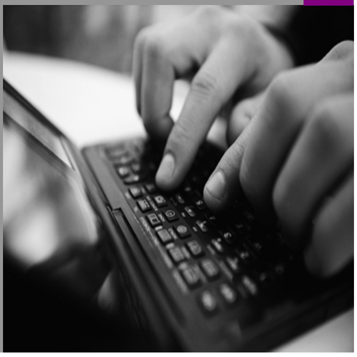


How-to Guide
SAP NetWeaver '04

How to… Expose A BW Query As Web Service In A UDDI Registry

Version 1.00 – January 2005

Applicable Releases:
SAP NetWeaver '04
(SAP BW 3.5)

THE BEST-RUN BUSINESSES RUN SAP

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.
SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.
SAP NetWeaver “How-To” Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

1 Business Scenario

You would like to expose information from a BW Query to the outside world. The consumers of this information should not have to have any knowledge about BW, e.g. how to run a query in a BW System. In our example the query returns the invoiced quantity for a given country and region.

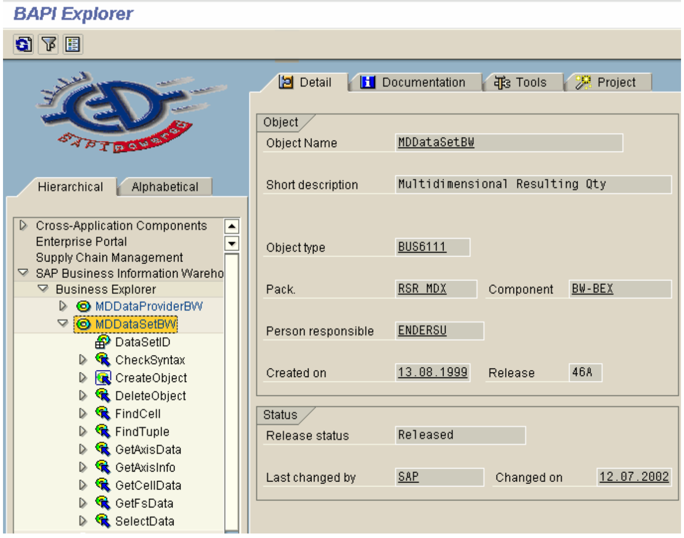
2 Introduction

Web Services can be used to provide simple scenarios that do not require any special knowledge about a component system. In SAP NetWeaver'04 all functionality is available to base a Web Service on a function module and expose it to a UDDI Registry. We deploy the OLAP BAPI in our own function module in order to run a given query with pre-defined variable parameters (country and region). The OLAP BAPI uses the MDX (Multi-Dimensional Expressions) standard for running queries in BW.

3 The Step By Step Solution

3.1 Implement Your Own Function Module Which Runs a Fixed Query

1. At first you may want to get an overview of the OLAP BAPI. In transaction BAPI you find the OLAP BAPI with its methods and documentation.



BAPI Explorer

Object

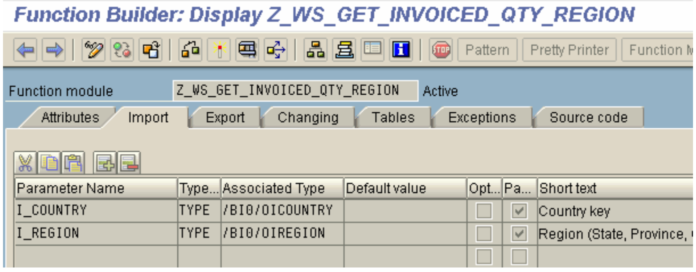
Object Name	MDDataSetBW
Short description	Multidimensional Resulting Qty
Object type	BUS6111
Pack.	RSR_MDX
Component	BW-BEX
Person responsible	ENDERSU
Created on	13.08.1999
Release	46A

Status

Release status	Released
Last changed by	SAP
Changed on	12.07.2002

- Cross-Application Components
- Enterprise Portal
- Supply Chain Management
- SAP Business Information Wareho
  - Business Explorer
    - MDDataProviderBW
    - MDDataSetBW
      - DataSetID
      - CheckSyntax
      - CreateObject
      - DeleteObject
      - FindCell
      - FindTuple
      - GetAxisData
      - GetAxisInfo
      - GetCellData
      - GetFsData
      - SelectData

2. Create your own function module in transaction SE37. Use import parameters you need for specifying filter values for the query.

