



Ostia Portus

Administration

Version 2012-12-17

December 2012

This document applies to Ostia Portus 2012-12-17 15:44:26 (MET) and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

© Copyright Ostia 2012.

All rights reserved.

The name Ostia Software Solutions and/or all Ostia Software Solutions product names are either trademarks or registered trademarks of Ostia Software Solutions. Other company and product names mentioned herein may be trademarks of their respective owners.

Table of Contents

Security	ix
1 Portus Security	1
2 Portus Control Centre	3
3 Using Portus with	5
4 Portus Server Admin	7
Stopping / Starting the Portus Server - Windows	8
Stopping / Starting the Portus Server - z/OS	8
Stopping the Portus Server - z/VSE	9
Stopping / Starting Portus Server - Linux, AIX and Solaris	9
Updating an expired license	9
5 Portus Statistics	11
Server Setup	12
Service Setup	14
Driver Setup	15
Monitoring	15
Reports	16
XML	18
WSDL	19
SOAP	20
URL	20
Example	21
Service Import	23
6 Portus Governance	27
7 Web Services in Portus	29
Overview	30
The Configuration Service	30
The Resource Administration Service	30
WSDL Access	31
8 Securing your server to IP address or hostname.	33
Example 1	49
Example 2	34
Example 3	35
Example 4	35
Example 5	36
9 Using Username and/or Password	37
Authenticate with backend database	38
Apache's htpasswd	38
Authenticate with RACF	39
Enabling Web Service Authorization	40
10 SSL Certificates	45
Setup	46
Introduction	46
Step 1: Set up your own CA	46

Step 2: Portus Server key and certificate	47
Server Configuration	47
11 The Portus (legacy) Servers View	55
Working with the Servers View	56
Refresh the Portus configuration	76
Save current Configuration to server file	76
Import the Portus Configuration from server	58
Export the Portus Server configuration	59
Export Portus Webservice definitions	61
Remove a Portus Server entry	80
Create new Services	62
Modify the Portus Server definition entry	66
Modify Global -> Security Options	66
Modify Global -> Transactional Options	67
Modify Global -> Settings	67
Modify tracing / messaging options	68
Add / Modify Drivers	69
12 Advanced Topics	73
How to install a server on a remote Windows	74
Running multiple instances of Portus	74
IPv6 Considerations	76
Reducing memory footprint using streaming	77
13 Working with the Portus View	79
Operations at the Server level	80
Operations at the Driver level	86
Operations at the Service level	87
Operations at the DataView / BusinessDataView levels	87
14 Drag-and-drop in the Portus View	89
Driver definition as a drag source	90
Service as a drag source	90
DataView and BusinessDataView as a drag source	92
15 Portus BusinessDataViews	93
Overview / Concepts	94
Using the BusinessDataView Webservice	99
Working with BusinessDataView Objects	201
Additional features	102
Summary	104
Creating BusinessDataViews (legacy)	106
BusinessDataView Editor hints	130
16 The Portus Control Center Action Log	139
Working with the Action Log	140
17 Setting Preferences for the Portus Control Centre	143
Working with the Preferences	144
18 The Portus (legacy) Configuration View	147
Working with the Configuration View	148

Filter displayed Services	150
Edit a Service Definition	151
Add a Service	151
Duplicate a Service Definition	153
Duplicate a Service Definition to other server	153
Remove a Service from the server configuration	254
Import a DataView	155
Export a DataView	156
Generate Java Wrapper for a Service	157
Register a Service with an UDDI Business Registry (UBR)	157
Create a Resource	158
Delete a Resource	159
Replicate a Service	160
Refresh a Service	263
Import Service Definition files	163
19 Service Configuration	165
Service Properties	167
Service context menu	168
Lifecycle Governance	171
20 Creating a SYSOBJH extract file	175
Generating a SYSOBJH extract (Windows)	176
Generating a SYSOBJH extract (Mainframe)	187
21 Portus DataViews	197
Create a DataView from scratch	198
Opening a Portus DataView for editing	199
Editing a Portus DataView	200
Editing "special" fields	205
Enhanced Type Conversion	322
Exporting a DataView to a Portus server	325
Importing an existing Portus DataView	211
Creating a XML Schema for a DataView	211
22 Using Portus with Adabas	215
Creating a Portus DataView from an Adabas FDT	217
Create DataView(s) from a SYSOBJH unload file	217
23 Using Portus with Natural	219
Natural Service generation	220
Other ways of defining a Service	220
Preparing the Natural environment	222
Type mapping	246
24 Using Portus to access MySQL database	247
Introduction	248
MySQL Connector	248
Portus Configuration	250
Accessing MySQL	252
25 Portus Service Creation	255

Initiating Service creation	256
Creating Service(s) from Adabas	258
Assigning a different DataView to existing service	259
Creating Services from SQL databases	261
Creating Services from Stored Procedures	262
Creating Services from a SYSOBJH extract file	263
Creating Service(s) from Natural	266
Creating Service(s) from 3rd party web services	270
26 Using Portus to access Microsoft SQL Server database	275
Introduction	276
SQL Server ODBC Setup	276
Portus Configuration	279
Accessing SQL Server using Portus	281
27 Using Portus to access the Sybase database	285
Introduction	286
ODBC Setup	286
Portus Configuration	287
Accessing SQL Server using Portus	289
28 Using Portus to access Shared Libraries or DLLs	293
Introduction	294
Building the DLL	294
Portus Configuration	295
Accessing DLL using Portus	303
29 Using Portus to invoke stored procedures	307
Introduction	308
Executing Stored Procedures using Portus	308
30 Using Portus with CICS	313
Creating a Portus DataView for a CICS program	314
31 Using Portus with LE	315
Creating a Portus DataView for a LE program	316
32 Using Portus with VSAM	319
Creating a VSAM Portus Service	320
33 Using Portus to access Web Services	321
Introduction	322
Accessing Web Services	322
34 Message Oriented Middleware Support in Portus	329
Introduction	330
35 Usage Governance	331
Introduction	332
Governance Data	332
Service collection	333
Controlling Collection	334
36 Lifecycle Governance	337
Concept	338
Web Service Versioning	339

Web Service Status	339
Practical Example	340
37 HTTP Proxy Driver	343
How to use	344
38 Monitor View	345

Security

Portus Security

Portus Control Centre

Using Portus with ...

Usage Governance

Portus Statistics

1 Portus Security

This section contains information on various aspects of the Portus Security facilities.

Portus makes use of the tried and trusted security support provided by Apache. This section will provide examples on how to secure Portus using Apache security. The security support should be added to the Portus Apache configuration file on your system. Depending on your system, this file is as follows:

Linux	[INSTALL_DIR]/apache2/conf/adabas_soa_gw.conf
Windows	[INSTALL_DIR]\httpd_softwareag_adabas_asg.conf. E.g. C:\Program Files (x86)\Ostia\Portus 4.2.1\httpd_softwareag_adabas_asg.conf
z/OS	<ASG_DS>.CONF(HTCONF) Note: Just editing and saving this file on z/OS is not sufficient. This file must be copied into the Portus file system (the <ASG_DS>.PFS dataset) and the Portus server restarted for the changes to take effect. To copy these changes into the Portus file system submit the job <ASG_DS>.JCL(SOAGINS3) <ASG_DS> is the HLQ where Portus has been installed.

Web Services in Portus

Securing your server to IP address or hostname

Using Username and/or Password

SSL Certificates

2 Portus Control Centre

This section describes how to work with the Portus Control Centre. This is the client tool used for administrating one or more Portus server(s).

Please read the Introduction to the Portus Control Centre before continuing.

The following sections will explain the individual Portus Eclipse perspectives and views available to the administrator, and how to work with each.

The Portus Administration Perspective

- **The Portus View**
- **Drag-and-drop support in the Portus View**
- **Service Properties**
- **Monitor View**

The Portus "Legacy" Perspective

- **The Servers View**
- **The Configuration View**
- **The Action Log View**
- **The Statistics View**

Other aspects:

- **The Portus "DataView"**
- **Setting Preferences**

3

Using Portus with ...

Adabas	Ultra-high performance, high-availability transactional database
Natural	Comprehensive high-productivity development environment
Shared Libraries or DLLs	Invoke 3gl programs available as DLLs or shared libraries on any platform
MySQL	An OpenSource relational database
MS SQL Server	The relational database for the Windows platform
Sybase	Adaptive Server Anywhere provides a full-featured SQL database and is part of the EAServer product set from Sybase
Stored Procedures	Subroutines available to applications accessing a relational database system
CICS	Invoke 3gl (sub)programs accessible through the TP-Monitor CICS
LE	Invoke LE (Language Environment) enabled 3gl (sub)programs on IBM mainframes
VSAM	Access data in VSAM datasets on IBM mainframes
Web Services	Access business logic, data and processes via Web Services
Message Oriented Middleware	Access data on Software AG EntireX Broker and IBM WebSphere MQ
HTTP Proxy (Pass-thru) Driver	A proxy driver to connect to external services and enable them to be called via Portus

4 Portus Server Admin

- Stopping / Starting the Portus Server - Windows 8
- Stopping / Starting the Portus Server - z/OS 8
- Stopping the Portus Server - z/VSE 9
- Stopping / Starting Portus Server - Linux, AIX and Solaris 9
- Updating an expired license 9

Most tasks will be handled by the **Portus Control Centre**, which presents a coordinated graphical view of the server. Tasks that are not possible from the Control Centre are documented here.

Stopping / Starting the Portus Server - Windows

The Portus server runs from within the Apache webserver. You may see a new icon in the system tray



If not, start the program by clicking **Start -> Portus vN -> Apache -> Monitor Apache Servers**

Clicking on this symbol will give you the option to choose your Portus Server and the options **Start**, **Stop**, and **Restart**. **Start** or **Stop** will be greyed out depending on the status of Portus.

It is strongly recommended to use **Start** and **Stop** to cycle Portus, *not* **Restart**

The green arrow above indicates that Portus is up and running.

If Portus is not running the icon looks like this



Stopping / Starting the Portus Server - z/OS

Portus can be stopped by issuing `/p jobname` at the terminal.

To start Portus, re-submit the ASGSTRT job.

Stopping the Portus Server - z/VSE

- Bring up the Attention Routine prompt for the Portus partition by issuing the AR command `MSG <ASG_partition_ID>`
- Shut down the server with the command `<ASG_partition_reply_id> EOJ`

Stopping / Starting Portus Server - Linux, AIX and Solaris

As Portus runs within Apache, bring down Apache to stop the Portus. Use the `apachectl` command, which will be located in the `[INSTALL_HOME]/apache2/bin` directory of your Portus installation. To start the server, issue the same command again with `start` as the first parameter.

Updating an expired license

If the Portus license has expired, please follow these steps to upgrade your license.

- *Linux/Solaris/HP/AIX* : Replace the existing `RisarisLicense.xml` in `[INSTALL_HOME]/config`
- *Windows* : Replace the existing `RisarisLicense.xml` `[INSTALL_HOME]/configuration`
- *z/OS* : Replace existing `INSTALL.HOME.CONF(ASGLIC)` and submit `INSTALL.HOME.JCL(SOAGINS4)`.

5 Portus Statistics

▪ Server Setup	12
▪ Service Setup	14
▪ Driver Setup	15
▪ Monitoring	15
▪ Reports	16
▪ XML	18
▪ WSDL	19
▪ SOAP	20
▪ URL	20
▪ Example	21
▪ Service Import	23

This section describes the statistics available in Portus.

The existing functionality helps determine the level of work flow activity through a server system. The infrastructure is also in place to enable other forms of statistics gathering such as performance monitoring, number of users and so on.

Work flow monitoring is set off by default. When on it will provide Key Performance Indicators (KPIs) on well know operations list, get, add, delete, update, invoke, select, selectCount, selectNext and selectEnd.

KPIs can be gathered at the server level, web service level or at the driver level.

The KPIs gathered for each operation are as follows:

- **total_count**

This is the number of times the operation has been called. If successful then the round trip time is added to total_time. If applicable high_time or low_time may be updated. If the operation is unsuccessful then errors_occurred is incremented.

- **total_time**

This is the accumulated round trip time for the operation.

- **high_time**

This is the highest round trip time for the operation.

- **low_time**

This is the lowest round trip time for the operation.

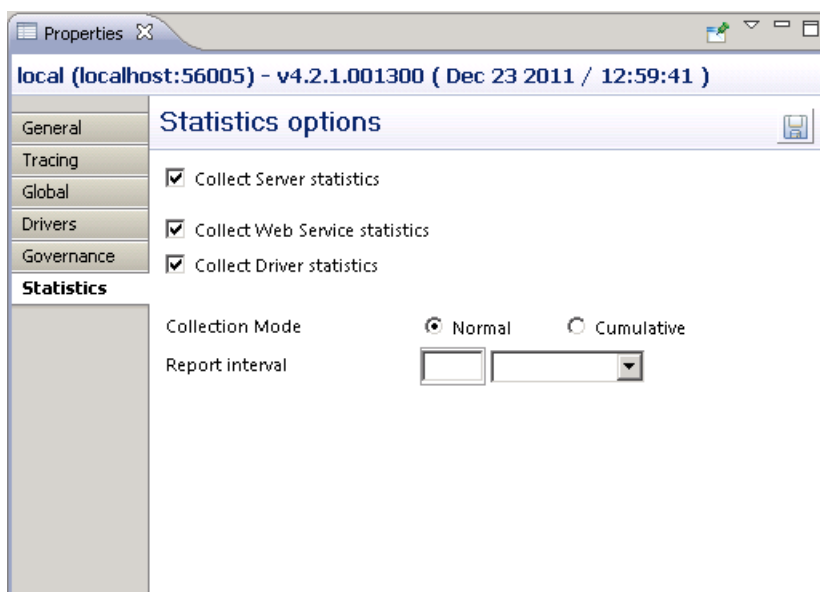
- **errors_occurred**

This is the number of times the operation has failed.

Server Setup

Setup is carried out using the Portus Control Centre.

Click on the server to be monitored in the list of Portus Servers. Then click on the Statistics tab within the Properties View as shown below:



- Collect Server statistics must be set on to enable any level of monitoring. This produces data at the server level.
- Collect Service statistics enables gathering at the web service level for all services.
- Collect Driver statistics enables gathering at the driver level.

Statistics can be collected in 2 modes, Normal and Cumulative. An interval can be set to control when a report is written.

Normal

Statistics are gathered from the point at which they are set on up to when the server is stopped or when they are set off or when the mode is changed to Cumulative or when the report interval has been reached. A file is written with the KPIs at that point. The name of the file has the format soag_normal_statistics_YYYY_MM_DD_HH_MM_SS_millisecond.xml e.g.

soag_normal_statistics_2008_10_09_12_56_11_364.xml

Cumulative

Statistics gathering continues over server stop/starts. When the server is stopped the data is written to the file soag_cumulative_statistics.xml. On restart the same file is read and used to initialise the statistics.

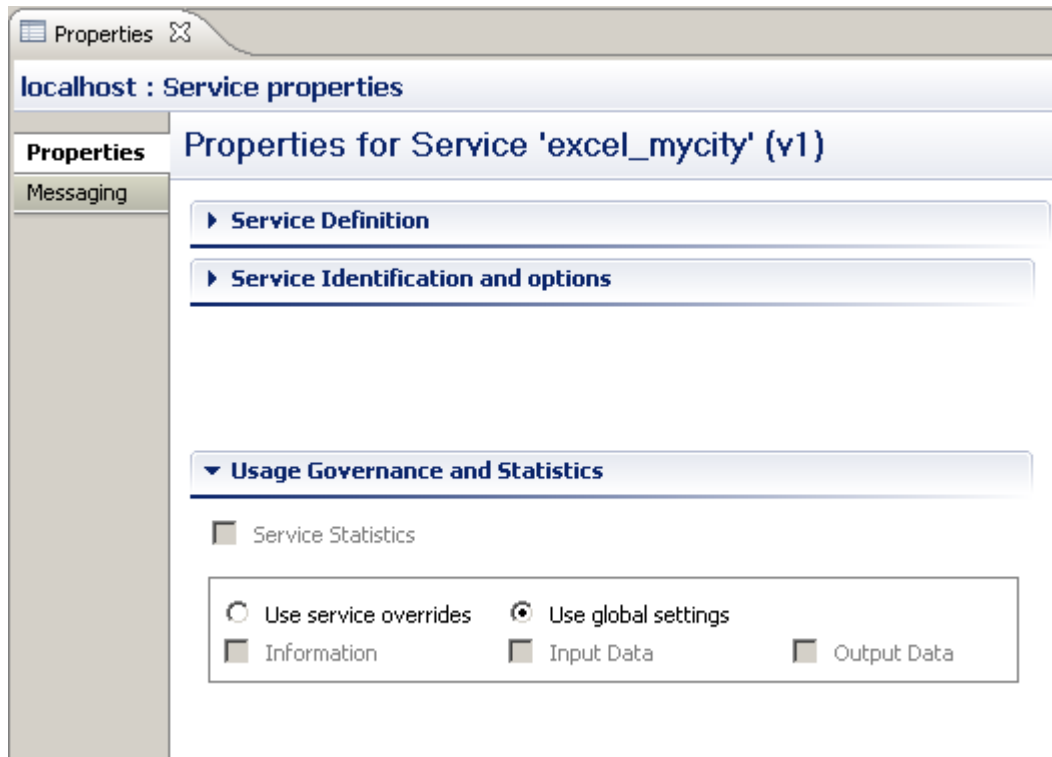
Report Interval

The default behaviour is for a report to be written when the server is stopped. Setting an interval enables reporting at set periods.

Service Setup

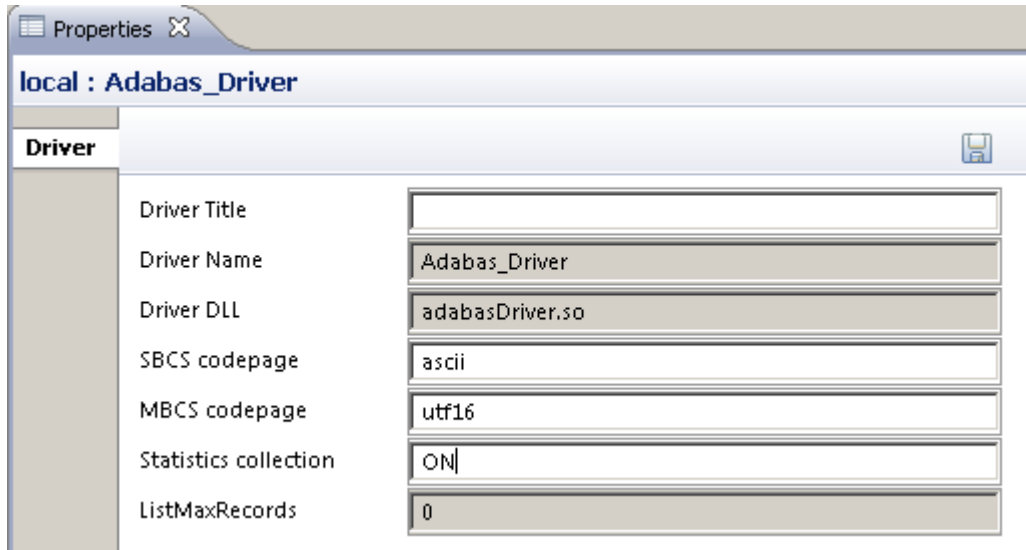
Click on the service to be monitored. In the Properties View set *Service Statistics* on/off. Note that the global setting overrides this setting and it will be greyed out if set on at the global level as shown below.

If you wish to monitor for a particular service set the global setting off and turn on at the service level.



Driver Setup

Click on the driver to be monitored. In the Properties View set Statistics collection to ON/OFF.



Monitoring

Statistics monitoring can be carried out on an ad hoc basis, again using the Portus Control Centre.

Open/click on the Portus Statistics view. If this view is not visible open menu items Window --> Show View --> Other... and select Portus Statistics under the Portus Administration folder.

The screenshot shows the SOA Gateway Statistics application interface. At the top, there are tabs for 'Navigator', 'Console', and 'SOA Gateway Statistics'. Below the tabs, there are three dropdown menus: 'Server' (set to 'local'), 'Group' (set to '*'), and 'Type' (set to '*'). There is also a 'Name' dropdown menu set to '*'. To the right of these dropdowns are two buttons: 'Reset Statistics' and 'Show Statistics'. Below the dropdowns is a section titled 'Server Statistics' which contains a table with the following data:

Group	Type	Name	Requests	Errors	Total time	Time high	Time low
Server	Request	list	2	0	2.034	1.984	0.050
Server	Request	get	0	0	0.000	0.000	0.000
Server	Request	delete	0	0	0.000	0.000	0.000
Server	Request	add	0	0	0.000	0.000	0.000
Server	Request	update	0	0	0.000	0.000	0.000
Server	Request	invoke	0	0	0.000	0.000	0.000
Server	Request	select	0	0	0.000	0.000	0.000
Server	Request	selectCount	0	0	0.000	0.000	0.000
Server	Request	selectNext	0	0	0.000	0.000	0.000
Server	Request	selectEnd	0	0	0.000	0.000	0.000
Service	adabas_EmployeesMini_...	list	0	0	0.000	0.000	0.000
Service	adabas_EmployeesMini_...	get	0	0	0.000	0.000	0.000
Service	adabas_EmployeesMini_...	add	0	0	0.000	0.000	0.000
Service	adabas_EmployeesMini_...	delete	0	0	0.000	0.000	0.000
Service	adabas_EmployeesMini_...	update	0	0	0.000	0.000	0.000
Service	adabas_EmployeesMini_...	invoke	0	0	0.000	0.000	0.000
Service	adabas_EmployeesMini_...	select	0	0	0.000	0.000	0.000
Service	adabas_EmployeesMini_...	selectCount	0	0	0.000	0.000	0.000
Service	adabas_EmployeesMini_...	selectNext	0	0	0.000	0.000	0.000

Pick the server from the Server listbox. Group, Type and Name can be wildcarded i.e. set to * or set to a particular value e.g. Adabas_Driver

Note that the wildcarding is asterisk only i.e. Adabas* is not acceptable.

Then click on the Show Statistics button to refresh the data.

The statistics can be reset at any time by selecting the Reset Statistics button.

Reports

The format of the normal and cumulative report is identical. Here is an extract from a normal file:

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Statistics>
  <Group id="Server">
    <Type id="Request">
      <Name id="list">
        <total_count>4</total_count>
        <errors_ocurred>0</errors_ocurred>
        <total_time>0.223</total_time>
        <high_time>0.097</high_time>
        <low_time>0.014</low_time>
      </Name>
      <Name id="get">
        <total_count>0</total_count>
        <errors_ocurred>0</errors_ocurred>
        <total_time>0.000</total_time>
        <high_time>0.000</high_time>
        <low_time>0.000</low_time>
      </Name>
      <Name id="delete">
        <total_count>0</total_count>
        <errors_ocurred>0</errors_ocurred>
        <total_time>0.000</total_time>
        <high_time>0.000</high_time>
        <low_time>0.000</low_time>
      </Name>
      <Name id="add">
        <total_count>0</total_count>
        <errors_ocurred>0</errors_ocurred>
        <total_time>0.000</total_time>
        <high_time>0.000</high_time>
        <low_time>0.000</low_time>
      </Name>
      <Name id="update">
        <total_count>0</total_count>
        <errors_ocurred>0</errors_ocurred>
        <total_time>0.000</total_time>
        <high_time>0.000</high_time>
        <low_time>0.000</low_time>
      </Name>
    </Type>
  </Group>
</Statistics>

```

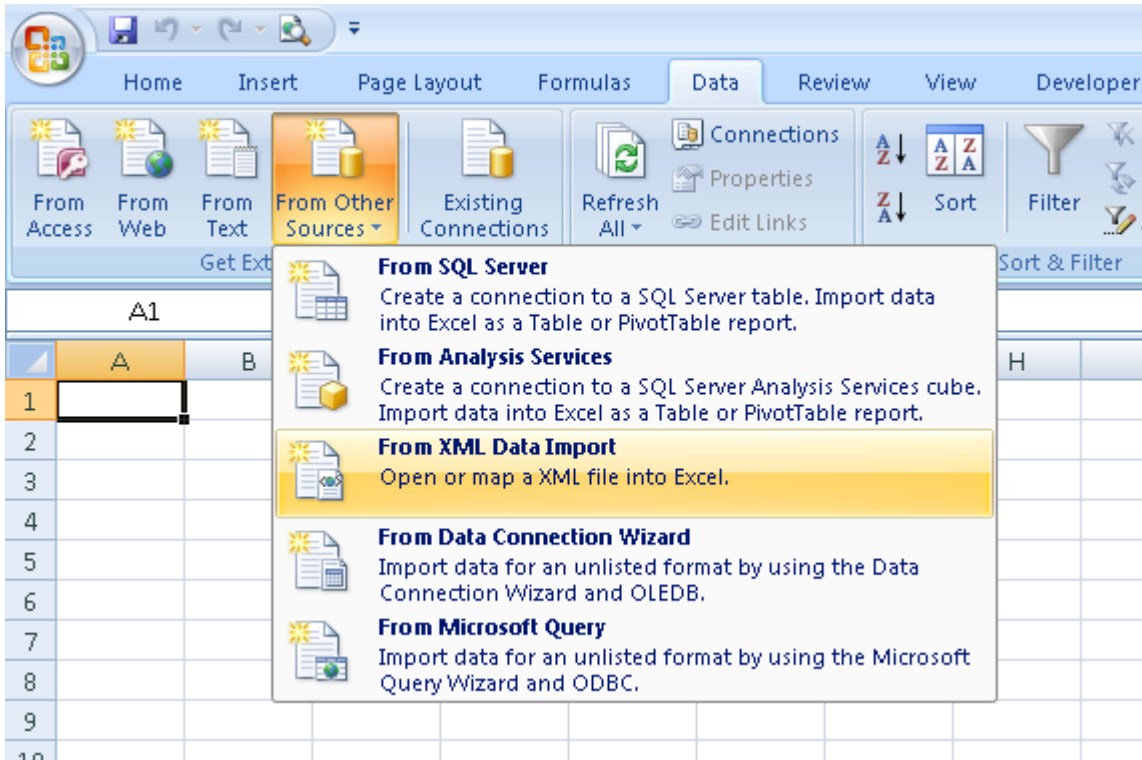
1,1 (0/100552) (xml:none,UTF-8)Nm r o UG 12/27Mb 2 error(s)11:00

The statistic reports are written to the configuration folder where Portus is installed.

XML

As the statistics report is an XML file there are many programs which can consume this format e.g. In Excel you can import the XML file from

the Portus configuration directory :



resulting in a very readable format::

	A	B	C	D	E	F	G	H	I
1	id	id2	version	id3	total_count	errors_occurred	total_time	high_time	low_time
2	Server	Request		list	4	0	0.223	0.097	0.014
3	Server	Request		get	0	0	0	0	0
4	Server	Request		delete	0	0	0	0	0
5	Server	Request		add	0	0	0	0	0
6	Server	Request		update	0	0	0	0	0
7	Server	Request		invoke	0	0	0	0	0
8	Server	Request		select	0	0	0	0	0
9	Server	Request		selectCount	0	0	0	0	0
10	Server	Request		selectNext	0	0	0	0	0
11	Server	Request		selectEnd	0	0	0	0	0
12	Service	adabas_EmployeesMini_veh	1	list	0	0	0	0	0
13	Service	adabas_EmployeesMini_veh	1	get	0	0	0	0	0
14	Service	adabas_EmployeesMini_veh	1	add	0	0	0	0	0
15	Service	adabas_EmployeesMini_veh	1	delete	0	0	0	0	0
16	Service	adabas_EmployeesMini_veh	1	update	0	0	0	0	0
17	Service	adabas_EmployeesMini_veh	1	invoke	0	0	0	0	0
18	Service	adabas_EmployeesMini_veh	1	select	0	0	0	0	0
19	Service	adabas_EmployeesMini_veh	1	selectCount	0	0	0	0	0
20	Service	adabas_EmployeesMini_veh	1	selectNext	0	0	0	0	0
21	Service	adabas_EmployeesMini_veh	1	selectEnd	0	0	0	0	0
22	Service	adabas_EmployeesFull_vehlist	1	list	0	0	0	0	0
23	Service	adabas_EmployeesFull_vehlist	1	get	0	0	0	0	0
24	Service	adabas_EmployeesFull_vehlist	1	add	0	0	0	0	0
25	Service	adabas_EmployeesFull_vehlist	1	delete	0	0	0	0	0
26	Service	adabas_EmployeesFull_vehlist	1	update	0	0	0	0	0
27	Service	adabas_EmployeesFull_vehlist	1	invoke	0	0	0	0	0
28	Service	adabas_EmployeesFull_vehlist	1	select	0	0	0	0	0
29	Service	adabas_EmployeesFull_vehlist	1	selectCount	0	0	0	0	0
30	Service	adabas_EmployeesFull_vehlist	1	selectNext	0	0	0	0	0
31	Service	adabas_EmployeesFull_vehlist	1	selectEnd	0	0	0	0	0
32	Service	test	1	list	0	0	0	0	0
33	Service	test	1	get	0	0	0	0	0
34	Service	test	1	add	0	0	0	0	0

WSDL

Access to statistics is also made available via the *statService* web service as described by the WSDL <http://server:port/statService?WSDL>.

SOAP

Statistics can be retrieved via a SOAP request using the statService web service. Using a WSDL consumer such as soapUI, as shown below, a request can be made using Group, Type and Name. As with the Control Centre, the '*' construct enables the entire set of statistics to be returned or a subset. The Reset option can be set to True or False as required. The default value is False if omitted.

The screenshot shows a SOAP request and response in soapUI. The request is sent to `http://localhost:56000/statService`. The response is received from `http://schemas.xmlsoap.org/ws/2002/04/secext`.

```

Request 1
http://localhost:56000/statService
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <stat:getStatsInputElement>
      <Group>*</Group>
      <Type>*</Type>
      <Name>*</Name>
      <Reset>False</Reset>
    </stat:getStatsInputElement>
  </soapenv:Body>
</soapenv:Envelope>

Response
http://schemas.xmlsoap.org/ws/2002/04/secext
<soapenv:Envelope xmlns:sec="http://schemas.xmlsoap.org/ws/2002/04/secext" xmlns:xml="http://www.w3.org/XML/1998/namespace" xmlns:rsp="com.SOAGateway/statServiceGroup">
  <soapenv:Body>
    <rsp:getStatsOutputElement xmlns:rsp="com.SOAGateway/statServiceGroup">
      <Statistics>
        <Group id="Server">
          <Type id="Request">
            <Name id="list">
              <total_count>0</total_count>
              <errors_ocurred>0</errors_ocurred>
              <total_time>0.000</total_time>
              <high_time>0.000</high_time>
              <low_time>0.000</low_time>
            </Name>
            <Name id="get">
              <total_count>0</total_count>
              <errors_ocurred>0</errors_ocurred>
              <total_time>0.000</total_time>
              <high_time>0.000</high_time>
              <low_time>0.000</low_time>
            </Name>
          </Type>
        </Group>
      </Statistics>
    </rsp:getStatsOutputElement>
  </soapenv:Body>
</soapenv:Envelope>

```

URL

The statistics can also be retrieved via a URL-based request in a browser such as Internet Explorer. The statService service exposes one method, GET. As with previous requests this can be restricted by Group, Type and Name. Again the '*' construct enables the entire set of statistics to be returned or a subset by specifying a value. The option is also provided to reset the current set of statistics. The example below restricts the search to the list method.

`http://localhost:56005/statService?GET&Group=*&Type=*&Name=list`

```

http://localhost:56005/statService?GET&Group=*&Type=*&Name=list - Windows Internet Explorer
http://localhost:56005/statService?GET&Group=*&Type=*&Name=list
http://localhost:56005/statService?GET&Group=*&Ty...

