SMC^3 Integration Cookbook for Web Services

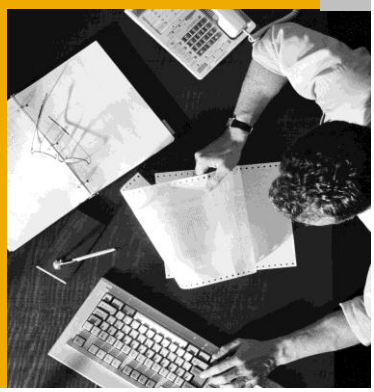
Integration of SMC^3 with SAP Transportation Management Using Web Services

Target Audience

- Technical Consultants
- Master Data/System Administrators

THE BEST-RUN Bbusinesses RUN SAP

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



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 starting point for technically integrating SMC³ with SAP Transportation Management (SAP TM) using Web services.

Relevant Applications

This cookbook is relevant for the charge management area of SAP Transportation Management (SAP TM).

Note

To integrate SMC³ into your system landscape, you must be running one of the following systems:

- SAP Transportation Management 8.0 Support Package 3
- SAP Transportation Management 8.1 Support Package 1
- SAP Transportation Management 9.0 Support Package 2

Constraints

This document does not:

- Describe the SMC³ concept, usage, or configuration in SAP TM applications (see White Paper for SMC³ Integration)
- Describe a technical connection to a specific, external Geo Service provider or product
- Give any recommendations on using an SMC³ 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	September 2011	Initial version
2.00	March 2012	Updated version
3.0	September 2012	Updated version

1.3 SMC³ Document Structure

The following documentation is applicable to SMC³ documents:

Maintaining Calculation Sheets

The Maintaining Calculation Sheets section of this document explains how the SMC³ call is integrated into the charge calculation.

SAP Knowledge Warehouse Documentation

For more information about agreements, see SAP Library for SAP Transportation Management under *Master Data* →*Charge Management* →*Setup of Master Data for Charge Management* →*Agreement*.

2 Technical Configuration

2.1 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.

Accessing Web Services with a Consumer 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 the 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 SMC³ 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.

2.2 Web Service Interface

You must check whether the Web service that you want to implement meets the requirements of SMC³. You must also check how the calls are to be realized and whether the Web services are synchronous or asynchronous.

2.3 Web Service Integration

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

Note

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

This section explains how you can integrate Web service calls into SAP NetWeaver 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

Web service calls require an ABAP class, which can be integrated into the framework. The request instances form the basis for determinations inside the process controller method (that is, preparation, determining logic, and processing). In principle, the method consists of the following blocks:

1. Preparation
    a. Instantiation of the proxy
    b. Cast of the `/SCTM/CL_REQUEST` instance into the required format for the service to be executed
    c. Preparation of the requests for the Web service
2. Determination logic
    a. Call of the Web service
    b. Logic for retrieving the result
3. 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 2-1 below.

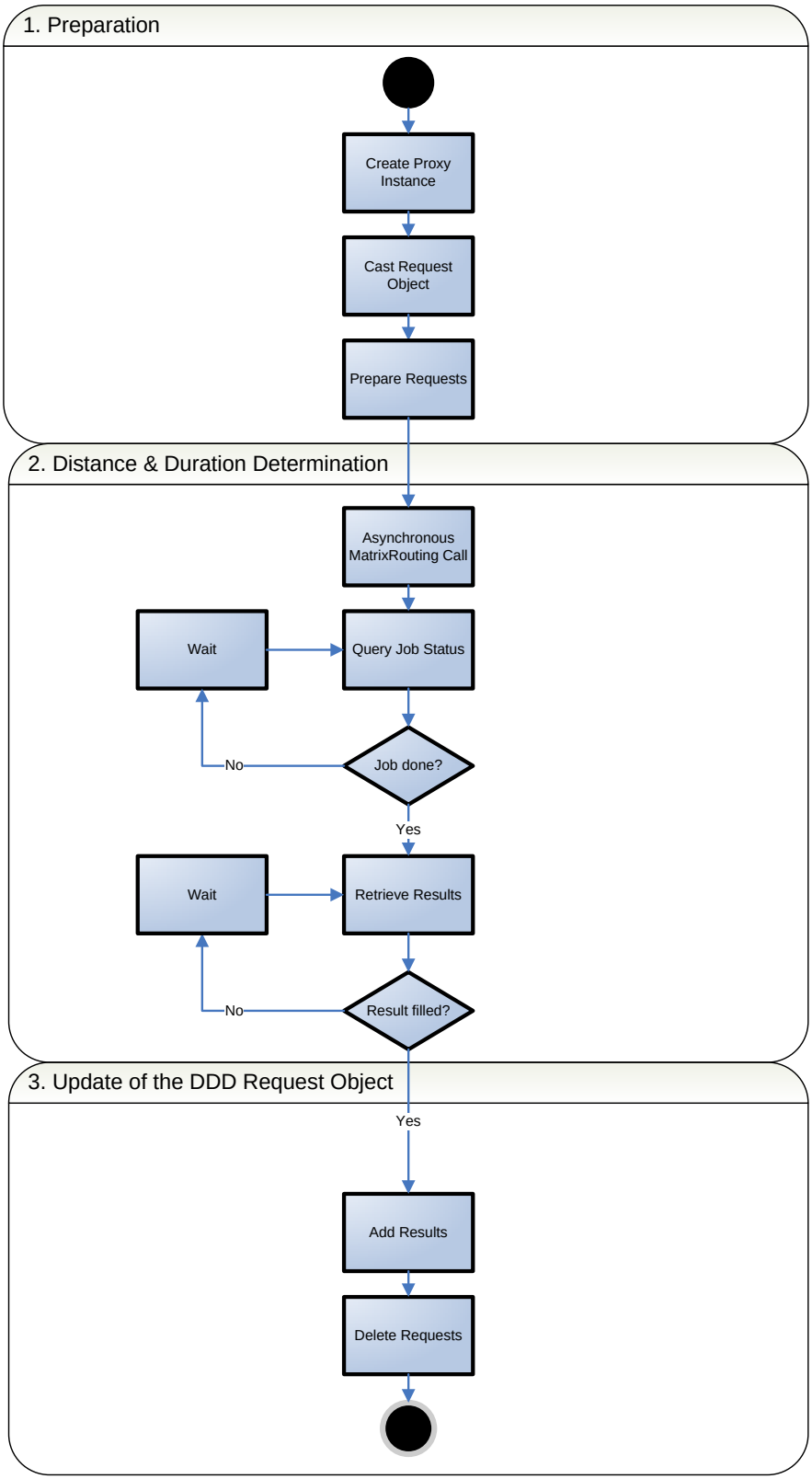


Figure 2-1: Example Activity Diagram for the Process Controller Method

2.4 Web Service Configuration

This section describes how to establish the connection to the Web service in your application system. This involves creating a consumer proxy together with a logical port.

A consumer 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 consumer proxy.