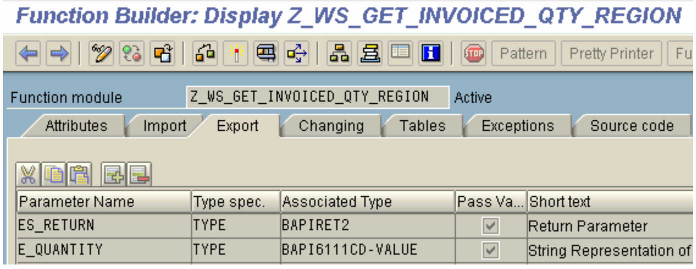


Function Builder: Display Z_WS_GET_INVOICED_QTY_REGION

Function module Z_WS_GET_INVOICED_QTY_REGION Active

Parameter Name	Type...	Associated Type	Default value	Opt...	Pa...	Short text
I_COUNTRY	TYPE	/BI0/OICOUNTRY		□	☑	Country key
I_REGION	TYPE	/BI0/OIREGION		□	☑	Region (State, Province,
				□	□	

3. Specify export parameters for the key figure value(s) and a return parameter for checking successful execution.



Function Builder: Display Z_WS_GET_INVOICED_QTY_REGION

Function module Z_WS_GET_INVOICED_QTY_REGION Active

Parameter Name	Type spec.	Associated Type	Pass Va...	Short text
ES_RETURN	TYPE	BAPIRET2	☑	Return Parameter
E_QUANTITY	TYPE	BAPI6111CD-VALUE	☑	String Representation of

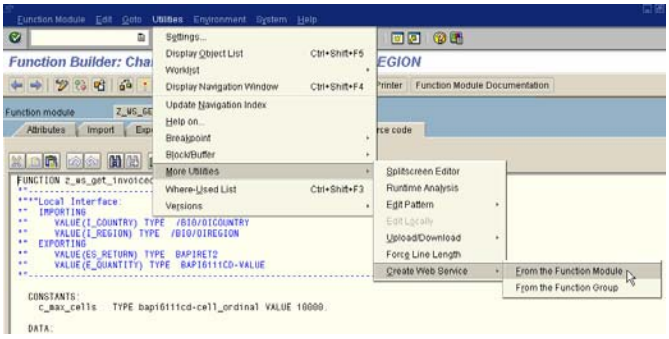
4. In the source code you have to create the MDX statement and run it with the OLAP BAPI. Please check the appendix for the complete sample code.

```
CONCATENATE i_country i_region
  INTO l_name SEPARATED BY space.
CONCATENATE '"' l_name '"'
  INTO l_name.
append_mdx 'SELECT [Measures].MEMBERS ON COLUMNS,'.
append_mdx 'FILTER ( [0REGION].MEMBERS,
  [0REGION].CURRENTMEMBER.NAME ='.
append_mdx l_name.
append_mdx ') ON ROWS'.
append_mdx 'FROM FABCUBE5/KWWS'.
```

3.2 Create a Web Service Using the Function Module

There are several ways to create a Web Service. You can start the Web Service Wizard directly (transaction WS_WZD_START). Alternatively, you can create Web Service in SE80 or directly start the Wizard from the Function Builder (SE37). Either way the result will be the same.

5. When editing the function module, select Utilities → More Utilities → Create Web Service → From the Function Module in transaction SE37.

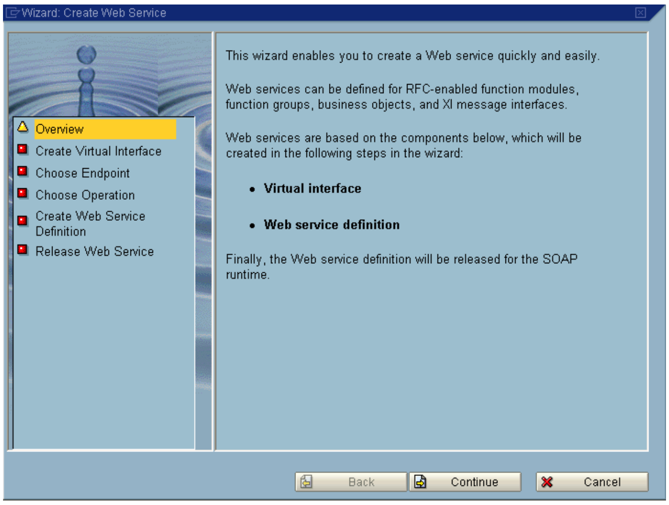


```
FUNCTION z_ws_get_invoiced
*"----------------------------
*"*"Local Interface:
*"  IMPORTING
*"     VALUE(I_COUNTRY) TYPE  /BI0/OICOUNTRY
*"     VALUE(I_REGION) TYPE  /BI0/OIREGION
*"  EXPORTING
*"     VALUE(ES_RETURN) TYPE  BAPIRET2
*"     VALUE(E_QUANTITY) TYPE  BAPI6111CD-VALUE
*"----------------------------------------------------------------------

  CONSTANTS:
    c_max_cells   TYPE bapi6111cd-cell_ordinal VALUE 10000.

  DATA:
```

6. The wizard starts and gives some introductory information. Press “Continue”.



This wizard enables you to create a Web service quickly and easily.