</Name>
</Type>
- <Type id="MySQL QE Driver">
  - <Name id="list">
    <total_count>0</total_count>
    <errors_occurred>0</errors_occurred>
    <total_time>0.000</total_time>
    <high_time>0.000</high_time>
    <low_time>0.000</low_time>
  </Name>
</Type>
- <Type id="naturalDriver_QE">
  - <Name id="list">
    <total_count>0</total_count>
    <errors_occurred>0</errors_occurred>
    <total_time>0.000</total_time>
    <high_time>0.000</high_time>
    <low_time>0.000</low_time>
  </Name>
</Type>
- <Type id="Adabas_Driver">
  - <Name id="list">
    <total_count>5</total_count>
    <errors_occurred>5</errors_occurred>
    <total_time>0.265</total_time>
    <high_time>0.171</high_time>
    <low_time>0.094</low_time>
  </Name>
</Type>
</Group>
- <Group id="Service">
  - <Type id="MyCity" version="1">
    - <Name id="list">
      <total_count>0</total_count>
      <errors_occurred>0</errors_occurred>
      <total_time>0.000</total_time>
      <high_time>0.000</high_time>
      <low_time>0.000</low_time>
    </Name>
  </Type>
</Group>
</Statistics>

```

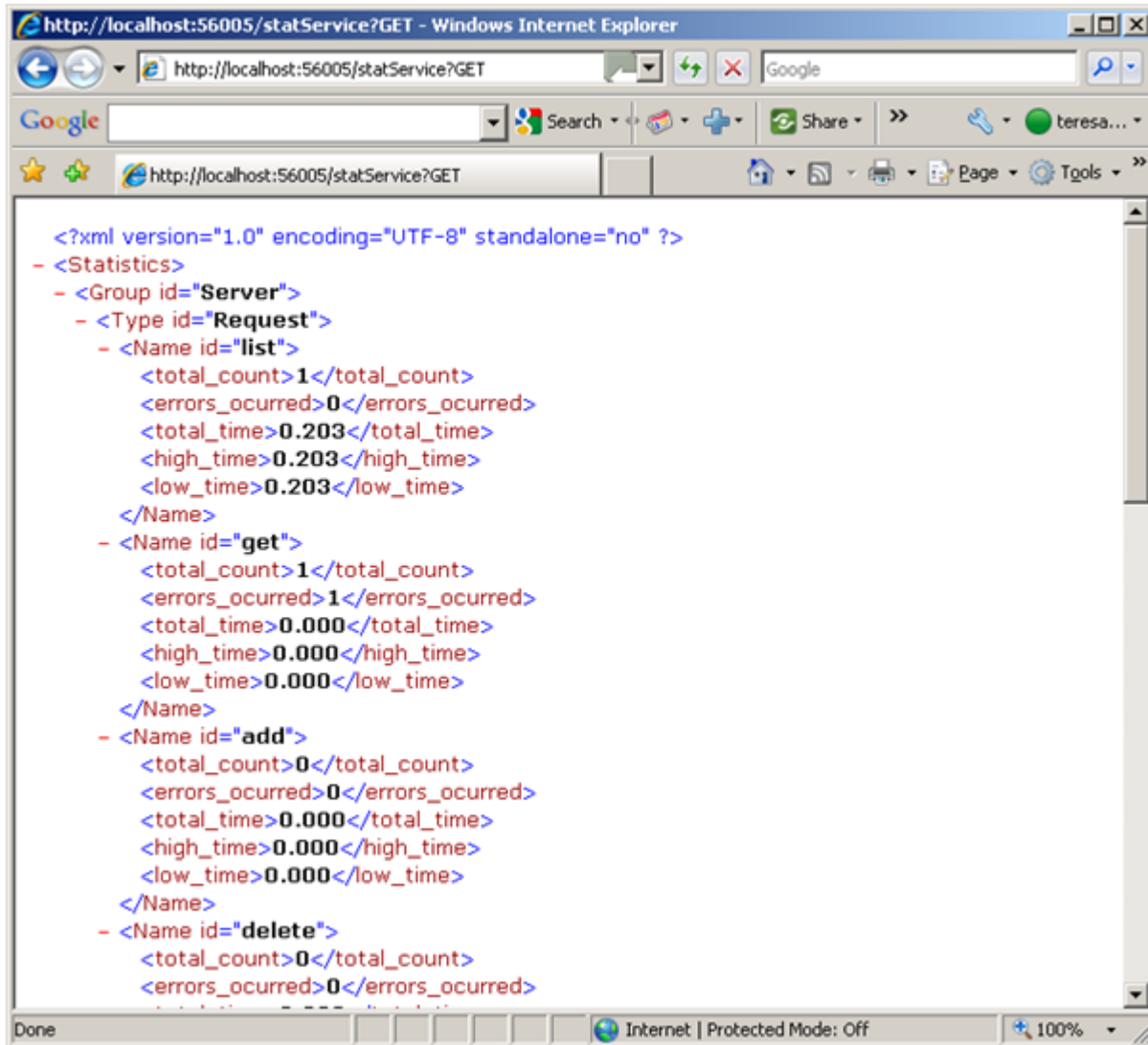
Example

Let us assume that the server has been started for the first time and that statistics collection has been set on for the first time. An Adabas database is available and a web service created to access it. Both queries below are issued via a browser such as IE. The LIST successfully retrieves the data for employees whose ids start with 200. The GET query fails because the syntax is incorrect.

http://localhost:56005/Adabas_Driver_1_9?GET=20020000

http://localhost:56005/Adabas_Driver_1_9?LIST&AA=200*

Now get the statistics so far by issuing a GET on statService as shown below. Note that the entry for the get request does not show any effect on the times. Note that for list, for the moment, all times are identical. The time is 0.203 or 203 milliseconds. This is the round trip time spent in the Portus Server.



```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Statistics>
- <Group id="Server">
- <Type id="Request">
- <Name id="list">
  <total_count>1</total_count>
  <errors_ocurred>0</errors_ocurred>
  <total_time>0.203</total_time>
  <high_time>0.203</high_time>
  <low_time>0.203</low_time>
</Name>
- <Name id="get">
  <total_count>1</total_count>
  <errors_ocurred>1</errors_ocurred>
  <total_time>0.000</total_time>
  <high_time>0.000</high_time>
  <low_time>0.000</low_time>
</Name>
- <Name id="add">
  <total_count>0</total_count>
  <errors_ocurred>0</errors_ocurred>
  <total_time>0.000</total_time>
  <high_time>0.000</high_time>
  <low_time>0.000</low_time>
</Name>
- <Name id="delete">
  <total_count>0</total_count>
  <errors_ocurred>0</errors_ocurred>

```

Let us issue the LIST and GET again, this time correcting the GET syntax and changing the LIST to retrieve ids starting with 20:

http://localhost:56005/Adabas_Driver_1_9?GET&AA=20020000

http://localhost:56005/Adabas_Driver_1_9?LIST&&AA=20*

Note the now successful entry for the get operation. Note also that for list the high time is now 0.234 as the query took slightly longer. As there have been only 2 list requests so far the total time equals the sum of low and high times.

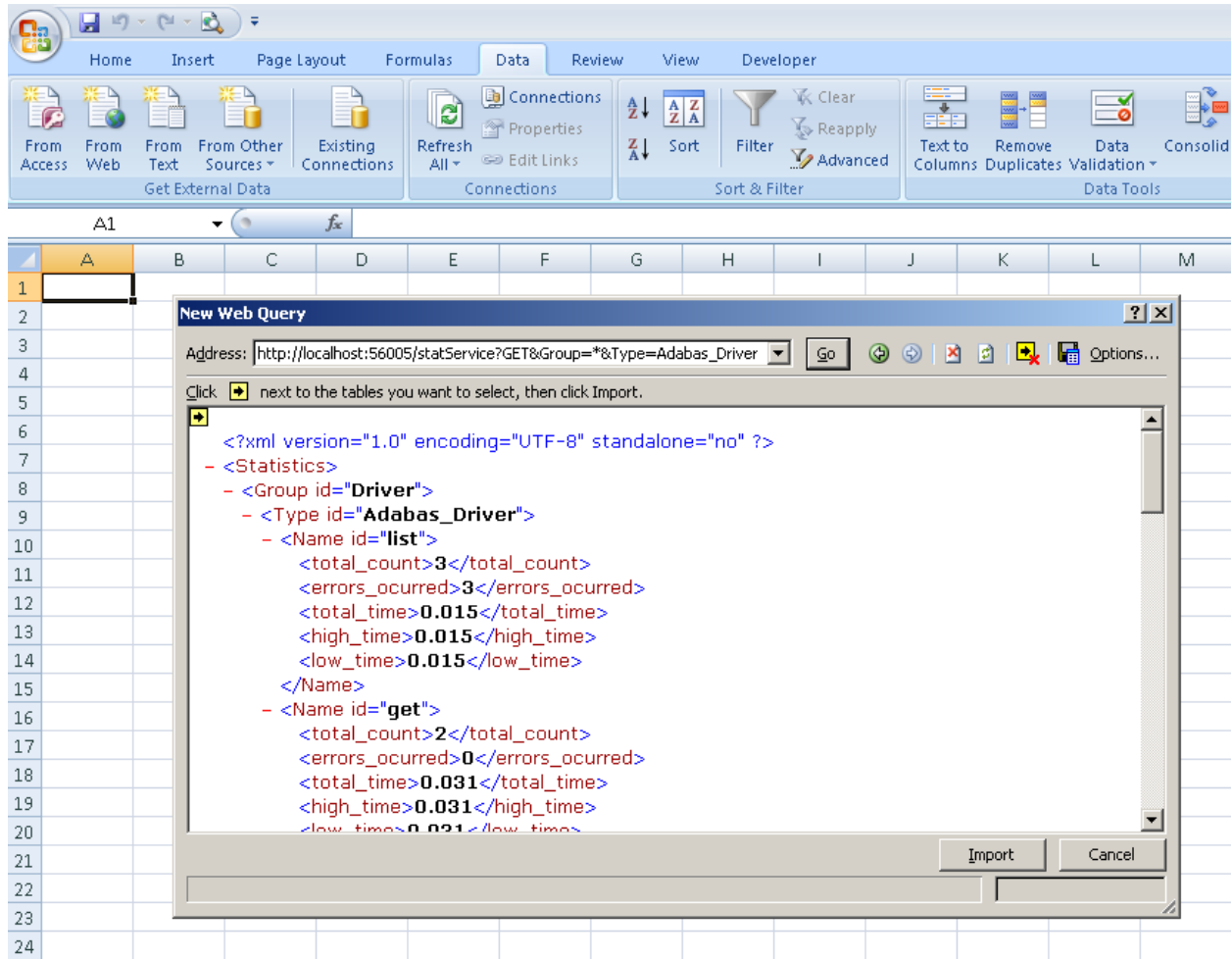

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Statistics>
- <Group id="Server">
- <Type id="Request">
- <Name id="list">
  <total_count>2</total_count>
  <errors_ocurred>0</errors_ocurred>
  <total_time>0.437</total_time>
  <high_time>0.234</high_time>
  <low_time>0.203</low_time>
</Name>
- <Name id="get">
  <total_count>2</total_count>
  <errors_ocurred>1</errors_ocurred>
  <total_time>0.016</total_time>
  <high_time>0.016</high_time>
  <low_time>0.016</low_time>
</Name>
- <Name id="add">
  <total_count>0</total_count>
  <errors_ocurred>0</errors_ocurred>
  <total_time>0.000</total_time>
  <high_time>0.000</high_time>
  <low_time>0.000</low_time>
</Name>
- <Name id="delete">
  <total_count>0</total_count>
  <errors_ocurred>0</errors_ocurred>

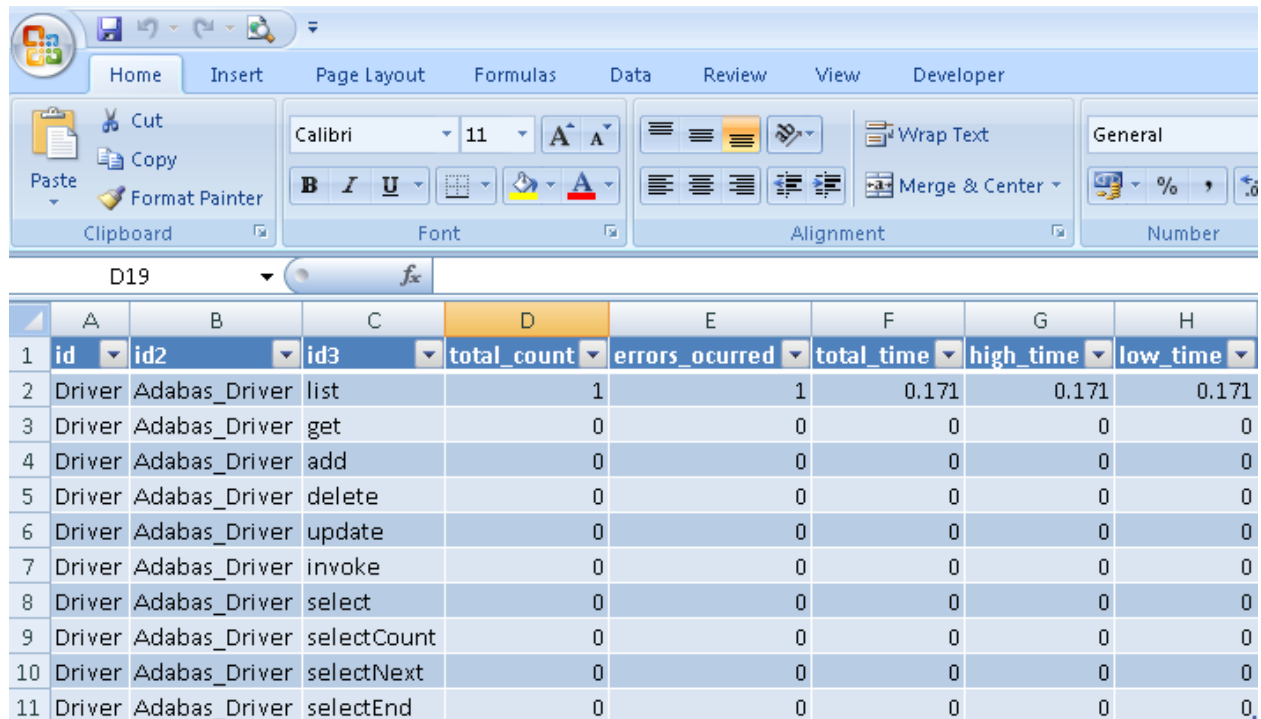
```

Service Import

The ability to use a URL request also enables us to use Excel to query the latest statistics. Open Excel and navigate to Data -> From Web. Enter the required request as shown below and select Go. The result should be returned as shown:

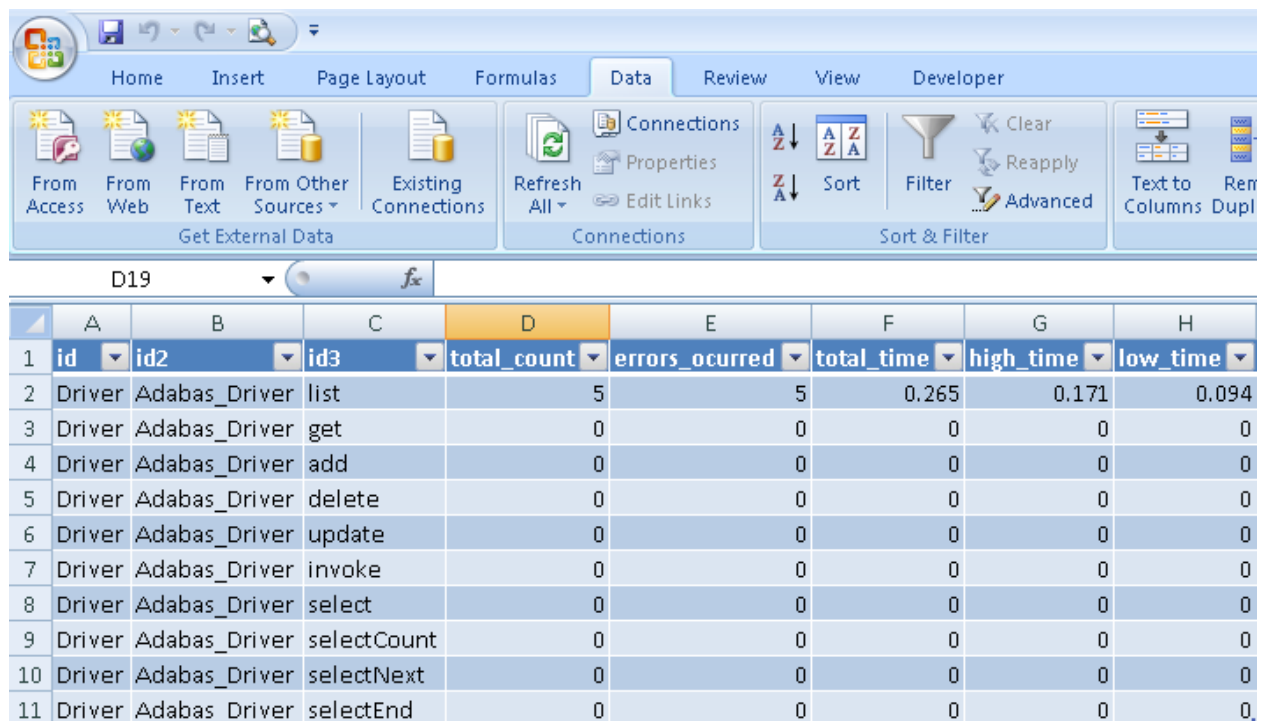


Select Import and accept the default location of A1.



	A	B	C	D	E	F	G	H
1	id	id2	id3	total_count	errors_ocurred	total_time	high_time	low_time
2	Driver	Adabas_Driver	list	1	1	0.171	0.171	0.171
3	Driver	Adabas_Driver	get	0	0	0	0	0
4	Driver	Adabas_Driver	add	0	0	0	0	0
5	Driver	Adabas_Driver	delete	0	0	0	0	0
6	Driver	Adabas_Driver	update	0	0	0	0	0
7	Driver	Adabas_Driver	invoke	0	0	0	0	0
8	Driver	Adabas_Driver	select	0	0	0	0	0
9	Driver	Adabas_Driver	selectCount	0	0	0	0	0
10	Driver	Adabas_Driver	selectNext	0	0	0	0	0
11	Driver	Adabas_Driver	selectEnd	0	0	0	0	0

The imported result can be refreshed to display changes in the statistic being queried. Select Data -> Refresh All.



	A	B	C	D	E	F	G	H
1	id	id2	id3	total_count	errors_ocurred	total_time	high_time	low_time
2	Driver	Adabas_Driver	list	5	5	0.265	0.171	0.094
3	Driver	Adabas_Driver	get	0	0	0	0	0
4	Driver	Adabas_Driver	add	0	0	0	0	0
5	Driver	Adabas_Driver	delete	0	0	0	0	0
6	Driver	Adabas_Driver	update	0	0	0	0	0
7	Driver	Adabas_Driver	invoke	0	0	0	0	0
8	Driver	Adabas_Driver	select	0	0	0	0	0
9	Driver	Adabas_Driver	selectCount	0	0	0	0	0
10	Driver	Adabas_Driver	selectNext	0	0	0	0	0
11	Driver	Adabas_Driver	selectEnd	0	0	0	0	0

6 Portus Governance

There are 2 ways to apply governance to Portus

- **Usage Governance**
- **Lifecycle Governance**

7 Web Services in Portus

- Overview 30
- The Configuration Service 30
- The Resource Administration Service 30
- WSDL Access 31

The methods supported by Portus are provided by Web Services. Thus, when an administrator uses the Control Centre to change the Portus configuration, they are calling a web service. It is the same when a user issues a request from .NET, PHP, etc, to access data; they are calling a web service.

Overview

Portus provides one dynamic and two static web services.

1. The static configuration service: `http://<hostname>:<port>/configurationService`
2. The static resource administration service: `http://<hostname>:<port>/resourceService`
3. The dynamic resource service: `http://<hostname>:<port>/<resource_name>` Appending “?WSDL” to the end of these URLs will return the WSDL for that particular web service, (see more information below). `<hostname>` and `<port>` (if required) will be the respective IP address/hostname and port of your Portus server. `<resource_name>` will be dynamically created based on the resource name you set in the configuration. E.g. `http://myServer/adabas/Employees` would be the endpoint for the “adabas/Employees” resource running on “myServer” (port 80).

The Configuration Service

The configuration service is used to create, delete and update the current configuration of Portus. It is also used to import and export the view of a particular resource, which in Eclipse terms is the XRD file. In Eclipse, the configuration service is used when using the Portus Import or Export wizards.

The Resource Administration Service

The resource administration service is used to create and delete resources based on a specific resource name. This should not be confused with defining a resource in the configuration. For example, the configuration service could be used (via Eclipse) to create a new resource name “adabas_newResource”. The resource administration service would be used to define an Adabas file on this resource. The resource administration service could also be used to delete (or drop) this Adabas file. In Eclipse, the resource administration service is used when running the “Portus Create Resource” or “Portus Delete Resource” wizards.

WSDL Access

An administrator can also prohibit usage of the service, but allow a client to view the description of the web service. A standard language called WSDL describes what this web service looks like. To access the WSDL for a web service, the client should append the “?WSDL” argument to the web service endpoint. E.g. to get the WSDL for the configuration service, the following URL could be entered into a browser: `http://myServer/configurationService?WSDL`. This provides the WSDL specifically for this web service. A client who understands WSDL will now know how to access this web service. Normally a ‘client’ will be software as the WSDL is not intended to be read by humans in the normal course of events. To use a web service, the client constructs and sends a SOAP request based on the WSDL. The WSDL also describes what the SOAP response will look like. In terms of security, the WSDL access is a HTTP GET request. The web service usage is HTTP POST request.

8

Securing your server to IP address or hostname.

▪ Example 1	49
▪ Example 2	34
▪ Example 3	35
▪ Example 4	35
▪ Example 5	36

This section will outline how to secure the Portus Web Services to a specific IP address/hostname.

All examples have been enclosed in <IfModule> directives. This means that the security directives will be ignored automatically if the Apache web server does not have Portus enabled.

Example 1

This example will show how to allow a request to the configuration service from only the local machine. This is local to where the Portus server is running, not where the Eclipse IDE is running (if different).

1. Edit the Portus Apache configuration file.
2. Enter the following directives.

```
<IfModule mod_xmiddle.c>
  <Location /configurationService>
    Allow from 127.0.0.1
    Deny from all
  </Location>
</IfModule>
```

3. Restart the server

This would only allow access to the configuration web service from the local machine. All remote clients would be denied access.

Example 2

Only allow a remote machine to configure the Portus server.

1. Edit the Portus Apache configuration file
2. Enter the following directives.

```
<IfModule mod_xmiddle.c>
  <Location /configurationService>
    Allow from adminHost
    Deny from all
  </Location>
</IfModule>
```

3. Restart the server

This example would only allow the machines adminHost to configure the Portus server. All others machines would be rejected access.

Example 3

Only allow a remote machine to configure Portus, but allow any client to access the Web Service WSDL.

1. Edit the Portus Apache configuration file
2. Enter the following directives:

```
<IfModule mod_xmiddle.c>
  <Location /configurationService>
    <Limit POST>
      Allow from adminHost
      Deny from all
    </Limit>
  </Location>
</IfModule>
```

3. Restart the server.

This would allow the machine adminHost to access the configuration, but would allow any client to access the configuration service WSDL.

Example 4

This example will show how to secure a specific resource. The examples 1, 2, and 3 above can also be applied to securing a resource. The only thing that has to change is the Location parameter. For example, using Example 1 as a basis; to only allow “adabas_Employees” to be accessed from the local machine, do the following:

1. Edit the Portus Apache configuration file
2. Enter the following directives.

```
<IfModule mod_xmiddle.c>
  <Location /adabas_Employees>
    Allow from 127.0.0.1
    Deny from all
  </Location>
</IfModule>
```

3. Restart the server

All remote access to the “adabas_Employees” resource would be denied. Note: This example will restrict access to the “adabas_Employees” service, not the XRD import or export. XRD import/export operations are provided by the configuration Web Service, to secure these operations see examples 1, 2 and 3.

Example 5

This example will show how to secure the resource administration service for Portus. Again, example 1-4 may be used once the Location parameter has been changed. For example, using Example 1 as a basis; to only allow the resource administration service to be accessed from the local machine, do the following:

1. Edit the Portus Apache configuration file
2. Enter the following directives.

```
<IfModule mod_xmiddle.c>
  <Location /resourceService>
    Allow from 127.0.0.1
    Deny from all
  </Location>
</IfModule>
```

3. Restart the server

All remote access to the “resourceService” resource would be denied.

9 Using Username and/or Password

- Authenticate with backend database 38
- Apache's htpasswd 38
- Authenticate with RACF 39
- Enabling Web Service Authorization 40

Authenticate with backend database

This is the simplest and most convenient form of authentication. When a request hits Portus the supplied security credentials are provided to the DBMS for authentication.

This uses the Portus Apache directive "SoaGatewaySecurityLevel (none|user|pass|ssl)".

The credentials can be provided on the HTTP challenge box (HTTP GET) or in the SOAP Headers (HTTP POST)

- Edit the Portus Apache configuration file
- Enter the following directives (putting the service name in the Location directive):

```
<IfModule mod_xmiddle.c>
  <Location /SERVICENAME>
    AuthType Basic
    AuthName "Username and password required"
    SoaGatewaySecurityLevel pass
    Require valid-user
  </Location>
</IfModule>
```

- Restart your Portus Server.

When `http://host:port/SERVICENAME?WSDL` is requested, a dialog box will appear where the DBMS credentials can be entered.

The DBMS credentials must be provided in the Header section of the the SOAP request

Apache's htpasswd

A password file can be created using Apache's htpasswd program. It is very important that this file is not placed in an area which is accessible to the web via the web server. It is recommended that the password file should be named "htpasswd". We recommend that the file is placed in the following locations

Linux	[SOAG_INSTALL_DIR]/apache2/bin/htpasswd
Windows	[SOAG_INSTALL_DIR]\Apache22\bin\htpasswd.exe

See [Apache 2.2 documentation](#) for more information about htpasswd.

Authenticate with RACF

There are a number of prerequisites to authenticating with RACF, ACF2 or Top Secret:

1. If you wish Portus to check directly with using the SAF interface, the Portus address space must be APF authorized.
2. If you have ADABAS installed and are using the ADABAS SAF Server, Portus can communicate with the SAF Gateway to authorize userids and passwords and to ensure that your ADABAS database is fully protected. In order to do this, you must make the ADABAS WAL Library available in the Portus STEPLIB on z/OS and relink the current SAFASG module as follows. Please ensure that the newly linked SAFASG is higher in the STEPLIB chain than the WAL library you are using.

```
//LINK EXEC PGM=IEWL,
// PARM='MAP,LET,LIST,XREF,NCAL,REUS,RENT'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=VIO,SPACE=(CYL,(1,1))
//SYSLMOD DD DSN=<target load librant>,DISP=SHR
//ADALIB DD DSN=<root>.ADAvrm.MVSLOAD,DISP=SHR
//WALLIB DD DSN=<root>.WALvrm.MVSLOAD,DISP=SHR
//SYSLIN DD *
INCLUDE WALLIB(SAFASG)
INCLUDE ADALIB(ADALNKR)
ENTRY SAFASG
MODE AMODE(31),RMODE(24)
NAME SAFASG(R)
/*
```

Please refer to the Adabas SAF Security documentation for more information and configuration options.

Portus must be configured to run with its security level at the “User” or above. Note that with “User” level, only userids will be checked which can be useful if all requests come from a trusted source but under normal circumstances, you should run with level “Password”.

Once Portus is running at Security Level = Password, then all requests must provide a password.

To pass security credentials from the Portus Control Centre, use the "Set Credentials For Server" option.

If you wish to use HTTP Headers to provide the credentials:

- Edit the HT\$CONF file
- Enter

```
<IfModule mod_xmiddle.c>
  <Location /configurationService>
    AuthType Basic
    AuthName "Username and password required"
    Require valid-user
  </Location>
</IfModule>
```

- Submit the JCL which copies this HT\$CONF to the Portus file system and restart the server.

If you wish to provide the credentials on the SOAP headers, there is no need to modify HT\$CONF

Enabling Web Service Authorization

 **Important:** The AuthUserFile is not required on z/OS.

There are 2 methods to pass the credentials to the Portus server

1. Using the SOAP Headers: Once the WSDL has been imported into your chosen SOAP client, you can add your credentials to the `<soap:Header>` section. No additional changes are required on the server side, and your request is processed using these credentials.
2. Using HTTP Headers: The Apache configuration file must be modified to include a `<Location ... >` directive. Now when a client requests this location, the server will "challenge" for the credentials. Examples for the `<Location ...>` directive are provided below.

Example 1

To request authorization on the configuration web service, perform the following steps, ensuring the `<<filename>>` is filled into the password file appropriate to your system.

- Edit the Portus Apache configuration file
- Enter the following directives:

```
<IfModule mod_xmiddle.c>
  <Location /configurationService>
    AuthType Basic
    AuthName "Username and password required"
    AuthUserFile <<filename>>
    Require valid-user
  </Location>
</IfModule>
```

- Restart your Portus Server.
- Open a web browser, and enter the URL `http://<host>:<port>/configurationService?WSDL` where `<host>` and `<port>` are the host and port (if required) where your Portus server is running.
- You will not be granted access to the WSDL unless you enter the correct credentials.

Example 2

To request authorization on the web service for a resource “adabas_Employees”, perform the following steps, ensuring the `<<filename>>` is filled into the `htpasswd` file appropriate to your system.

- Edit the Portus Apache configuration file
- Enter the following directives:

```
<IfModule mod_xmiddle.c>
  <Location /adabas_Employees>
    AuthType Basic
    AuthName "Username and password required"
    AuthUserFile <<filename>>
    Require valid-user
  </Location>
</IfModule>
```

- Restart your Portus Server.
- Open a web browser, and enter the URL `http://<host>:<port>/adabas_Employees?WSDL` where `<host>` and `<port>` are the host and port (if required) where your Portus server is running.
- You will not be granted access to the WSDL unless you enter the correct credentials.

Example 3

The following example will demonstrate how to access the adabas_Employees web service from PHP. This resource should be set up to only allow access when the client has the correct user name and password. You should change this program to use the host and port that your Portus server is running on.



Important: This program assumes that personnel_id field of the “adabas_Employees” resource has been set up to be the one and only primary key.

```
<?php
ini_set( "soap.wsdl_cache_enabled", 0);

$soapClient = new SoapClient(
    "http://localhost:8080/adabas_QE_Employees?WSDL",
    array('login'="asg", 'password'=>"boston1") );

$adabasEmployeeGetKey = array('personnel_id'=>50005000);

try{
    $results = $soapClient->get($adabasEmployeeGetKey);
}
catch( Exception $e){

    print "An exception occurred!\n";
    print "Code : ";
    print_r( $e->faultcode);

    print "\nString : ";
    print_r( $e->faultstring);

    print "\n ";

    exit;
}
?>
```

If the PHP request works, then the results of the get operation will be printed using PHP print_r function.

Or in the case of an error:

X-Powered-By: PHP/5.1.2 Content-type: text/html

An exception occurred!

Code : HTTP


String : Unauthorized Request

The Apache error_log should have more information about why this request was rejected

Example 4

It is also possible to add a username and password to a SOAP Request.

On z/OS, the username and password will be authenticated against RACF. This allows request to be not only authenticated, but also authorized to run with the required credentials.

 **Important:** It is recommended that Portus's security level should be set to "Password". This can be set either by the global setting in the control centre, or via the Portus Apache SoaGatewaySecurityLevel directive.