2.4.1 Consumer Proxy

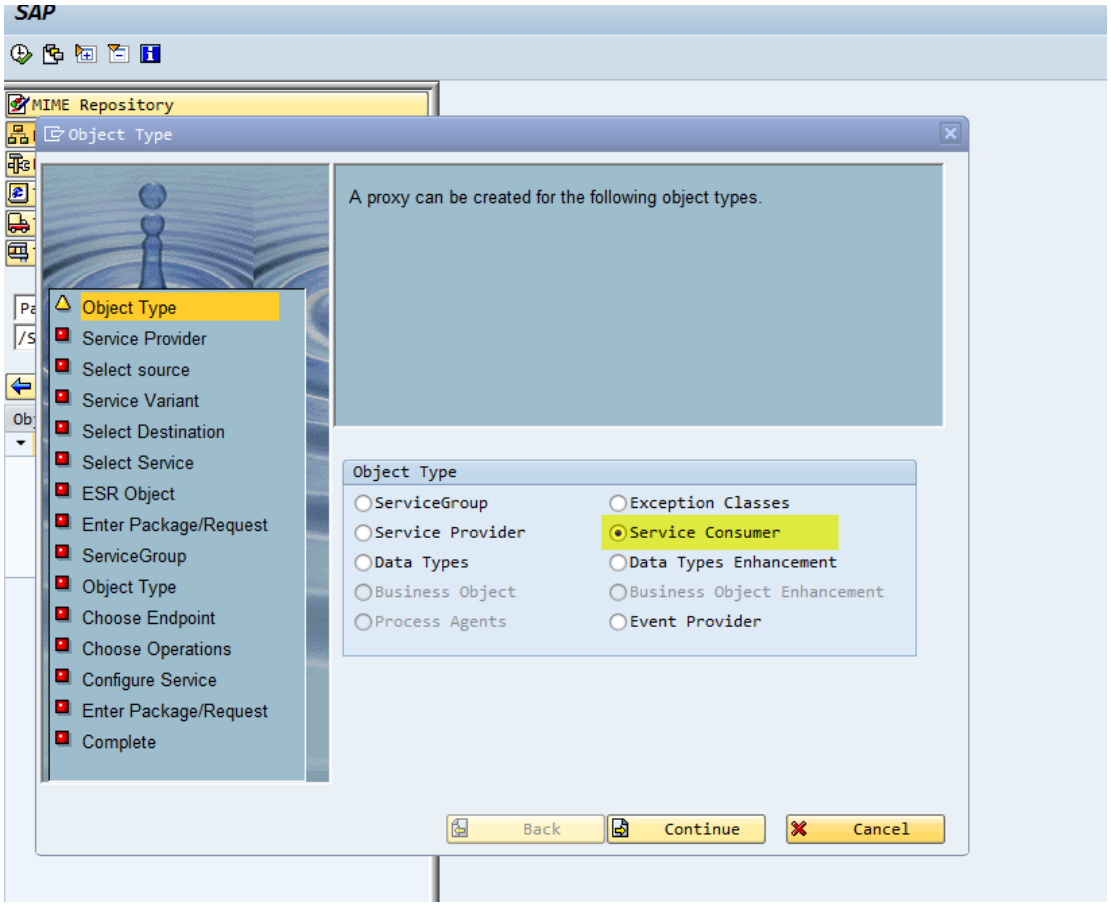
Note

These steps only apply if you are using SAP TM 8.0 SP02 or lower or SAP TM 8.1 SP00.

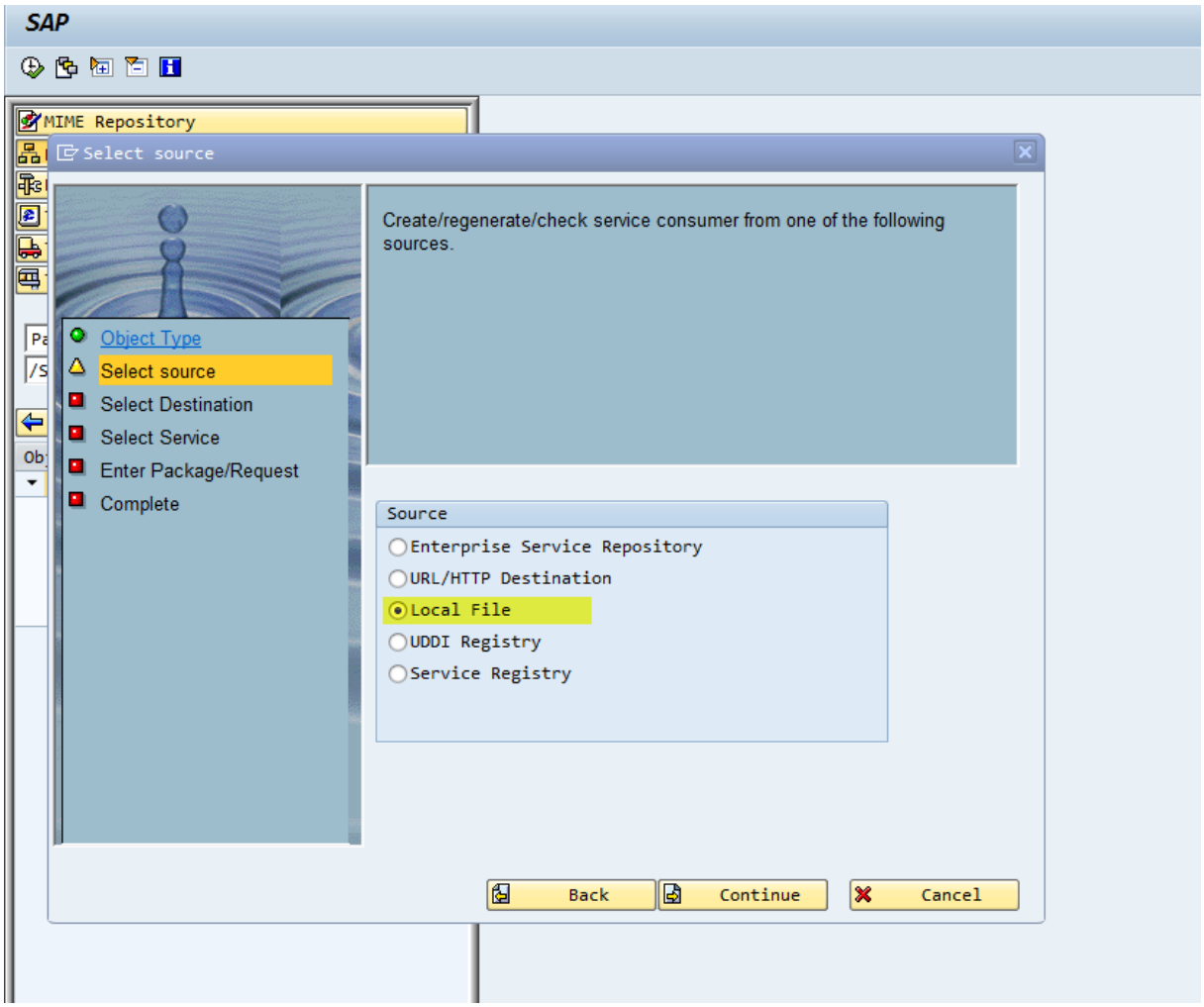
Due to legal restrictions, we do not provide the consumer proxy for the SMC³ Web service itself as part of the standard SAP TM offering. To obtain the WSDL file for the RateWare Web service, contact your SMC³ administrator and then import the WSDL file to generate the consumer proxy in your system landscape. To do so, proceed as follows:

1. Call transaction `SE80`.
2. Enter the package name in which you want to create the consumer proxy, for example, **`ZZSMC3`**.
3. Right-click the package name and choose *Create* →*Enterprise Service* from the context menu.
   The *Enterprise Service Creation* wizard appears.

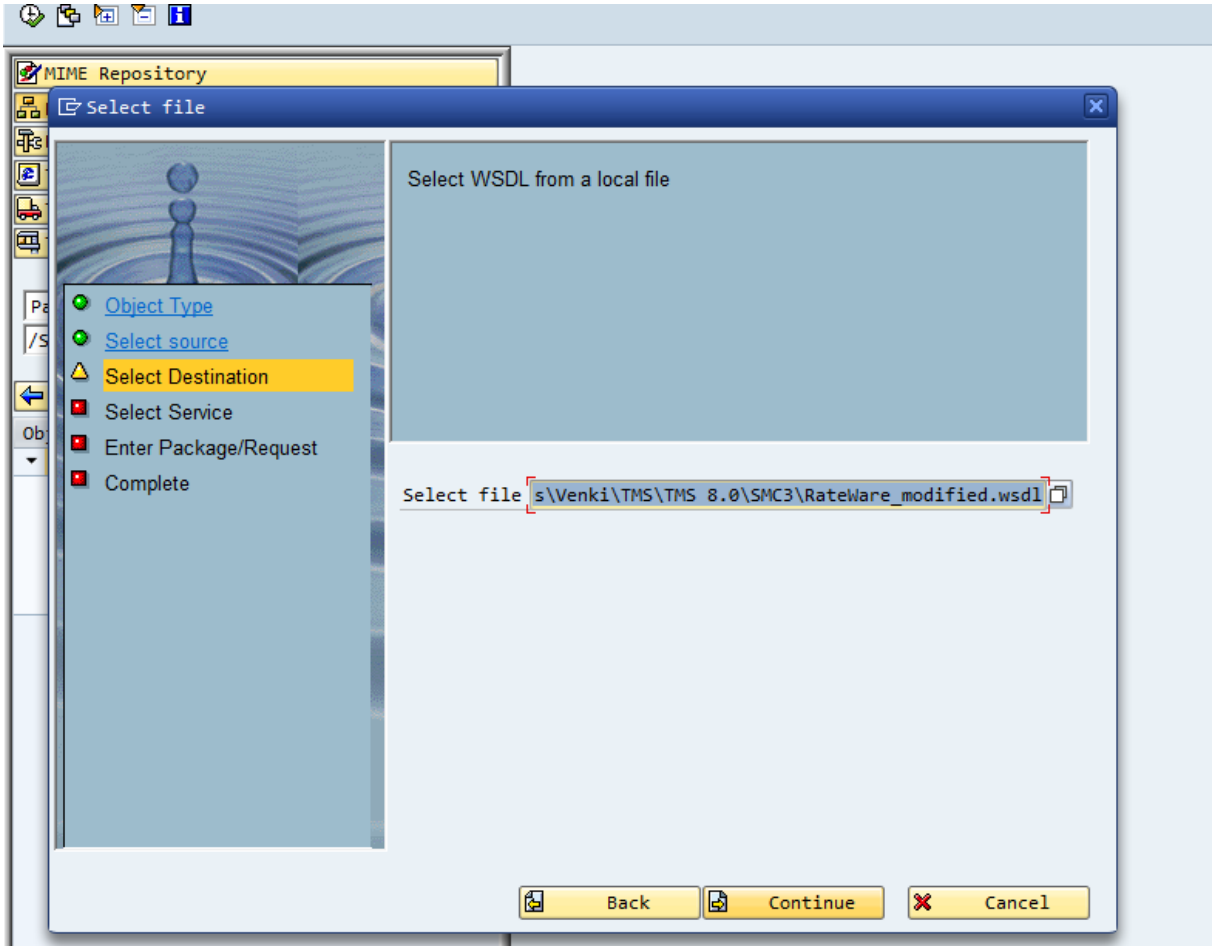
4. Choose the *Service Consumer* radio button.



