

Connecting a 3rd Party to SAP ECC Using HTTP/XML Format



Applies to:

Custom integration between 3rd Party systems (i.e. Java, .Net) and SAP ERP. For more information, visit the [Web Services homepage](#).

Summary

This document demonstrates how to send XML IDocs into SAP ECC from a non SAP 3rd party system using the HTTP IDOC XML interface. For knowledge on how to send IDocs from SAP using the HTTP IDOC XML Interface we would like to refer you to the article published by Arvind Kugasia titled "Cost effective and Quick Communication between SAP and 3rd Party systems using IDOC HTTP XML Interface". This document assumes you are familiar with SOAP and web services.

Audience: Technical Consultants, SAP Architects, IS Architects, Developers and IT Managers.

This procedure addresses issues caused by the phasing out of WSADMIN and WSCONFIG by SAP.

Author: Jonathan Eaton, Ricardo Poeta

Company: Capgemini

Created on: 11 September 2009

Author Bio



Jonathan Eaton is the cluster leader for SAP Integration at Capgemini Netherlands and IS Architect and an MBSC CITP. He has over 10 years experience in designing, building and advising clients with regards to complex integration landscapes in both SAP and non SAP environments. His tools of experience are Sonic MQ, SAP PI/XI, Sterling Integration Suite, Gentran NT, SAP BC, MS BizTalk and Tibco.



Ricardo Poeta is a Consultant in SAP PI/XI with experience in complex PI landscapes and support & implementation projects. He has over 3 years consultancy experience in a wide variety of business positions. He is a member of the SAP Integration Cluster at Capgemini Netherlands.

Table of Contents

Introduction	3
Architecture.....	3
Procedure	4
Setup the RFC Designations.....	4
Setup Idoc Processing Ports.....	5
Partner Profiles.....	6
Configure the Function module.	7
Configure the web service	9
Activation of the web service.....	13
Retrieving the WSDL	14
Testing the Web Service.....	16
Example of Idoc XML.....	18
Related Content.....	19
Disclaimer and Liability Notice.....	20

Introduction

Enterprises embarking on integration with SAP have faced limitations in the possible cost effective methods for IDoc communication with SAP ECC platforms. This is especially the case when it involves low volume, single interface messaging (i.e. sales order processing). Traditionally this would be achieved by using middleware such as SAP PI, Sterling Integration Suite or BizTalk. However such communication can be achieved at a much lower cost by using standard function modules provided by SAP (located in the WAS).

We would not recommend this solution for more complex integration scenarios (multiple interfaces and high volume architectures). In such cases middleware will provide better value results.

Architecture

Architecture uses a 3rd party for communication of IDoc with SAP ECC using the SOAP protocol. The IDoc is an XML message within an XML message. Due to this it MUST be first encoded into Base64 and then inserted into the XML Message within the SOAP protocol.

The receiving SAP system reads the base64 data type and transforms it back into native IDoc format.

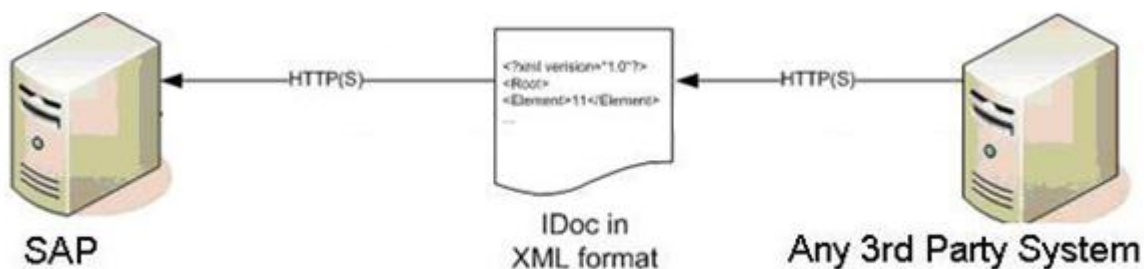


Fig 1: Architectural Diagram

Procedure


Setup the RFC Designations.