 **Important:** On z/OS, the Portus dataset must be authorized.

Consider the following SOAP Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:xmid="http://www.risaris.com/namespaces/xmiddle"
                  xmlns:sec="http://schemas.xmlsoap.org/ws/2002/04/secext">
  <soapenv:Header>
    <sec:Security>
      <UsernameToken>
        <Username>ASG</Username>
        <Password>BOSTON1</Password>
      </UsernameToken>
    </sec:Security>
  </soapenv:Header>
  <soapenv:Body>
    <xmid:adabasEmployeeListElement>
      <personnel_id>400001*</personnel_id>
    </xmid:adabasEmployeeListElement>
  </soapenv:Body>
</soapenv:Envelope>
```

If this SOAP request is sent to z/OS, Portus will attempt to authenticate this user with RACF, providing the username "ASG" and the password "BOSTON1".

10 SSL Certificates

- Setup 46
- Introduction 46
- Step 1: Set up your own CA 46
- Step 2: Portus Server key and certificate 47
- Server Configuration 47

This section describes how to use SSL Certificates with Portus running on Linux.

Setup

To enable the use of SSL Certificates for your Portus, [openssl](#) must be installed.

SSL support is not automatically built into Portus due to export restrictions in certain countries. Please contact your Portus representative to get access to a SSL enabled version.

Introduction

The Apache module `mod_ssl` provides strong cryptography for the SOA Gateway via Secure Sockets Layer (SSL v2/v3) and the Transport Security Layer (TLS v1) protocols with the help of the SSL/TLS implementation library `openssl`. This section will help you to secure your Portus using `mod_ssl`. In order to run a secure server, you need a private key and a certificate for the server. In a commercial environment, it would be advisable to purchase a signed certificate from a well-know Certificate Authority (CA), such as Thawte or Verisign. For the purpose of this section, we will become the CA and generate our certificates using the `openssl` toolkit. Some terms used in this section are outside of the scope of the documentation, and will not be explained in detail. For more information on SSL, and corresponding keys or certificates, see [here](#)

Step 1: Set up your own CA

Firstly we will setup our own CA, and generate a certificate and a key that can be used to sign other certificates.

Generate the key, entering a password when prompted:

```
openssl genrsa -des3 -out myCa.key 2048
```

Generate the X.509 certificate:

```
openssl req -new -x509 -key myCa.key -out myCa.crt
```

Enter the password you added when creating the key (when prompted).

Enter the information would you like to appear on your CA certificate. You should now have your CA key, `myCa.key`, and a CA certificate, `myCa.crt` in the current directory.

Optionally you may view the certificate by typing the command


```
openssl x509 -in myCa.crt -text -noout
```

Step 2: Portus Server key and certificate

This step will create a key and certificate for the Portus server.

Rather than creating a certificate directly, we will create a certificate request, then use the CA key we made in Step 1 to sign the server certificate.

Generate the key, entering a password when prompted:

```
openssl genrsa -des3 -out asg-server.key 1024
```

Generate the server certificate request

```
openssl req -new -key asg-server.key -out asg-server.csr
```

Sign the certificate request with our CA information and generate our server certificate. Note: For this certificate, the “Common Name” should be the hostname of the server this certificate is going to be used on.

```
openssl x509 -req -in asg-server.csr -out asg-server.crt -sha1 -CA myCa.crt -CAkey myCa.key -CAcreateserial -days 3650
```

Optionally, you can view the server certificate you’ve created with the command:

```
openssl x509 -in asg-server.crt -text -noout
```

You should see that the Certificate issuer is your CA Company.

Server Configuration

Before importing the key and certificate into the Portus server, we need to enable SSL support. Choose next section depending on your system.

SSL Configuration

The SSL configuration file should be located in your Portus installation directory. Your Portus representative will provide you with a version of this file to suit your system. The following is an example:

```
<IfDefine SSL>
<IfDefine !NOSSL>
<IfModule mod_ssl.c>

AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-cr1 .cr1

SSLPassPhraseDialog builtin

SSLSessionCache          shmcb:/usr/local/soaGateway/apache2/logs/ssl_scache
SSLSessionCacheTimeout  600

SSLMutex sem
#SSLMutex file:/usr/local/soaGateway/apache2/logs/ssl_mutex

SSLRandomSeed startup builtin
SSLRandomSeed connect builtin

<VirtualHost _default_:443>

    DocumentRoot "/srv/www/htdocs"
    ErrorLog /usr/local/soaGateway/apache2/logs/error_log
    TransferLog /usr/local/soaGateway/apache2/logs/access_log
    ServerName <<hostname>>

    SSLEngine on

    SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

    SSLCertificateFile /usr/local/soaGateway/apache2/certs/asg-server.crt
    SSLCertificateKeyFile /usr/local/soaGateway/apache2/keys/asg-server.key

    SSLCACertificateFile /usr/local/soaGateway/apache2/certs/myCa.crt

    <Files ~ "\.(cgi|shtml|phtml|php3?)$" >
        SSLOptions +StdEnvVars
    </Files>
    <Directory "/srv/www/cgi-bin">
        SSLOptions +StdEnvVars
    </Directory>

        SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown ↵
downgrade-1.0 force-response-1.0

        CustomLog /usr/local/soaGateway/apache2/logs/ssl_request_log  ssl_combined
```

```
</VirtualHost>
</IfModule>
</IfDefine>
</IfDefine>
```

For the `ServerName <<hostname>>` directive ensure `<<hostname>>` is the hostname of your machine. This should match the “Common Name” of the `asg-server.crt` created earlier.

Take the “`asg-server.crt`” and copy it into `/usr/local/soaGateway/apache2/certs/` directory Take the “`asg-server.key`” and copy it into the `/usr/local/soaGateway/apache2/keys` directory.

Stop Apache (`apache2ctl stop`). Start Apache with SSL support (`apache2ctl sslstart`). Enter the pass phrase for the server key.

Open a browser and enter “`https://<hostname>:<port>/configurationService?WSDL`” where host-name and port are location where your Portus is running.

You should be asked do you wish to accept the certificate. Click “Accept”. The WSDL will be downloaded to the browser over a secure connection.

To disable SSL support on this Portus, stop the server (`apache2ctl stop`) and start the server normally (`apache2ctl start`)

Troubleshooting

- Cannot connect to `https://` page
 - Ensure the “`Include ssl.conf`” directive has been added to `httpd.conf`
 - Ensure that you have started apache with the “`sslstart`” parameter
 - Check Apache logs for error (see `/usr/local/soaGateway/apache2/logs/*`)
 - Ensure that you have connectivity to that particular hostname.
- Function not implemented: Cannot create SSLMutex
 - Change the SSLMutex directive to “`file:/usr/local/soaGateway/apache2/logs/ssl_mutex`”

Example

The following is a PHP program to connect to an SSL enabled web service provided by Portus. Note: You must have openssl support in your PHP installation. To check if you do, run the following PHP program.

```
<?php phpinfo(); ?>
```

You should check the “configure command” section. If there is no `--with-openssl` option, then you need to download PHP and build the requirements into it. See instructions [here](#)

This example uses an SSL enabled endpoint (`https://`) the user name and password set up earlier in the documentation.

If this username and password are not required, remove the array type from the `soapClient` constructor. E.g. `$soapClient = new SoapClient(https://localhost:8080/adabas_QE_Employees?WSDL);`

```
<?php
ini_set( "soap.wsdl_cache_enabled", 0);

$soapClient = new SoapClient(
    "https://localhost:8080/adabas_QE_Employees?WSDL",
    array('login'=>"asg", 'password'=>"boston1") );

$adabasEmployeeGetKey = array('personnel_id'=>50005000);

try{
    $results = $soapClient->get($adabasEmployeeGetKey);
}
catch( Exception $e){

    print "An exception occurred!\n";
    print "Code : ";
    print_r( $e->faultcode);

    print "\nString : ";
    print_r( $e->faultstring);

    print "\n ";

    exit;
}

print_r($results);

?>
```

Client verification using SSL

This section outlines how to create and use a SSL client certificate. This certificate must be digitally signed by the CA that the server trusts, and the user must import the certificate into their web service client program. We will use the OpenSSL toolkit to create this client certificate.

Step 1: Generate client key and certificate

Generate the client's key:

```
openssl genrsa -des3 -out asg-client.key 1024
```

Generate the client's certificate request:

```
openssl req -new -key asg-client.key -out asg-client.csr
```

Sign (using "our" CA) and generate the client's certificate. Note: For this certificate, the "Common Name" should be the hostname of the server this certificate is going to be used on.

```
openssl x509 -req -in asg-client.csr -out asg-client.crt -sha1 -CA myCa.crt -CAkey myCa.key -CAcreateserial -days 3650
```

Step 2: Generate the PKCS12 cert

The industry standard in client certificates is the Public Key Cryptography Standard 12 (PKCS12) encoding. These are binary files which again can be generated using the OpenSSL toolkit.

Generate the PKCS12 encoded certificate. The "export password" that is prompted for here is the password that the user needs to know when they import this certificate into the program.

```
openssl pkcs12 -export -in asg-client.crt -inkey asg-client.key -name "Portus Client" -out asg-client.p12
```

You can optionally view the created certificate with the command:

```
openssl pkcs12 -in asg-client.p12 -clcerts -nokeys -info
```

Step 3: Apache Configuration

Apache must be configured to only allow clients who have the correct certificate. For the purposes of this example, we will only all the resource "secure_adabas_employees" to be accessed by a client with the correct certificate.

Perform the following steps:

- Edit the Portus Apache configuration file
- Enter the following directives.

```
<IfModule mod_xmiddle.c>
  <Location /secure_adabas_employees>
    SSLVerifyClient require
    SSLVerifyDepth 1
  </Location>
</IfModule>
```

■ Restart the server

To test this, attempt to access this resource's WSDL. Open a browser and enter the following: `https://<host>:<port>/secure_adabas_employees?WSDL`" where <host> and <port> (if required) are the hostname and port your SOA Gateway is running on. You should be rejected by the server, and see a validation error message in Apache's `error_log`.

Step 4: Import Client Certificate

Firstly we will import the certificate into a browser and access the WSDL.

■ Firefox

- Tools -> Options -> Advanced -> Security -> View Certificates -> Import
- Choose you PKCS12 client certificate and enter the password.

■ Internet Explorer

- Tools -> Internet Options -> Privacy -> Certificates -> Import
- Choose the PKCS12 client certificate and enter the password

Now when you attempt to get the WSDL for `secure_adabas_employees` you should be able to accept the certificate signed by "our" CA company, and then view the WSDL.

If there are any errors in doing this, check Apache's `error_log` for messages. Also ensure that the certificate import has worked, and you are accessing the correct URL. Finally ensure that the <Location> directive in `httpd.conf` is correct. Remember this is case sensitive!

PHP Example

The following PHP example accesses the "secure_adabas_employees" resource, which has been secured above.

PHP will not accept a PKCS certificate. Instead, it requires a file containing both the x509 client key and cert. To create this file, copy `asg-client.crt` to a new file and append the contents of `asg-client.key` to `asg-client.crt`. These files will have been created in 1.5.8. E.g.

```
cat asg-client.crt > asg-newCert.crt
```

```
cat asg-client.key >> asg-newCert.crt
```

Or, on Windows, use Notepad.exe to create asg-newCert.crt..



Important: There is a bug in Apache version 2.0.x which prevents this PHP example from working properly. This bug has been fixed in Apache version 2.2

```
<?
ini_set("soap.wsdl_cache_enabled", "0"); // disabling WSDL cache

$soapClient = new SoapClient(
    "https://lxbre/secure_adabas_employees?WSDL",
    array( 'local_cert'=> "asg-newCert.crt" ) );

$adabasEmployeeGetKey = array('personnel_id'=>50005000);

try{
    $results = $soapClient->get($adabasEmployeeGetKey);
}
catch( Exception $e){

    print "An exception occurred!\n";
    print "Code : ";
    print_r( $e->faultcode);

    print "\nString : ";
    print_r( $e->faultstring);

    print "\n ";

