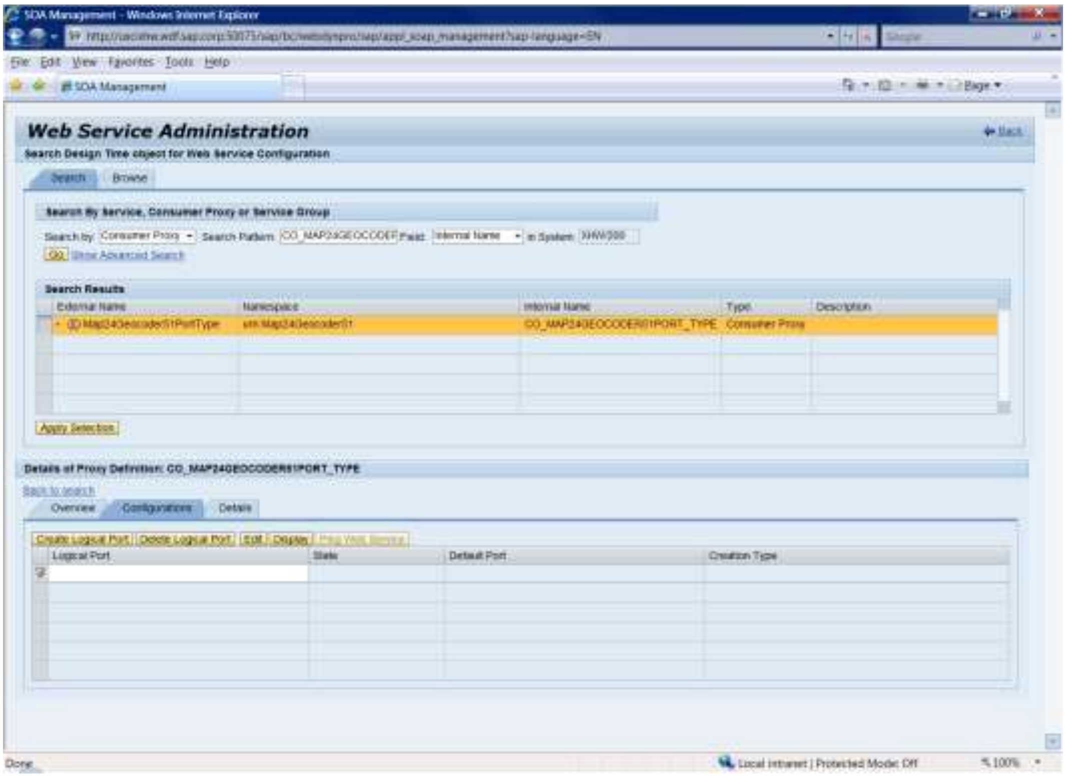
Search Results

External Name	Namespace	Internal Name	Type	Description
Map24Geocoder51PortType	urn:Map24Geocoder51	CO_MAP24GEOCODER51PORT_TYPE	Consumer Proxy	

Details of Proxy Definition: CO_MAP24GEOCODER51PORT_TYPE

Object Status	Services: 0 / Endpoints: 0
Porttype Namespace	urn:Map24Geocoder51
Porttype Name	Map24Geocoder51PortType
Internal Name	CO_MAP24GEOCODER51PORT_TYPE
SOAP Application	urn:sap-com:soap:runtime:application:client:710
Package Name	$TMP

4. Choose *Go*.



Web Service Administration

Search Design Time object for Web Service Configuration

Search By Service, Consumer Proxy or Service Group

Search by: Consumer Proxy Search Pattern: CO_MAP24GEOCODER Field: Internal Name in System: XHW/200

Search Results

External Name	Namespace	Internal Name	Type	Description
Map24Geocoder51PortType	urn:Map24Geocoder51	CO_MAP24GEOCODER51PORT_TYPE	Consumer Proxy	