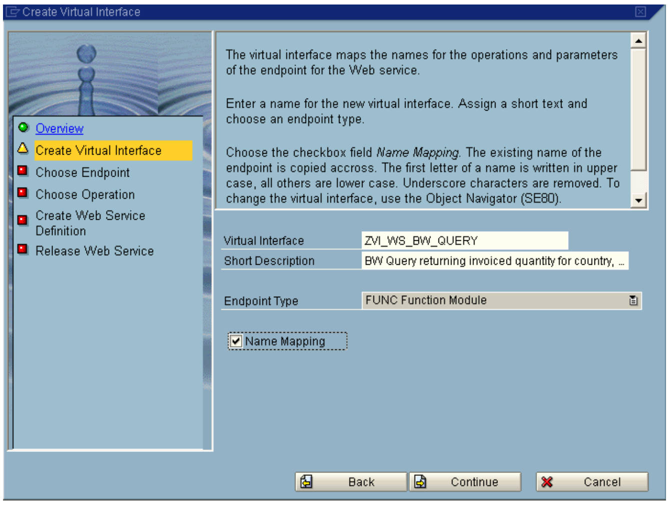
Web services can be defined for RFC-enabled function modules, function groups, business objects, and XI message interfaces.

Web services are based on the components below, which will be created in the following steps in the wizard:

- **Virtual interface**
- **Web service definition**

Finally, the Web service definition will be released for the SOAP runtime.

7. The virtual interface is created. Specify a name, short description, specify the Endpoint Type (already defaulted to “Function Module” if you start from transaction SE37). Press “Continue”.



The virtual interface maps the names for the operations and parameters of the endpoint for the Web service.

Enter a name for the new virtual interface. Assign a short text and choose an endpoint type.

Choose the checkbox field *Name Mapping*. The existing name of the endpoint is copied accross. The first letter of a name is written in upper case, all others are lower case. Underscore characters are removed. To change the virtual interface, use the Object Navigator (SE80).

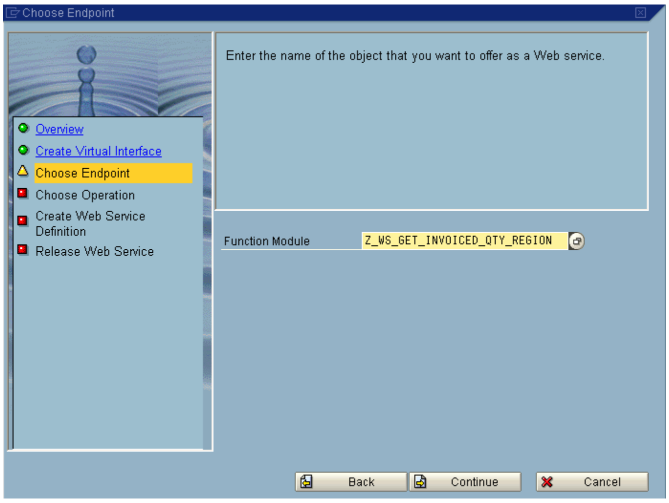
Virtual Interface: ZVI_WS_BW_QUERY

Short Description: BW Query returning invoiced quantity for country, ...

Endpoint Type: FUNC Function Module

☑ Name Mapping

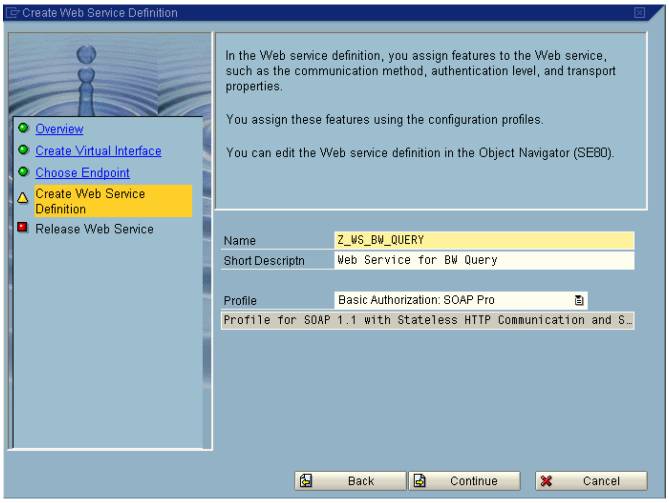
8. Specify the function module (defaulted correctly if you start from transaction SE37). Press "Continue".