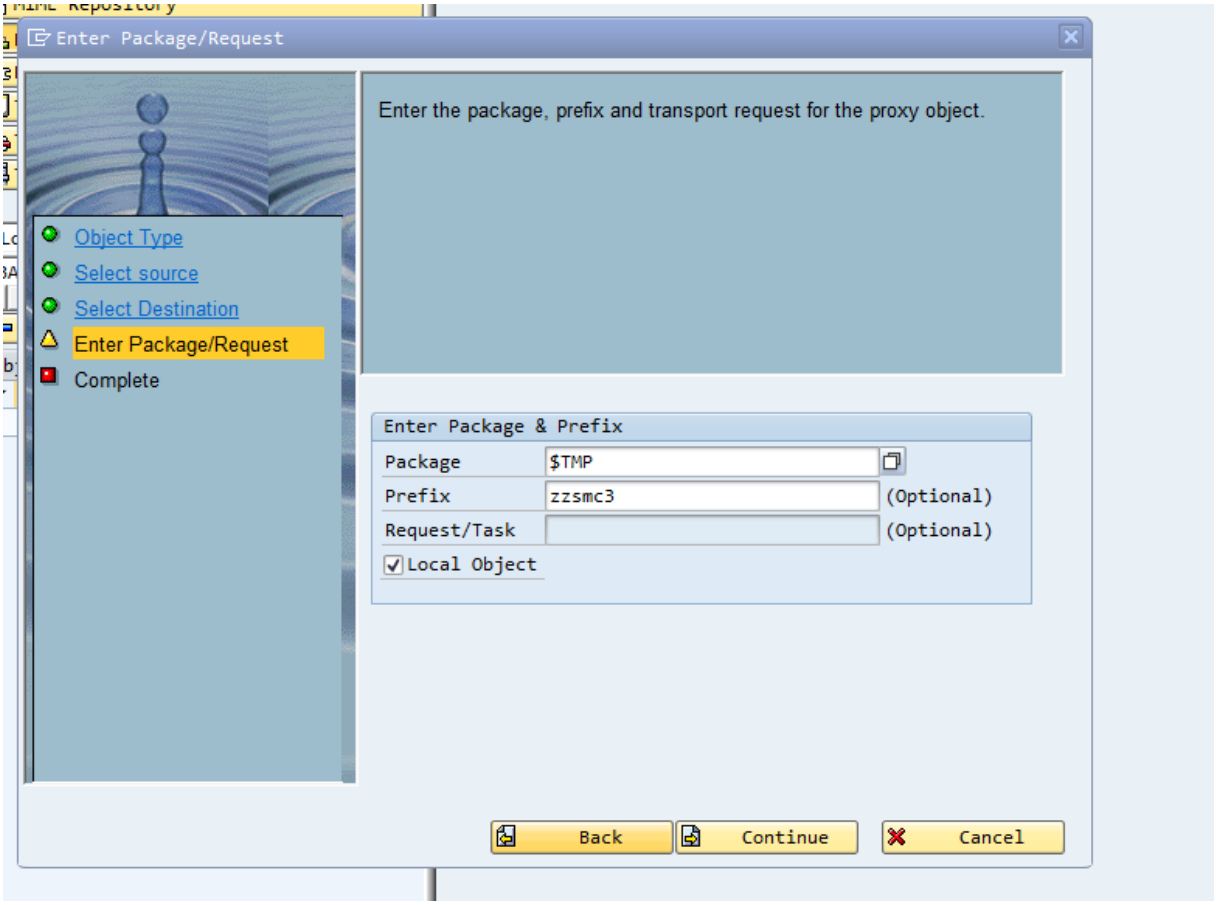
5. Choose the source of the WSDL file and choose the *Local File* radio button.



6. Specify the file path for the WSDL file in the local file system.



7. Enter the package details as required.



8. Choose *Complete.*

This generates the consumer proxy for the SMC³ Web service (for example, `ZZSMC3CO_RATE_WARE_XLPORT_TY`). The proxy has two operations that can be used to determine the rate:

- `LTLRATE_SHIPMENT_MULTIPLE`
- `REQUEST_SET_WITH_CONTRACTS`

Depending on whether you use the first or the second operation, implement the `<METHOD_NAME>` or `<METHOD_NAME>_ENH` calculation method respectively as given below. In this document, we assume that you use the operation `REQUEST_SET_WITH_CONTRACTS` and so method `<METHOD_NAME>_ENH` is used.

2.4.1.1 Implementing Calculation Methods

Note

These steps only apply if you are using SAP TM 8.0 SP02 or lower or SAP TM 8.1 SP00.

To calculate charges using SMC³, you must carry out the following steps manually:

1. Copy standard class `/SCMTMS/CL_TCC_GET_RATE_SMC3` and create a new class such as `ZCL_TCC_GET_RATE_SMC3`.
2. Navigate to method `CALL_SMC3_ENH` and uncomment the shipped code:
   a. Change the type of variable `LO_PROXY` from `/SCMTMS/CO_RATE_WARE_XLPORT_TY` to `ZZSMC3CO_RATE_WARE_XLPORT_TY` (that is, the name for the consumer Web service proxy that you generated above) and save the method implementation.
   b. Change the signature of the method as follows:
      - Change the type of `IS_REQUEST` from `ANY` to `ZZSMC3REQUEST_SET_WITH_CONTRA1` (importing data type of the operation in `SPROXY`).
      - Change the type of `ES_RESPONSE` from `ANY` to `ZZSMC3REQUEST_SET_WITH_CONTRAC` (exporting data type of the operation in `SPROXY`).
3. Navigate to method `MAP_DATA_TO_SMC3_ENH` and uncomment the shipped code.
   a. Change the following definitions:
      - `/SCMTMS/SHIPMENT_REQUEST` to `ZZSMC3SHIPMENT_REQUEST`
      - `/SCMTMS/LINE_ITEM` to `ZZSMC3LINE_ITEM`
      - `/SCMTMS/LTLREQUEST_DETAIL` to `ZZSMC3LTLREQUEST_DETAIL`
      - `/SCMTMS/REQUEST_SET` to `ZZSMC3REQUEST_SET`
      - `/SCMTMS/FREIGHT` to `ZZSMC3FREIGHT`
   b. Save the method.
   c. Change the signature of method `CS_SMC3_REQUEST` to `ZZSMC3REQUEST_SET_WITH_CONTRA1` (importing data type of operation in `SPROXY`).
4. Navigate to method `MAP_DATA_FROM_SMC3_ENH`, uncomment the shipped code, and save your changes.
5. Change the signature of the method so that the type of `IS_SMC3_SHIP_RESP` is `ZZSMC3REQUEST_SET_WITH_CONTRA1` (importing data type of the operation in `SPROXY`).
6. Save and activate class `ZCL_TCC_GET_RATE_SMC3`.

2.4.1.2 Configuration

The following objects are shipped as standard with SAP TM 8.0:

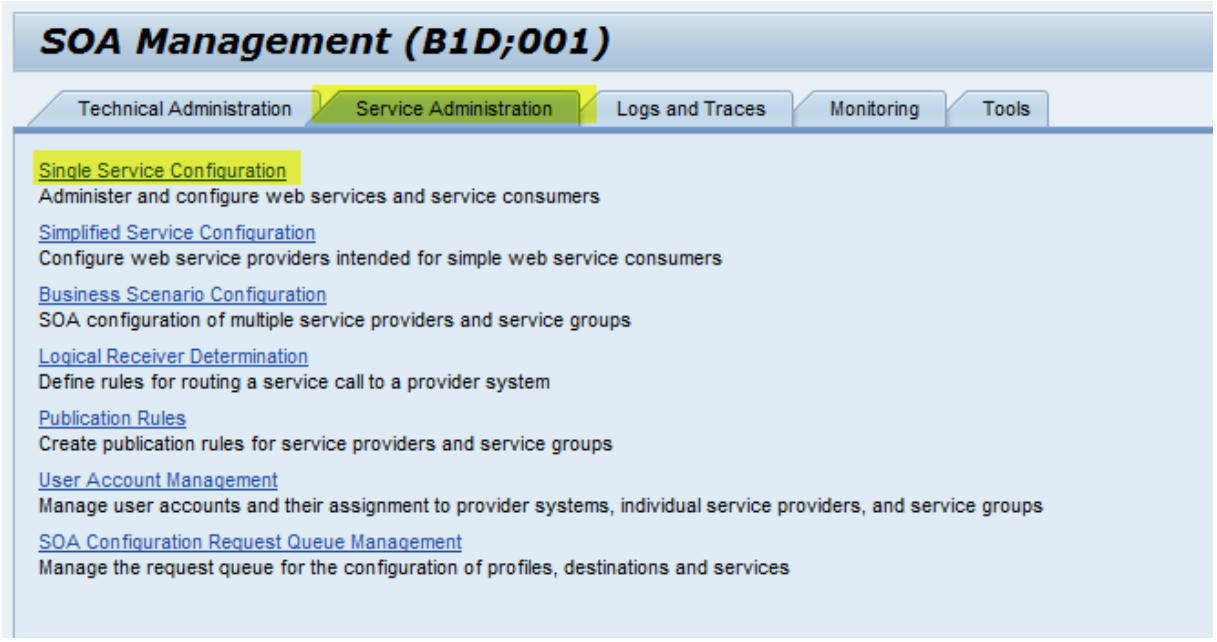
- Report `/SCMTMS/TCC_SEC_STORE_LOGON`, which enables customers to store the SMC³ license key in the secure store
- View cluster `/SCMTMS/VC_SMC3`, which provides configuration options for the charge types, accessorials, and connection parameters for the SMC³ call

You must carry out a set of configuration steps to implement these features on the customer side. These steps are explained in more detail below.