    exit;
}


print_r($results);
?>
```


11 The Portus (legacy) Servers View

▪ Working with the Servers View	56
▪ Refresh the Portus configuration	76
▪ Save current Configuration to server file	76
▪ Import the Portus Configuration from server	58
▪ Export the Portus Server configuration	59
▪ Export Portus WebService definitions	61
▪ Remove a Portus Server entry	80
▪ Create new Services	62
▪ Modify the Portus Server definition entry	66
▪ Modify Global -> Security Options	66
▪ Modify Global -> Transactional Options	67
▪ Modify Global -> Settings	67
▪ Modify tracing / messaging options	68
▪ Add / Modify Drivers	69

The Servers View lists all defined servers along with their status.

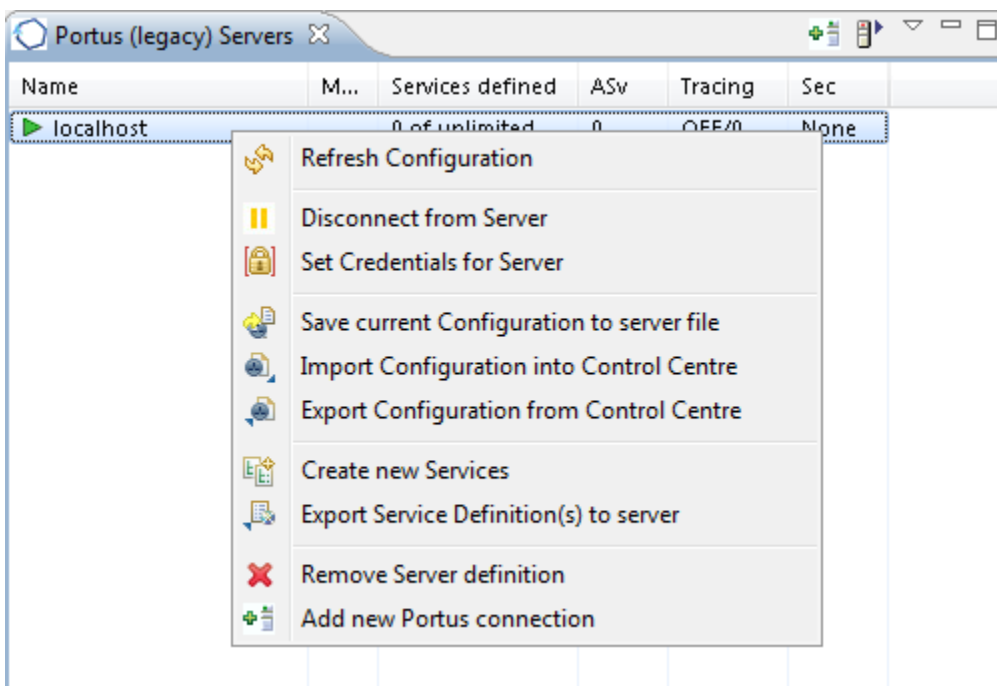
This view is your top level of configuration of one or many Portus servers.

 **Note:** The Servers view is part of the Portus "legacy" perspective. This perspective is superseded by the new "Portus perspective", you are strongly encouraged to familiarise yourself with, and start using the new perspective as early as possible, only the latter will be enhanced further some features are available in the Portus perspective *only*.

Working with the Servers View

There are 2 ways to perform actions on a defined server.

- The first method is to left-click the server, and use the Properties view. In this view you can:
 1. [Get some more detailed information on the server](#)
 2. [Set tracing parameters](#)
 3. [Set security levels](#)
 4. [Set transactional parameters](#)
 5. [Set server settings](#)
 6. [Add or modify the drivers used to access resources](#)
- Using the second method, you may right click on a server entry and bring up the context menu.



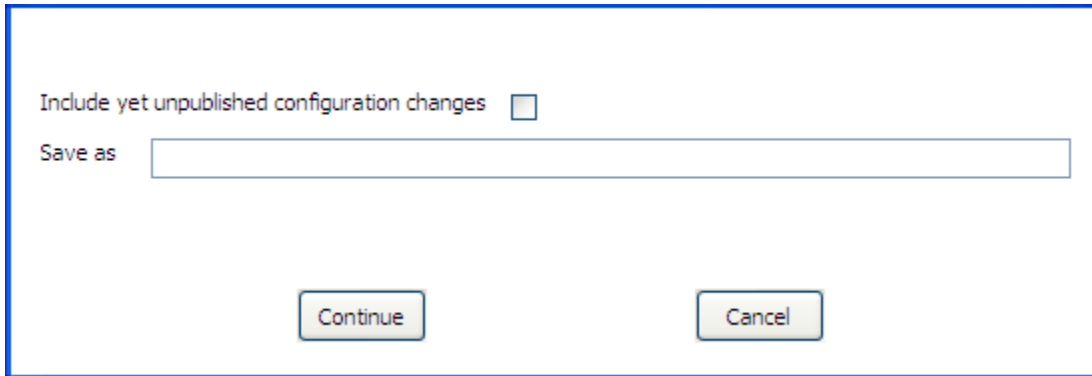
Function	Description
Refresh Configuration	A server's configuration may be modified outside of your own Control Centre session, thus it may be necessary to refresh the local copy.
Disconnect from Server	When many servers are defined, "inactive" servers will still be contacted when you start the Control Centre. This may take quite a while until the TCP/IP timeout is reached, so the connection to knowingly "inactive" servers can be "suspended", which will cause them to be ignored until the connection is reestablished. For disconnected servers the "Disconnect" context menu item will change to "Connect to Server".
Set Credentials for Server	Set the login credentials (HTTP and SOAP User Id and password) for secured Portus servers. Important: To use HTTP credentials you must have a <Location /configurationService> directive enabled.
Save current Configuration to server file	Changes applied to a Portus Server configuration from the Control Centre are volatile, thus need to be "saved" - written to the currently active or a different configuration file, otherwise changes made on-the-fly are lost when the server is restarted.
Import Configuration into Control Centre	Takes a snapshot of the current "online" configuration, for backup purposes (locally or stored in a CVS repository for version control, e.g.), copying or exporting to a different server etc.
Export Configuration from Control Centre	Sends (exports) a local copy of an imported configuration to a Portus server. The server must be restarted for the exported configuration to be activated.
Create new Services	Create Portus Resource related definition elements - DataViews and XML Schemata (XSDs) from native resource definition data, e.g. an Adabas FDT, SQL Tables, etc
Export Service Definition(s) to server	Export Portus Resource related definition elements - DataViews, XML Schemata (XSDs) and Stylesheets (XSLs).
Remove Server definition	Deletes the selected entry from the Servers View.
Add new Portus connection	Define a new Portus server to the Control Centre

Refresh the Portus configuration

When a Portus Server configuration is modified "on-the-fly", for example by modifying, adding or deleting resources, these changes are volatile, they are not preserved over a server restart. Additionally, a server configuration may have been modified by someone else. In either case, your local representation of the Portus server configuration may no longer be accurate, issue a Refresh to synchronize it by selecting the "Refresh Configuration" option on the context menu. A message will be written to the Action Log as well as the status line after the refresh operation is completed.

Save current Configuration to server file

As the Portus Server configuration is volatile - not preserved over a restart - the configuration has to be "saved" (written to disc) in order to have it available for subsequent server executions. Selecting the "Save current Configuration to server file" function from the context menu brings up the following dialog:



The dialog box has a white background and a blue border. At the top, it says "Include yet unpublished configuration changes" followed by an unchecked checkbox. Below that is the label "Save as" followed by a long, empty text input field. At the bottom, there are two buttons: "Continue" on the left and "Cancel" on the right.

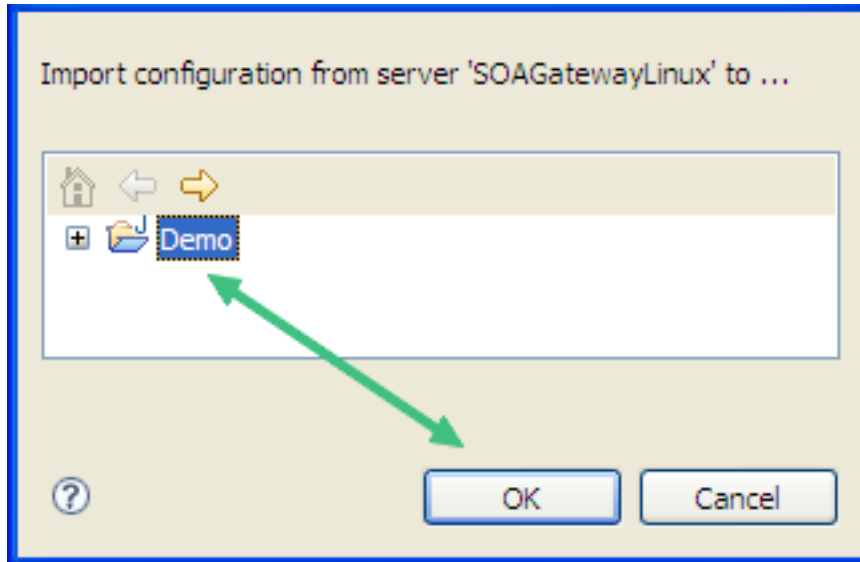
Leave the **Save as** field empty to overwrite the currently active configuration file, or opt to write to a different file (specify the name in the "Save as" field), in which case the respective Portus Server environment variables have to be adjusted to point to the new file. Click "Continue" to save the configuration, the successful operation will be confirmed by a message in the status line.

To include unpublished configuration changes in the saved copy, check the respective box.

Import the Portus Configuration from server

The current "live" Portus Server configuration can be imported to the local file system, e.g. for archiving purposes (locally, saved in a CVS repository etc.), copying / exporting to another Portus server etc.

1. Select the "**Import Configuration from server**" option from the context menu
2. Select a target directory, click **OK**.



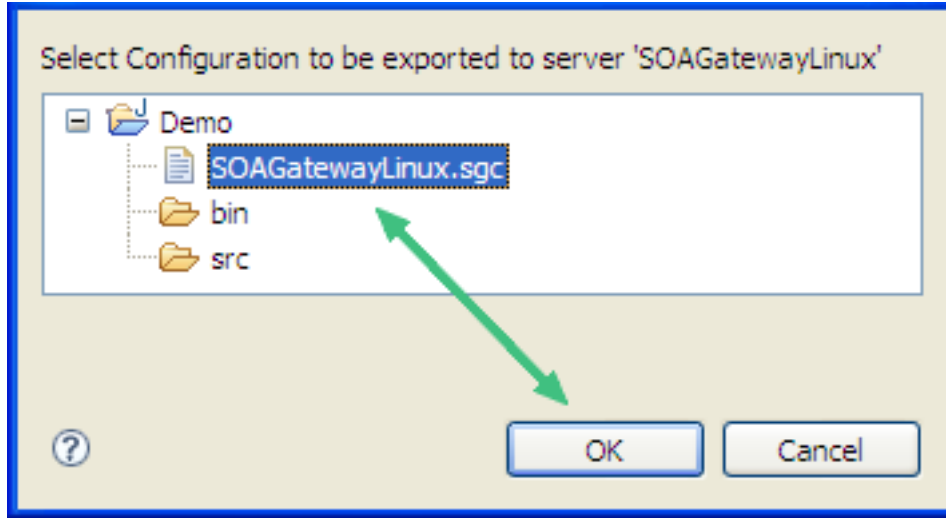
3. Successful import will be confirmed with a message in the status line as well as the Log View

Server 'SOAGatewayLinux' configuration imported to /Demo/SOAGatewayLinux.sgc

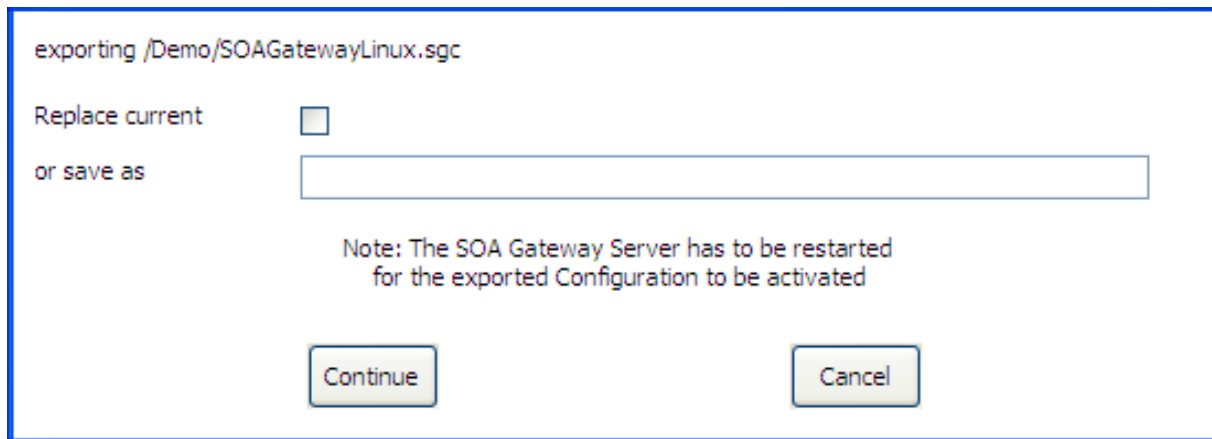
Export the Portus Server configuration

An imported configuration can be exported back to any Portus Server.

1. Select the "Export Configuration to server" option from the context menu
2. Select an exported configuration, click "OK".



3. Specify the export destination, either the currently active, or a different Portus configuration file, click "Continue" to perform the export operation.



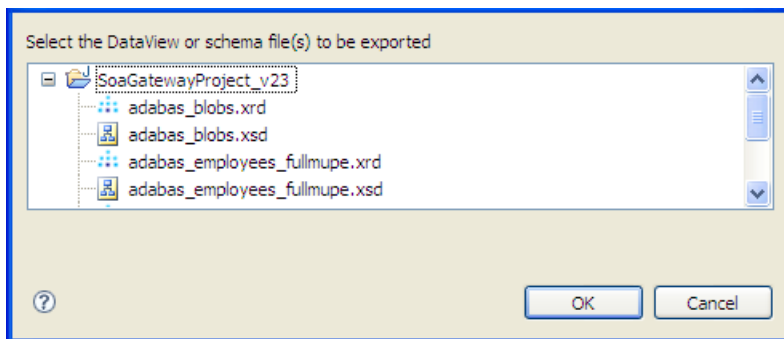
4. Successful export will be confirmed with a message in the status line

Server 'SOAGatewayLinux' configuration exported from /Demo/SOAGatewayLinux.sgc

Export Portus Webservice definitions

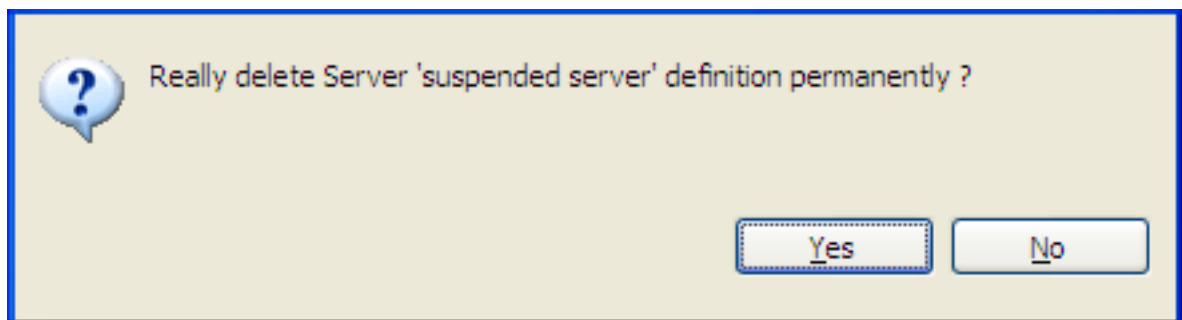
Export Portus Webservice related definition elements - DataViews (XRDs) and/or XML Schemata (XSDs) and/or Stylesheets (XSLs).

1. Select the **export Webservice Definitions** option from the context menu
2. Select a DataView (XRD), Schema (XSD) or Stylesheet (XSL), click **OK**.



Remove a Portus Server entry

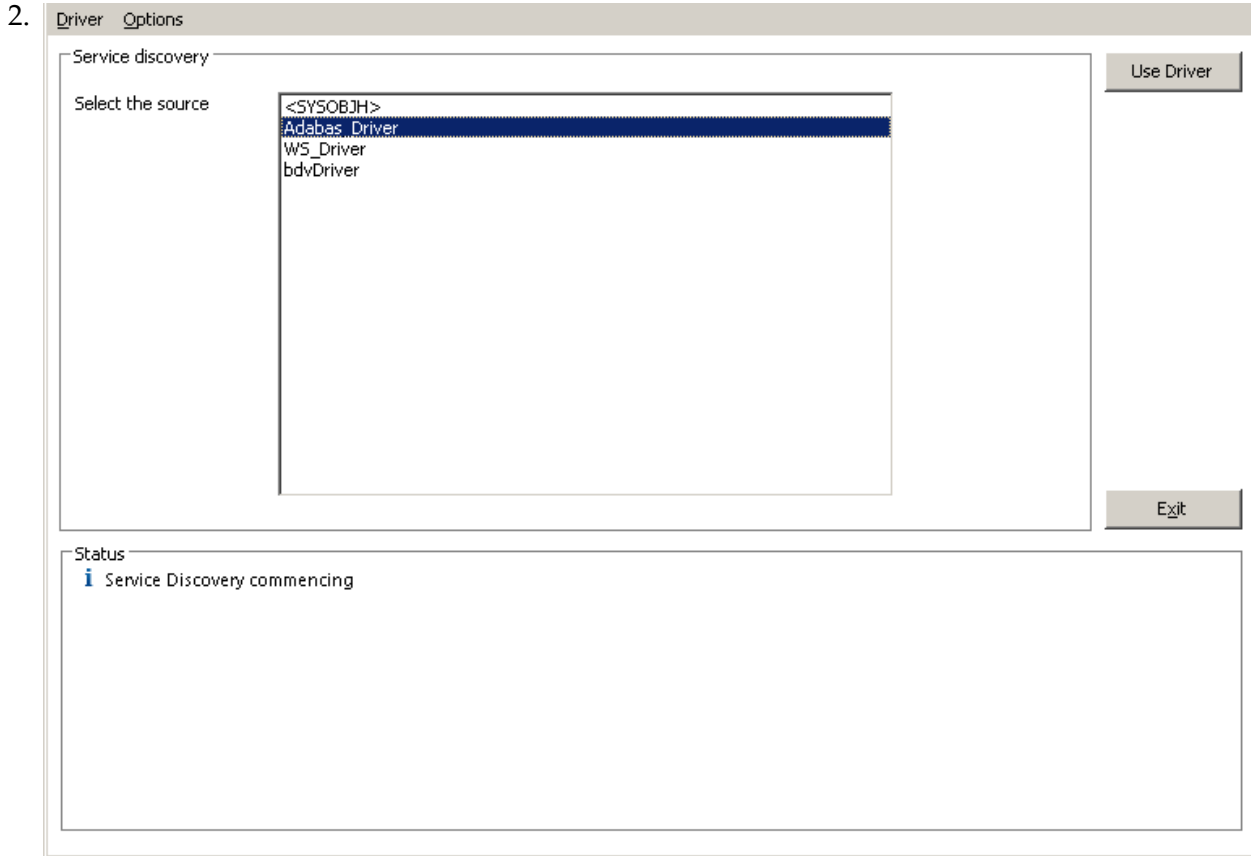
1. Select the "Remove Server definition" option from the context menu
2. Confirm by clicking the "Yes" button.



3. The server entry is now deleted and removed from the Servers View

Create new Services

1. Select the "Creat new Services" option from the context menu



Select the driver to be used for creating/importing the resource(s), then click 'Next'.

Example: Create a new Adabas WebService

For Adabas drivers, the following dialog is presented

Driver Options

Service discovery

Discover

Input & option(s)

Database Id

File range from to

Password

Generate ISN as DataView field

Advanced option(s)

Influence generated Service name(s)

Exit

Status

- i Please specify selection parameters
- i Service Discovery commencing

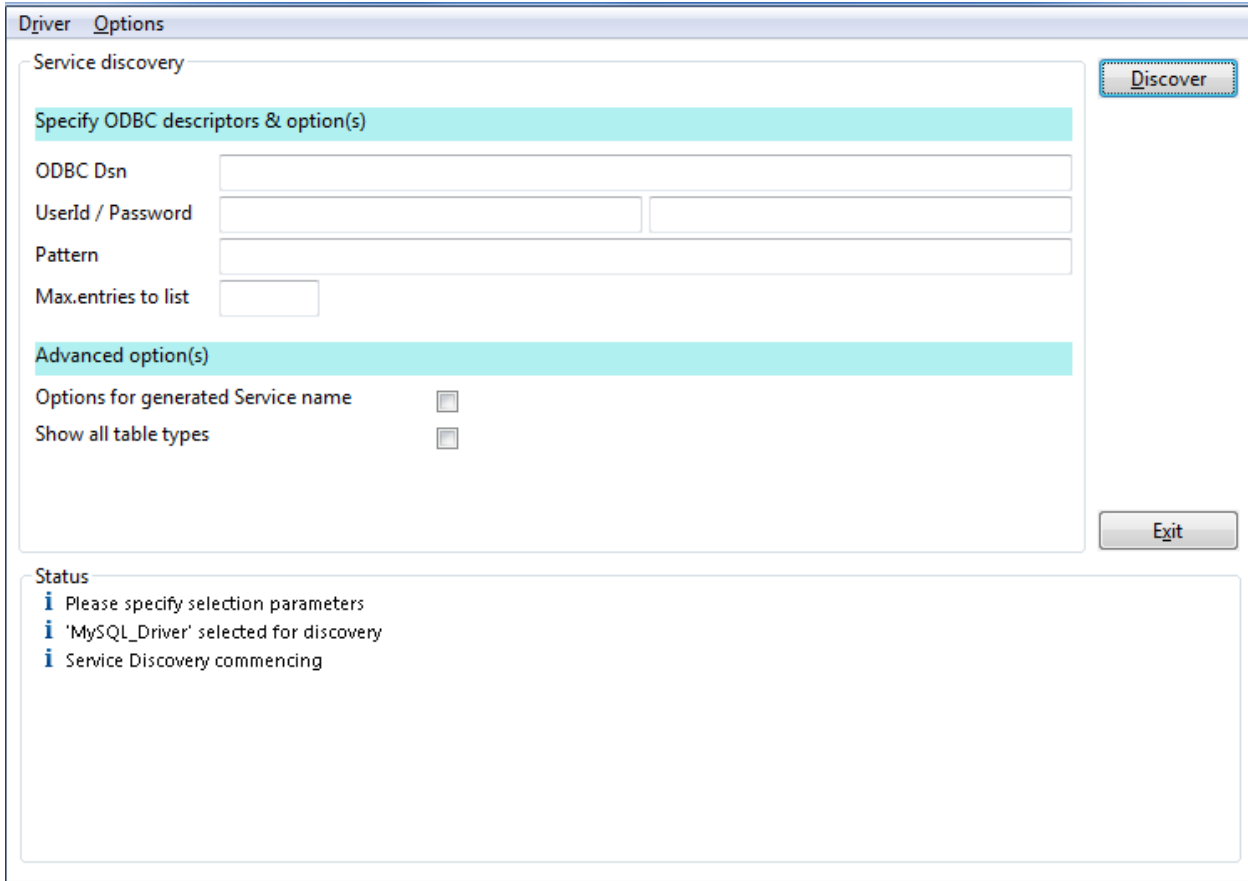
Specify

- The database ID to "discover" web services from
- The file number to begin discovery at
- The file number to end discovery at. Any existing files in this from-to range will then be enabled as Web Services.

For a comprehensive description please refer to the Portus WebService Creation section.

Example: Create a new MySQL WebService

For MySQL, PostgreSQL, and other ODBC-based Drivers, the following dialog will be displayed

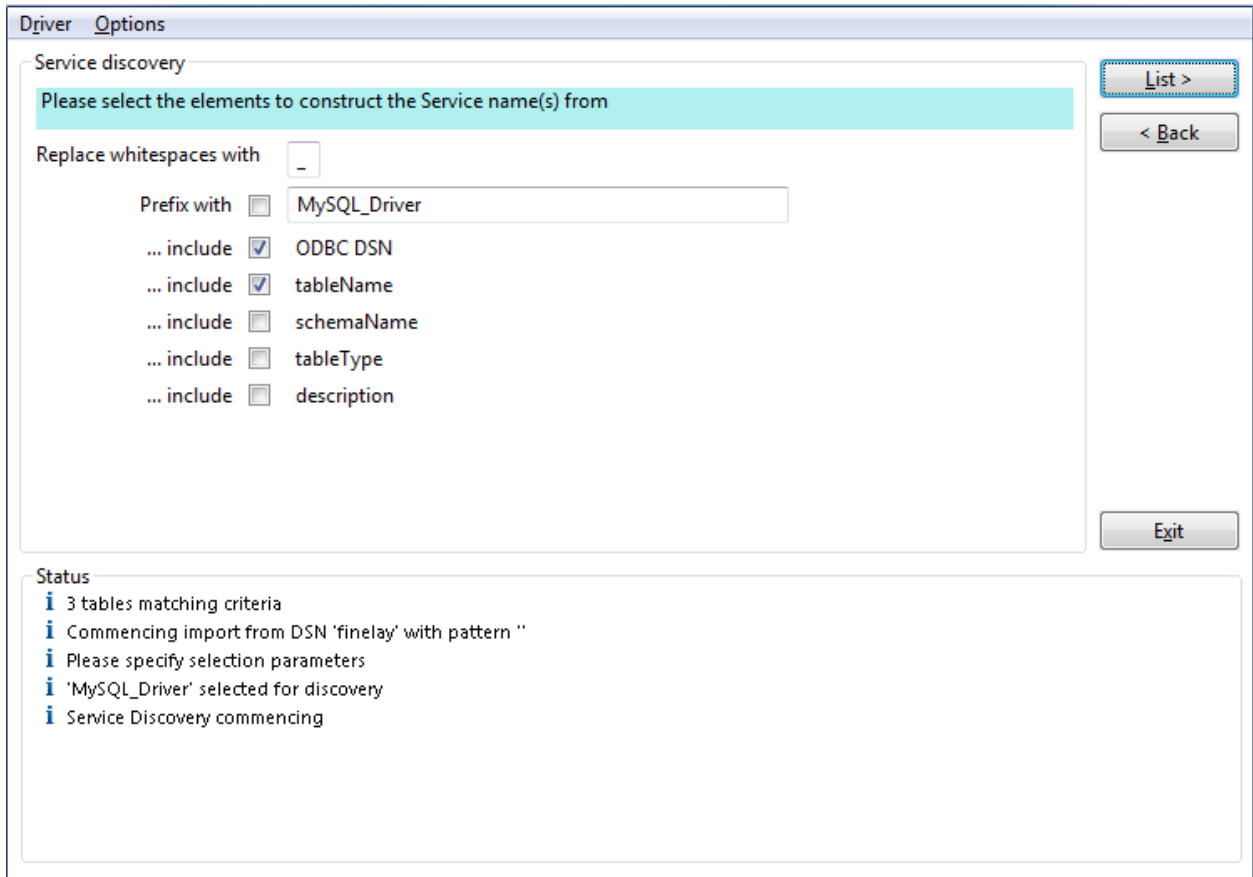


Specify

- The ODBC DSN required
- The credentials to access the database catalog, if required
- A value used to pattern match tables. The percentage (%) will match any sequence of zero or more characters. The underscore (_) will match any single character. In our above example, "C%" will match the tables "City", "Country", but not "Department". You may also leave this value blank to match all tables.
- Max. Tables is a hard-limit on the amount of tables returned.

Click the **Discover** button to start the import/generation process.

When "Options for generated Service name" is checked, the following dialog will appear after having clicked the **Discover** button



Here you can control how the names of automatically created WebServices will be assembled. Specify the replacement character for whitespaces in table names, decide if (and what) prefix is to be attached, select which catalog elements describing a table are to be included in the WebService name.

Click **Continue** to use the current settings as per this dialog, or click **Cancel** to use the default settings.

When "Show all table types" is checked, the following dialog will appear after having clicked the **Discover** button

Click **Continue** to use the current settings as per this dialog, or click **Cancel** to use the default settings.

WebService definition(s), DataViews as well as XSDs, for the selected file(s) will be generated

For a comprehensive description please refer to the Portus WebService Creation section.

Modify the Portus Server definition entry

In this section you can modify the friendly-name, host and port that the Portus server is running on. You may also view some detailed information about the server.

To modify the Portus server entry:


1. In the Portus Server view, left-click the server you wish to modify.
2. Select the **General** tab in the Properties view
3. Modify parameters as required, click **Save**.

The screenshot shows the 'Server Properties' dialog box with the 'General' tab selected. The 'Name' field is set to 'localhost' and has a 'Save' button next to it. The 'Host' field is also set to 'localhost'. The 'Port' field is set to '56421' and the 'Communic. Timeout (sec)' field is set to '20'. There are two checkboxes: 'Autosave' is checked with the label 'Configuration after Service modifications', and 'Disconnect' is unchecked with the label 'from server when exiting the ControlCentre'.

Note: you can also view the Portus Status, and license information here too. This information is not modifiable.

Modify Global -> Security Options

In this section located in the **Global** properties you can set the Security Level that Portus runs at.

 **Important:** We recommend you contact your Portus support team before modifying these parameters.

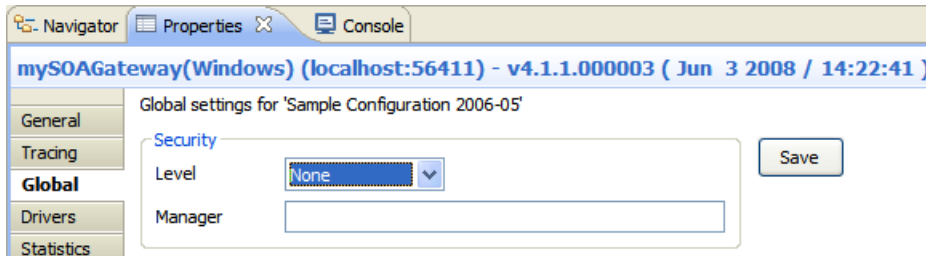
To open the Security view, left-click the server you wish to modify. In the Properties view, select the **Global** tab. Security options are in the **Security** group.

There are 4 security levels that Portus can run at

1. None: No Portus security.
2. Userid: All requests must have a user ID.
3. Password: All requests must have a user ID and password.

4. SSLCert: All requests must have a SSL Certificate.

Under normal circumstances the user will be authenticated by sending the credentials to the underlying resource (for example the ODBC database). It is also possible to provide a 3rd party DLL which can authenticate the credentials. This DLL name should be added in the `Manager` text box.



Modify Global -> Transactional Options

In this section you can set the Transaction Manager that Portus uses to handle data integrity.



Important: We recommend you contact your Portus support team before modifying these parameters.

To open the Transaction view, left-click the server you wish to modify. In the Properties view, select the `Global` tab. Transaction options are in the `Transaction` group.

The screenshot shows the 'Transaction' group in the Properties view. The 'Mode' dropdown menu is set to 'internal'. The 'Manager' and 'Parameters' text boxes are empty.

Modify Global -> Settings

In this section you can set threshold values for Portus global variables.

Left-click the server you wish to modify. In the Properties view, select the `Global` tab. Server settings are in the `Settings` group.


Settings	
Context Time-to-live	<input type="text" value="300"/>
Configuration write timer	<input type="text" value="0"/>
SELECT record limit	<input type="text" value="0"/>
Trim repeating elements	<input type="checkbox"/>

The following can be set here

- Context Time-to-live: specifies the time (in seconds) after which orphaned contexts (for example due to timeouts or broken connections) will be considered "dead" and cleaned up.
- Configuration write timer: The in-core copy of the Portus configuration will be automatically saved to file after the specified interval (in seconds) elapsed. A value of 0 means the current configuration will never be saved back to the configuration file.
- SELECT record limit: Limit the amount of records returned from a "select" or "select next" call.
- Trim repeating elements. This option only applies to Natural-based web services. Repeating array elements will be trimmed based on the first empty field found. For example, if the service returns . . .
`<myArray>ONE</myArray><myArray>TWO</myArray><myArray/><myArray/><myArray/> . . . ,`
when this option is turned on the output will be trimmed to . . .
`.<myArray>ONE</myArray><myArray>TWO</myArray>`
- Trim repeating elements: This option only applies to Natural-based web services. Repeating array elements will be trimmed based on the first empty field found. For example, if the service returns . . .
`<myArray>ONE</myArray><myArray>TWO</myArray><myArray/><myArray/><myArray/> . . . ,`
when this option is turned on the output will be trimmed to . . .
`.<myArray>ONE</myArray><myArray>TWO</myArray>`

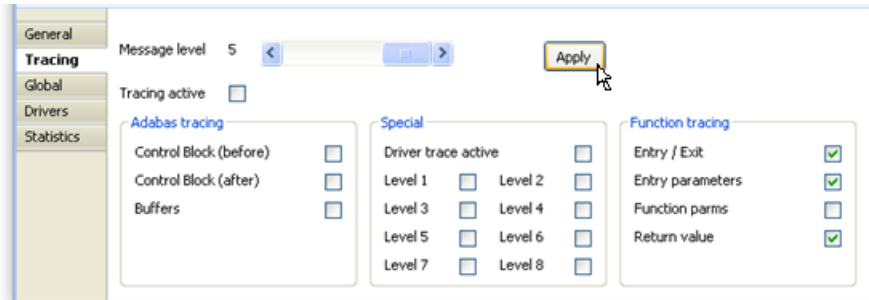
Modify tracing / messaging options

The message level set for a Portus Server determines the amount and detail of information written to the server log.

 **Important:** Trace options are debugging instruments which should not be modified unless instructed to do so by support personnel, continuous tracing will adversely effect the performance of the Portus Server.

To change either the message level or trace options:

1. Select the "Tracing" tab from the Properties view
2. Modify the message level or tracing option(s), as appropriate, click "Apply" to send the changes to the Portus Server.



Add / Modify Drivers

This section describes how you add/remove/modify the drivers that Portus uses to talk to the underlying resource whether that be Adabas, Natural, MySQL, PostgreSQL, etc

To open the Drivers view, left-click the server you wish to modify. In the Properties view, select the **Drivers** tab.

From here you can add, remove and modify drivers, or view detailed information about a particular driver.

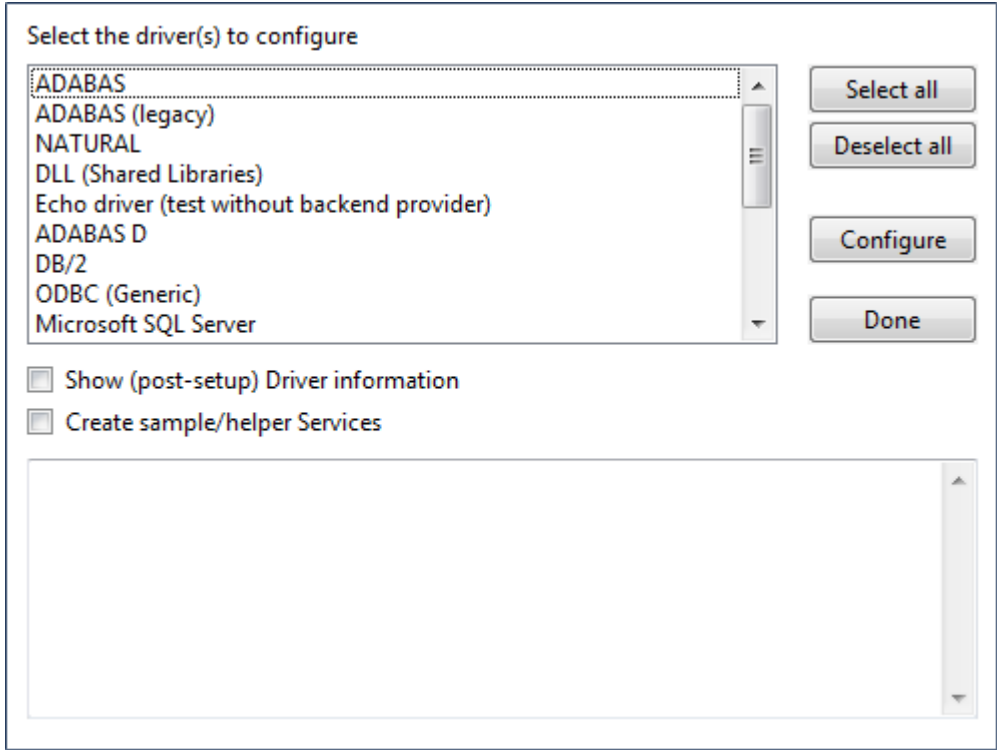
Driver Name	Type	Executable	SBCS Codepage	MBCS Codepage
Adabas_Driver	adabas	adabas81Driver.so	ascii	utf16
bdvDriver	bdvDriver	bdvDriver.so	latin1	utf16
DLL_Driver	sharedlib	dllDriver.so	ascii	utf16
MS_SQLServer_Driver	mssqlserver	MSSQLServerDriver.so	latin1	utf16
MySQL_Driver	mysql	MySQLDriver.so	latin1	utf16
Natural_Driver	natural	naturalDriver.so	ascii	utf16
WS_Driver	WSDriver	wsDriver.so	latin1	utf16

You may left-click a driver entry and select **Details** to display more information about this driver.

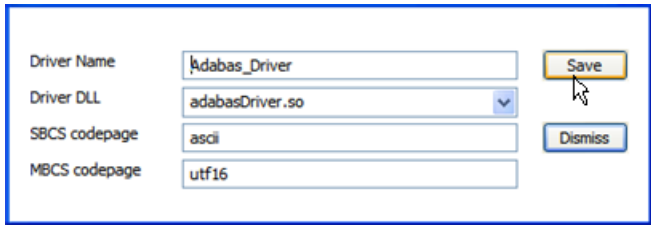
To modify an existing driver, left-click the driver, and select **Modify**

To add a new driver, click the **Add...** button.

The Driver Definition Wizard presents a list of licensed drivers, select one to all and press the **Configure** button.



The Driver Definition dialog will come up with preset values. Modify as appropriate.



Press the **Save** button to define the driver.

If the new driver does not appear, check the error log.

If the driver requires any options, you will be prompted to enter these after you click **Save** once, specify them as appropriate, click **Save** again to actually define the driver.

When "Show (post-setup) Driver Information" is selected, additional information may be displayed after the driver add operation has completed:

The following table illustrates information about all the drivers Portus supports on Linux and Windows systems

Suggested Driver Name	Library Name	Suggested SBCS	Suggested MBCS	Notes / Other Options	More Info
Adabas_Driver	adabasDriver.so	ascii	utf16	ListMaxRecords is optional	
Natural_Driver	naturalDriver.so	ascii	utf16		here
AdabasD_Driver	AdabasdDriver.so	latin1	utf16		
DB2_Driver	DB2Driver.so	latin1	utf16		
DLL_Driver	dllDriver.so	ascii	utf16		
Echo_Driver	echoDriver.so	ascii	utf16		
Generic_ODBC_Driver	odbcDriver.so	latin1	utf16		
Sybase_Driver	SybaseDriver.so	latin1	utf16		
MS_SQLServer_Driver	MSSQLServerDriver.so	latin1	utf16		
MySQL_Driver	MySQLDriver.so	latin1	utf16		
Oracle_Driver	OracleDriver.so	latin1	utf16		
PostgreSQL_Driver	PostgreSQLDriver.so	latin1	utf16		
Stored_Procedures_Driver	storedProcsDriver.so	latin1	utf16		

And the following outlines the driver information for Portus running on mainframe (z/OS or z/VSE) systems

Suggested Name	Library Name	Suggested SBCS	Suggested MBCS	Notes / Other Options / Default values	More Info
Adabas_Driver	ADADRVR	CP1141	utf16	ListMaxRecords is optional	
Natural_Driver	NATDRVR	CP1141	utf16	Natural Batch Pgm : NATBATCH Init Params : ETID=OFF Pre-Init Sessions : 2 Max. Sessions: 8 Natural Library Name : SYSSOA Natural Steplibs: SYSSOAEX,SYSEXT Natural Security: No Important: Natural libraries are required in the Portus server STEPLIB, the Natural Batch Nucleus used MUST NOT be LE-enabled (specify LE370=NO).	here
CICS_Driver	CICSDRVR	CP1141	utf16	CICS APPLID: specify target CICS system application id	

Suggested Name	Library Name	Suggested SBCS	Suggested MBCS	Notes / Other Options / Default values	More Info
				CICS EXCI library is required in the Portus steplib.	
VSAM_Driver	VSAMDRVR	CP1141	utf16		
LE_Driver	LEDRVR	CP1141	utf16	Runtime Options: TRAP(OFF,NOSPIE),RPTOPTS(ON),RPTSTG(ON) envMax: 50 preEnvInit: 5	
DLL_Driver	DLLDRVR	CP1141	utf16		
DB2_Driver	DB2DRVR	CP1141	utf16		

To remove an existing driver, select the driver, and click **Remove**

12

Advanced Topics

- How to install a server on a remote Windows 74
- Running multiple instances of Portus 74
- IPv6 Considerations 76
- Reducing memory footprint using streaming 77

These tasks would typically be carried out by an experienced Portus user.

How to install a server on a remote Windows

Typically when you install Portus on Windows, the Control Centre and the Portus server are on the same machine. If you wish to install Portus on a remote Windows here are the steps involved.

1. Transfer win32Build.msi to the target machine.

The msi can be located at `../eclipse/features/com.SOAGateway.feature.install.x86.windows_4.2.1.NN`.

where `../eclipse` is the folder from which your Control Centre is launched.

2. Transfer the Portus license to the target machine.
3. Logon to the target machine and run:

```
msiexec /I c:\temp\win32Build.msi /L*v c:\temp\installLog.txt SERVERPORT=56000 LICENSES-RC=c:\temp\LIC.xml
```

Assuming that:

- The msi and your license are in `c:\temp\`
- You want the server to run on port 56000
- Your license is named `LIC.xml`
- The server prerequisites have already been installed. Check here for these.

Running multiple instances of Portus

This section outlines the steps required to run multiple instances of Portus, either of the same, or different versions, on a single operating system image.

- [Multiple instances of Portus on Linux](#)
- [Multiple instances of Portus on z/OS](#)

Multiple instances of Portus on z/OS

The simplest way for setting up multiple Portus servers on z/OS is to execute the Deployment Wizard once for every server, specifying distinct HLQs (High Level Qualifiers) for the datasets, TCP/IP port numbers etc., then execute the setup jobs.

This will also create multiple copies of the Portus load library, which is not required, so the start job / procedure for the various instances of the server can be modified to all use the same loadlib in its STEPLIB.

Multiple instances of Portus on Linux

This step-by-step guide assumes good knowledge of the Apache webserver on a Linux platform.

The following steps outline how to copy the existing Portus installation, and then bring up both the existing and new installations using the same Apache webserver.

- If running, stop Portus. See here on how to do this.
- Make a duplicate of the current installations files and directories. For example, if installed in `/home/bre/soaGateway`, then

```
mkdir /home/bre/soaGatewayNew
```

```
cp -R /home/bre/soaGateway/* /home/bre/portusNew
```

- In the new installation, change the Apache Listen directive, specifying a yet unused port number. This directive can be found in the `httpd.conf` file in the `apache2/conf` directory.
- Again, edit the new `httpd.conf` file and change the Include directive on the last line to pick up the new installation.
- Edit new `adabas_soa_gw.conf` file in the new installation. Modify required directives to pick up the new files and directories in the new installation.
- Add a fully qualified PidFile directive to the new `httpd.conf` (if one is not already present). This pid file should point into the new installation.

For example

```
<IfModule !mpm_netware.c>
PidFile /home/bre/portusNew/apache2/logs/httpd.pid
</IfModule>
```

- Modify the new `xmiddleEnv.sh` file , updating its contents to use the new installation.
- Edit the new `envvars` file, updating its contents to use the new installation.
- In the current installation (*not* the one you have just created) copy the `apachectl` control script. For example

```
cp /home/bre/portus/apache2/bin/apachectl ↵  
/home/bre/portus/apache2/bin/apachectl_new
```

Modify the new file to pick up the new *httpd.conf* and new *envvars* file.



Note: You do not need to change the path to the httpd binary.



Note: Therefore you may remove the new Apache libraries and binaries. For example, these are located under */home/bre/portus/apache2/lib* and */home/bre/portus/apache2/bin/*. Do not remove the *apache2/bin/envvars* file.

- Use the 2 apachectl scripts to start Portus
- Verify that the Portus start-up messages (found in the Apache logs) refer to the new Portus configuration file.

IPv6 Considerations

IPv6 (Internet Protocol version 6) is the latest revision of the Internet Protocol (IP), the primary communications protocol upon which the entire Internet is built. It is intended to replace the older IPv4, which is still employed for the vast majority of Internet traffic as of 2012. IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 running out of addresses.

IPv6 Portus is supported on all non-Windows platforms. The underlying hosting infrastructure of Portus is provided by an Apache web server, therefore IPv6 configuration changes take place in the Apache configuration.

Further information can be found [here](#)

To make Portus listen on the IPv6 interface, provide the IPv6 address of the host machine on the Listen directive in the Apache configuration file. This file is `[INSTALL_DIR]/apache2/conf/httpd.conf` or `[INSTALL_DS].CONF(HT$CONF)` depending on your system.

E.g.

Listen on port 56001 on this machine : `Listen [2607:f0d0:1002:11::4]:56001`

Listen on port 56001 for ONLY local IPv6 connections: `Listen [::1]:56001`

More than one Listen directive can be applied, but the ports must be unique

To connect to this server from the Portus Control Centre, add a new server connection with the IPv6 address surrounded by [and] and the port number of your server

ServerName	<input type="text" value="My IPv6 Server"/>
Host / IP	<input type="text" value="[2607:f0d0:1002:11::4]"/>
Port	<input type="text" value="56001"/>
Timeout (sec)	<input type="text" value="10"/>
Autodisconnect	<input type="checkbox"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Reducing memory footprint using streaming

When Portus handles a request it builds XML structures internally to hold the response payload. This data can become very large, and on systems where memory is limited, unexpected results can occur. To resolve this, Portus has the ability to stream back data as it is retrieved from the database. The Portus streaming uses HTTP chunked transfer encoding as a transfer mechanism. This is only available in HTTP clients that understand HTTP 1.1 or above. When data is streamed back to the client, Portus does not have to build large internal structures to handle the payload, therefore the memory footprint is considerably less.

When streaming is enabled, it only applies when the client issues a LIST request, with all keys wildcarded. E.g. a REST request with LIST&ID=*


 **Important:** Portus may use extra CPU resources when streaming is enabled.

Streaming can be enabled by adding the following line in the Apache configuration