Enter the name of the object that you want to offer as a Web service.

Function Module Z_WS_GET_INVOICED_QTY_REGION

9. Now the Web Service itself is created. Specify name, short description and choose an authorization profile. Press "Continue".



In the Web service definition, you assign features to the Web service, such as the communication method, authentication level, and transport properties.

You assign these features using the configuration profiles.

You can edit the Web service definition in the Object Navigator (SE80).

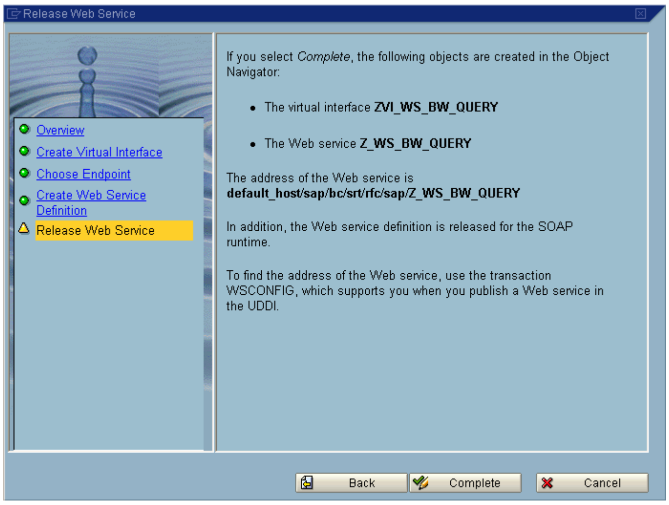
Name Z_WS_BW_QUERY

Short Descriptn Web Service for BW Query

Profile Basic Authorization: SOAP Pro

Profile for SOAP 1.1 with Stateless HTTP Communication and S...

10. Check all settings and press "Complete". All objects will then be created.



If you select *Complete*, the following objects are created in the Object Navigator:

- The virtual interface **ZVI_WS_BW_QUERY**
- The Web service **Z_WS_BW_QUERY**

The address of the Web service is **default_host/sap/bc/srt/rfc/sap/Z_WS_BW_QUERY**

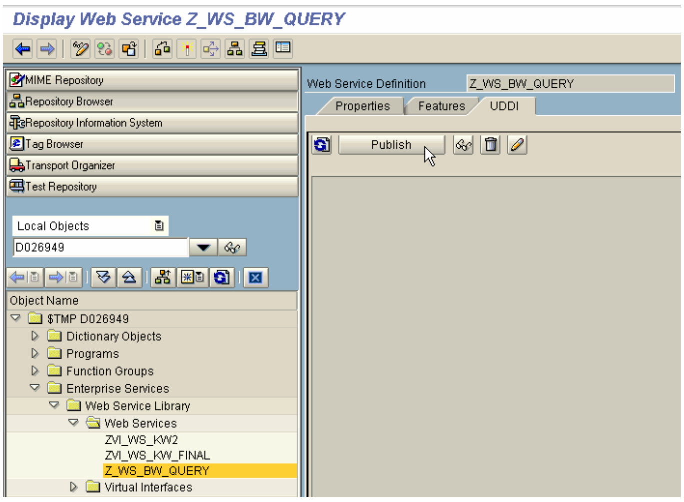
In addition, the Web service definition is released for the SOAP runtime.

To find the address of the Web service, use the transaction WSCONFIG, which supports you when you publish a Web service in the UDDI.

3.3 Publish the Web Service to a UDDI Registry

The Web Service is now available and ready for use. You don't have to do anything in addition. Publishing the Web Service in a UDDI Registry makes the Web Service known to people who have access to that registry. Let's assume that we have a UDDI Registry available. Please check the appendix if the UDDI Registry is not known or not available.