2.4.2 Logical Ports

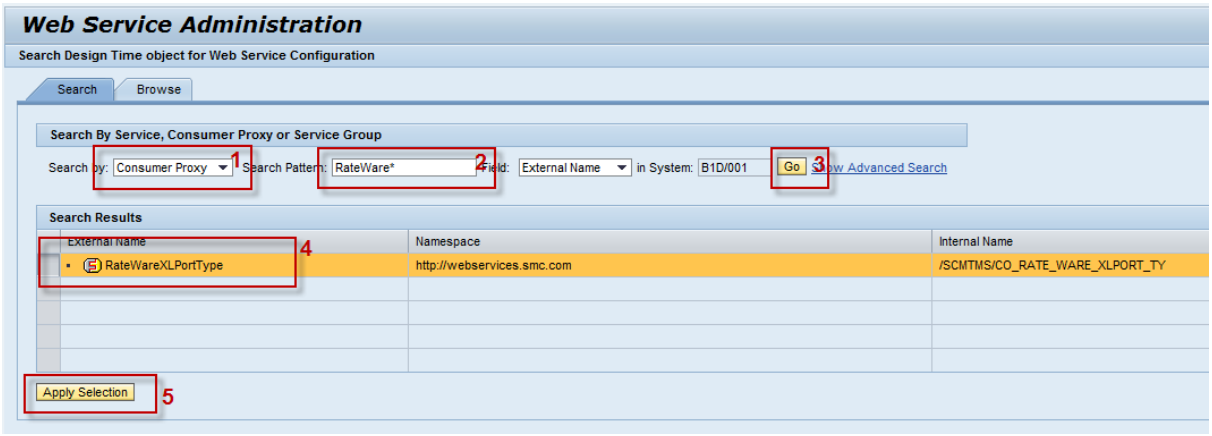
To complete the integration process, you must first use transaction `SOAMANAGER` in the SAP back-end system to define the bindings or logical ports. To do so, proceed as follows:

1. Call transaction `SOAMANAGER` and choose *Single Service Configuration.*



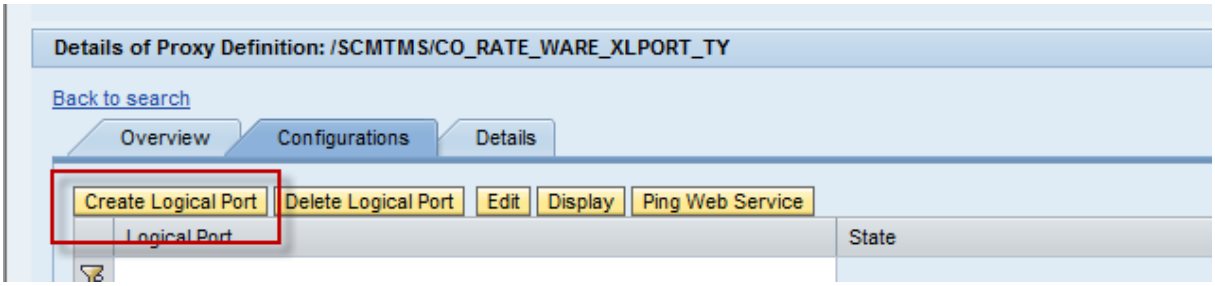
2. Find the consumer proxy with the external name `RateWareXLPortType`.

3. Choose the *Apply Selection* button.

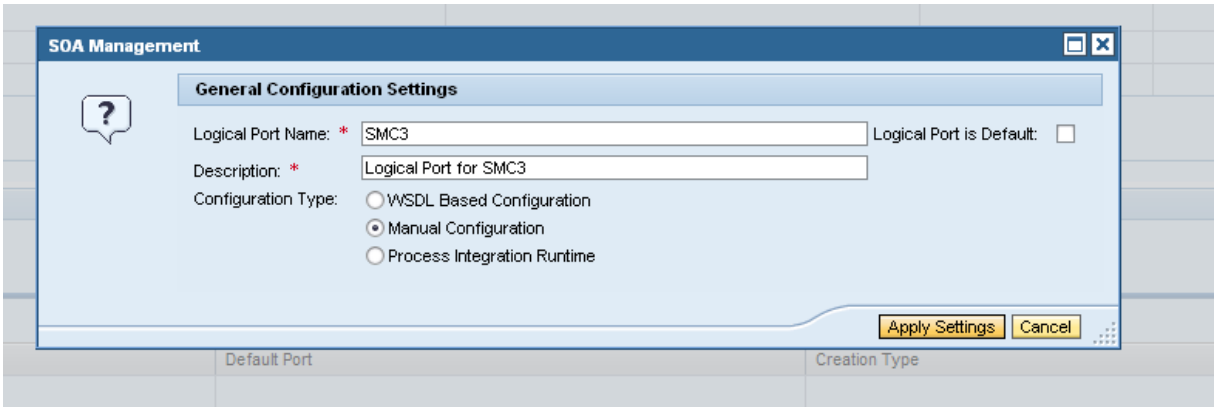


4. Choose the *Configurations* tab page and check whether a logical port is already assigned to the Web service (only one logical port is necessary).

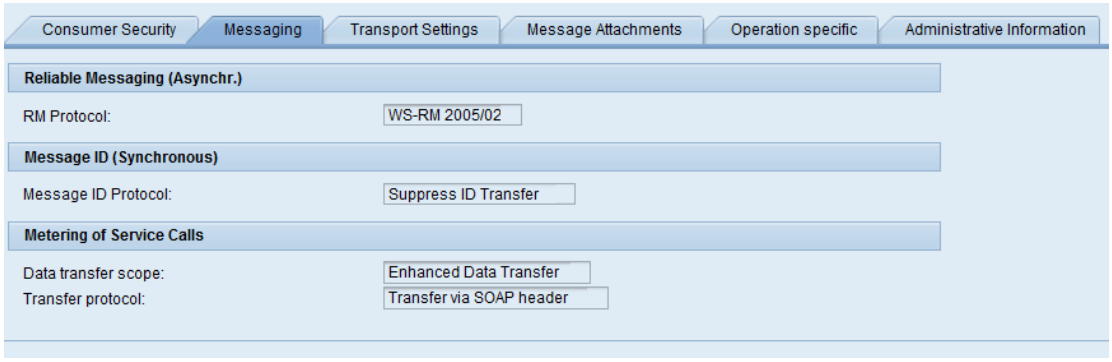
5. If a logical port is not assigned, choose the *Create Logical Port* button.