Details of Proxy Definition: CO_MAP24GEOCODER51PORT_TYPE

Logical Port	State	Default Port	Creation Type

5. Select the relevant record from the search results.
6. Choose *Apply Selection*.
7. In the *Details of Proxy Definition* screen area, choose the *Configurations* tab page.
8. Choose *Create Logical Port*. A dialog box appears.
9. Enter the following data:
    a. Logical Port Name: **`<name>`**
    b. Description: **`<description>`**
    c. Configuration Type: *WSDL Based Configuration*
    d. WSDL Base: Via HTTP Access or Via File
    e. WSDL Location: **`<URL>`** or **`<file path>`**
10. Choose *Apply Settings*.

⚠ If you access the Internet using a proxy server, you need to enter the *Name of the Proxy Host* and the *Port Number of the Proxy Host* on the *Transport Settings* tab page of the logical port.

3.2.3 Proxy Test

You can test the client proxy from within the ABAP Workbench. To do this, proceed as follows:

1. Open the client proxy in the Repository Browser.
2. In the menu, choose *Proxy* →*Test*. A dialog box appears.
3. Select *Use Runtime*.
4. Enter the name of the logical port.
5. Enter the name of the method for the test.
6. Select *Generate template data*.
7. Choose *Execute*. A screen with the request message appears.
8. Modify the request message if necessary.
9. Choose *Execute*.

3.3 Web Service Calls

This section explains how you call Web service methods in ABAP. From now on, Web service operations and Web service methods are used interchangeably.

3.3.1 Client Proxy Instantiation

Before you can call Web service methods, you must create the client proxy.

Listing 1: Instantiating a Client Proxy

```
DATA: lo_proxy TYPE REF TO co_proxy_port_ty.

CREATE OBJECT lo_proxy
  EXPORTING
    logical_port_name = 'GIS1'.
```

3.3.2 Web Service Method Call

The following listing shows how you can call a specific Web service method in ABAP.

Listing 2: Calling a Web Service Method

```
CALL METHOD lo_proxy->execute_determination
  EXPORTING
    input  = ls_request
  IMPORTING
    output = ls_response.
```

⚠ The names of Web service methods may be truncated due to the limit of 30 characters in ABAP.

3.3.3 Web Service Job Status Request

If an asynchronous Web service method is called, you may want to retrieve the status of the calculation. To do so, you need the job ID that is part of the response of the calculation method.

Listing 3: Retrieving the Job Status of an Asynchronous Calculation

```
lv_job_id = ls_response-parameters-async_response-job_id.

ls_status_request-parameters-async_response-job_id = lv_job_id.

CALL METHOD lo_proxy->async_job_status
  EXPORTING
```

```
    input  = ls_status_request
  IMPORTING
    output = ls_status_response.
```

3.3.4 Retrieving Result of Asynchronous Web Service Calls

After the query of the job status has returned *JOB_DONE,* you can retrieve the result of an asynchronous calculation with a separate operation.

Listing 4: Retrieving the Result of an asynchronous Calculation

```
CALL METHOD lo_proxy->async_job_retrieve_calculate1
  EXPORTING
    input  = ls_job_retrieve_request
  IMPORTING
    output = ls_job_retrieve_response.
```

⚠ If the response structure is not filled immediately after the method has been called, wait a moment and try to the retrieve the result again. We recommend that you wrap the retrieve operation in a loop and wait a certain amount of time between the calls.

Example

The following figure shows an example of the workflow for an asynchronous *MatrixRouting* operation. To improve the overview, we have omitted the initial steps and the processing of the result, as well as the error handling.

First, you start the calculation with an asynchronous MatrixRouting operation. The method returns the corresponding job ID so that you can query the status until the calculation has finished successfully. The job status is in this case *JOB_DONE*. Second, you can retrieve the result of the calculation with a separate operation. Repeat this step if the response structure is not filled immediately after the calculation. You can then process the results. For more information, see the following section.