```
SoaGatewayStreaming on
```

And stopping and starting the server

This directive can also be applied to specific services using the <Location /SERVICENAME > directive

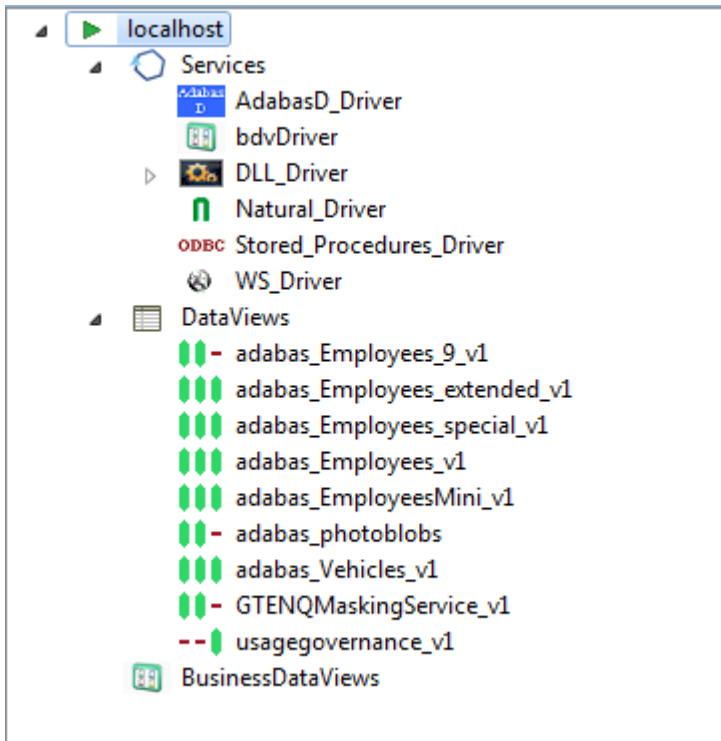
 **Important:** Not all clients understand HTTP 1.1 or indeed support the chunked transfer encoding.

13 Working with the Portus View

- Operations at the Server level 80
- Operations at the Driver level 86
- Operations at the Service level 87
- Operations at the DataView / BusinessDataView levels 87

The Portus View lists, and allows operation on

- **Defined servers** along with their respective status
- **Drivers** defined on each server
- **Services** defined on each driver
- **DataViews and BusinessDataViews**



Operations at the Server level

This level can be expanded to display all the Drivers defined on this Portus Server.

There are 2 ways to perform actions on a defined server.

- Right-click a server entry and bring up the context menu.

The context menu looks like this for a server which you are not connected to

1. *Connect to Server*: Before administrative task can be carried out on a server from the Control Centre a connection needs to be established to it.
2. *Add new Portus connection*: Define a new Portus server to the Control Centre
3. *Remove Connection definition*: Deletes the selected entry from the Servers View.

4. *Hide this Server* Hides the selected entry from the Servers View. Can be unhidden if required.

The context menu changes to the following for a server which you are connected to

1. *Refresh Configuration*: When a Portus Server configuration is modified "on-the-fly", for example by modifying, adding or deleting resources, these changes are volatile, they are not preserved over a server restart. Additionally, a server configuration may have been modified by someone else. In either case, your local representation of the Portus server configuration may no longer be accurate, issue a Refresh to synchronize it by selecting the "Refresh Configuration" option on the context menu. A message will be written to the Action Log as well as the status line after the refresh operation is completed.
2. *Disconnect from Server*: When many servers are defined, "inactive" servers will still be contacted when you start the Control Centre. This may take quite a while until the TCP/IP timeout is reached, so the connection to knowingly "inactive" servers can be "suspended", which will cause them to be ignored until the connection is reestablished. For disconnected servers the "Disconnect" context menu item will change to "Connect to Server".
3. *Set Credentials for Server*: Set the login credentials (HTTP and SOAP User Id and password) for secured Portus servers.



Important: To use HTTP credentials you must have a <Location /configurationService> directive enabled.

4. *Add driver to Server*:

5. *Add new Portus connection*: Define a new Portus server to the Control Centre
6. *Remove Server definition*: Deletes the selected entry from the Servers View.
7. *Hide this Server* Hides the selected entry from the Servers View. Can be unhidden if required.

- Left-click the server, and use the Properties view.

Here you can

1. **View server status, modify parameters**
2. **Set tracing and message level parameters**
3. **Change global settings**
4. **Add or modify driver(s)** used to access resources
5. Modify Governance Settings
6. Modify Statistical Settings
7. Modify Messaging Settings

The Server entry can be expanded to show the sub-levels - Services, DataViews and Business-DataViews.

Server status display / parameter modification

In this section you can modify the friendly-name, host and port that the Portus server is running on. You may also view some detailed information about the server.


To modify the Portus server entry:

1. In the Portus Server view, left-click the server you wish to modify.
2. Select the **General** tab in the Properties view
3. Modify parameters as required, click **Save**.

Note: you can also view the Portus Status, and license information here too. This information is not modifiable.

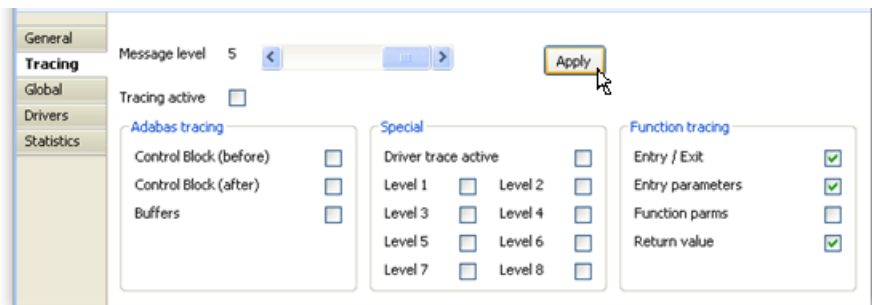
Modify tracing and message level options

The message level set for a Portus Server determines the amount and detail of information written to the server log.

 **Important:** Trace options are debugging instruments which should not be modified unless instructed to do so by support personnel, continuous tracing will adversely effect the performance of the Portus Server.

To change either the message level or trace options:

1. Select the "Tracing" tab from the Properties view
2. Modify the message level or tracing option(s), as appropriate, click "Apply" to send the changes to the Portus Server.



Modify global settings

Global settings for the entire Portus Server can be changed here.



Important: We recommend you contact your Portus support team before modifying these parameters.



Security Settings

There are 4 security levels that Portus can run at

1. None: No Portus security.
2. Userid: All requests must have a user ID.
3. Password: All requests must have a user ID and password.
4. SSLCert: All requests must have a SSL Certificate.

Under normal circumstances the user will be authenticated by sending the credentials to the underlying resource (for example the ODBC database). It is also possible to provide a 3rd party DLL which can authenticate the credentials. This DLL name should be added in the Manager text box.

Transactional Options

You can set the Transaction Manager that Portus uses to handle data integrity.

Server Settings

You can set specific options for the server

1. Context Time-to-live: specifies the time (in seconds) after which orphaned contexts (for example due to timeouts or broken connections) will be considered "dead" and cleaned up.
2. Configuration write timer: The in-core copy of the Portus configuration will be automatically saved to file after the specified interval (in seconds) elapsed. A value of 0 means the current configuration will never be saved back to the configuration file.
3. Select record limit: Limit the amount of records returned from a "select" or "select next" call.

4. Trim repeating elements. This option only applies to Natural-based web services. Repeating array elements will be trimmed based on the first empty field found. For example, if the service returns
`<myArray>ONE</myArray><myArray>TWO</myArray><myArray/><myArray/><myArray/>`,
 when this option is turned on the output will be trimmed to
`.<myArray>ONE</myArray><myArray>TWO</myArray>`
5. Max internal length: An upper limit on the size of a field in the XRD.
6. Messaging sys retry count: If using Portus to connect to a messaging system such as EntireX or MQ, it may be that a failure occurs attempting to connect due to a database not being started. A retry count can be set to allow the database to be started and the connection to succeed.
7. Messaging sys retry time: This sets the time to wait until another attempt is made to connect to whatever system has initially caused the messaging system to throw an error on connection.

Add / modify drivers

This section describes how you add/remove/modify the drivers that Portus uses to talk to the underlying resource whether that be Adabas, Natural, MySQL, PostgreSQL, etc

To open the Drivers view, left-click the server you wish to modify. In the Properties view, select the **Drivers** tab.

From here you can add, remove and modify drivers, or view detailed information about a particular driver.

Driver Name	Type	Executable	SBCS Codepage	MBCS Codepage
Adabas_Driver	adabas	adabas81Driver.so	ascii	utf16
bdvDriver	bdvDriver	bdvDriver.so	latin1	utf16
DLL_Driver	sharedlib	dllDriver.so	ascii	utf16
MS_SQLServer_Driver	mssqlserver	MSSQLServerDriver.so	latin1	utf16
MySQL_Driver	mysql	MySQLDriver.so	latin1	utf16
Natural_Driver	natural	naturalDriver.so	ascii	utf16
WS_Driver	WSDriver	wsDriver.so	latin1	utf16

You may left-click a driver entry and select **Details** to display more information about this driver.

To modify an existing driver, left-click the driver, and select **Modify**

To add a new driver, click the **Add...** button.

The Driver Definition Wizard presents a list of licensed drivers, select one to all and press the **Configure** button.

Preset values are presented , modify as appropriate.

Press the **Save** button to define the driver.

If the new driver does not appear, check the error log.

If the driver requires any options, you will be prompted to enter these after you click **Save** once, specify them as appropriate, click **Save** again to actually define the driver.

When "Show (post-setup) Driver Information" is selected, additional information may be displayed after the driver add operation has completed:

The following table illustrates information about all the drivers Portus supports on Linux and Windows systems

Suggested Driver Name	Library Name	Suggested SBCS	Suggested MBCS	Notes / Other Options	More Info
Adabas_Driver	adabasDriver.so	ascii	utf16	ListMaxRecords is optional	
Natural_Driver	naturalDriver.so	ascii	utf16		here
AdabasD_Driver	AdabasdDriver.so	latin1	utf16		
DB2_Driver	DB2Driver.so	latin1	utf16		
DLL_Driver	dllDriver.so	ascii	utf16		
Echo_Driver	echoDriver.so	ascii	utf16		
Generic_ODBC_Driver	odbcDriver.so	latin1	utf16		
Sybase_Driver	SybaseDriver.so	latin1	utf16		
MS_SQLServer_Driver	MSSQLServerDriver.so	latin1	utf16		
MySQL_Driver	MySQLDriver.so	latin1	utf16		
Oracle_Driver	OracleDriver.so	latin1	utf16		
PostgreSQL_Driver	PostgreSQLDriver.so	latin1	utf16		
Stored_Procedures_Driver	storedProcsDriver.so	latin1	utf16		

And the following outlines the driver information for Portus running on mainframe (z/OS or z/VSE) systems

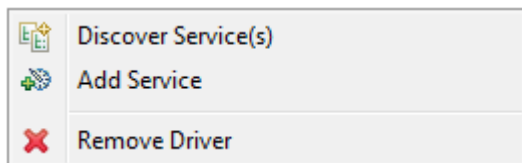
Suggested Name	Library Name	Suggested SBCS	Suggested MBCS	Notes / Other Options / Default values	More Info
Adabas_Driver	ADADRVR	CP1141	utf16	ListMaxRecords is optional	
Natural_Driver	NATDRVR	CP1141	utf16	Natural Batch Pgm : NATBATCH Init Params : ETID=OFF Pre-Init Sessions : 2	here

Suggested Name	Library Name	Suggested SBCS	Suggested MBCS	Notes / Other Options / Default values	More Info
				Max. Sessions: 8 Natural Library Name : SYSSOA Natural Steplibs: SYSSOAEX,SYSEXT Natural Security: No Important: Natural libraries are required in the Portus server STEPLIB, the Natural Batch Nucleus used MUST NOT be LE-enabled (specify LE370=NO).	
CICS_Driver	CICSDRVR	CP1141	utf16	CICS APPLID: specify target CICS system application id CICS EXCI library is required in the Portus steplib.	
VSAM_Driver	VSAMDRVR	CP1141	utf16		
LE_Driver	LEDRVR	CP1141	utf16	Runtime Options: TRAP(OFF,NOSPIE),RPTOPTS(ON),RPTSTG(ON) envMax: 50 preEnvInit: 5	
DLL_Driver	DLLDRVR	CP1141	utf16		
DB2_Driver	DB2DRVR	CP1141	utf16		

To remove an existing driver, select the driver, and click **Remove**

Operations at the Driver level

The following context functions are available at the driver level



1. Discover Service(s) - This starts the Discovery Wizard which can be used to create web services.
2. **Add Service**
3. Remove Driver, which is only possible when no service exists for the driver

Add a Service

Normally Services are created using the Discovery Wizard, but it is also possible to define these manually.

Right-click the driver you wish to use to create the server with, and select **Add Service**

1. Enter the name of the Service. This will have an impact on the URL used to access the WSDL, and the endpoint the service uses.
2. The Control Centre will automatically create an "empty" DataView or BusinessDataView for you, based on the type of the driver, and will open the editor on the newly created view.
3. A tutorial on creating a DataView from scratch is available [here](#)
4. Once the file is saved, the (Business)DataView will be automatically exported back to the server and a XSD be generated., if the respective option is enabled
5. Select the new service, and edit the Service Properties to suit your back-end resource.

Operations at the Service level

Further information on service related functions and topics can be found in the Service Configuration section

Operations at the DataView / BusinessDataView levels

By right-clicking on one of the items in the list, you can

1. Import them from the server into your local workspace. Select *Import ...*
2. Delete them from the server.



Notes:

- a. This operation cannot be undone.
- b. Deleting a (Business)DataView referenced by a service will render the referencing service unusable.

14 Drag-and-drop in the Portus View

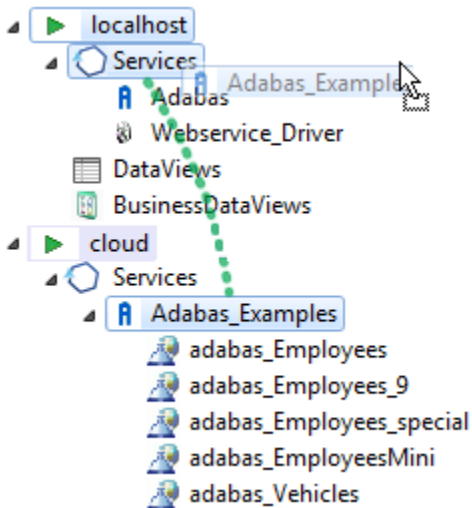
- Driver definition as a drag source 90
- Service as a drag source 90
- DataView and BusinessDataView as a drag source 92

The Portus View incorporates extensive drag-and-drop support to copy between servers, and even to rearrange elements within a server environment.

The following operations are possible

Driver definition as a drag source

Dragging a driver to the *Services* element of another server



will copy the driver definition as well as all services defined using this driver, along with all definition files ((Business)DataViews, XSDs, XSLs)

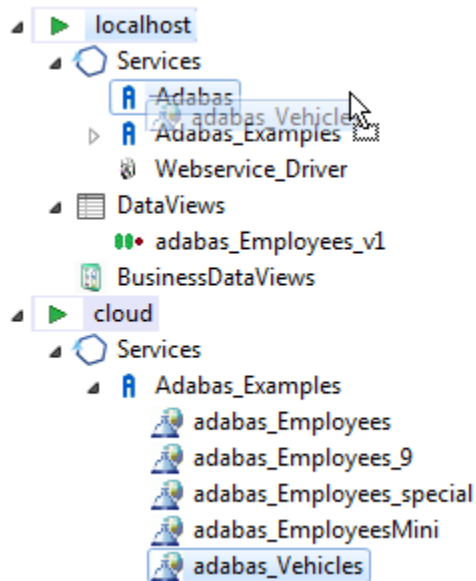
Service as a drag source

- Dragging services from one server to another

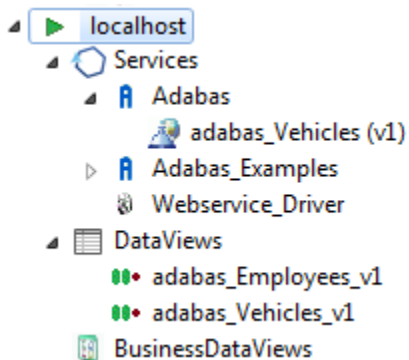
- Dragging services from one driver to another on the same server

Dragging services from one server to another

Service(s) can simply be copied from one server to another.



the service(s) will be copied along with all definition files ((Business)DataViews, XSDs, XSLs)



Note: The source and target drivers must be of the same type, otherwise an error message will be thrown.

Dragging services from one driver to another on the same server

In order to change the driver used for a service, just drag-and-drop it as required.

DataView and BusinessDataView as a drag source

(Business)DataViews can be dropped as follows

- Dropping a (Business)DataView on a Service
- Dropping a (Business)DataView on a Server definition

Dropping a (Business)DataView on a Service

A DataView or BusinessDataView dropped on a service will replace the respective view currently used by that service.

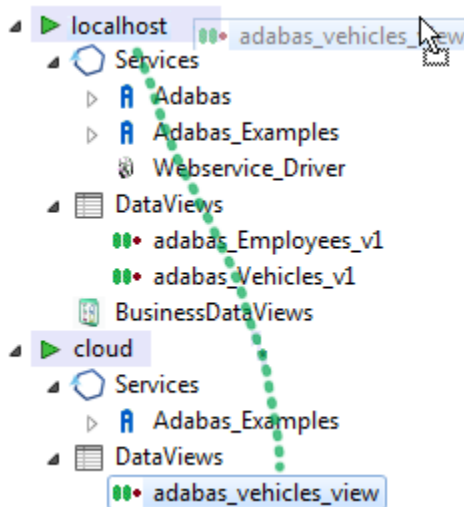
This can either be a View located on the same server as the service, or one to be copied from a different server.



Note: The service needs to be *refreshed* after this change, otherwise the WSDL will still reflect the previous state of the service.

Dropping a (Business)DataView on a Server definition

The (Business)DataView will be copied from one server to another



15 Portus BusinessDataViews

▪ Overview / Concepts	94
▪ Using the BusinessDataView Webservice	99
▪ Working with BusinessDataView Objects	201
▪ Additional features	102
▪ Summary	104
▪ Creating BusinessDataViews (legacy)	106
▪ BusinessDataView Editor hints	130

Portus enables physical assets (tables, files, etc.) to be exposed as "atomic" WebServices, which is useful when direct access to these resources is required. However, it is often the case that a single physical table or file does not represent an object that is of use in a business sense. A typical business function will generally consist of data from more than one physical file/table/...

For this purpose, Portus introduces the concept of *Business Data Views* (BDVs). This will enable the building of "compound" services around logical views of data, combining the relevant elements from the underlying physical assets. This way WebServices can be created to list, add, update, delete, get and select objects that will represent data in a business sense as against the physical view.

Overview / Concepts

All businesses have enormous amounts of data which is exposed via various interfaces and applications, however, rarely, if ever, does one application, one database table or one file represent a complete piece of business related data. The reason for this is straightforward in that most IT projects tend to focus on the 'data model' and the efficiency of getting at data as the primary driver when designing such systems. This is still core to system design today to ensure that the physical architecture can handle the loads anticipated for a given system. However, it generally means that new projects must deal with data models that are sometimes hard to understand from a business perspective. Portus technology now offers the ability to map the existing data and business logic assets into composite views that reflect the business view of the data. This means that newer projects can deal with business related concepts and objects when viewing or updating the core assets.

A Simple Example

Taking a very simple example where an organization has two tables or files containing information about the owner of a vehicle and the vehicle details itself. Assume the data model here is as follows:

1. The Owner Table or File (The "Personnel" File)



Note: This contains details about the owners of a vehicle. For the purposes of this illustration, we will assume the following fields:

- The owners unique ID which identifies the owner and will be referred to as the OID in this example. This is the primary key for this table/file.
- The owners First Name which will be referred to as 'Name' in this example.
- The owners surname which will be referred to as 'Surname' in this example.

2. The Vehicle Table or File



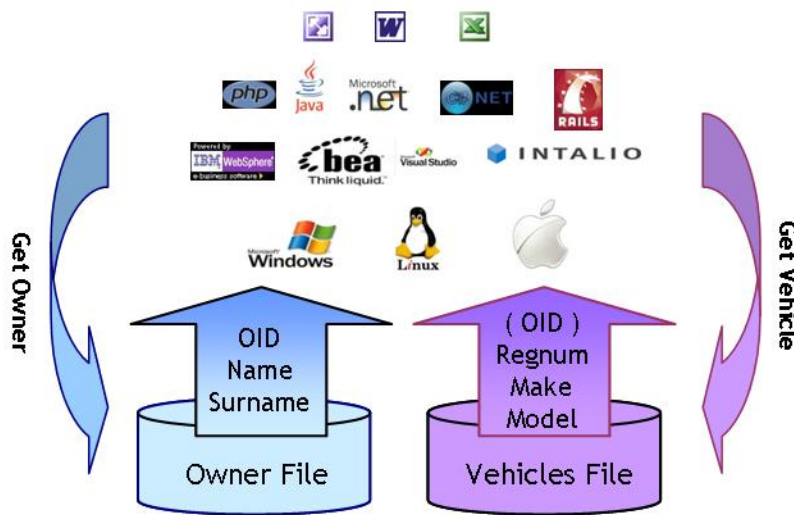
Note: This will contain details about the vehicles that are owned by owners. For the purposes of this illustration, we will assume the following fields:

- The owners unique ID which identifies the owner and will be referred to as the OID in this example. This is the foreign key for this table/file which links it to the "owner" file.
- The vehicles registration number which will be referred to as 'Regnum' in this example. This is the primary key for this table/file.
- The vehicles manufacturer which will be referred to as 'Make' in this example.
- The vehicle's model will be referred to as 'Model' in this example.

In the following sections, the traditional approach is discussed and the issues there are with such an approach followed by the Portus approach.

The traditional approach

In order to access and update this data, applications must understand that there are two different tables or files and must understand the association between the files. They must also understand the requirements to keep these files consistent. In this case, when an owner is added with a OID number, the vehicles file must have a vehicle entry with that particular OID number. The access will normally be direct as per the following diagram:



Traditional Implementation

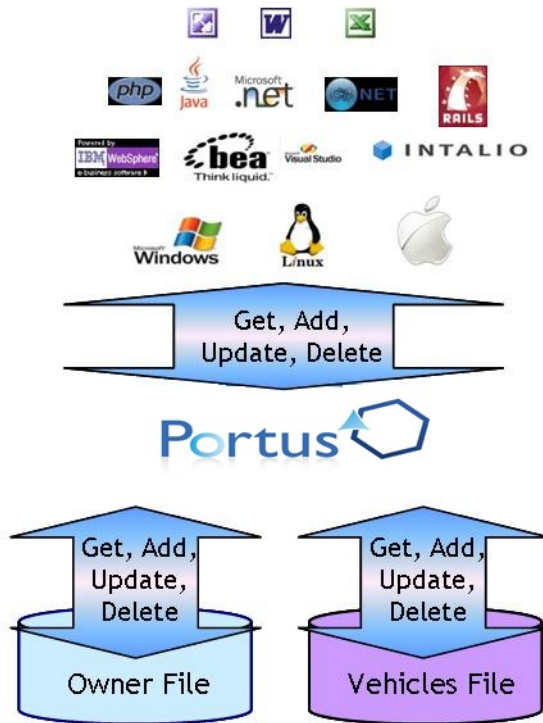
There are a number of issues with this scenario:

1. For each request to get the data there is a requirement to make two requests to the back end database across what will normally be a busy network.
2. The application needs to understand that the 'OID' returned is used to get the associated entry in the Vehicles table/file. While in this case, the OID number may be of value to the client, it

may not be necessary for the client to see or know this. In many cases, keys are used simply as a means to locate a unique record and have no value outside the database itself. However, these must still be understood and manipulated by the programmer.

3. For traditional access, software must be installed on the client system to access the back end databases. This costs time and money to keep up to date and, in many cases, has additional license costs associated with it.
4. If the data model is not clear to the programmers developing on the client side, the databases may end up with inconsistent data.
5. If the data is in multiple databases, there may be an issue with synchronizing updates to those databases unless an infrastructure supporting the two-phase-commit (2PC) protocol is available and in use.

The Portus approach

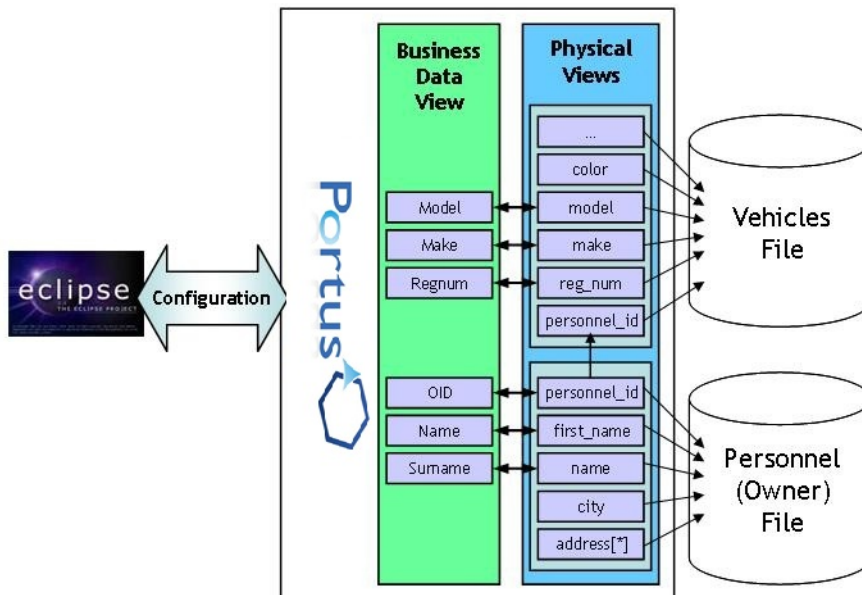


This scenario offers a number of substantial advantages:

1. Only one trip to the back end system is required to retrieve or update the composite object.
2. The programmer does not have to know or understand internal links between tables or files and just has to deal with the fields and columns of relevance to the application.
3. Portus needs to be installed on the server where the data is resident only, thus no software has to be installed or maintained on the client system apart from the client business logic itself.
4. The owner of the data defines the model and thus programmers always deal with the business data view and need no understanding of how the data model looks in the background.
5. Portus can take care of any transactional issues with data from different databases using the transactional capabilities of the backend system. This requires two-phase-commit (2PC) infrastructure to be present on the system, Portus will interface with it to guarantee transaction integrity.

Creating a BusinessDataView Service

A BusinessDataView based WebService is created in Portus based on existing "physical" WebServices. The physical service is the service that is created to directly expose a table, file or piece of business logic. This means that existing services defined to Portus may be reused in one or more Business Data Views. The Eclipse Control Center supplied with Portus is used to create the business data view as illustrated in the following diagram:



As can be seen from the above, the BusinessDataView is built as a separate entity to reflect the "business entity" and is then linked to the physical data views. Some notes about the concepts:

1. Only the fields / columns that are required in the BusinessDataView must be selected from the physical view. In the example above, city and address in the Owner (Personnel) table/file and color (and others) in the Vehicles table/file are not used in the BusinessDataView.
2. Fields or columns in the physical view may be passed and parsed internally though not form part of the actually BusinessDataView. This is useful for passing internal keys to data which can sometimes be meaningless values simply to find data and thus have no value in being shown to a user.
3. This is all achieved using a GUI provided as part of the Eclipse Control Center with Portus.

Using the BusinessDataView WebService

Once the BusinessDataView Service has been created, it can be used in precisely the same way as any other Portus WebService. If the service is called 'OwnerVehicle' for example, the WSDL for the SOAP requests may be returned using the following URL:

http://<host>:<port>/OwnerVehicle?WSDL

This WSDL will expose the standard database CRUD services offered today for physical services by Portus namely ADD, DELETE, UPDATE, GET, LIST and the various SELECT methods. In the same way, to use the REST based interface, the following request will return the BusinessDataView for the Owner id '12345678':

http://<host>:<port>/OwnerVehicle?F=GET&OID=12345678

All owners with an ID starting 1234 could be listed using the following URL:

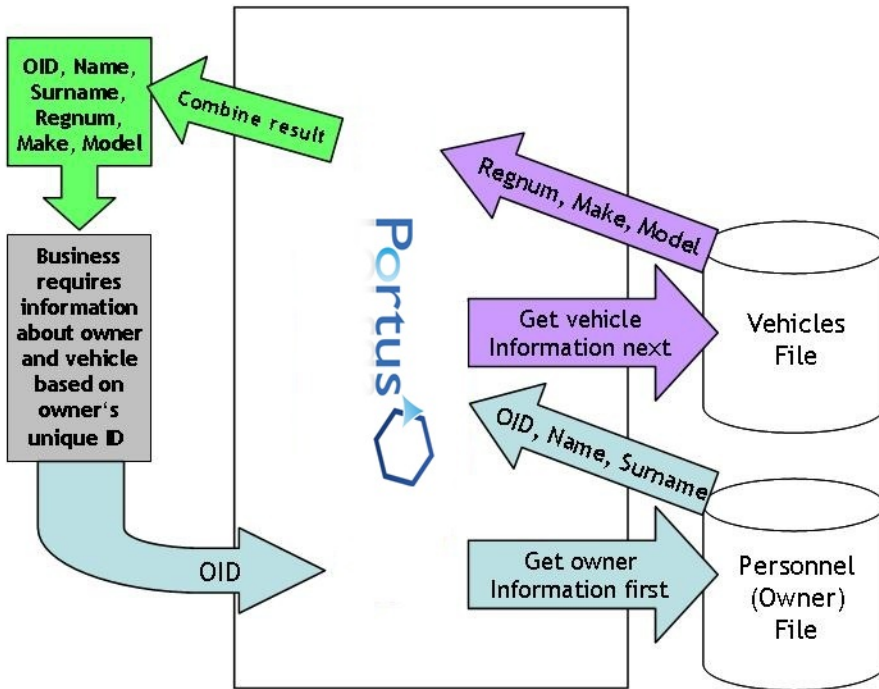
*http://<host>:<port>/OwnerVehicle?F=LIST&OID=1234**

Working with BusinessDataView Objects

The following subsections show how the BusinessDataView Service processes the various types of requests.

Accessing a BusinessDataView Object

Portus can make this sample BusinessDataView object available as follows using the standard REST or SOAP Based access available with Portus:

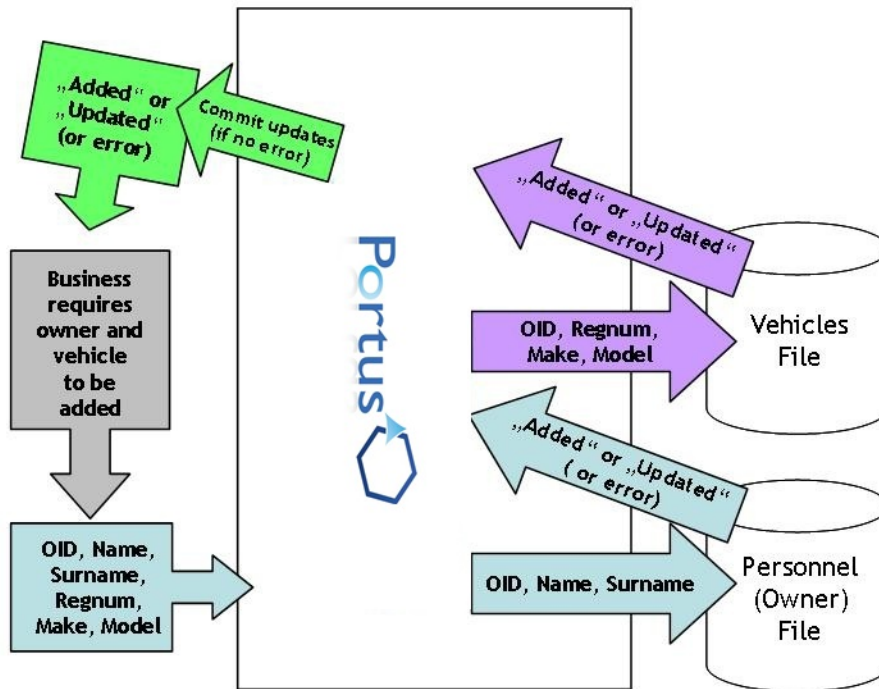


Some notes about the above:

1. If the key for the vehicles file is not the same as the input key (here: OID) for the owner file it may be derived from data on the owner's table/file.
2. The OID only appears once in the output but it doesn't have to appear at all.
3. Any data from either table/file may be returned as part of the BusinessDataView object
4. Using the Portus LIST functionality, it would be possible to get a list of owners and their vehicles using this service, or all vehicles for a single owner.

Adding or updating a BusinessDataView Object

When a new owner and vehicle(s) must be added or updated, Portus can achieve this as one standard REST or SOAP request:

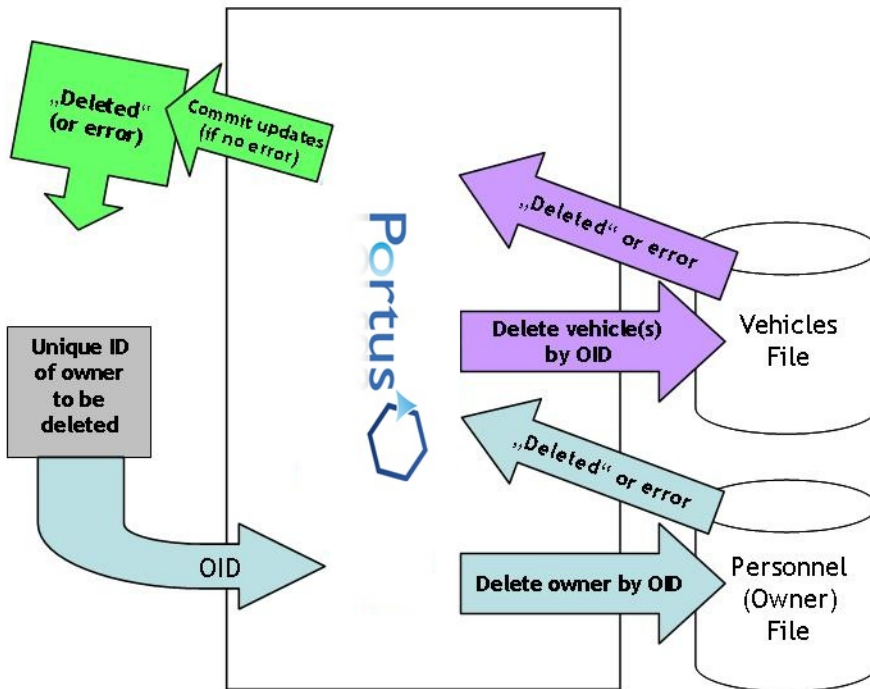


Some notes about the above:

1. Either both adds to the physical database work or neither do using a transactional semantic and thus the database remains consistent. When more than one database is involved in the transaction two-phase-commit (2PC) capabilities are required to guarantee consistent transactions.
2. All data is supplied in one request, this avoids the need for two round trips between the client and the server to add or update the objects (physical DataViews).

Deleting a BusinessDataView Object

When an owner and vehicle(s) must be deleted, Portus can achieve this as one standard REST or SOAP request:



Some notes about the above:

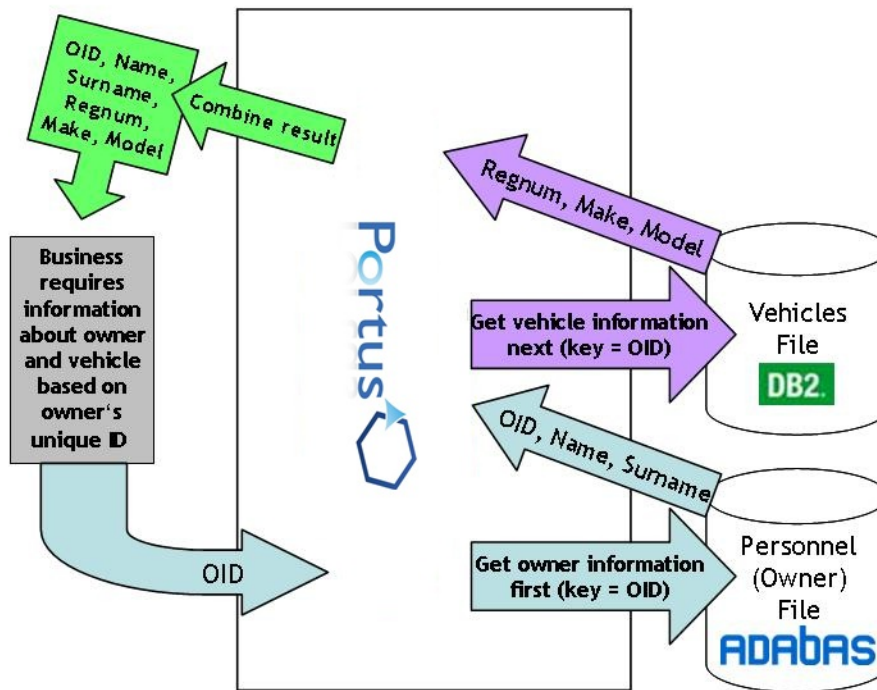
1. Either both (all) deletes from the physical database work or neither do using a transactional semantic and thus the database remains consistent.
2. All data is supplied in one request, this avoids the need for two round trips between the client and the server to delete the objects (physical DataViews).
3. While only primary keys for all affected files are required to delete the records, the same BDV used for add or update requests can be used.

Additional features

Portus offers a number of additional features in addition to enabling the creation of Business-DataViews based on "simple" scenarios as outlined in the previous sections

Data in Multiple Databases

When building a BusinessDataView, the data does not need to reside in the same database or even database technology. For example, the following illustrates how this would look with some data in ADABAS and some data in DB2:

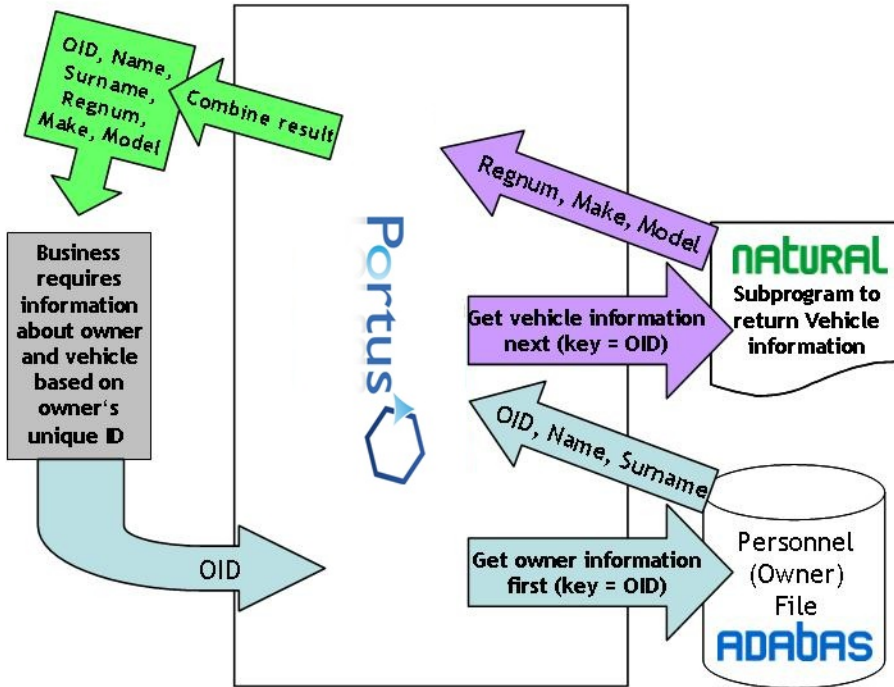


Some notes about the above:

1. Adds, updates and deletes are also possible with this configuration but a transaction manager such as RRMS is required to ensure integrity is maintained between the databases.
2. Databases can be replaced and or added as appropriate without the external business logic changing.

BusinessDataViews Including Business Logic

When building a BusinessDataView, the data must not be retrieved directly from a database. Portus can use any service defined by it as input for a business data view object. For example, the following illustrates how this would look with some data in ADABAS and a Natural subprogram providing additional data:



Some notes about the above:

1. Adds, updates and deletes are also possible in which case the application will be given an indication of the type of operation when it is called.
2. The business logic may be running in (amongst others) CICS, COBOL, PL1, C etc.
3. It is possible to create a BusinessDataView service based on business logic services only so no direct database access is required.

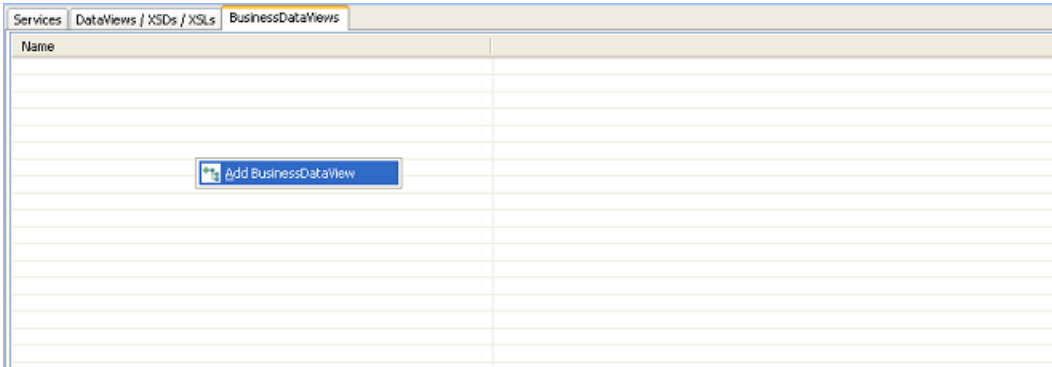
Summary

It is clear that the Business Data View concept in Portus can provide an incredible level of flexibility to an organization in the following ways:

- Avoid the need for new projects to understand legacy data models. Enable them to work with the business model which is more natural for them
- Create Business Data View models from multiple sources including different databases and business logic as required.
- Ensure consistency in your data by exposing services that prevent applications creating inconsistencies in the data.

- Create abstract business services that may be used again and again from any technologies including Java, c#.net, vb.net, php, Ruby, Excel, Word, Internet Explorer and so on.
- Secure your assets based on the business view of the data and not the physical view of the data which is where the security is really required.
- Using the Portus messaging features with MQ or EntireX, enable batch updates of your business objects that can run as part of your batch processing.

Creating BusinessDataViews (legacy)



To create a *BusinessDataView* select the target server, in the ConfigurationView select the Business-DataViews tab

Right-click the ConfigurationView

Enter a name for the BusinessDataView to be created, click OK

Prerequisite - the BusinessDataView driver

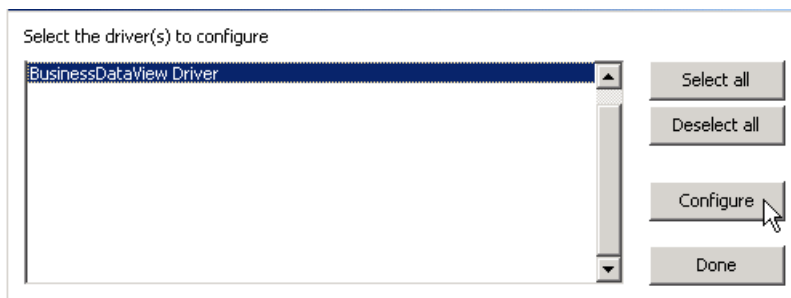
Exposing BusinessDataViews as WebServices requires a BusinessDataView driver to be defined, if there is none already, proceed as follows

- In the *Drivers* tab of the Server properties, click the *Add* button.

or

Right-click on the server and select *Add driver to server*

- Select the *BusinessDataView Driver* and click *Configure*.



- Change as appropriate and click Save.

Driver Title	<input type="text" value="BusinessDataView"/>	<input type="button" value="Save"/>
Driver Name	<input type="text" value="bdvDriver"/>	
Driver DLL	<input type="text" value="bdvDriver.so"/>	<input type="button" value="Close"/>
SBCS codepage	<input type="text" value="latin1"/>	
MBCS codepage	<input type="text" value="utf16"/>	

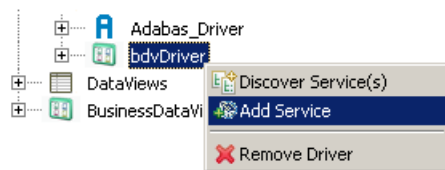
- Click *Done* to close the Driver definition Wizard.

Example 1

The Adabas demo file "Employees" is keyed on personnel number and the "Vehicles" file also has a personnel number which determines which car is owned by which employee. We wish to define a BusinessDataView which gives us personnel id and surname of the employee and, if present, one instance of their vehicles car registration and date acquired.

Creating the BusinessDataView

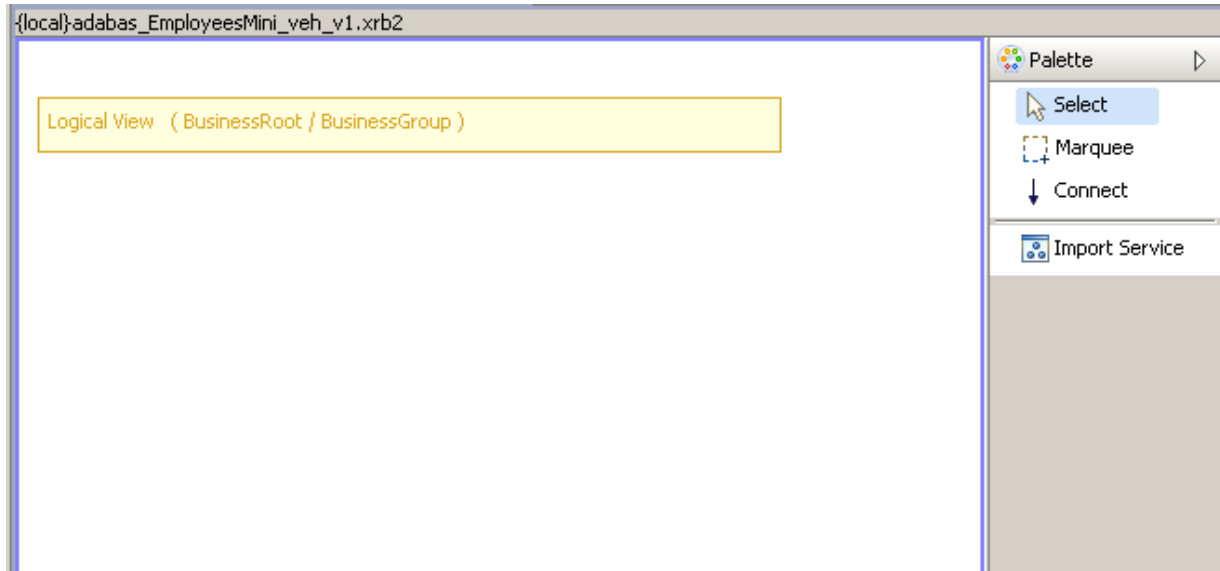
- In the Servers View, right-click on bdvDriver under Services and select 'Add Service'



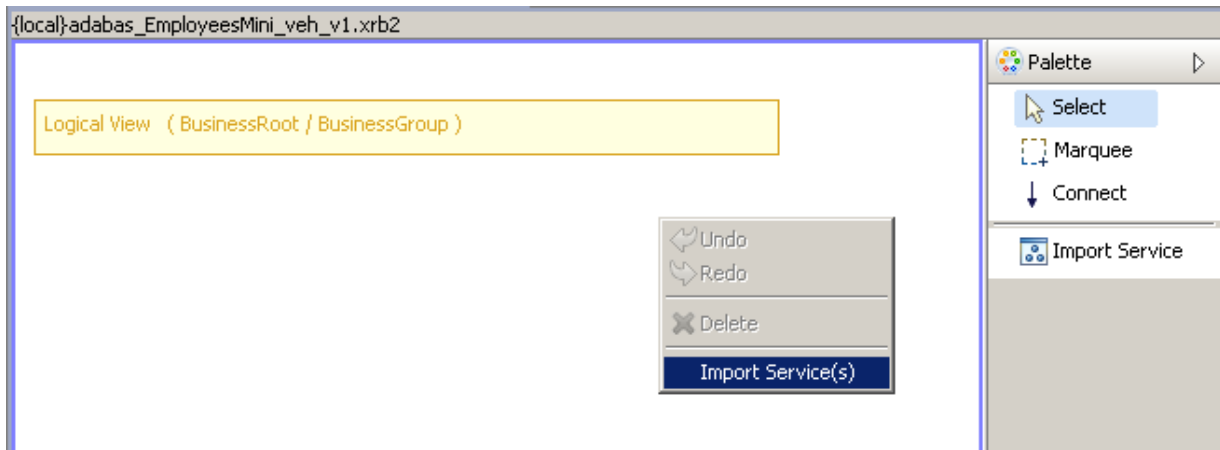
- Name it "adabas_EmployeesMini_veh", click OK

Enter name for new Service	
<input type="text" value="adabas_EmployeesMini_veh"/>	
<input type="button" value="OK"/>	<input type="button" value="Cancel"/>

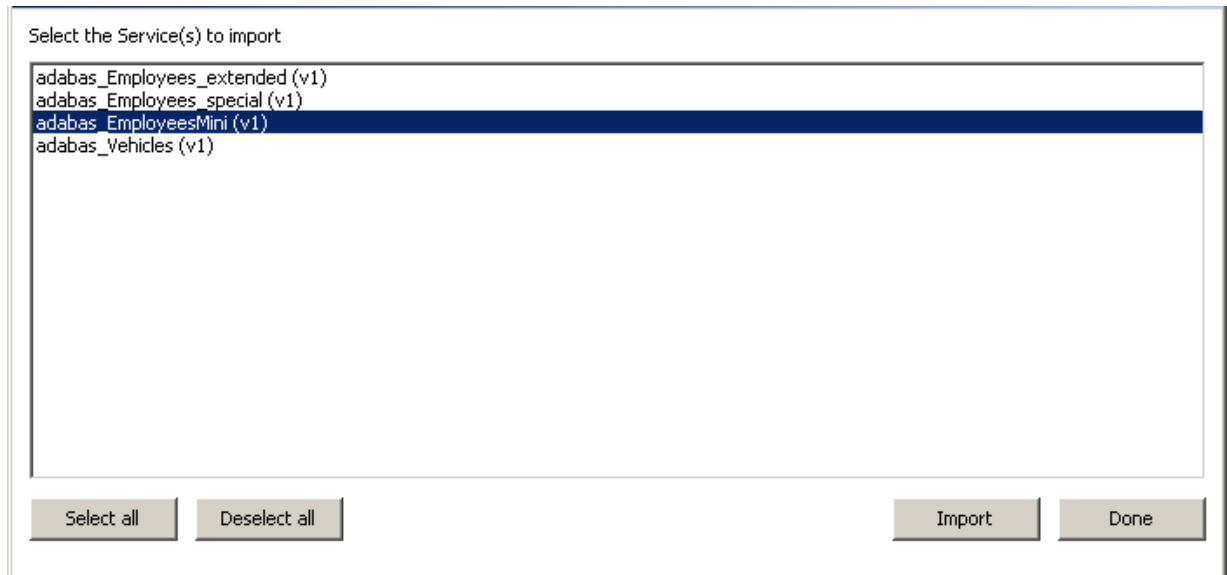
- This will open the BusinessDataView Editor with an initially empty "Logical View" created



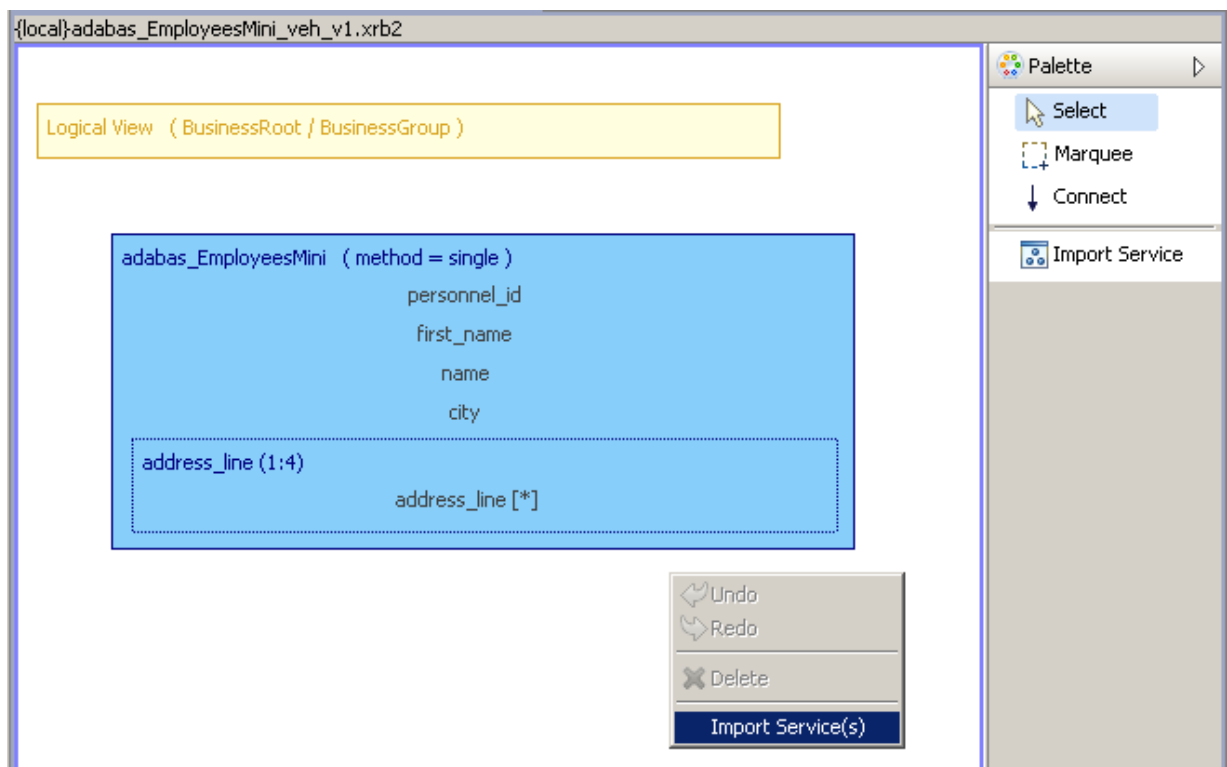
- Right-click anywhere within the blank area of the BusinessDataView Editor, select the *Import Service(s)* function from the context menu



- Select the *adabas_EmployeesMini* WebService service and click *Import*.



- A visual representation of the elements of this web service appears as shown below. Right-click beneath this view as shown and select the *Import Service(s)* function from the context menu



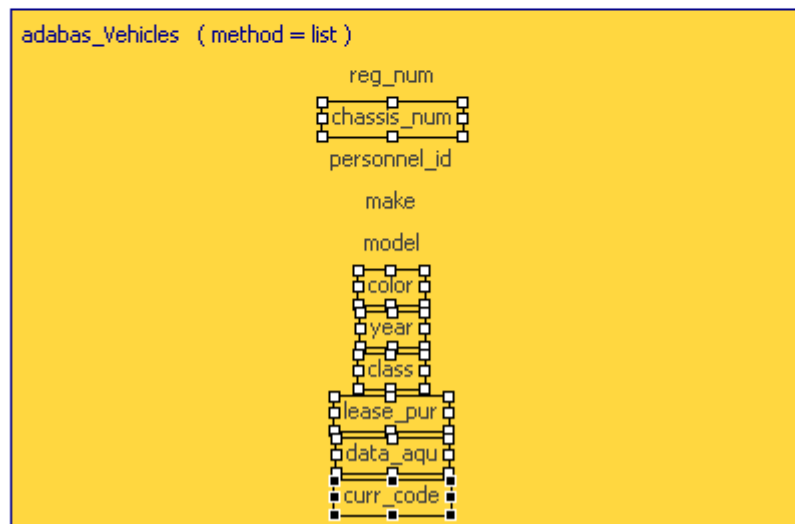
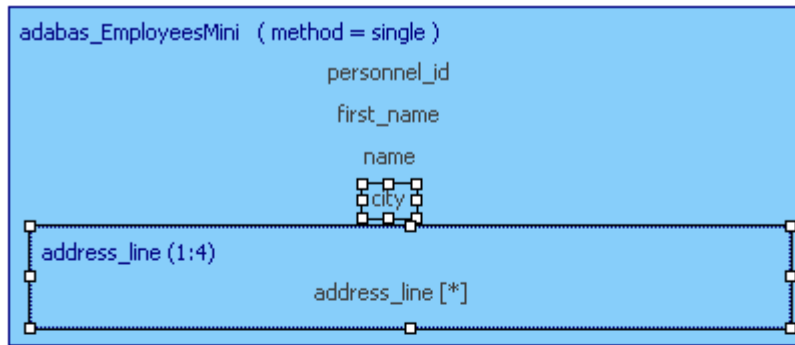
- Select the *adabas_Vehicles* WebService service and click *Import*.



- Select the "Physical View" items not required for the time being (shown selected below) and hit the key to remove them. There is *no* requirement whatsoever to use all of the "base WebService" fields within the BusinessDataView's "physical resource(s)". Note that there is also no requirement to remove them as they can stay in the Physical View if so desired.



Note: Select the *city* item first and, holding down the Ctrl key, select the other items.

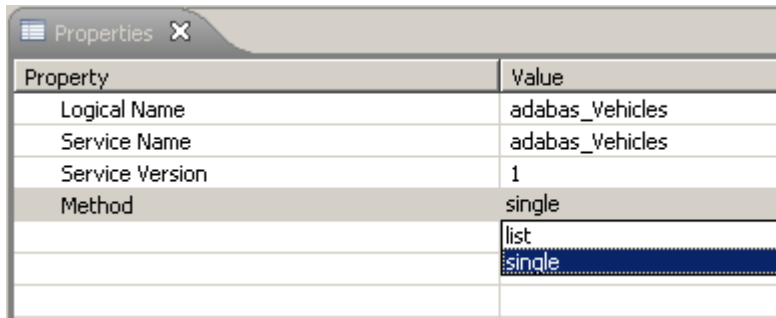


- This will leave you with the following items. For this example we are only interested in one vehicle per employee so we need to change the method from *list* to *single*. Our next example will demonstrate the use of the list method.

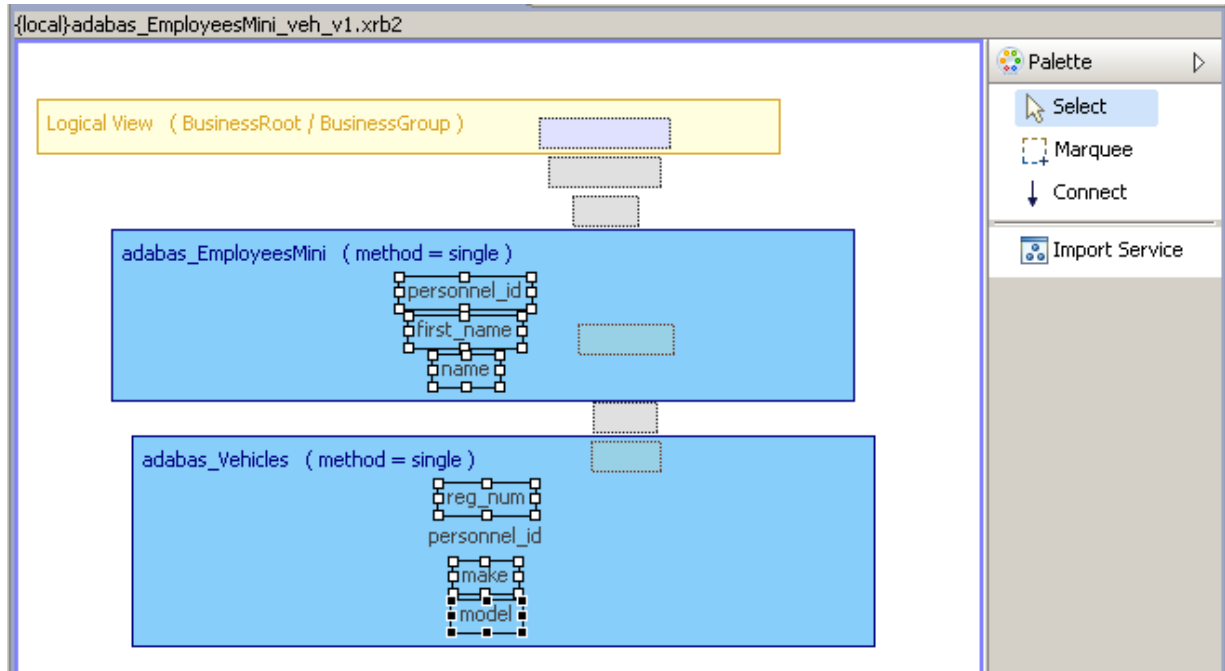
```
adabas_EmployeesMini ( method = single )
    personnel_id
    first_name
    name
```