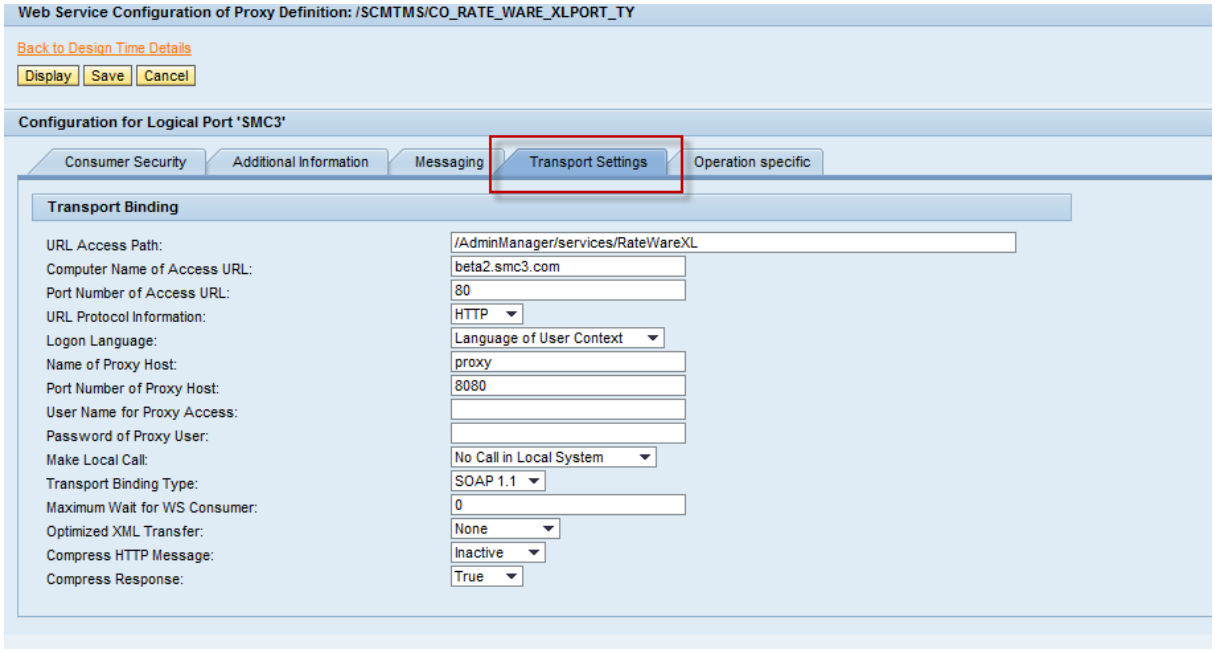
6. Enter the name of the logical port (for example, SMC³), choose the configuration type *Manual Configuration,* and choose *Apply Settings.*



7. Go to the *Messaging* tab and set the *Message ID Protocol* to *Suppress ID Transfer* as shown below.



8. Choose the *Transport Settings* tab page and enter the information as shown below. Note that the value entered in the *Computer Name of Access URL* field is based on information provided by SMC³. You must confirm this path with SMC³ before carrying out a customer implementation.



9. Save your settings.

2.4.3 Testing the Consumer Proxy

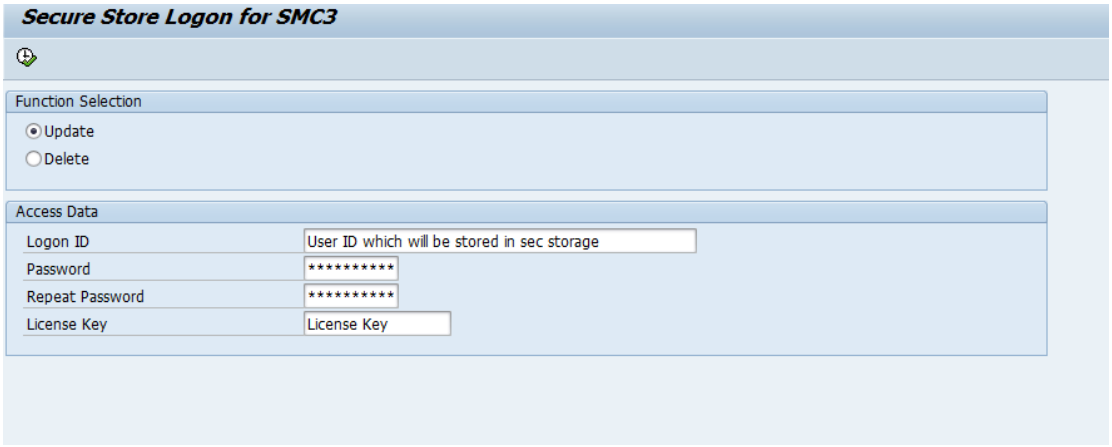
You can test the consumer proxy from within the ABAP Workbench. To do this, proceed as follows:

1. Open the consumer proxy in the Repository Browser.
2. Choose *Proxy →Test*. A dialog box appears.
3. Choose *Use Runtime*.
4. Enter the name of the logical port.

5. Enter the name of the method to test.
6. Choose *Generate Template Data*.
7. Choose *Execute*. A screen with the request message appears.
8. Modify the request message if necessary.
9. Choose *Execute*.

2.4.4 Defining User Access

1. Call transaction `SE38` and execute report `/SCMTMS/TCC_SEC_STORE_LOGON`.



2. Enter your logon ID (provided by SMC³), password (used to access the SAP Secure Store in the SAP back end), and license key (provided by SMC³).
3. Choose *Execute* to save the information in the secure store.

2.5 Web Service Calls

Note

This section only applies if you are using SAP TM 8.0 SP02 or lower or SAP TM 8.1 SP00.

In this section, you learn how to call Web service methods in ABAP. From now on, Web service operations and Web service methods are used interchangeably.

2.5.1 Consumer Proxy Instantiation

As a prerequisite for calling Web service methods, the consumer proxy needs to be created.

Listing 1: **Instantiating a Consumer Proxy**

```
DATA: lo_proxy TYPE REF TO co_proxy_port_ty.

CREATE OBJECT lo_proxy
  EXPORTING
    logical_port_name = 'SMC3'.
```

2.5.2 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.
```

Caution

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

2.5.3 Web Service Job Status Request

If an asynchronous Web service method is called, you may want to retrieve the status of the calculation. To this end, 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.
```