11. In transaction SE80 locate and double click on your Web Service, choose the UDDI tab and press "Publish".

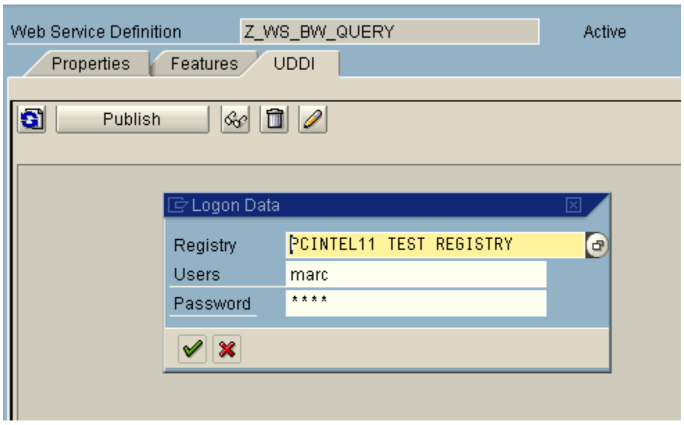


Display Web Service Z_WS_BW_QUERY

Web Service Definition: Z_WS_BW_QUERY

Local Objects: D026949

12. Choose the registry you want to use for publishing and specify user and password of the UDDI Registry if necessary.



Web Service Definition: Z_WS_BW_QUERY Active

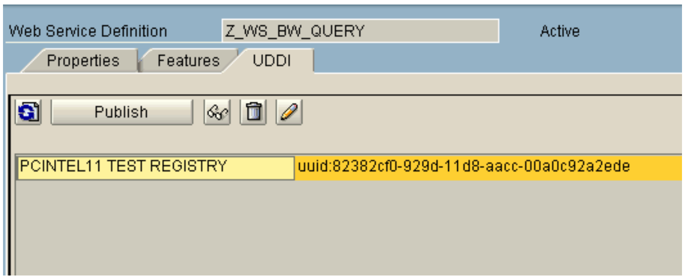
Logon Data

Registry: PCINTEL11 TEST REGISTRY

Users: marc

Password: ****

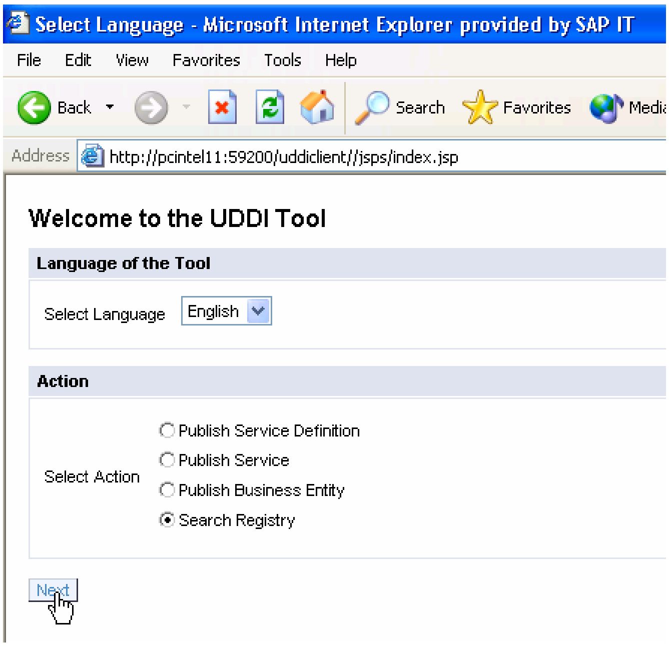
13. A line is added to the list specifying the UDDI Registry and a generated UUID. The Web Service is now available and can be found in the UDDI Registry.



Web Service Definition: Z_WS_BW_QUERY Active

PCINTEL11 TEST REGISTRY	uuid:82382cf0-929d-11d8-aacc-00a0c92a2ede

14. If the UDDI Registry of a SAP J2EE Engine is used you can use the link http://<servername>:<port>/uddiclient. Choose “Search Registry”and press “Next”.