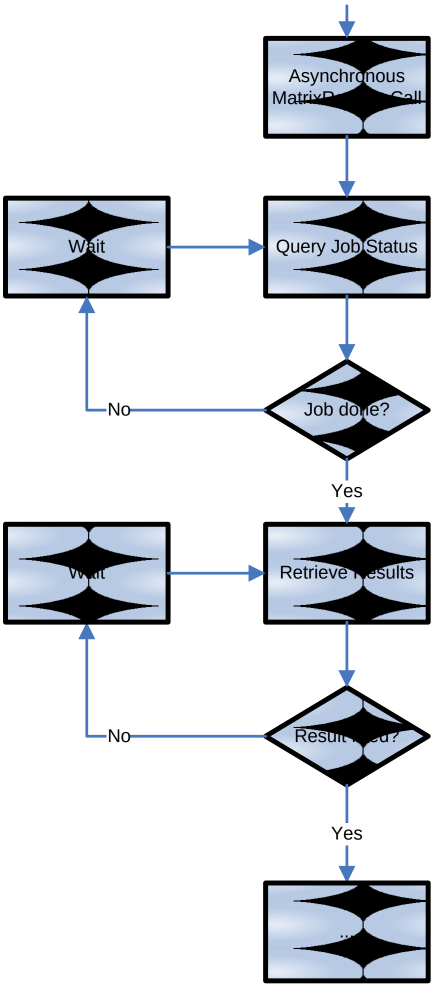


Figure 3-2: Example Activity Diagram of an Asynchronous Matrix Routing Operation

4 Common Problems

The following sections contain answers to common problems that can occur during the integration of the Web service.

4.1 Web Service Configuration

Name problems occurred, names adjusted

Symptom

You have generated a client proxy from a WSDL document as described in Section 3.2. The message area of the ABAP Workbench (transaction `SE80`) contains the warning *Name problems occurred, names adjusted*.

Reason

In ABAP, the names of data elements and methods are restricted to a length of 30 characters.

Solution

Navigate to the *External View* tab page on the *Display Service Consumer* screen. This shows the original names of the data elements and methods provided by the Web service as well as the abbreviated names that have to be used in ABAP.

You can open the *Display Service Consumer* screen by double-clicking the client proxy in the *Repository Browser* of the ABAP Workbench (transaction `SE80`).

The XSD type <type> does not correspond exactly to the ABAP type <type>

Symptom

You have generated a client proxy from a WSDL document as described in Section 3.2. On the *Warnings* tab page of the *Display Service Consumer* screen, a warning message is issued such as *The XSD type <type> does not correspond exactly to the ABAP type <type>*.

Reason

The Web service description uses standard XML data types that are not typically available in ABAP, for example, the data type *double*. The client proxy generator, therefore, maps these data types to corresponding ABAP types.

Solution

Ignore this warning.

Restrictions to the value range ignored

Symptom

You have generated a client proxy from a WSDL document as described in Section 3.2. On the *Warnings* tab page of the *Display Service Consumer* screen, the warning message *Restrictions to the value range ignored* is displayed.

Reason

In the WSDL, it is possible to specify a range for an input parameter of a Web service method. However, this cannot be reflected in ABAP.

Solution

Ignore this warning.

ICM_HTTP_CONNECTION_FAILED

Symptom

You execute the proxy test as described in Section 3.2.3 and a CX_SOAP_CORE exception is raised. The error text reads *Error when calling SOAP Runtime functions: SRT: Processing Error in Internet Communication Framework: ("ICF Error when receiving the response: ICM_HTTP_CONNECTION_FAILED")*.

Reason

A connection to the Internet could not be established.

Solution

Check that you have configured the logical port as explained in Section 3.2.2. If you access the Internet using a proxy server, you must enter the correct proxy settings in the transport settings of the logical port.