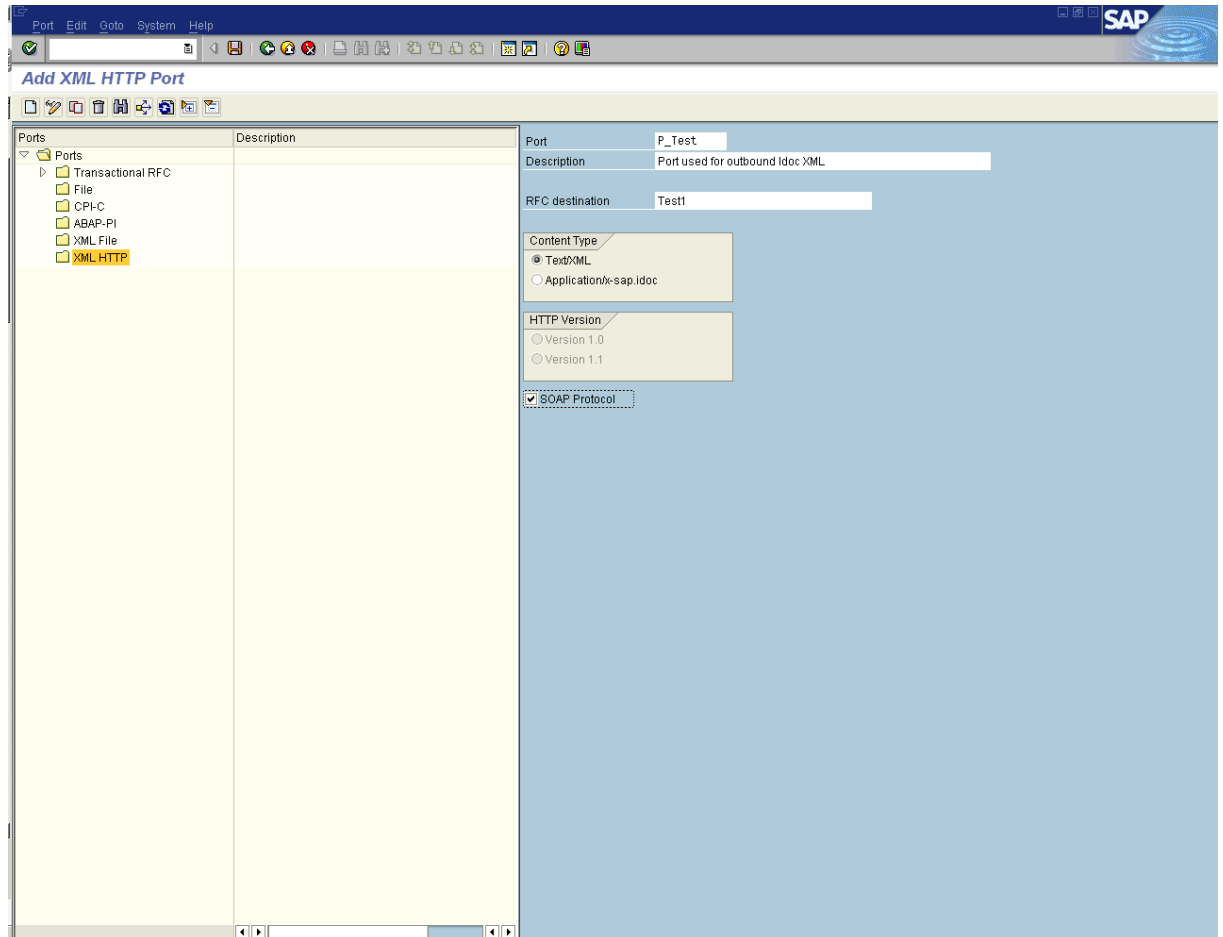
- 1) Configure the RFC connection (SM59) - create a new connection of type G (HTTP connection to external server).
- 2) Fill in the RFC destination, Service No. (the port number for outbound interfaces) and Target Host.
- 3) Where no target host exists, or outbound integration is not necessary you may use a phantom host name

The screenshot shows the SAP SM59 'RFC Destination' configuration screen. The browser window title is 'RFC Destination'. The main content area is divided into several sections:

- Connection Test:** A button with a magnifying glass icon.
- RF Destination:** A table with one entry: 'Test1'.
- Connection Type:** A dropdown menu set to 'G HTTP Connection to External Serv'.
- Description:** Three text input fields labeled 'Description 1', 'Description 2', and 'Description 3'.
- Administration / Technical Settings / Logon & Security / Special Options:** A set of tabs.
- Target System Settings:** A section with three fields: 'Target Host' (server1.capgemini.com), 'Service No.' (8000), and 'Path Prefix'.
- HTTP Proxy Options:** A section with a 'Global Configuration' button and five fields: 'Proxy Host', 'Proxy Service', 'Proxy User', 'Proxy PW Status' (is initial), and 'Proxy Password'.

Setup Idoc Processing Ports

- 1) Go to SAP transaction *WE21*.
- 2) Define a port for IDoc processing by selecting the  button
- 3) Fill in the *Port*, *Description* (optional), *RFC destination* and tick the *SOAP protocol* checkbox.
- 4) Select the content Type Text/XML.
- 5) Save.



Partner Profiles

Define all the partner profiles for the partners that receive your messages (WE20).

The screenshot displays the SAP 'Partner profiles' configuration window. The left-hand pane shows a hierarchical tree of partner types. The right-hand pane is divided into several sections for configuring a specific partner profile.

Partner Profiles Tree:

- Partner Profiles
 - Partner Type SP: Service provider (IS-U deregulat
 - Partner Type B: Bank
 - Partner Type BP: Benefits provider
 - Partner Type GP: Business Partner
 - Partner Type KU: Customer (Selected)
 - Partner Type LJ: Vendor
 - Partner Type LS: Logical system
 - RHDDONT: RHDDONT
 - RHDDONT: RHDDONT
 - RHDPONT: RHDPONT
 - RHDPONT: RHDPONT
 - RHDPONT: RHDPONT
 - RHDPONT: RHDPONT
 - RHDPONT: RHDPONT
 - UCOCLNT210: UCO client 210
 - Partner Type US: User (first 10 characters, no che

Partner Configuration Details:

- Partner No.: 100223 TEST
- Partn.Type: KU Customer
- Post processing: permitted agent
- Classification: [Dropdown]
- Ty.: [User Icon] User
- Agent: POETAR Ricardo POETA
- Lang.: NL Dutch

Outbound parmrts. Table:

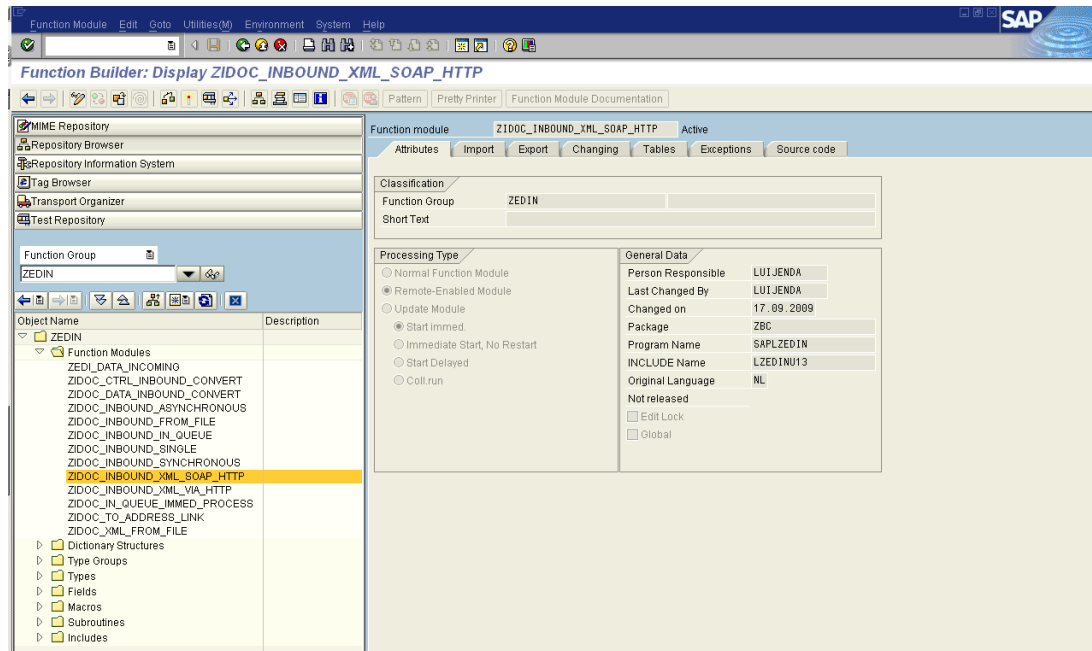
Partner Role	Message Type	Message var.	MessageFu.	Test

Inbound parmrts. Table:

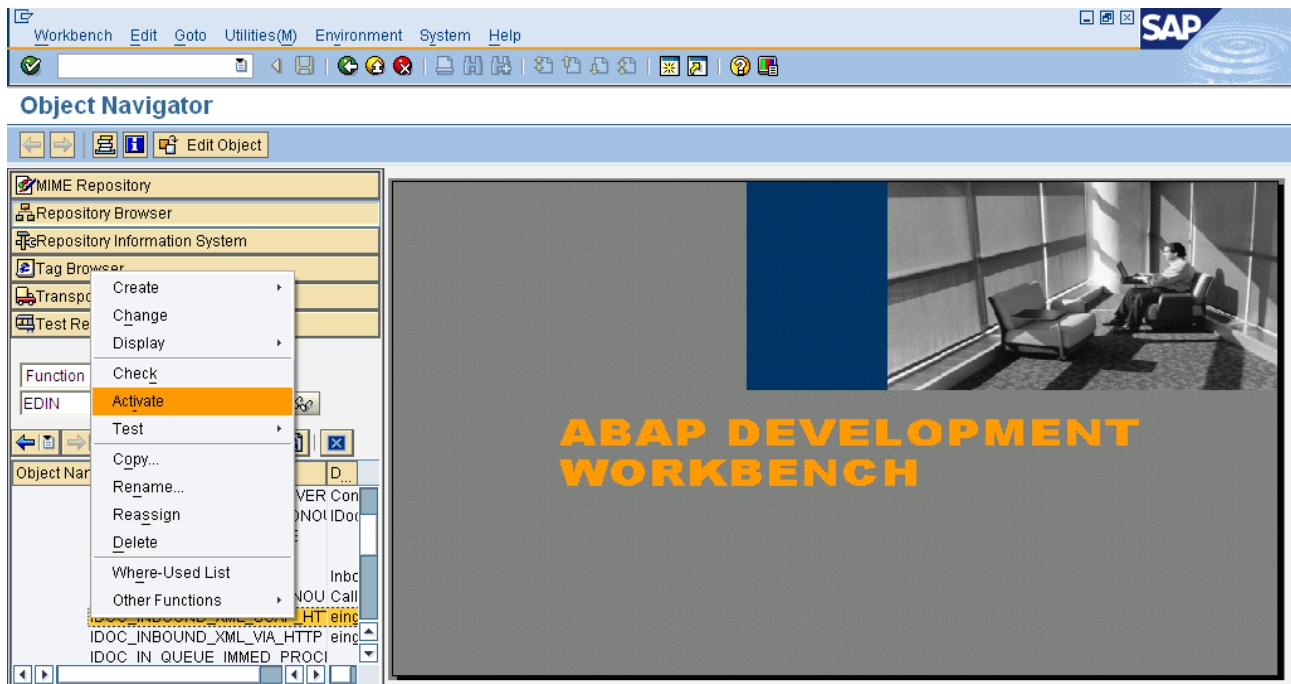
Partner Role	Message Type	Message var.	MessageFu.	Test
	ORDERS			

Configure the Function module.

- 1) To find the appropriate function module use SAP Transaction *SE80*.
- 2) Select the function group *EDIN* (generic RFC Idoc handling functions used by SAP adapters) and create a new function group based on it called *ZEDIN* (right click on it and choose copy).
- 3) Within the function group *ZEDIN* is a function module called *ZIDOC_INBOUND_XML_SOAP_HTTP* (previously called *IDOC_INBOUND_XML_SOAP_HTTP*). Select this function.
- 4) Make this function module RFC enabled (Processing Type "Remote-Enabled Module").