Select Language - Microsoft Internet Explorer provided by SAP IT

Address http://pcintel11:59200/uddiclient//jsps/index.jsp

Welcome to the UDDI Tool

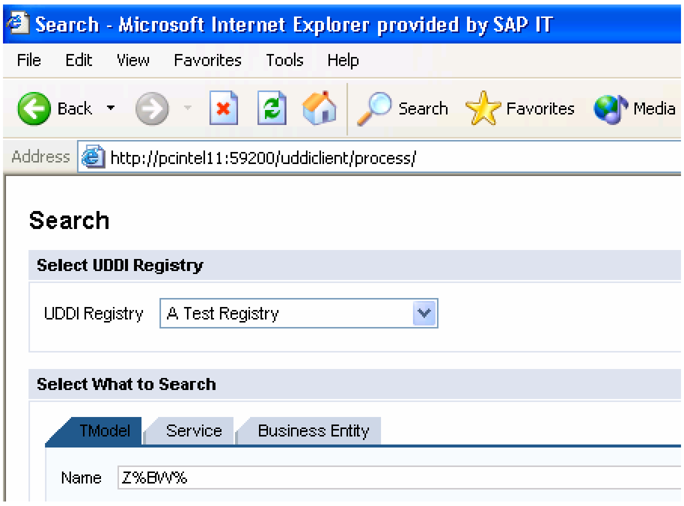
Language of the Tool

Select Language English

Action

Select Action
○ Publish Service Definition
○ Publish Service
○ Publish Business Entity
● Search Registry

15. Choose the registry. Mind that the names of the (links to the) registries differ between the one defined before in the transaction and the one defined in the J2EE Engine itself. Nevertheless they can point to the same physical registry. Specify your search criteria using “%” as a wildcard and press “Search” at the bottom.



Search - Microsoft Internet Explorer provided by SAP IT

Address http://pcintel11:59200/uddiclient/process/

Search

Select UDDI Registry

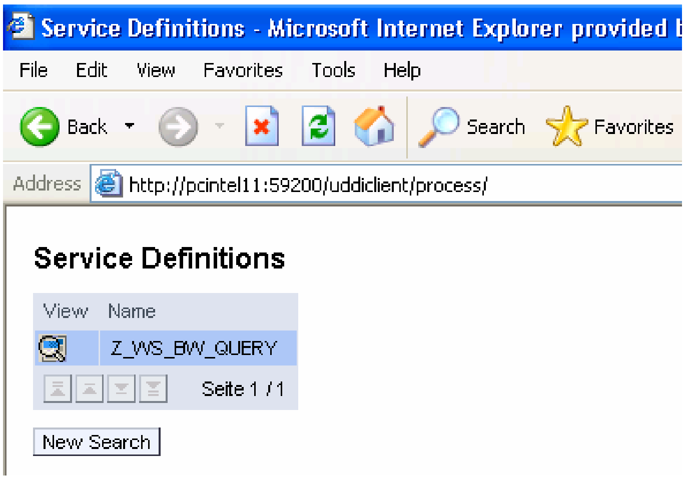
UDDI Registry A Test Registry

Select What to Search

TModel | Service | Business Entity

Name Z%BW%

16. You will find your Web Service in the registry.



Service Definitions - Microsoft Internet Explorer provided b

Address http://pcintel11:59200/uddiclient/process/

Service Definitions

View	Name
🔍	Z_WS_BW_QUERY
	Seite 1 / 1

4 Appendix

4.1 Sample Code for BW Query Function Module

```
FUNCTION z_ws_get_invoiced_qty_region.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     VALUE(I_COUNTRY) TYPE  /BI0/OICOUNTRY
*"     VALUE(I_REGION) TYPE  /BI0/OIREGION
*"  EXPORTING
*"     VALUE(ES_RETURN) TYPE  BAPIRET2
*"     VALUE(E_QUANTITY) TYPE  BAPI6111CD-VALUE
*"----------------------------------------------------------------------

  CONSTANTS:
    c_max_cells   TYPE bapi6111cd-cell_ordinal VALUE 10000.

  DATA:
    l_name        TYPE bapi6111mdx-line,
    l_datasetid   TYPE bapi6111gen-obj_handle,
    ls_mdx        TYPE bapi6111mdx,
    lt_mdx        TYPE TABLE OF bapi6111mdx WITH DEFAULT KEY,
    ls_cell_data  TYPE bapi6111cd,
    lt_cell_data  TYPE TABLE OF bapi6111cd WITH DEFAULT KEY.

  DEFINE append_mdx.
    ls_mdx = &1.
    append ls_mdx to lt_mdx.
  END-OF-DEFINITION.

* Compose MDX statement
  CONCATENATE i_country i_region INTO l_name SEPARATED BY space.
  CONCATENATE '"' l_name '"' INTO l_name.

  append_mdx 'SELECT'.
  append_mdx '[Measures].MEMBERS ON COLUMNS,'.
  append_mdx 'FILTER ( [0REGION].MEMBERS, [0REGION].CURRENTMEMBER.NAME ='.
  append_mdx l_name.
  append_mdx ') ON ROWS'.
  append_mdx 'FROM FABCUBE5/KWWS'.

  CLEAR es_return.

* Create MDX data set
  CALL FUNCTION 'BAPI_MDDATASET_CREATE_OBJECT'
    IMPORTING
      datasetid    = l_datasetid
      return       = es_return
    TABLES
      command_text = lt_mdx.

  CHECK es_return IS INITIAL.

* Run MDX data selection
  CALL FUNCTION 'BAPI_MDDATASET_SELECT_DATA'
    EXPORTING
      datasetid = l_datasetid
    IMPORTING
      return    = es_return.

  CHECK es_return IS INITIAL.
```

```
* Get MDX result set
  CALL FUNCTION 'BAPI_MDDATASET_GET_CELL_DATA'
    EXPORTING
      datasetid  = l_datasetid
      start_cell = 0
      end_cell   = c_max_cells
    IMPORTING
      return     = es_return
    TABLES
      cell_data  = lt_cell_data.

  CHECK es_return IS INITIAL.

* Get return value
  CLEAR e_quantity.

  READ TABLE lt_cell_data INTO ls_cell_data INDEX 1.
  IF sy-subrc = 0.
    e_quantity = ls_cell_data-value.
  ENDIF.

ENDFUNCTION.
```