2.5.4 Retrieving the 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.
```

Caution

The response structure is sometimes not filled immediately after the method call. Should this be the case, wait a moment and try to the retrieve the result again. It is advisable to wrap the retrieve operation in a loop and wait a certain amount of time between the calls.

Example

The following figure shows how the workflow of an asynchronous *MatrixRouting* operation may appear. For a better understanding, the initial steps and the processing of the result as well as the error handling are omitted.

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. In this case, the job status is *JOB_DONE*. In the next step, you can retrieve the result of the calculation with a separate operation. Repeat this step if the response structure was not filled immediately after the calculation. You can then process the results. For more information, see the following section.

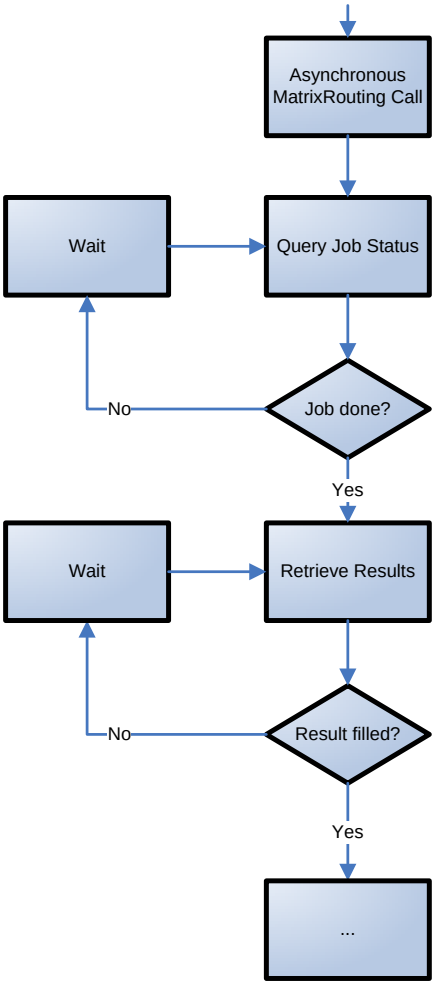


Figure 2-2: Example Activity Diagram of an Asynchronous Matrix Routing Operation

3 Business Configuration

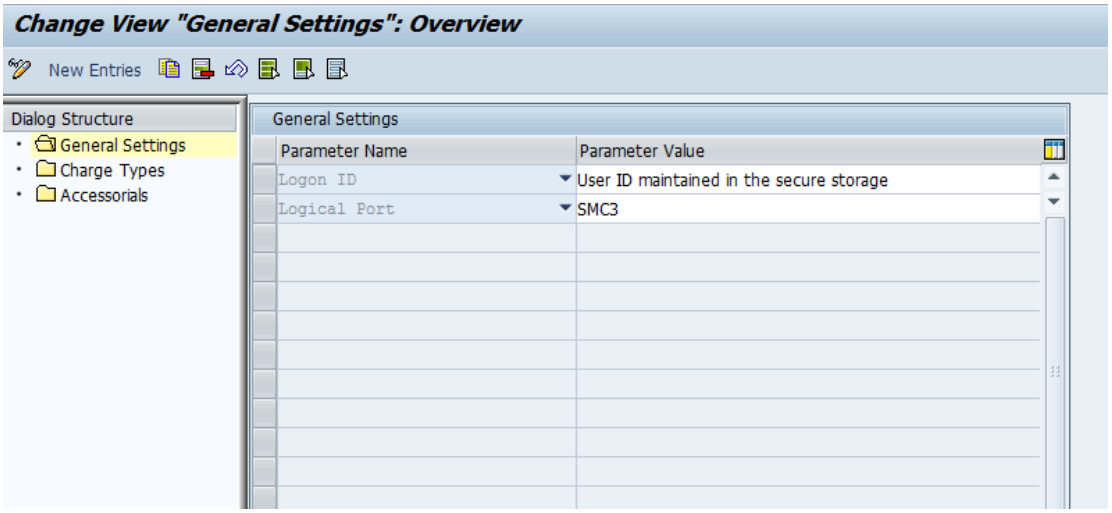
SMC³ RateWare benefits shippers, distributors, logistics service providers, and transportation exchanges in the following ways:

- Less than truck load (LTL) shipment costs and prices can be managed along with carrier-specific discounts, minimum charges, surcharges, deficit ratings, and freight of all kinds (FAK) charges.
- Manual charges can be added to SAP TM alongside SMC³ charges.

To obtain information from SMC³ and include it in a freight agreement or forwarding agreement, you must configure the following settings.

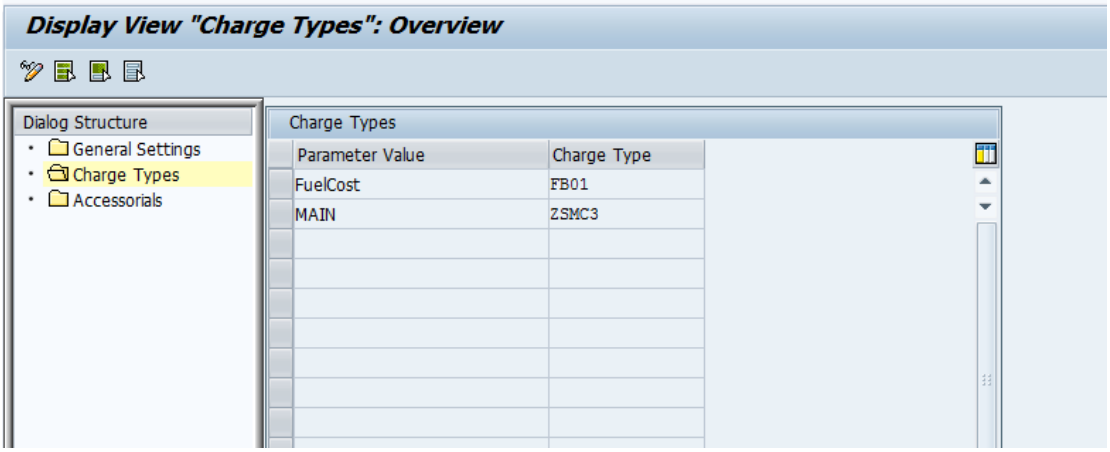
3.1 Configuring Charge Calculation

1. Call transaction `SM34` to access view cluster `/SCMTMS/VC_SMC3` and maintain the charge calculation-specific program parameters for SMC³.

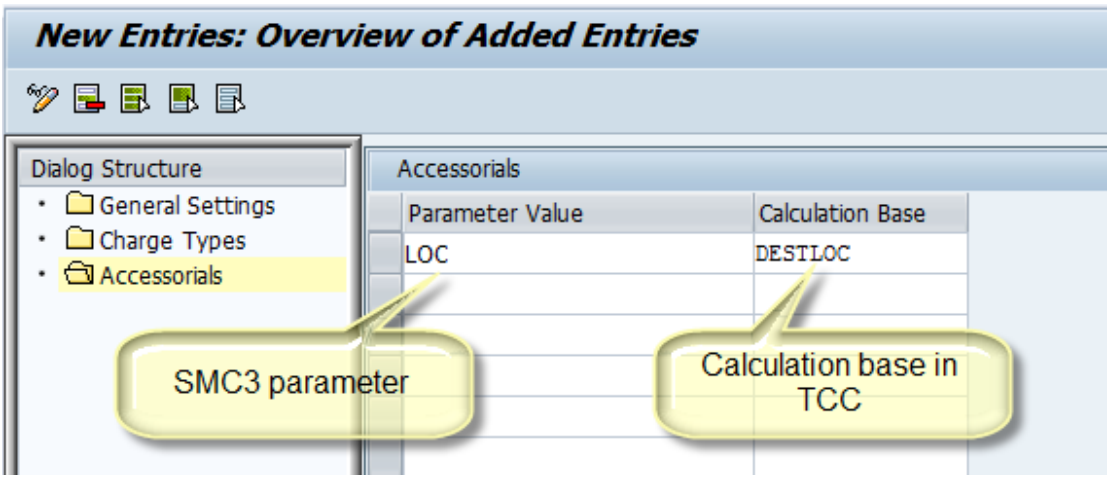


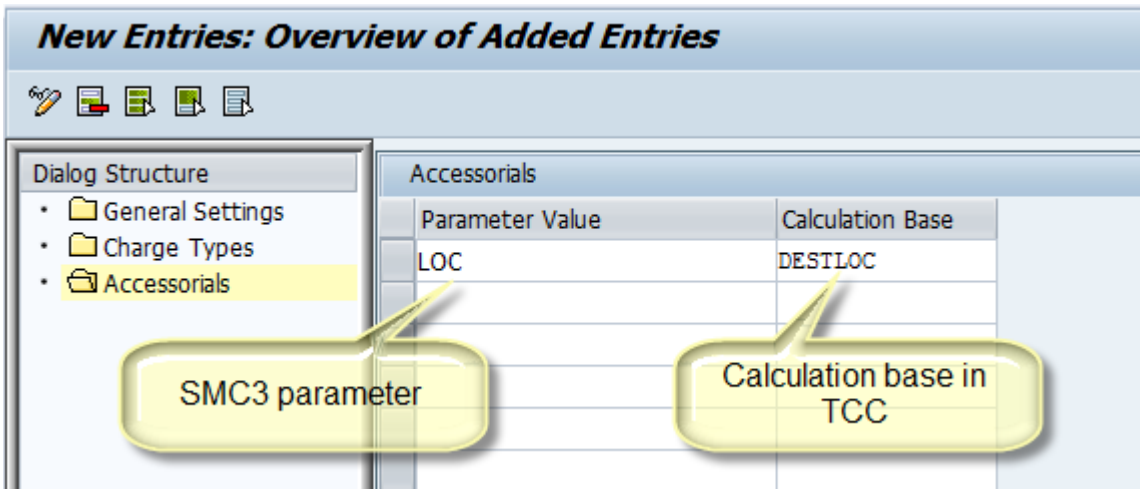
2. Enter the logon ID and logical port data as specified in the previous steps for the corresponding program parameters.
3. Enter the charge types and discounts that must be taken into account in the calculation results. Note that charge type must match the accessorial returned from the response of SMC³. We are yet to confirm the accessorial returned from SMC³. To begin with, FuelCost is the accessorial returned from SMC³. This is valid only for the SMC³

contract scenario.

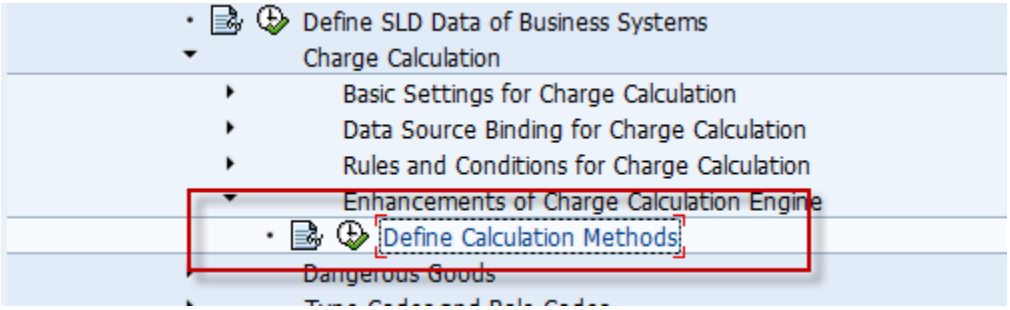


4. If accessorials are calculated using flexible parameters at SMC³, you can also map the SMC³ parameters to the calculation bases in charge calculation so that appropriate values are transferred to SMC³ for the accessorial calculation.

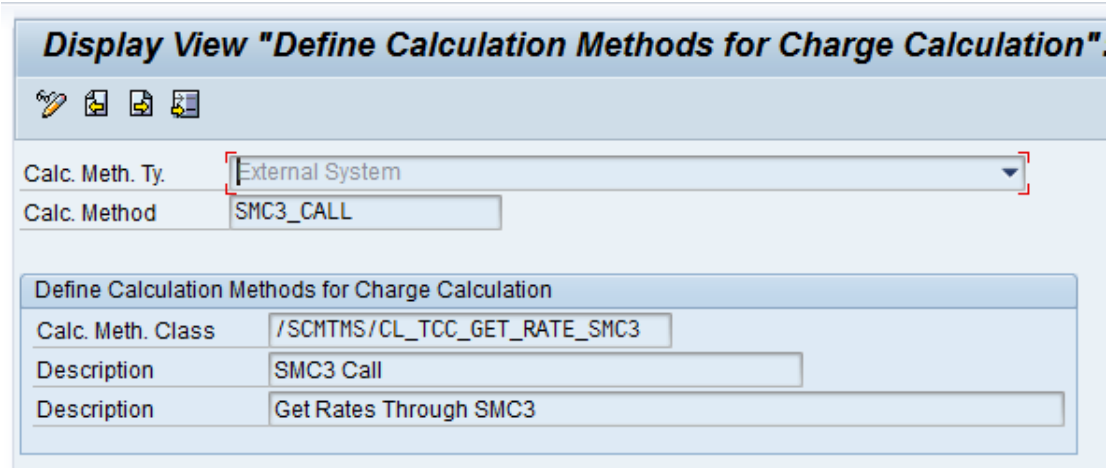




5. If you are using SAP TM 8.0 SP02 or lower or SAP TM 8.1 SP00, add a new calculation method to invoke the SMC³ call.

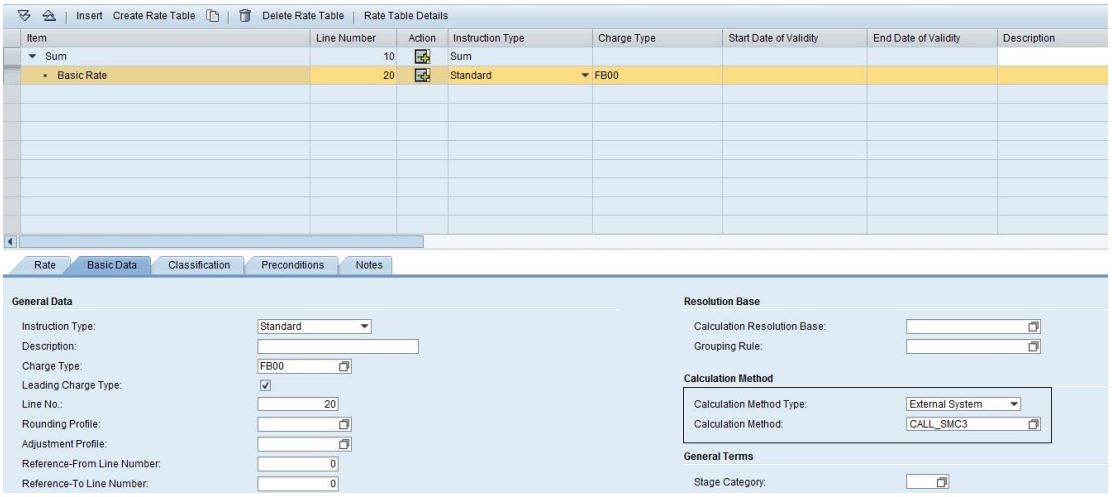


6. If you are using SAP TM 8.0 SP02 or lower or SAP TM 8.1 SP00, use standard class `/SCMTMS/CL_TCC_GET_RATE_SMC3` created as described above.

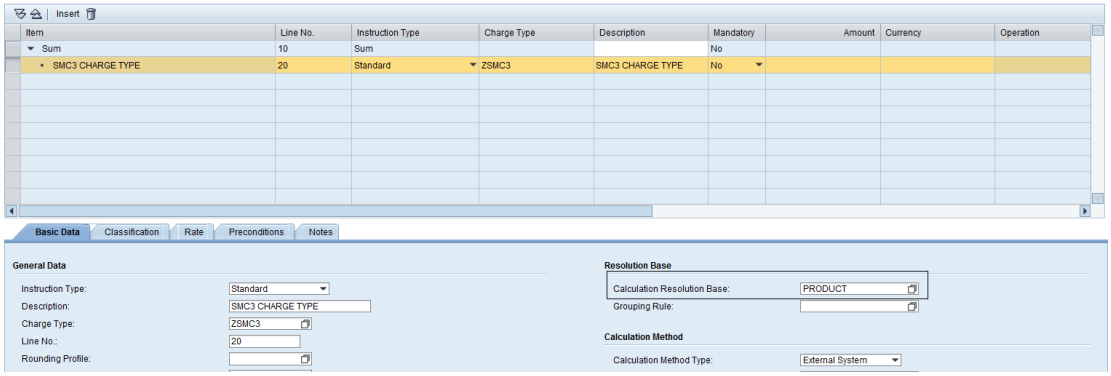


3.2 Maintaining Calculation Sheets

The SMC³ call is integrated into charge calculation using the `CALCULATION_METHOD` assignment at the calculation sheet line item. In our example, charge element `FB00` can be attached to any resolution base in the calculation sheet with the calculation method type `External System` as defined in the step below.