- 5) Activate the function by right clicking on it and selecting the Activate tab.



Inside the import and export tags, you need to check the *Pass Value* checkbox for every parameter

Function Module Edit Goto Utilities (M) Environment System Help

SAP

Function Builder: Display ZIDOC_INBOUND_XML_SOAP_HTTP

Function module: ZIDOC_INBOUND_XML_SOAP_HTTP Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Typing	Associated Type	Default value	Opti	Pass Value	Short text	Lon
XML_STREAM	TYPE	XSTRING		<input type="checkbox"/>	<input checked="" type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>		

UCO (2) 210
vmuco
INS

Configure the web service

- 1) In SAP Transaction `SE37` select function module `ZIDOC_INBOUND_XML_SOAP_HTTP`

It is important to remember that the web service should be generated for each client you will transport to (It is not transportable).

- 2) Go to *Utilities->More Utilities->Create Web Service->From the Function Group*:
- 3) In the "Provide Service Definition" window enter in a name for your web service.
- 4) In the tab *Kurzbeschreibung* enter in a description for your web service.
- 5) Press continue.

Provide Service Definition details

Enter a name and a short description for the Web Service and choose an end-point type.

To change the Web Service, use the ABAP Workbench (transaction SE80).

Object Type

- Choose Endpoint
- Choose Operations
- Configure Service
- Enter Package/Request
- Complete

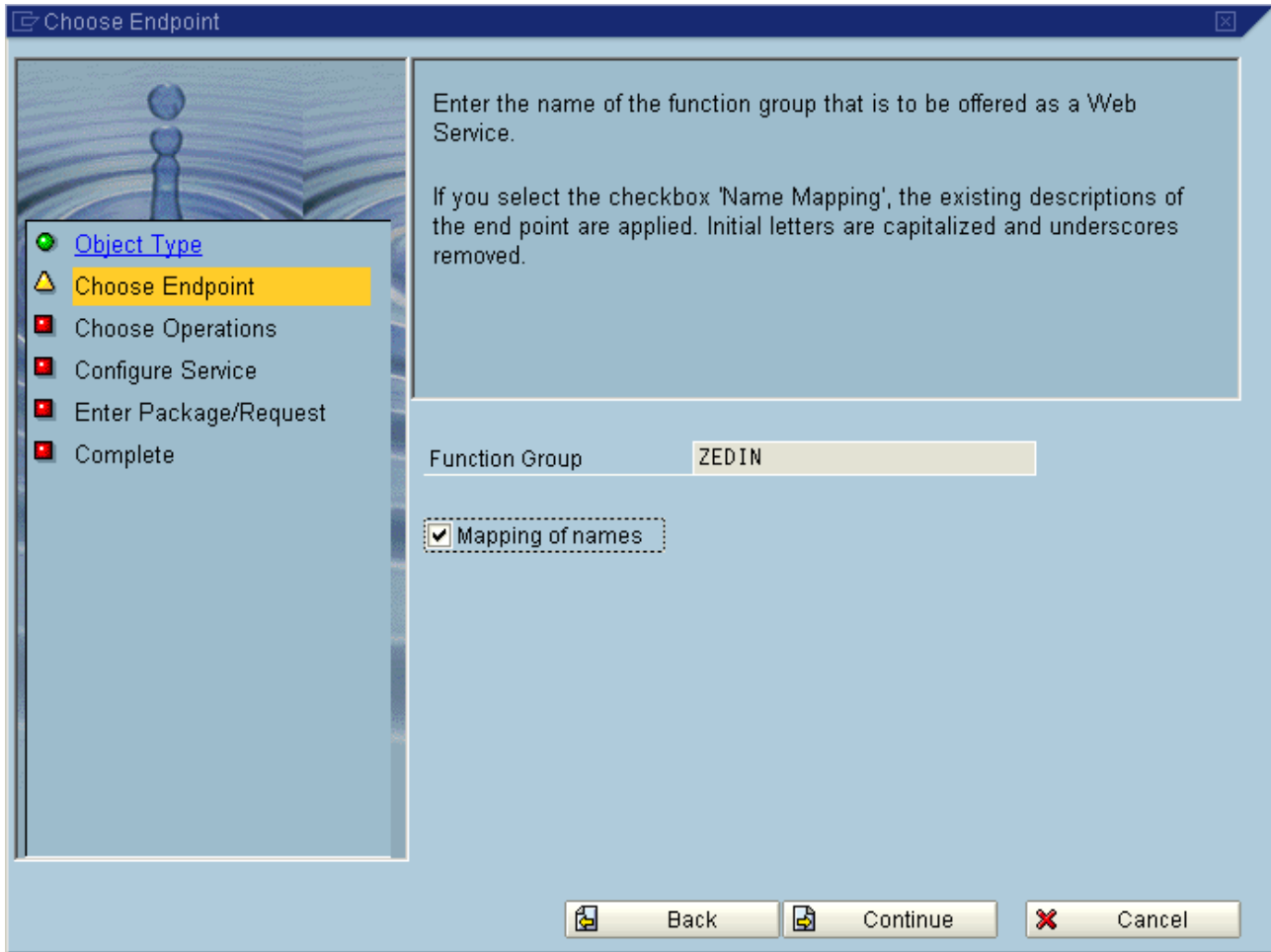
Service Definition: ZIDOC_XML_SOAP_WS

Kurzbeschreibung: IDOC INBOUND XML SOAP

Endpoint Type: Function Group

Back Continue Cancel

- 6) Tick the box "Name Mapping".
- 7) Select continue.