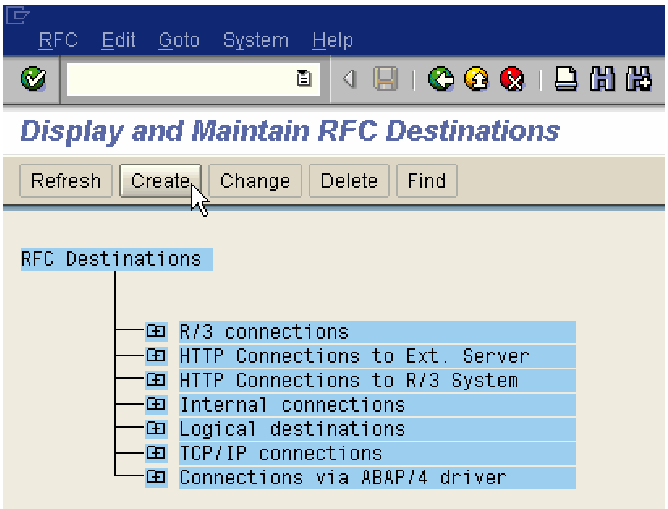
4.2 Activating the UDDI Server Built Into J2EE Engine

In SAP NetWeaver'04 documentation navigate to NetWeaver → Application Platform → Java Technology in SAP Web Application Server → Administration Manual → Server Administration → SAP J2EE Engine Administration → Web Services Container Service → Managing the UDDI Server. Follow the steps described in order to activate the UDDI Server functionality. The UDDI user you create is the one you need for step 12.

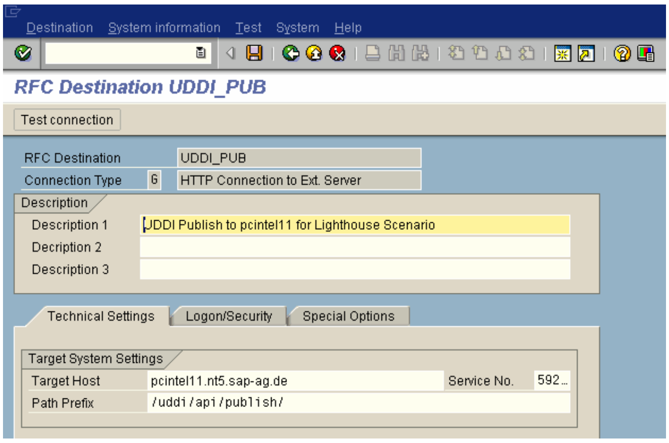
4.3 Creating a Link to the UDDI Registry

For publishing a Web Service (e.g. from transaction SE80) a link to the UDDI Registry must be defined.

1. In transaction SM59 create new RFC destinations for publishing to inquiring from the UDDI Registry.



2. Specify name, description, and connection type “G”. Fill in the server of the J2EE Engine (if the UDDI Registry of the J2EE Engine is used) and as Service No. specify the port of the J2EE Engine. The default path prefix for the J2EE UDDI Registry is displayed in the screen shot.