You can maintain the calculation resolution base as either *Product* or *Container*. If you choose *Product* as the resolution base, the system creates a charge line for every product and determines charges for every product. If you choose *Root* as the resolution base, the system consolidates the products with the same NMFC (National Motor Freight Classification) code and determines the charges for the total weight.

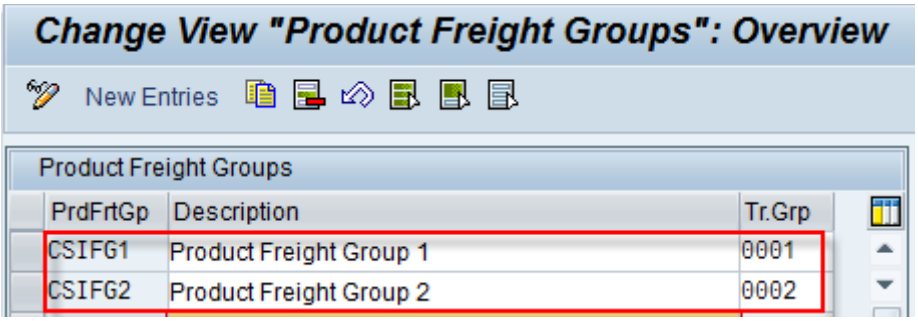


Runtime Aspects

When charges are calculated, charge element `ZSMC3` shown in step 3 above is handled in the same way as a standard charge element. However, when the rate lookup occurs (when the calculation sheet is processed), call bundling is carried out to ensure that a mass-enabled external call is triggered for SMC³. This results in improved system performance.

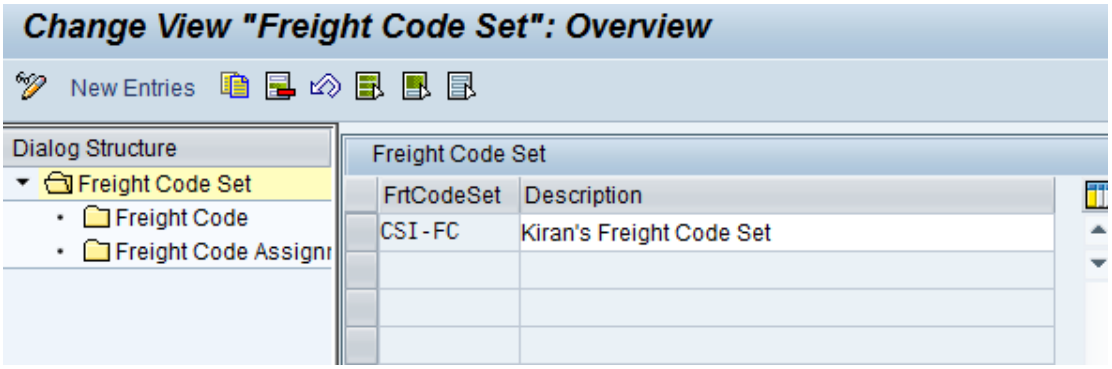
3.3 Creating Product Freight Groups

1. In Customizing for SCM Basis, choose *Master Data* →*Transportation Lane* → *Transportation Service Provider Profile* →*Define Product Freight Groups*.
2. In the *PrdFrtGp* field, enter the name of your product freight group.
3. In the *Tr.Grp* field, enter the name of your transportation group.

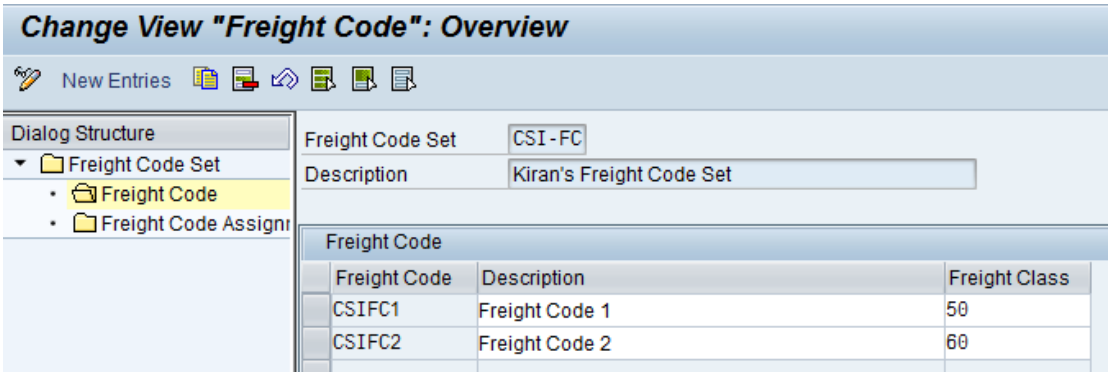


3.4 Defining Freight Code Sets

1. In Customizing for SCM Basis, choose *Master Data* →*Transportation Lane* → *Transportation Service Provider Profile* →*Define Freight Code Sets, Freight Codes, and Determination*.
2. In the *FrtCodeSet* field, enter the name of your freight code set.

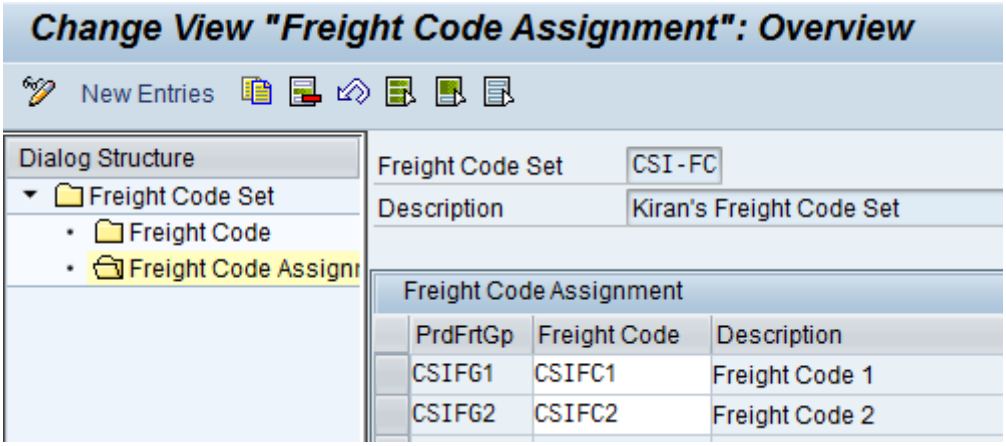


3. Select your freight code set and choose *Freight Code* from the dialog structure.
4. Assign your freight code to a freight class.



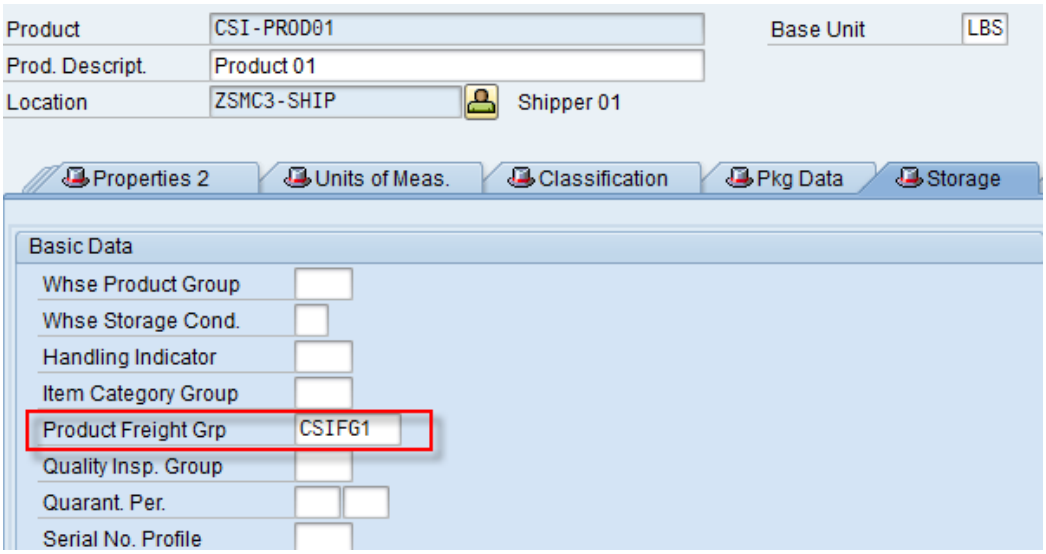
5. Select your freight code and choose *Freight Code Assignment* from the dialog structure.

6. Assign the product freight group that you created above to your freight code.



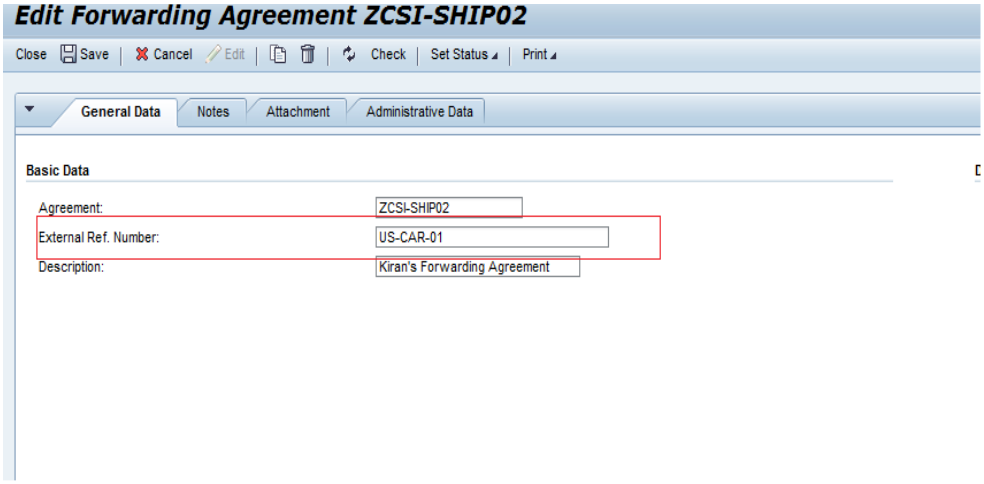
Note

There must be at least two products with different product freight groups. To assign your product to a product freight group, call transaction `/SAPAPO/MAT1`, enter your product, and choose the *Storage* tab page. Enter your product freight group in the *Basic Data* screen area.



3.5 Creating Agreements

The contract ID from SMC³ needs to be maintained in the *External Reference Number* field of the agreement as shown below for the contract scenario.



- If the Contract ID is not maintained, the fall back is to the shipment scenario where the ID should match the calculation sheet name.

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 generated a consumer proxy from a WSDL document as described in Section 2.4. In the message area of the ABAP Workbench (transaction `SE80`), the system displays 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 contains 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.

Note

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

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

Symptom

You generated a consumer proxy from a WSDL document as described in Section 2.4. On the *Warnings* tab page of the *Display Service Consumer* screen, the system displays a warning message such as *The XSD type <type> does not exactly correspond to the ABAP type <type>*.