Choose Endpoint

Enter the name of the function group that is to be offered as a Web Service.

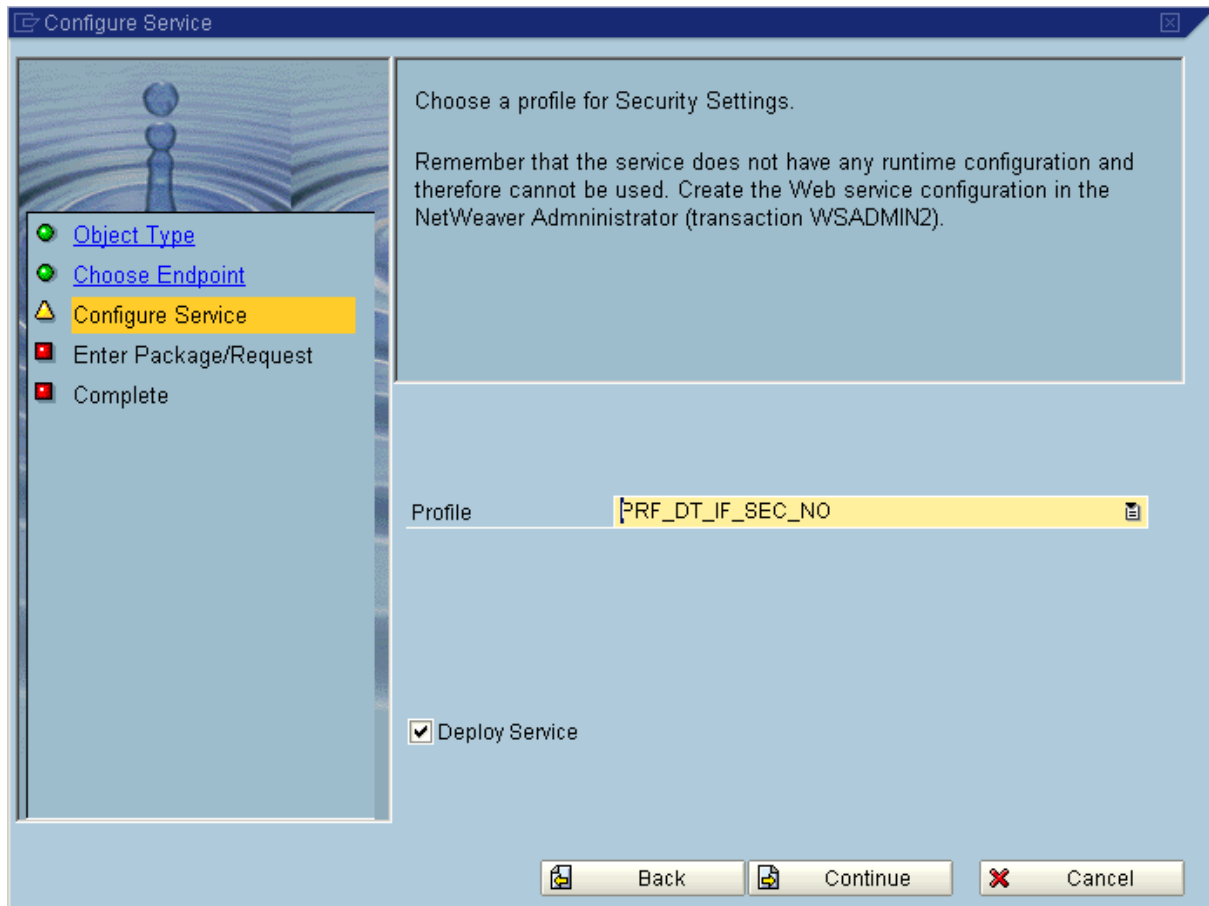
If you select the checkbox 'Name Mapping', the existing descriptions of the end point are applied. Initial letters are capitalized and underscores removed.