```
adabas_Vehicles ( method = list )
    reg_num
    personnel_id
    make
    model
```

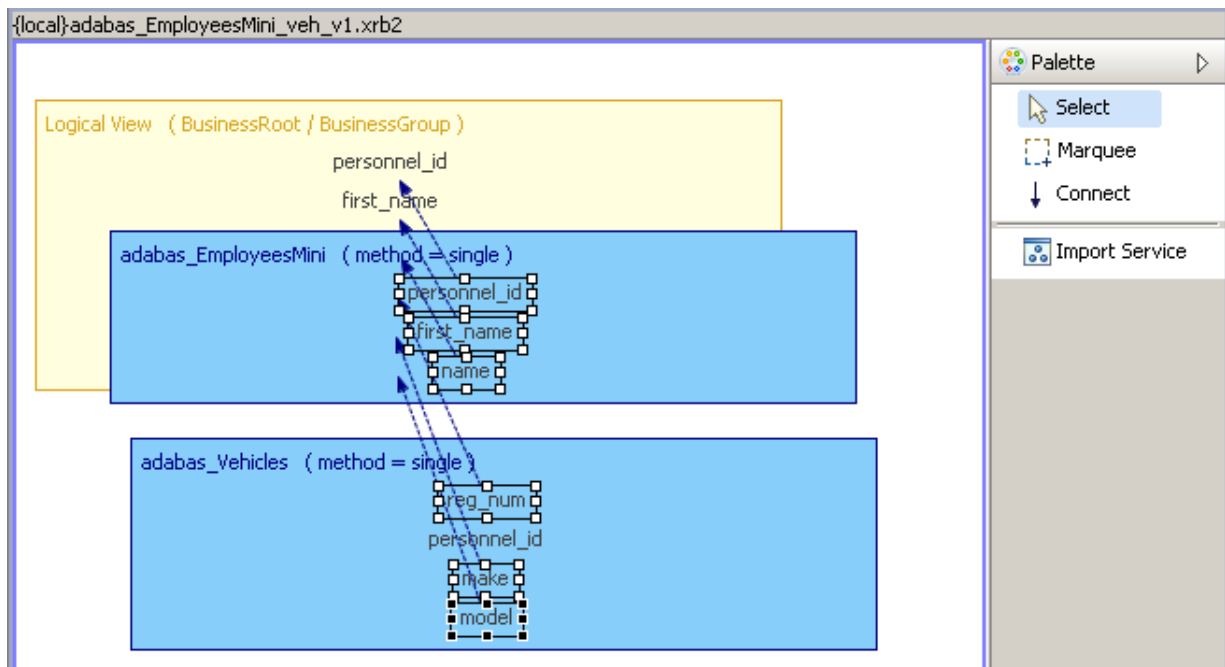
- Click on an empty area within the adabas_Vehicles physical view. The Properties view should display the properties for this Physical Resource. If the Properties view is not visible, select menu item Window -> Show View -> Properties. Change the Method to single as shown. Ctrl+S to save.



- Select all fields, *but NOT the personnel_id field within adabas_Vehicles*. Drag them across to the "Logical View" (LV) and drop them there. Note that there is a shadow view of the items selected. There is also a red circle with a line through it (not shown here) which disappears when the drop into the LV is possible.



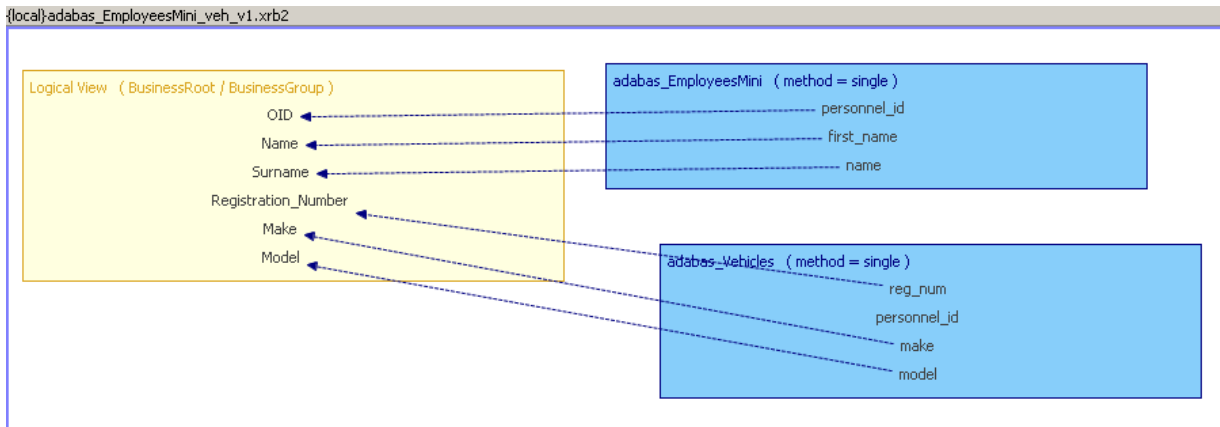
- This will add the selected fields to the "Logical View", and also create their connections to the source "Physical View" fields at the same time. Note that the various views can be moved around to get a clearer picture. It is a good idea to save the BDV at this stage (Ctrl+S).



- Now go to the "Logical View" and select the `personnel_id` field. In the Properties view change the node name to `OID` by changing its XML-Name value.

Property	Value
Logical Node properties	
XML-Name	OID
Field direction	In/Out
Max. Occurrences	1
Key Type	primary
Service reference	
Service	adabas_EmployeesMini
Node	/personnel_id

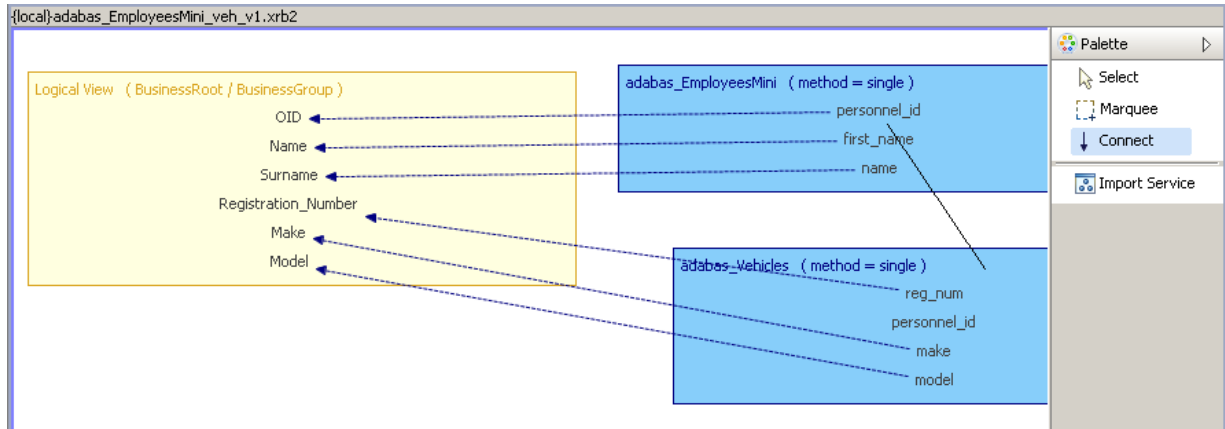
- Change all field names as required, the picture should now look like this. Note again that this is not compulsory. The names can be left as is if so desired.



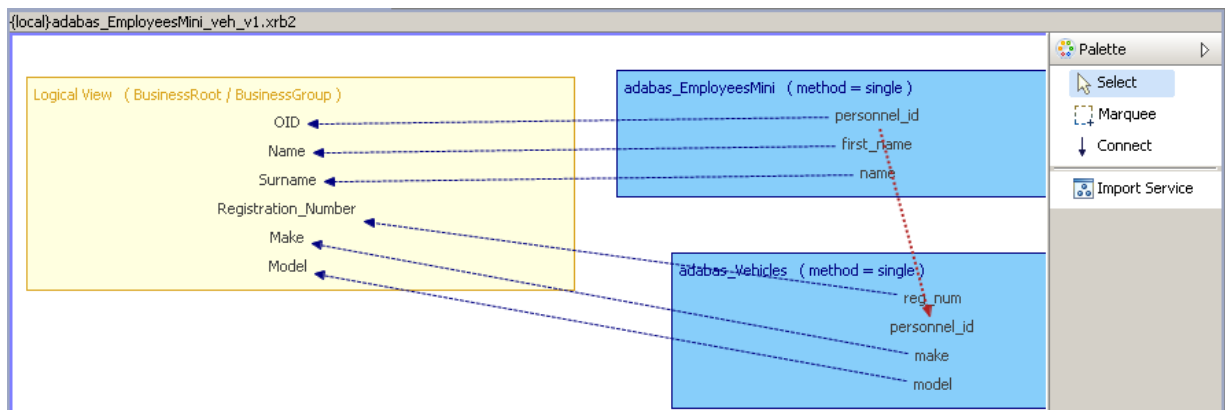
- Recall that this BDV will enable us to query employee details and their vehicle details. There are 2 steps to creating this link in the BDV. The first step involves setting the appropriate field to be a primary or secondary key. In this example we need to set the OID field in the LV to primary. This is done by selecting the field in the LV and setting the key to primary in the associated Properties view as follows:

Property	Value
Logical Node properties	
XML-Name	OID
Field direction	In/Out
Max. Occurrences	1
Key Type	primary
Service reference	
Service	primary
Node	/personnel_id

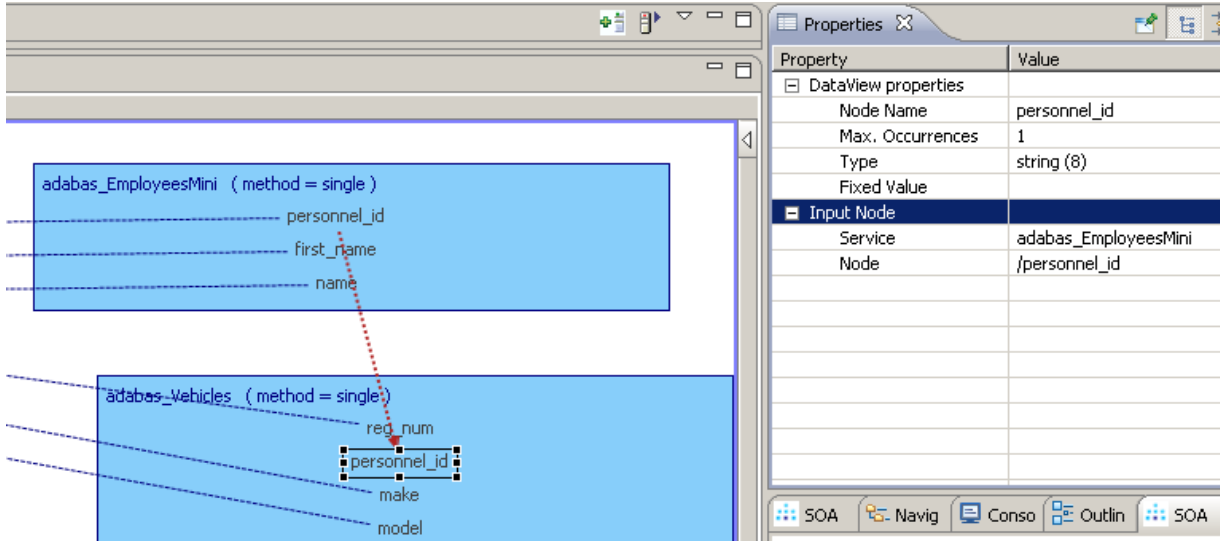
- The next step involves creating a connection or link between the appropriate field of one physical resource to a field in another physical resource. Select *Connect* item from the Palette. Select the *personnel_id* field of *adabas_EmployeesMini* view and move the cursor away. As you can see this marks the starting point for the connection:



- Now click the *personnel_id* field within the *adabas_Vehicles* view.



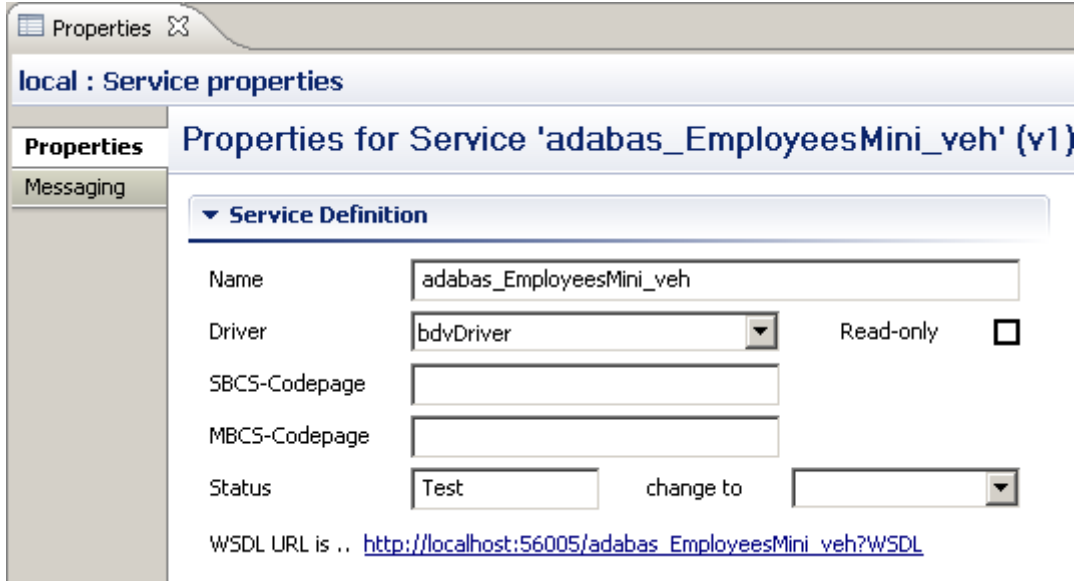
- This "links" the "base (physical) WebServices" together and indicates that the *personnel_id* from *adabas_EmployeesMini* is to be passed on and input into the *personnel_id* field of *adabas_Vehicles*. When the WebService based on the BusinessDataView is invoked, the "Employees" file is read first, the *personnel_id* taken from the result is then used to access the related Vehicles record.



- Click the *Save* icon, or File->Save, or Ctrl+S to save the BusinessDataView on your Portus server

Test the Webservice

- Open the URI next to *WSDL URL is ..* in the Properties view.



- The WSDL for a Webservice based on a BusinessDataView looks just like one for a "basic" (= "physical") Webservice, exposing all available access functions (add, delete, get, list, select, etc).