Reason

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

Solution

In general, you can ignore this warning.

Restrictions to the value range ignored

Symptom

You generated a consumer proxy from a WSDL document as described in Section 2.4. On the *Warnings* tab page of the *Display Service Consumer* screen, the system displays the warning *Restrictions to the value range ignored*.

Reason

In 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 2.4.3 and a `CX_SOAP_CORE` exception is raised. The error 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

No Internet connection could be established.

Solution

Make sure you configured the logical port as explained in Section 2.4.2. If you access the Internet via a proxy server, it is important to maintain the correct proxy settings in the transport settings of the logical port.

4.2 Web Service Calls

Element <element> missing

Symptom

You tried to call a Web service method in ABAP and an exception of the type `CX_ST_GROUP_MISSING_CASE:XSLT` with 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 typically a problem on the Web service provider side. In ABAP, you can try to locate the error by having a look at 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 configured the logical port as described in Section 2.4.2. 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 consumer 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 consumer proxy or the name of the logical port is not specified correctly at the instantion of the consumer proxy.

Solution

Check if you created a logical port for the consumer proxy as described in Section 2.4 and make sure that you specify the correct name of the logical port when instantiating the consumer proxy.

Response of an asynchronous operation is empty

Symptom

You called an asynchronous operation of the Web service to retrieve the result after the query of the job status returned *JOB_DONE.* The obtained response structure is not filled completely.

Reason

Unknown

Solution

Sometimes the response structure is not ready immediately after the calculation has finished. Wait and try to retrieve the response again.

The query of the operation status returns JOB_FAILED

Symptom

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

Reason

One possible reason could be that invalid data was included in the calculation. The Web service aborts the whole job when a single coordinate is wrong and sets the job status to *JOB_FAILED*.

Solution

Check or update the data used in the call.

5 References

List of Documents

The following table lists all documentation mentioned in this Cookbook.

Title	Location
Process Controller	▶ http://help.sap.com →*SAP Business Suite* →*SAP Transportation Management* →*English/German*↵. Choose *SAP Library* and then choose the following path: ▶ *SAP Business Suite* →*SAP Transportation Management (SAP TM)* →*SCM Basis* →*Process Controller*↵.

List of SAP Notes

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

SAP Note Number	Title	Description
No SAP Notes currently required		

6 Appendix

6.1 Transactions and Reports

Transactions

Transaction Code	Description
SOAMANAGER	SOA Manager
SE80	Object Navigator
SE38	ABAP Editor
SM34	View Cluster Maintenance

Reports

Report Code	Description
/SCMTMS/TCC_SEC_STORE_LOGON	Secure Store Logon for SMC³