Function Group

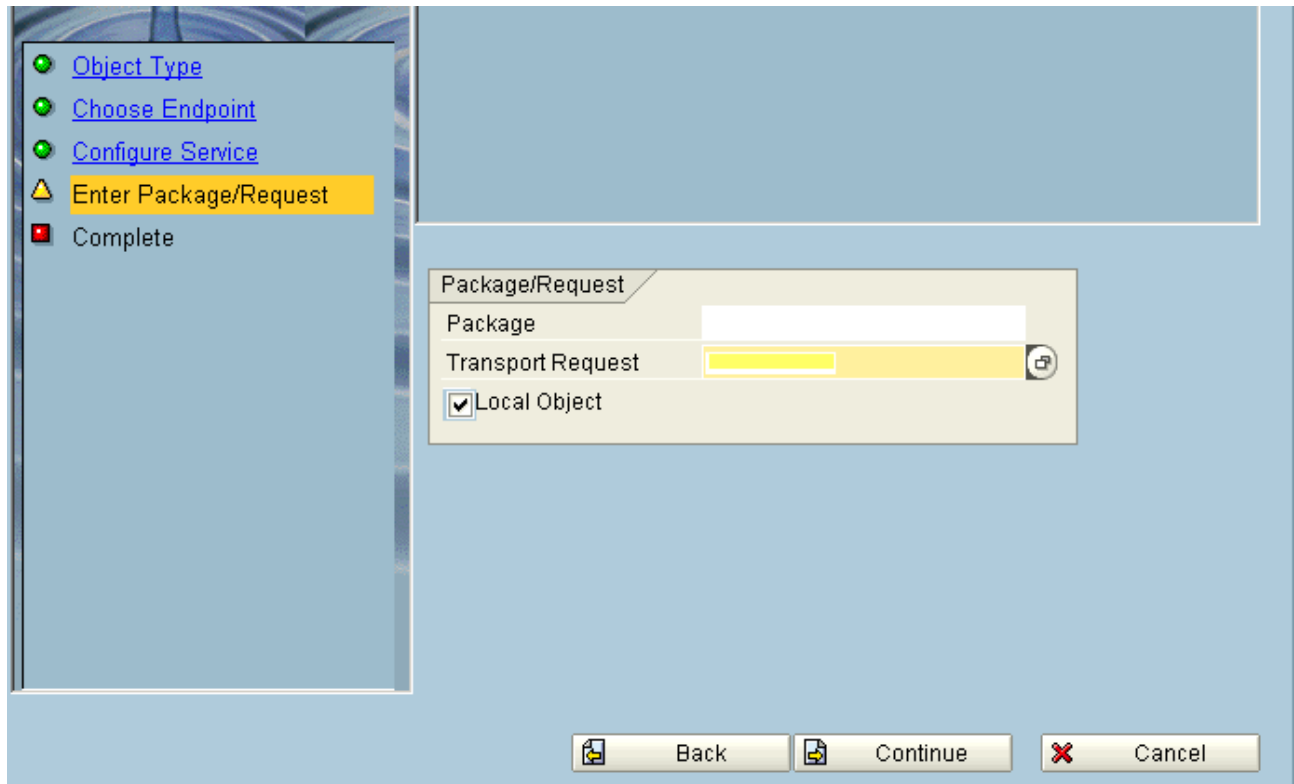
Mapping of names

Back Continue Cancel

- 8) Choose what kind of security you want to implement. As we are only testing, we won't implement a secure service.
- 9) Tick the box deploy service.
- 10) Click on continue.

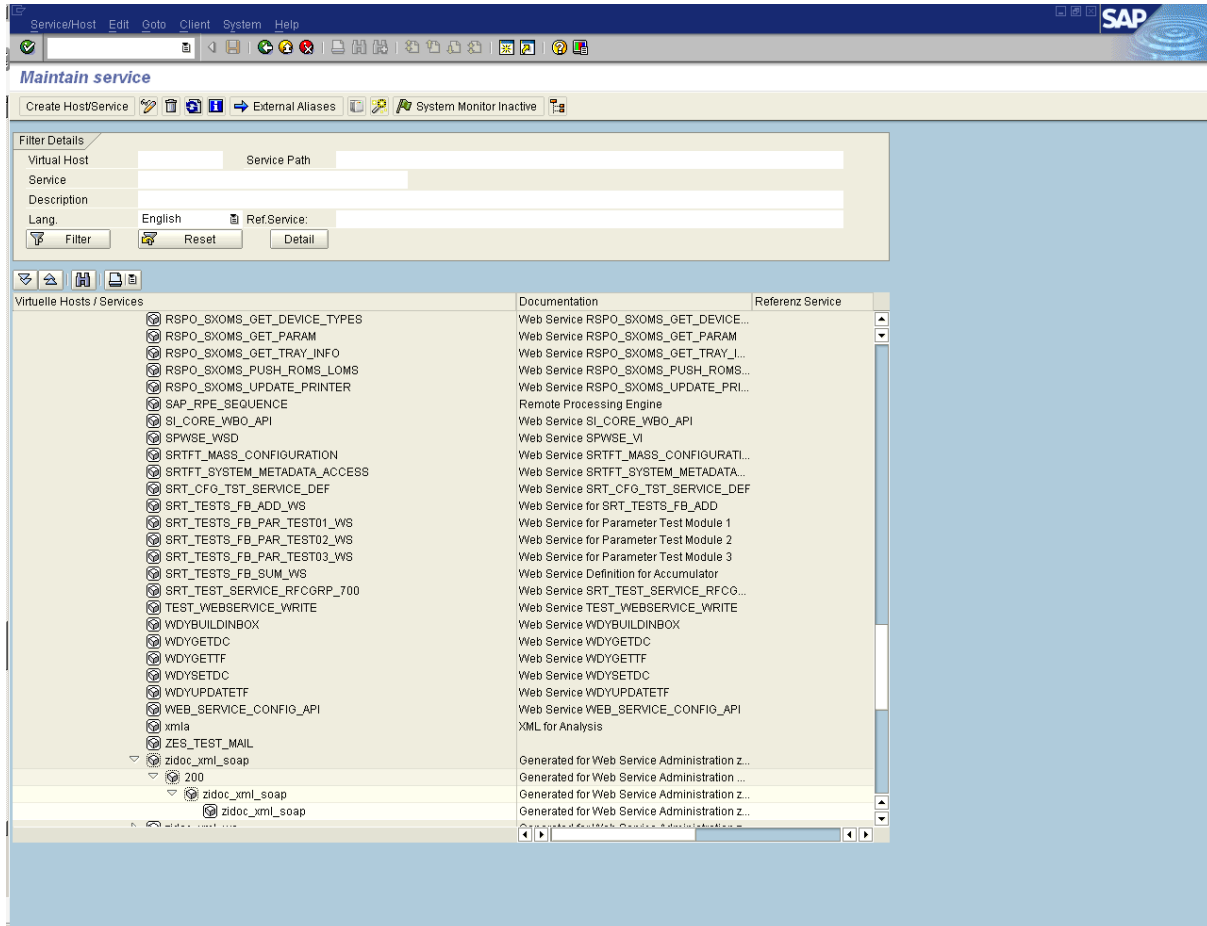


- 11) Keep the item as local package (select Local Object).
- 12) Click on continue.
- 13) The web service is now complete.