```

</xs:sequence>
</xs:complexType>
<xs:element name="BusinessGroupListElement" type="asg:BusinessGroupKeyType" />
<xs:element name="BusinessGroupGetElement" type="asg:BusinessGroupPrimaryKeyType" />
<xs:element name="BusinessGroupAddElement" type="asg:BusinessGroupRootType" />
<xs:element name="BusinessGroupDeleteElement" type="asg:BusinessGroupPrimaryKeyType" />
<xs:element name="BusinessGroupUpdateElement" type="asg:BusinessGroupRootType" />
<xs:element name="BusinessGroupSelectElement" type="asg:BusinessGroupSelectType" />
<xs:element name="BusinessGroupSelectNextElement" type="asg:BusinessGroupSelectNextType" />
<xs:element name="BusinessGroupSelectCountElement" type="asg:BusinessGroupSelectType" />
<xs:element name="BusinessGroupSelectEndElement" type="asg:BusinessGroupSelectEndType" />
- <xs:element name="BusinessRootElement">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="BusinessRoot">
- <xs:complexType>
- <xs:sequence>
- <xs:element maxOccurs="unbounded" name="BusinessGroup">
- <xs:complexType>
- <xs:sequence>
<xs:element name="OID" type="xs:string" />
<xs:element name="Name" type="xs:string" />
<xs:element name="Surname" type="xs:string" />
<xs:element name="Registration_Number" type="xs:string" />
<xs:element name="Make" type="xs:string" />
<xs:element name="Model" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

- Change the WSDL directive in the URI to *LIST&OID=20021**, this will show the surname for the selected Employees record(s), plus the (first) car's registration, make and model for every employee whose OID (personnel_id) begins with 20021.

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <BusinessRoot xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <BusinessGroup>
  <OID>20021000</OID>
  <Name>DALIAH</Name>
  <Surname>MILLS</Surname>
  <Registration_Number>31108350</Registration_Number>
  <Make>CHRYSLER</Make>
  <Model>PLYMOUTH</Model>
</BusinessGroup>
- <BusinessGroup>
  <OID>20021100</OID>
  <Name>ROBERT</Name>
  <Surname>JONES</Surname>
  <Registration_Number>31111905</Registration_Number>
  <Make>GENERAL MOTORS</Make>
  <Model>CHEVROLET</Model>
</BusinessGroup>
- <BusinessGroup>
  <OID>20021200</OID>
  <Name>WILLIE</Name>
  <Surname>SENKO</Surname>
  <Registration_Number>31113139</Registration_Number>
  <Make>FORD</Make>
  <Model>GRANADA</Model>
</BusinessGroup>
- <BusinessGroup>
  <OID>20021300</OID>

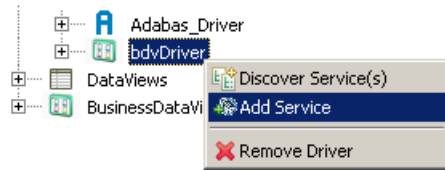
```

Example 2

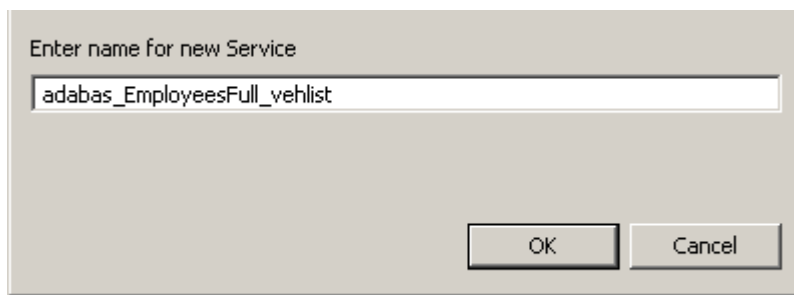
We now define a BusinessDataView which provides more employee details plus a list of vehicle(s) for the employee..

Create the BusinessDataView

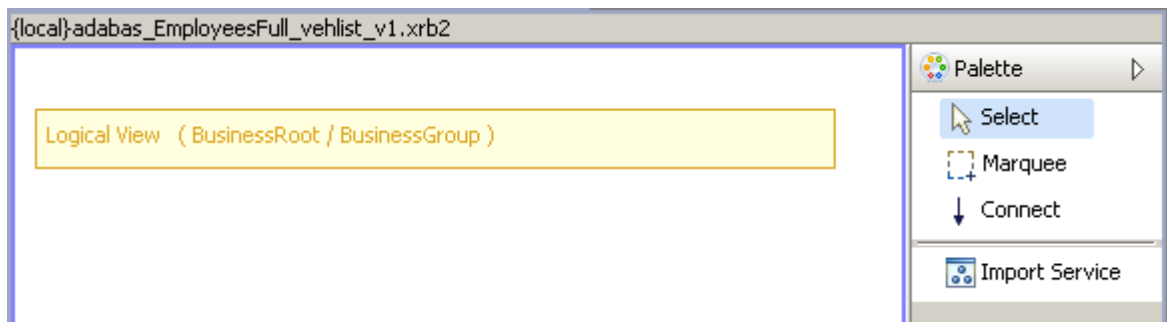
- In the Servers View, right-click on bdvDriver under Services and select 'Add Service'



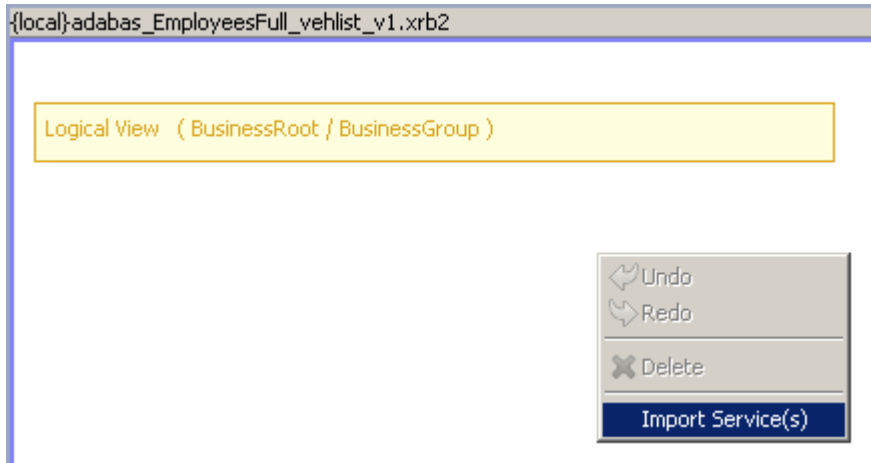
- Name it *adabas_EmployeesFull_vehlist* and click OK



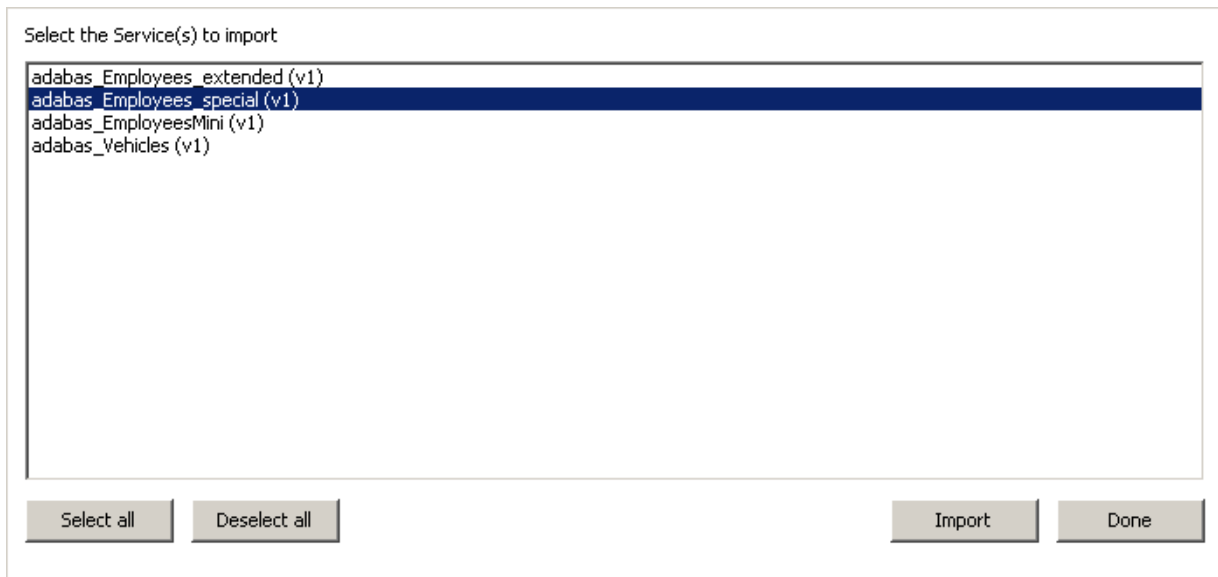
- This will open the BusinessDataView Editor with an initially empty "Logical View" (LV) created



- Right-click anywhere within the blank area of the BusinessDataView Editor, select the *Import Service(s)* function from the context menu



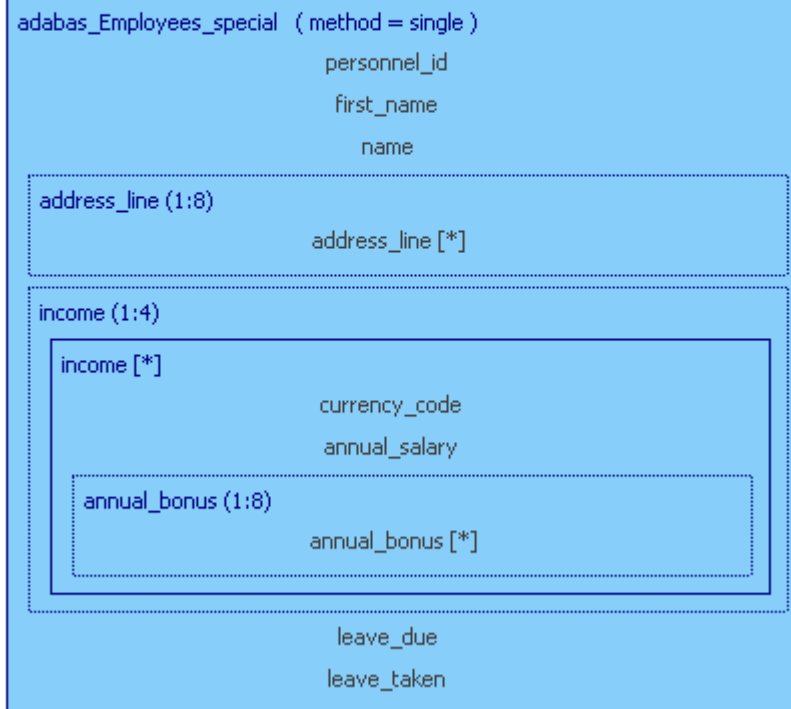
- Select the *adabas_Employees_special* service, click *Import*



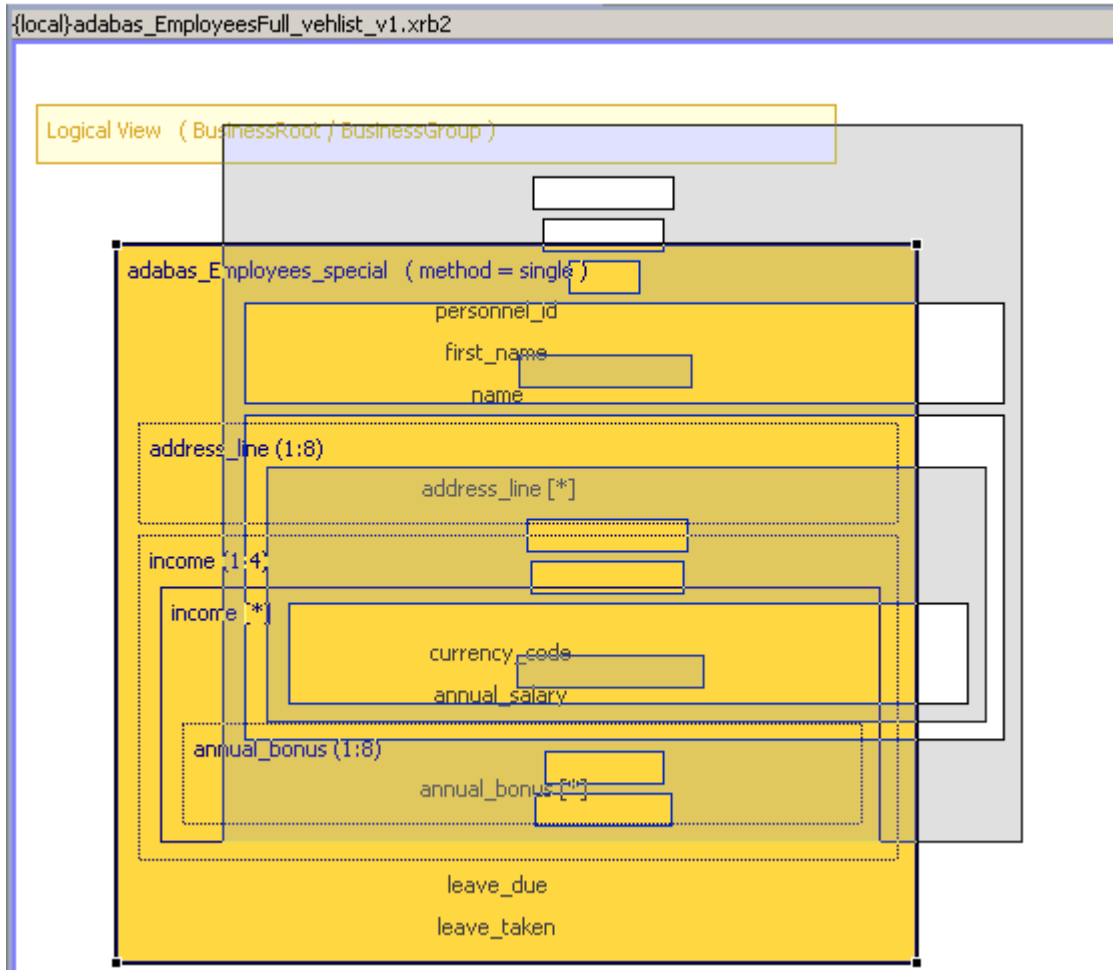
- Again in this example, we wish to use only certain fields in our BDV. Select the items not required and hit the ** key to remove them. Select the first item and, holding down the Ctrl key select the others. This can be done in batches if so desired. Note again that this is not compulsory to do. Here are the fields we wish to retain for this BDV:

{local}adabas_EmployeesFull_vehlist_v1.xrb2

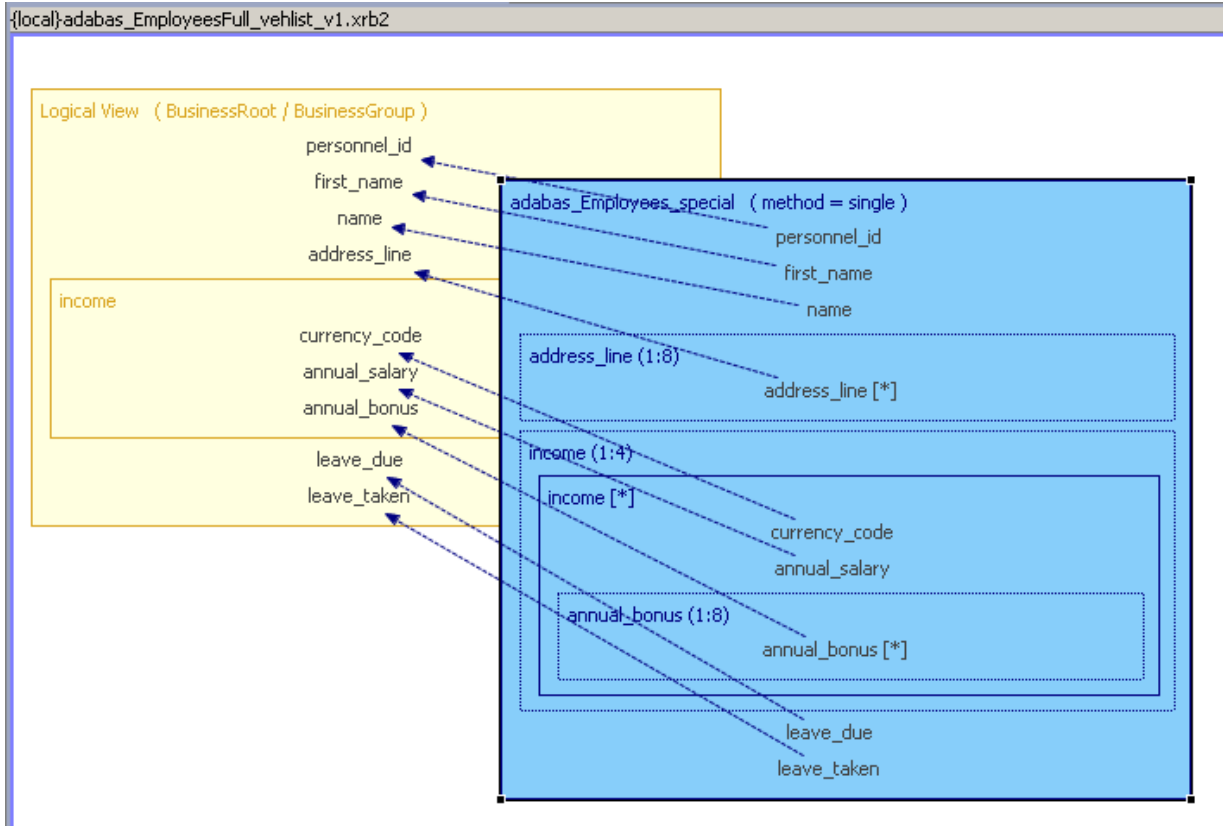
Logical View (BusinessRoot / BusinessGroup)



- It is good practice to save frequently (Ctrl+S, select the Save icon, File->Save). Select the entire `adabas_Employees_special` view, drag it to the LV and drop it there.



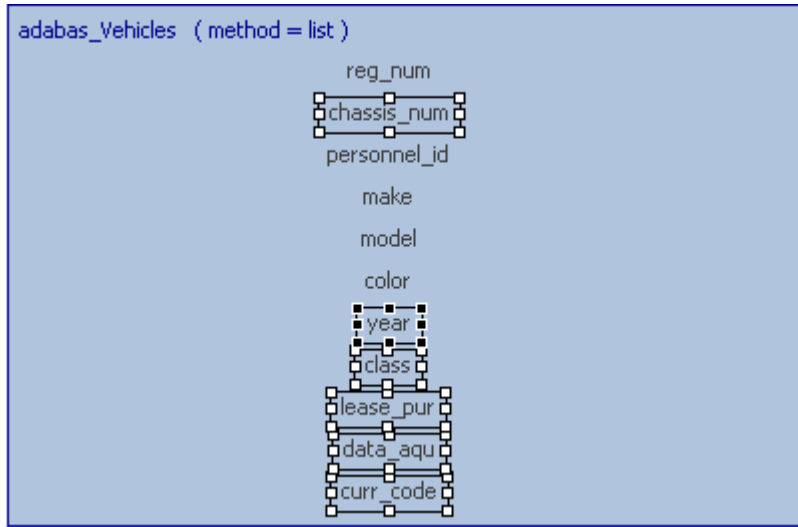
- This will add the fields to the LV, and also create their connections to the source "Physical View" (PV) fields at the same time.



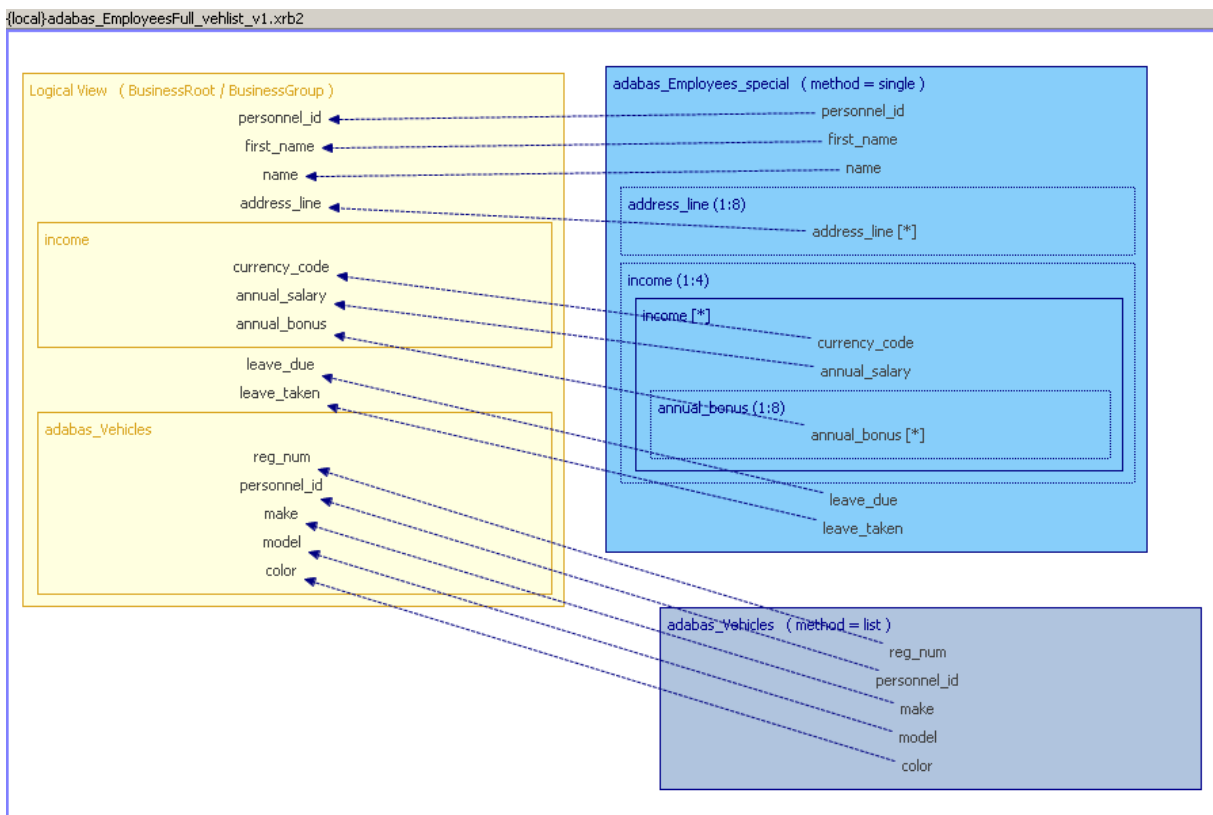
- Import the *adabas_Vehicles* WebService.



- Select the nodes which are NOT to be included in the "Logical View", as shown here, hit the key or right-click and select *Delete*.



- Drag the *adabas_Vehicles* view and drop it into the "Logical View".

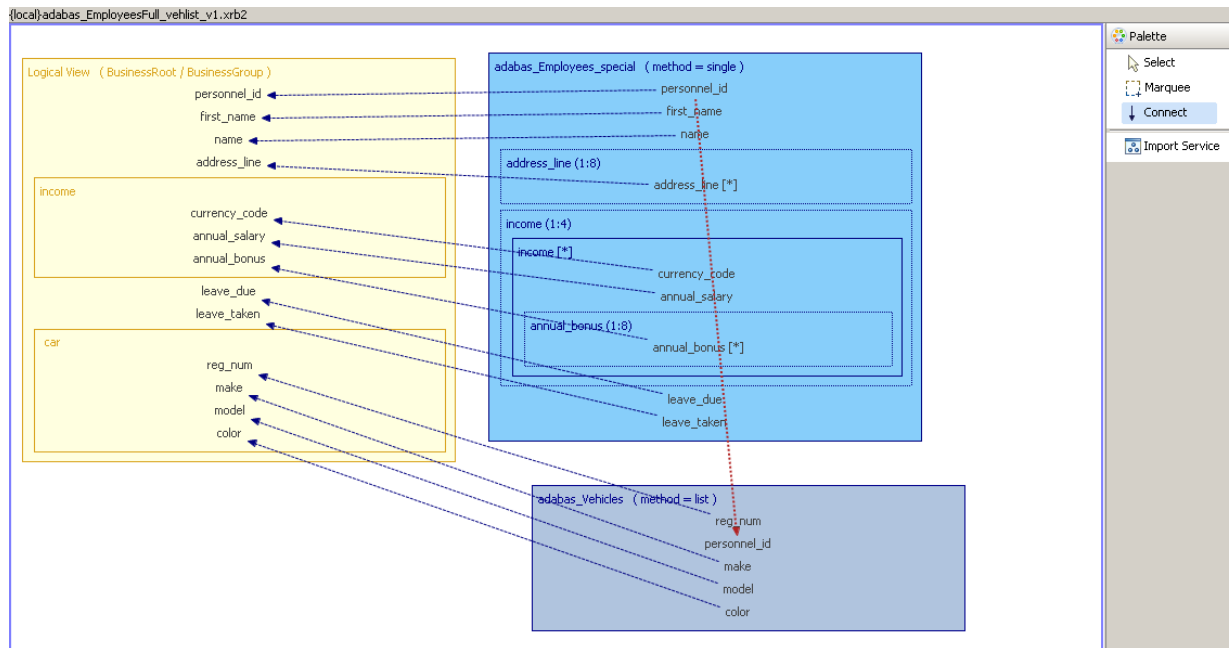


- Note that the elements from the PR have been wrapped in a container or logical group which by default is given the same name as the source PR. This happens automatically when the PR method is set to list. The group name can be changed as follows. Click the newly created group to select it. In the Properties View change the XML-Name from "adabas_Vehicles" to "car".

Property	Value
Logical Group properties	
XML-Name	adabas_Vehicles
Field direction	In/Out
Max. Occurrences	1
Service reference	
Physical Service	adabas_Vehicles
Physical Node	

- At this point we have 2 elements in the LV named *personnel_id*. As shown in Example 1 a connection can be made in the PRs which enables the value for the source PR to be used as input to the target PR. Therefore we have no need for *personnel_id* in the CARS group so select the field and delete it. Connect the *personnel_id* fields from *adabas_Employees_special* and *adabas_Vehicles* as follows.

Select the *Connect* item from the Palette. Select the *personnel_id* field of *adabas_Employees_special* view and move the cursor away. Now select *personnel_id* field of *adabas_Vehicles*. The BDV should now look like this:



- We have already seen that the LV can be a subset of the underlying data. It can also represent that data in a different format e.g. the fields names can be changed and occurrences of fields such as *address_line* can be reduced.

For these changes select the field and open its Property View.

Rename the *personnel_id* field within the LV to *PID* (optional).

Change the key type for the PID field to *primary* (mandatory).

Property	Value
Logical Node properties	
XML-Name	PID
Field direction	In/Out
Max. Occurrences	1
Key Type	primary
Service reference	
Service	adabas_Employees_special
Node	/personnel_id

- Change the *Max. Occurrences* property for the *address_line* field to 3.

Property	Value
Logical Node properties	
XML-Name	address_line
Field direction	In/Out
Max. Occurrences	3
Key Type	none
Service reference	
Service	adabas_Employees_special
Node	/address_line

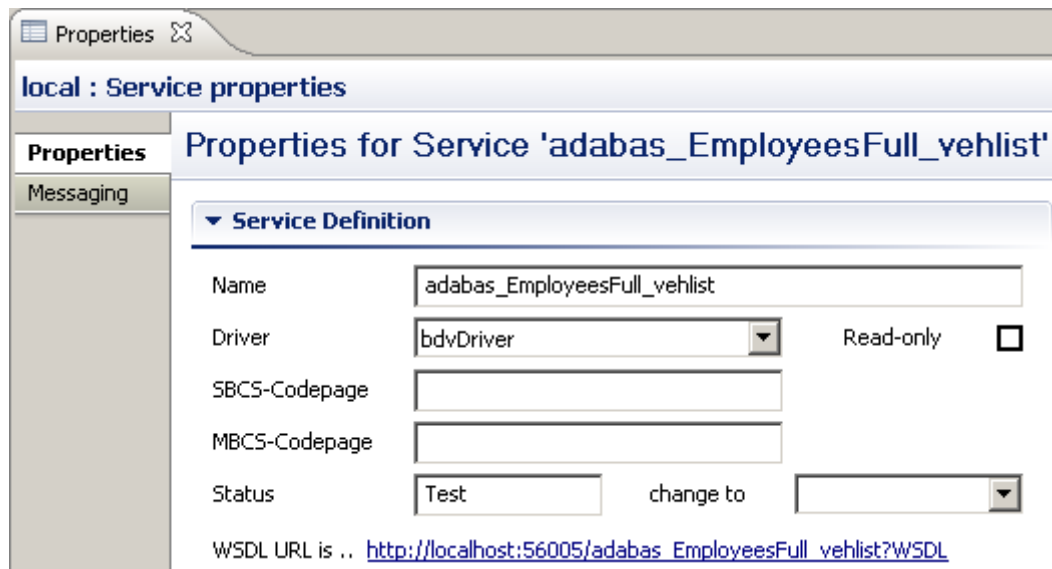
- Change the *Max. Occurrences* property for the *income* group field to 1.

Property	Value
Logical Group properties	
XML-Name	income
Field direction	In/Out
Max. Occurrences	1
Service reference	
Physical Service	adabas_Employees_special
Physical Node	/income

- Click the *Save* icon, or File->Save, or Ctrl + S to save the BusinessDataView on your Portus server

Test the Webservice

- Click the URI next to *WSDL URL is ...* (This can be found in the Service properties for the BDV just created. Select the web service under bdvDriver as shown and open the Property View.

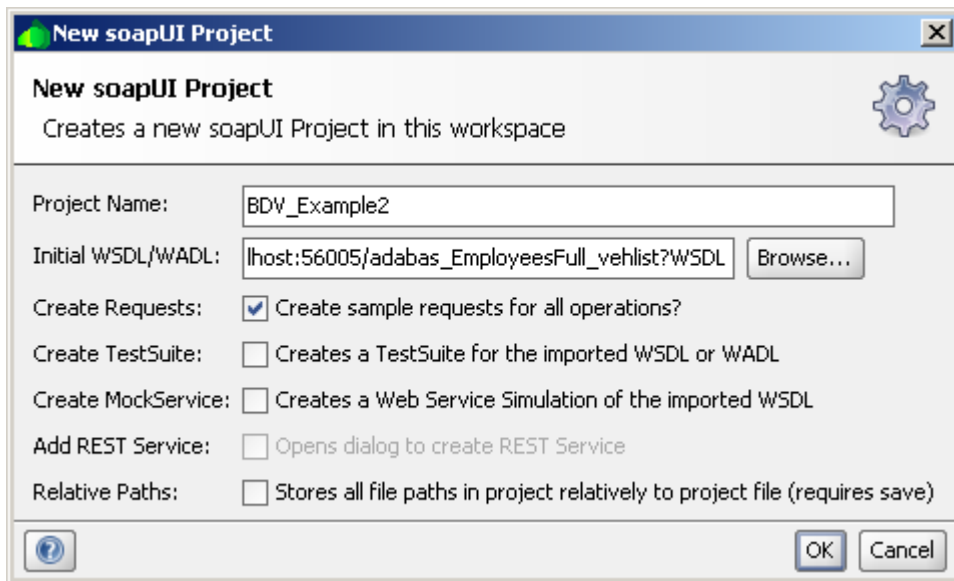


- The WSDL for a Webservice based on a BusinessDataView looks just like one for a "basic" (= "physical") Webservice, exposing all available access functions (add, delete, get, list, select, etc).

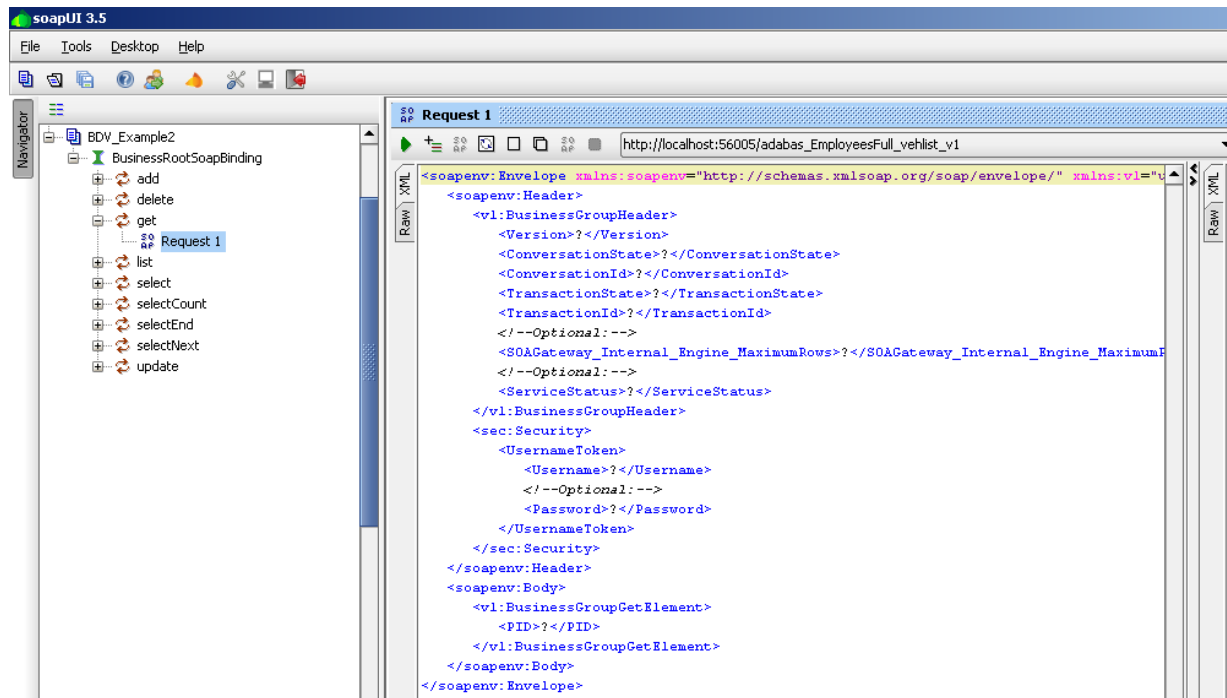
```