RFC Destination UDDI_PUB

RFC Destination		UDDI_PUB
Connection Type	6	HTTP Connection to Ext. Server

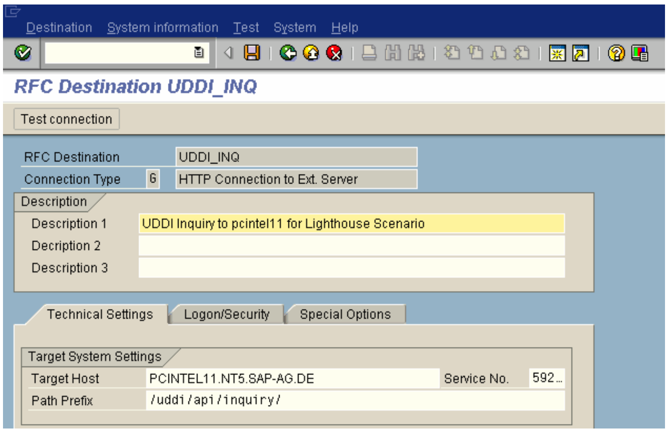
Description

Description 1	UDDI Publish to pcintel11 for Lighthouse Scenario
Decription 2	
Description 3	

Target System Settings

Target Host	pcintel11.nt5.sap-ag.de	Service No.	592..
Path Prefix	/uddi/api/publish/		

3. Create a similar entry for inquiring the Registry.



RFC Destination UDDI_INQ

RFC Destination		UDDI_INQ
Connection Type	6	HTTP Connection to Ext. Server

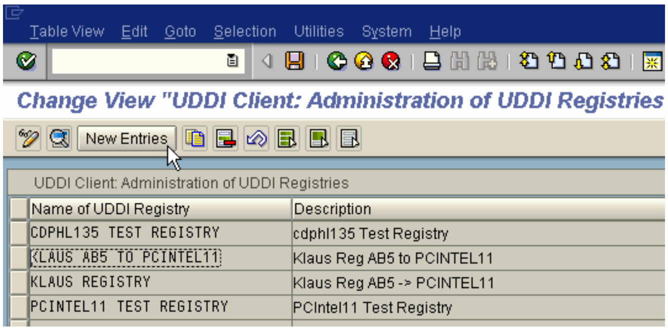
Description

Description 1	UDDI Inquiry to pcintel11 for Lighthouse Scenario
Decription 2	
Description 3	

Target System Settings

Target Host	PCINTEL11.NT5.SAP-AG.DE	Service No.	592..
Path Prefix	/uddi/api/inquiry/		

4. In transaction SUDDIREG switch to change mode and create a new entry.

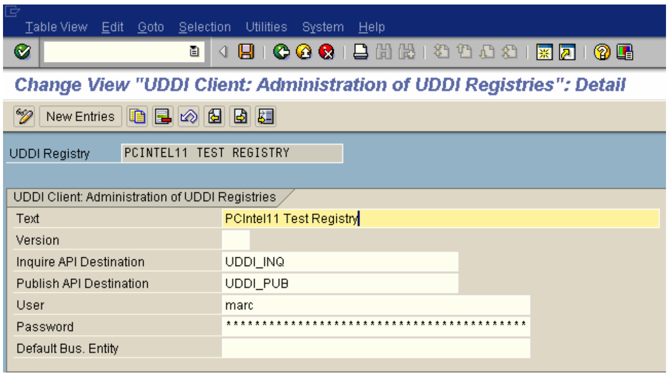


Change View "UDDI Client: Administration of UDDI Registries

UDDI Client: Administration of UDDI Registries

Name of UDDI Registry	Description
CDPHL135 TEST REGISTRY	cdphl135 Test Registry
KLAUS AB5 TO PCINTEL11	Klaus Reg AB5 to PCINTEL11
KLAUS REGISTRY	Klaus Reg AB5 -> PCINTEL11
PCINTEL11 TEST REGISTRY	PCIntel11 Test Registry

5. Specify the RFC destinations created before in the Inquire and Publish API Destinations. You can specify a user and a password (the one created for the UDDI Server). Save your new entry.



Change View "UDDI Client: Administration of UDDI Registries": Detail

UDDI Registry: PCINTEL11 TEST REGISTRY

UDDI Client: Administration of UDDI Registries

Text	PCIntel11 Test Registry
Version	
Inquire API Destination	UDDI_INQ
Publish API Destination	UDDI_PUB
User	marc
Password	**
Default Bus. Entity	

5 Glossary

BAPI	Business Application Programming Interface
J2EE	Java 2 Platform, Enterprise Edition
MDX	MultiDimensional eXpressions
OLAP	Online Analytical Processing
RFC	Remote Function Call
UDDI	Universal Description, Discovery, and Integration