Activation of the web service.

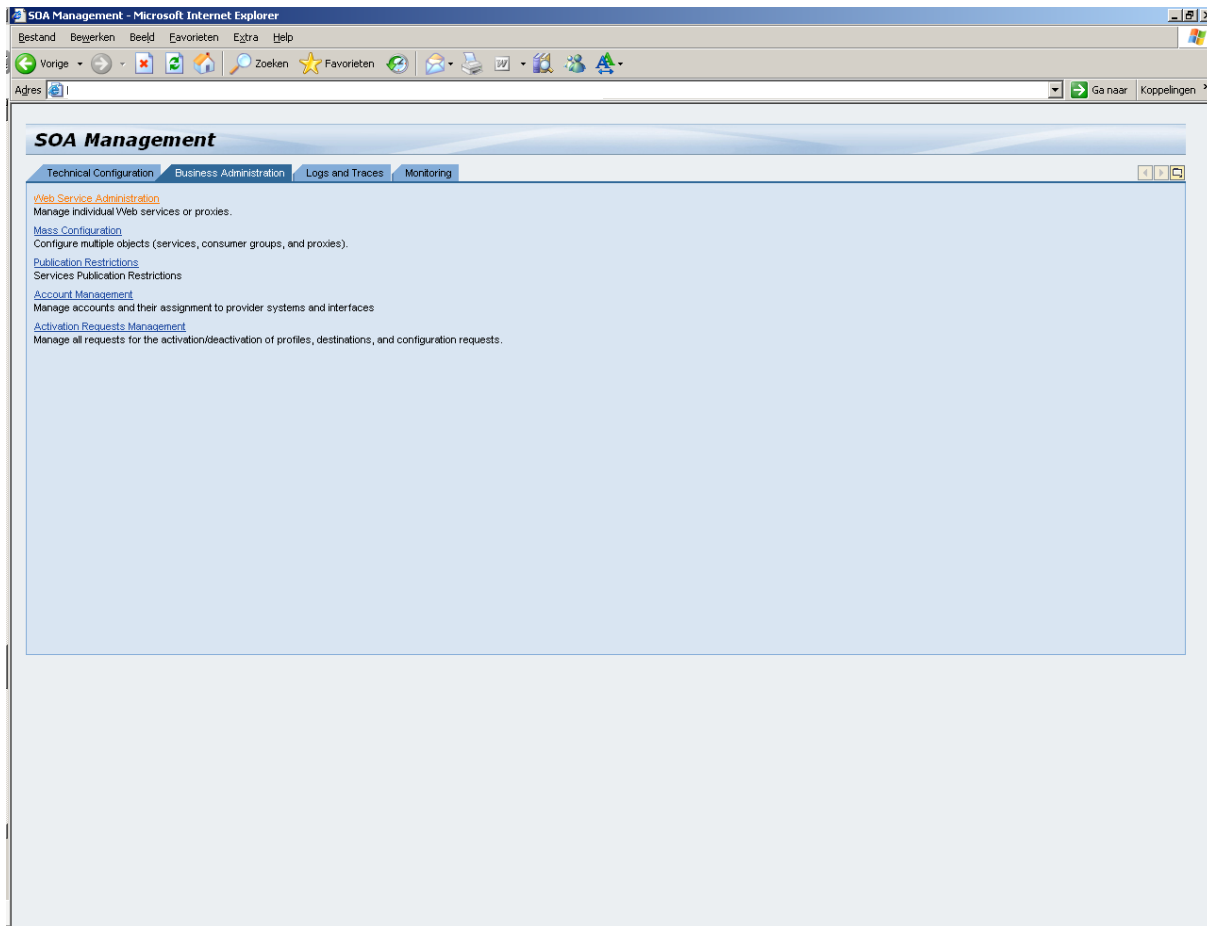
- 1) Go to the SAP Transaction *SICF*.
- 2) Select the web service You will find your service under *default_host/sap/bc/srt/rfc/sap*:
- 3) Right-click on the web service and select *Activate Service* (for testing purposes. you might as well specify the logon data, by double clicking on the service -> *Logon Data* tab, otherwise, you can also specify them later in your test client)



Retrieving the WSDL

Now we need to retrieve the WSDL in SOAMANAGER

- 1) Run transaction the SAP transaction *SOAMANAGER*.
- 2) The screen below will appear.
- 3) Select the tab "Business Administration".
- 4) Select Web Services Administration.



Note: You might find problems to open the SOAMANAGER if some key services are disabled. If so, when you run it, an error will appear and it will specify which service is not activated. Just go to SICF, locate the missing service and activate it

- 5) When you reach *Web Service Administration*, search for your service, select it and click on *Apply Selection*:
- 6) You should have in the *Object Status: Services: 1 / Endpoints: 1*
- 7) Click on *Open WSDL document for selected binding* and there you have your WSDL.
- 8) Load the WSDL into your SOAP test client.