4.2 Web Service Calls

Element <element> missing

Symptom

You have tried to call a Web service method in ABAP and an exception of the type `CX_ST_GROUP_MISSING_CASE:XSLT` with the error text *Element '{<namespace>}'<element>' missing* was raised.

Reason

The Web service method returned a response message that does not conform to the WSDL document.

Solution

This is usually a problem on the Web service provider side. In ABAP, you can try to locate the error by examining the response message after the Web service method call was returned. You need to compare it to the method description in the WSDL document.
To obtain the response message from the Web service payload protocol, you first have to announce the payload consumption. Then, you can retrieve the payload, and with it the response message (see Listing 5:).

Listing 5: Reading the Response Message of a Web Service Call

```
DATA: lo_payload_protocol TYPE REF TO if_wsprotocol_payload,
      lo_payload          TYPE REF TO if_ws_payload,
      lv_payload_response TYPE string.
```

```
lo_payload_protocol ?=
   lo_proxy->get_protocol( if_wsprotocol=>payload ).
lo_payload_protocol->announce_payload_consumption( ).

lo_payload =
   lo_payload_protocol->get_sent_response_payload( ).
lv_payload_response = lo_payload->get_xml_text( ).
```

If there are XML elements in the response message without a qualified namespace and the parameter *elementFormDefault* is set to *qualified* in the WSDL document, check whether you have configured the logical port as described in Section 3.2.2. It is important that you set the parameter *useXmlNS* to *true* in the transport settings of the logical port.

No logical port exists

Symptom

You tried to instantiate a client proxy and an exception of type `CX_AI_SYSTEM_FAULT` with the error text *No logical port '`<logical port>`' exists for the proxy class '`<proxy class>`'* was raised.

Reason

No logical port exists for the client proxy or the name of the logical port is not specified correctly at the instantiation of the client proxy.

Solution

Check whether you created a logical port for the client proxy as described in Section 3.2 and make sure that you have specified the correct name of the logical port when instantiating the client proxy.

Response of an asynchronous operation is empty

Symptom

You have called an asynchronous operation of the Web service to retrieve the result after the query of the job status returned *JOB_DONE.* The response structure obtained is incomplete.

Reason

Unknown

Solution

The response structure may not be ready immediately after the calculation has finished. Wait and attempt to retrieve the response again.

The query of the operation status returns JOB_FAILED

Symptom

You have called the *async_job_status* operation to query the status of an asynchronous calculation and obtained the result *JOB_FAILED*.

Reason

Invalid data may have been included in the calculation. The Web service aborts the entire job if a single coordinate is incorrect and sets the job status to *JOB_FAILED*.

Solution

Check or update the data used in the call.

5 References

List of Documentations

The following table lists all documentation mentioned in this cookbook.

Title	Where to Find
Process Controller	http://aiokeh.wdf.sap.corp:1080/SAPIKS2/navTreeOpen.sap?_SCLASS=IWB_STRUCT&_SLOIO=6CB897A6A4896244BAC1693F429CE5B3&IWB_INDUSTRY=SCMTM&TMP_IWB_TASK=DISPLAY&RELEASE=800&LANGUAGE=EN&_SEQNUM=7&_LOIO=FF73B4E1E6434EAABB175419E192ED53&_CLASS=XDP_STRUCT

List of Documents

The following table lists all documents mentioned in this Cookbook.

Title	Where to Find
White Paper Using Geographical Services with SAP SCM 7.0	See Note 1685381
Transportation Network Performance Guide	Not yet available

List of SAP Notes

Use the following SAP Notes to implement additional functions. You can access the SAP Notes on SAP Service Marketplace at **http://service.sap.com/notes**.

SAP Note Number	Title	Description
1685381	GIS Integration: Concept and Technology	Central note for all required information regarding GIS services

6 Appendix

6.1 Transactions and Reports

Transactions

Transaction Code	Description
SOAMANAGER	SOA Manager

Report

Report	Description