The screenshot shows the SAP SOA Management Web Service Administration interface in a Microsoft Internet Explorer browser. The page title is "Web Service Administration". Below the title, there is a search bar with the search pattern "*XML_ws*" and a dropdown menu for "Field" set to "Internal Name". The search results table shows one entry:

Internal Name	External Name	Namespace	Type	Description
ZIDOC_XML_WS	zidoc_xml_ws	urn:sap-com:document:sap:rfc:functions	Service	idoc xml inbound

Below the search results, there is a section titled "Details of Service Definition: ZIDOC_XML_WS". This section has tabs for "Overview", "Configurations", "Classifications", and "Details". The "Overview" tab is selected, showing the following information:

Object Status: Services: 1 / Endpoints: 1
 Porttype Namespace: urn:sap-com:document:sap:rfc:functions
 Porttype Name: zidoc_xml_ws
 Internal Name: ZIDOC_XML_WS
 SOAP Application: URN:SAP-COM:SOAP:RUNTIME:APPLICATION:RFC:710
 Package Name: ZBC

At the bottom of the details section, there are three links: "Open porttype WSDL document", "Open WSDL document for selected binding", and "Open Web Service navigator for selected binding". A dropdown menu below these links shows the selected service: "ZIDOC_XML_WS::zidoc_xml_ws".

If the message is successful, you should get something like below as a return message.

The screenshot displays the SoapUI 3.0.1 interface. The main window shows the XML response for 'Request 2'. The response is a SOAP message with the following structure:

```

<?xml version='1.0' encoding='UTF-8'>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header/>
  <soap:Body>
    <n0:ZidocInboundMailSoapHttpResponse xmlns:n0="urn:sap-com:document:sap:soap:functions:uc-style">
      <Assign>
        <item>
          <TransferId>0000000000000000</TransferId>
          <DbId>0000000000203017</DbId>
        </item>
      </Assign>
    </n0:ZidocInboundMailSoapHttpResponse>
  </soap:Body>
</soap:Envelope>
  
```

The 'Request Properties' table is also visible:

Property	Value
Name	Request 2
Description	
Message Size	2647
Encoding	UTF-8
Endpoint	http://vmuco.domstad...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	
WSS:Password Type	

At the bottom, the status bar shows 'response time: 1502ms (397 bytes)' and a timestamp of '13:21'.

Example of Idoc XML.

```

<?xml version='1.0'?><!DOCTYPE ORDERS05 []><ORDERS05><IDOC BEGIN="1"><EDI_DC40
SEGMENT="1"><DIRECT>2</DIRECT><IDOC TYP>ORDERS05</IDOC TYP><MESTYP>ORDERS</MESTYP><SNDR
OR>BLAH</SNDR POR><SNDR PRT>KU</SNDR PRT><SNDR PRN>0000117026</SNDR PRN><RCVPRT>LS</RCVPRT><RCV
PRN>BLAHLNT124</RCV PRN><CRE DAT>20070122</CRE DAT><CRETIM>163914</CRETIM></EDI_DC40><E1
EDK01><EDI_BSART>ZID</EDI_BSART></E1EDK01><E1EDK14
SEGMENT="1"><QUALF>008</QUALF><ORGID>1030</ORGID></E1EDK14><E1EDK14
SEGMENT="1"><QUALF>007</QUALF><ORGID>00</ORGID></E1EDK14><E1EDK14
SEGMENT="1"><QUALF>006</QUALF><ORGID>10</ORGID></E1EDK14><E1EDK14
SEGMENT="1"><QUALF>012</QUALF><ORGID>ZID</ORGID></E1EDK14><E1EDK03
SEGMENT="1"><ID DAT>022</ID DAT><DATUM>20070122</DATUM></E1EDK03><E1EDK03
SEGMENT="1"><ID DAT>023</ID DAT><DATUM>20070201</DATUM></E1EDK03><E1EDK03
SEGMENT="1"><ID DAT>016</ID DAT><DATUM>20070201</DATUM></E1EDK03><E1EDK03
SEGMENT="1"><ID DAT>012</ID DAT><DATUM>20070201</DATUM></E1EDK03><E1EDK03
SEGMENT="1"><ID DAT>024</ID DAT><DATUM>20070201</DATUM></E1EDK03><E1EDKA1
SEGMENT="1"><PAR VW>AG</PAR VW><PARTN>0000117026</PARTN></E1EDKA1><E1EDKA1
SEGMENT="1"><PAR VW>WE</PAR VW><PARTN>0000117026</PARTN></E1EDKA1><E1EDK02
SEGMENT="1"><QUALF>017</QUALF><BELNR>K-52637</BELNR></E1EDK02><E1EDP01
SEGMENT="1"><E1EDP02
SEGMENT="1"><QUALF>052</QUALF><BELNR>BLAHJET1</BELNR><ZEILE>1</ZEILE></E1EDP02><E1EDP
02 SEGMENT="1"><QUALF>002</QUALF><BELNR>BLAHJET2</BELNR></E1EDP02><E1EDP02
SEGMENT="1"><QUALF>047</QUALF><BELNR>BLAHJET3</BELNR></E1EDP02><ACTION>001</ACTION><M
ENGE>100</MENGE><MENE E>PCE</MENE E><NETWR>8.94</NETWR><ABGR T>Bedrijfs
Yacht</ABGR T><E1EDP19 SEGMENT="1"><QUALF>002</QUALF><ID TNR>20013601</ID TNR><KTEXT>UGH
I hope this works</KTEXT></E1EDP19></E1EDP01></IDOC></ORDERS05>

```

Related Content

[Cost-Effective and Quick Communication between SAP and 3rd Party Systems using IDOC HTTP XML Interface](#)

[Architecture of the SAP Web AS](#)

[SAP NetWeaver Administration](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.