<?xml version="1.0" encoding="UTF-8" ?>
- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/s
targetNamespace="uri://localhost:56005/adabas_EmployeesFull_vehlist" xmlns:tns="uri://localhost:5600
xmlns:asg="urn:namespaces:com.risaris/xmiddle/resources/xrb/v1.0.4" xmlns:wsse="http://schemas.x
xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:xop="http://www.w3.org/2004/08/xop/in
- <types>
+ <schema targetNamespace="http://schemas.xmlsoap.org/ws/2002/04/secext" xmlns="http://www.w3
xmlns:tns="http://schemas.xmlsoap.org/ws/2002/04/secext">
- <schema targetNamespace="urn:namespaces:com.risaris/xmiddle/resources/xrb/v1.0.4" xmlns="http://
xmlns:tns="urn:namespaces:com.risaris/xmiddle/resources/xrb/v1.0.4">
  <import namespace="http://www.w3.org/2005/05/xmlmime"
    schemaLocation="http://localhost:56005/SOA_GATEWAY_CONFIGURATION_DIRECTORY/schema/p
- <xs:complexType name="BusinessGroupRootType">
  - <xs:sequence>
    <xs:element maxOccurs="1" minOccurs="1" name="BusinessGroup" type="asg:BusinessGroupType" />
  </xs:sequence>
</xs:complexType>
<xs:element name="BusinessGroupListElement" type="asg:BusinessGroupKeyType" />
<xs:element name="BusinessGroupGetElement" type="asg:BusinessGroupPrimaryKeyType" />
<xs:element name="BusinessGroupAddElement" type="asg:BusinessGroupRootType" />
<xs:element name="BusinessGroupDeleteElement" type="asg:BusinessGroupPrimaryKeyType" />
<xs:element name="BusinessGroupUpdateElement" type="asg:BusinessGroupRootType" />
<xs:element name="BusinessGroupSelectElement" type="asg:BusinessGroupSelectType" />
<xs:element name="BusinessGroupSelectNextElement" type="asg:BusinessGroupSelectNextType" />
<xs:element name="BusinessGroupSelectCountElement" type="asg:BusinessGroupSelectType" />
<xs:element name="BusinessGroupSelectEndElement" type="asg:BusinessGroupSelectEndType" />
  
```

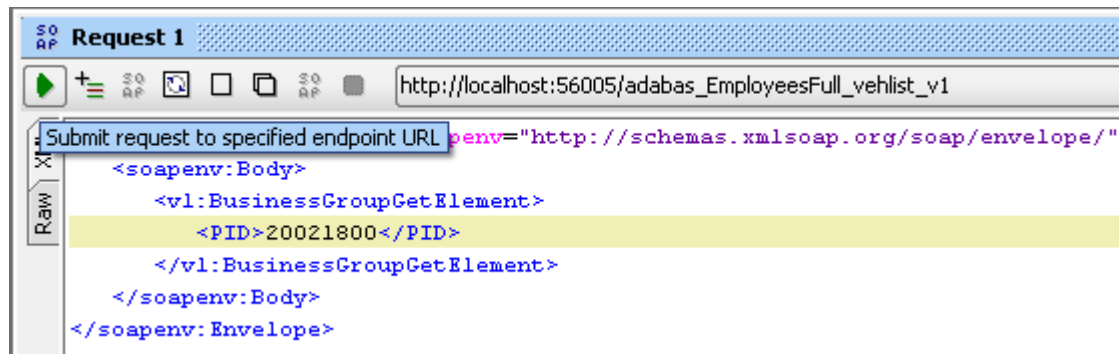
- We will now use *soapUI* (see the Introduction in the Tutorials section) to access the BDV-based WebService. First of all, create a new soapUI project based on the WSDL URL shown above.



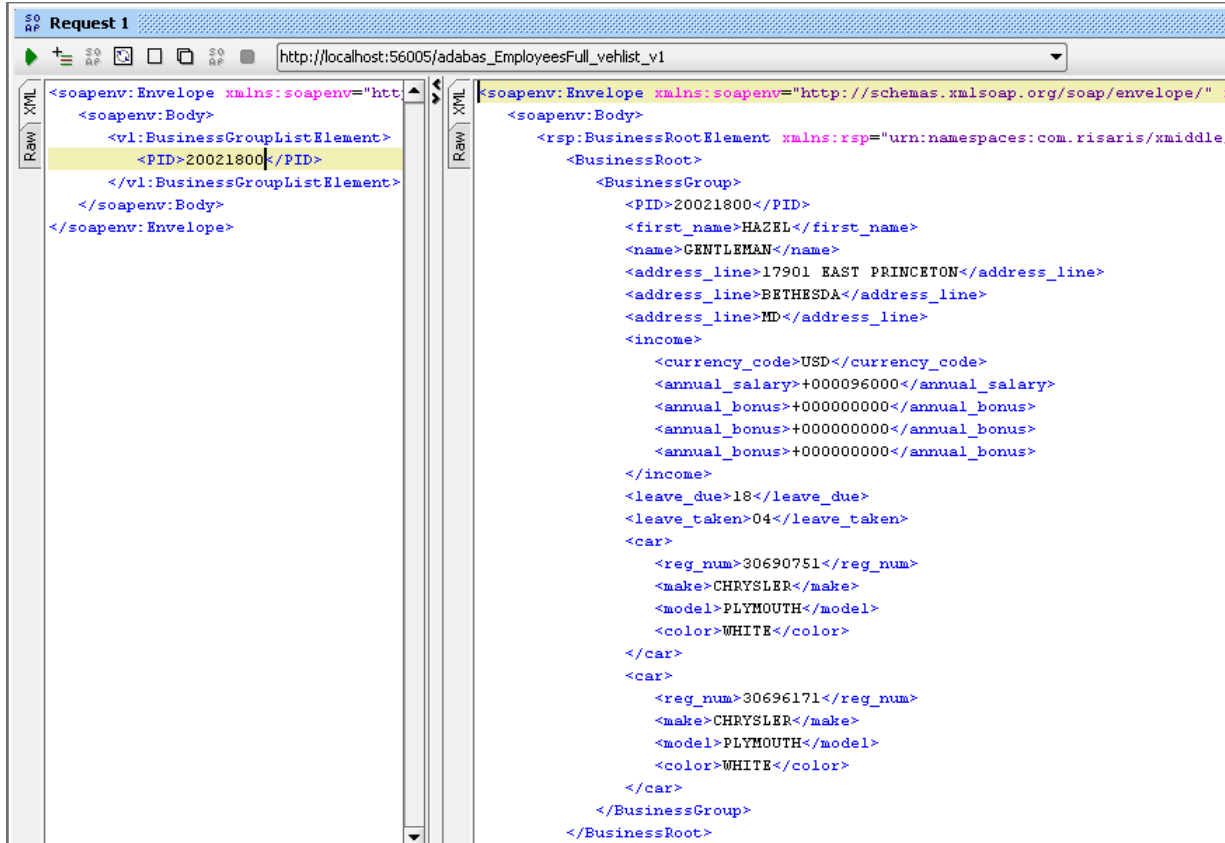
- Double-click the default request created for the *get* method, this opens up the request window.



- Remove the entire *soapenv:Header* section from the request, specify a *PID* of 20021800. Click the submit icon to send the request to the Portus server.



- The result will be a compound set of data from both the *Employees* and *Vehicles* files.

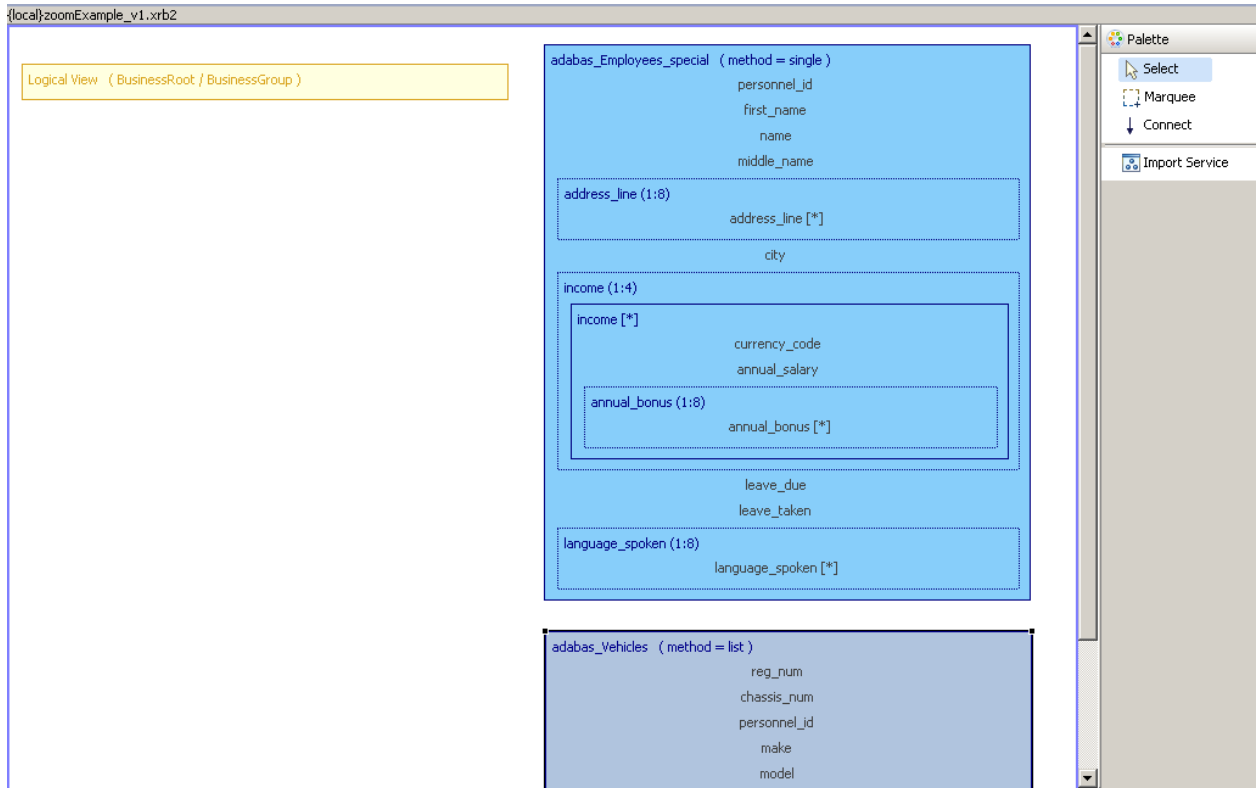


BusinessDataView Editor hints

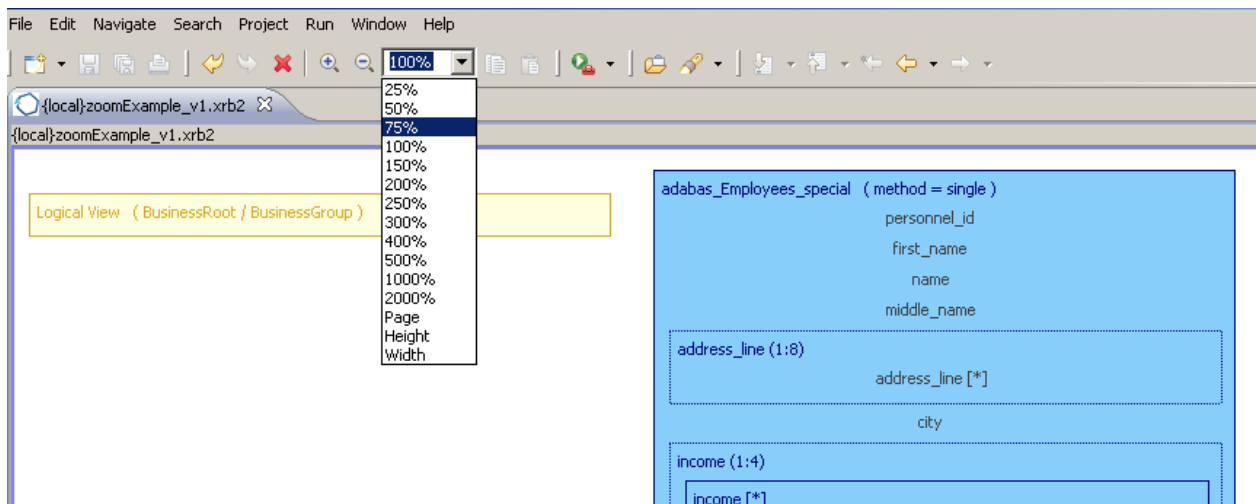
This section provides you with a number of general purpose hints for making your life easier in respect to editing BusinessDataViews.

Zooming

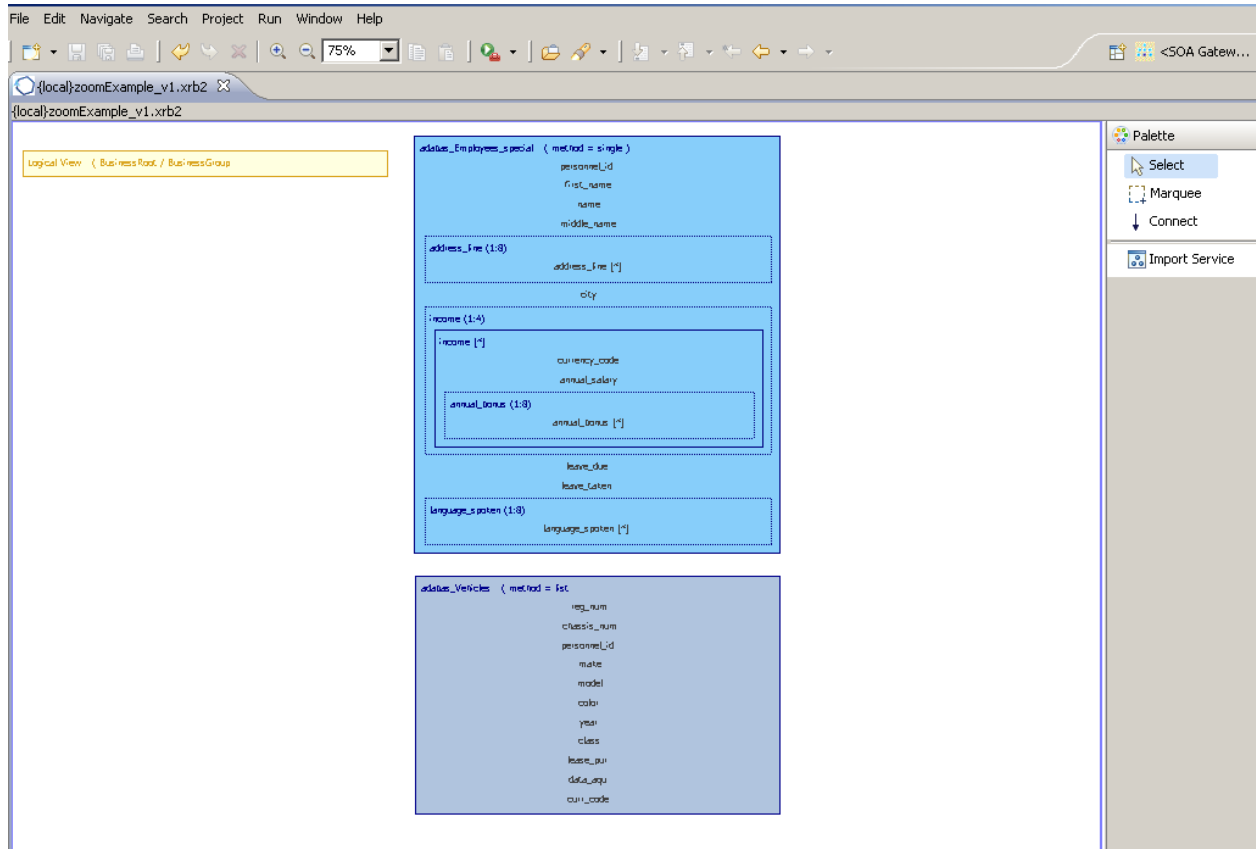
When the BusinessDataView is too big, or the screen is too small to display it in its entirety at a resolution of 100%, or if you want to see some element(s) in more detail, the zoom level can be changed accordingly.



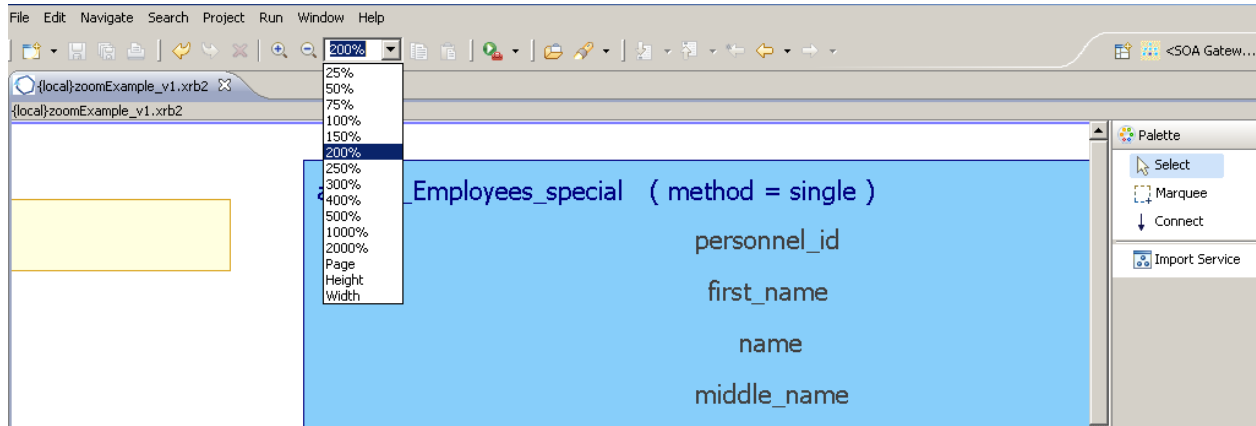
The initial display at 100%, not all details are visible.



Reduce the zoom level, from the combo box located in the toolbar area, as required to see the entire graph.



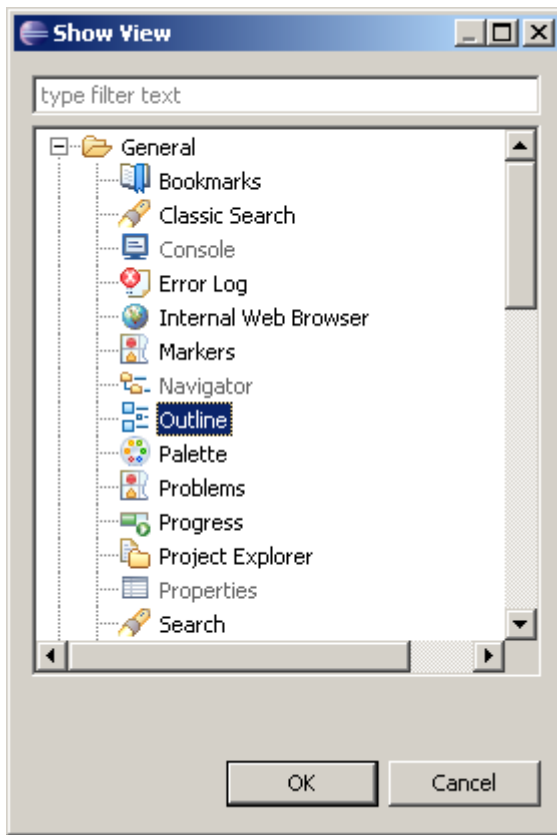
Increase the zoom level to a value above 100% to see more details.



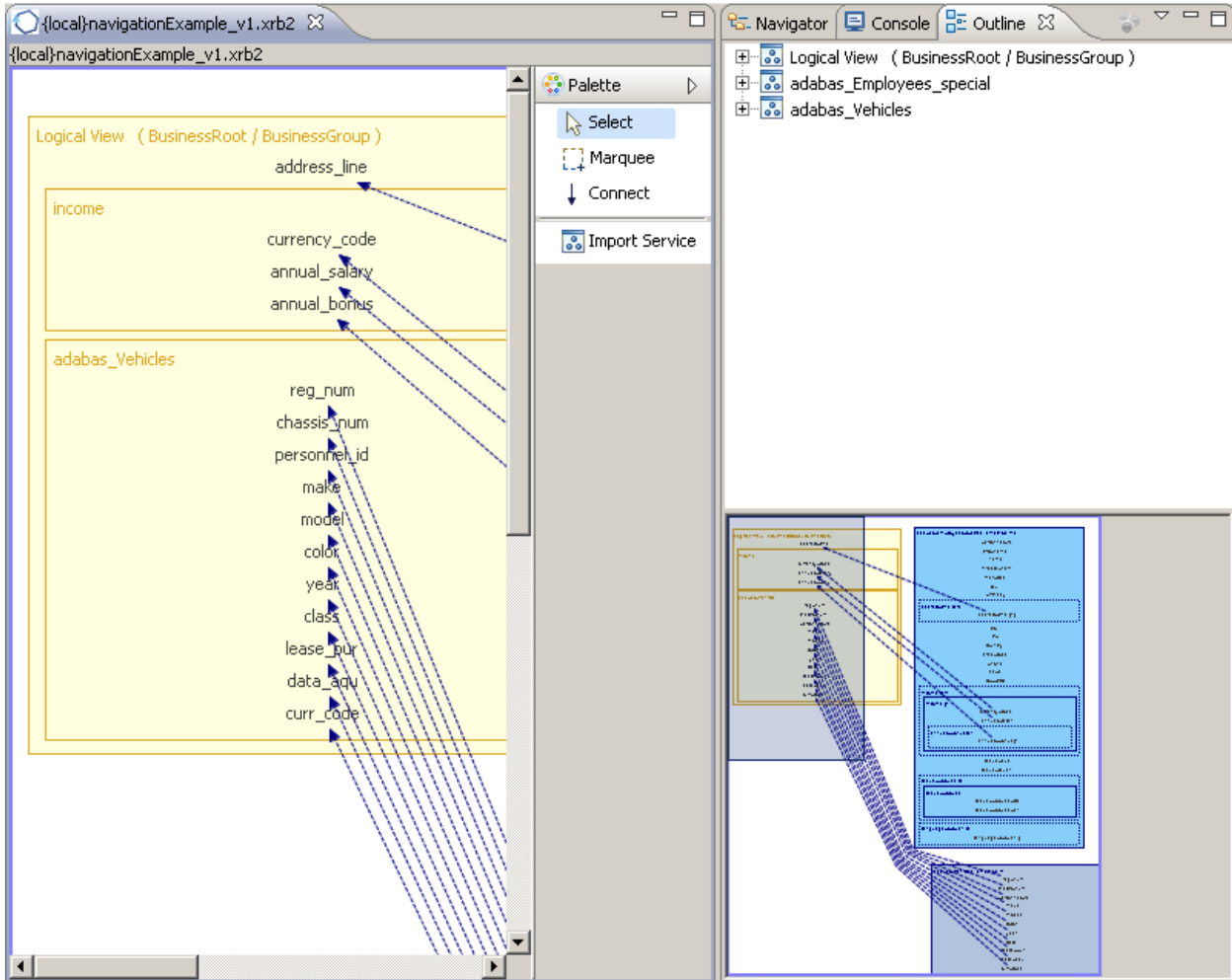
Making the BDV elements visible

In a busy BDV where there are several elements, it can be tedious to navigate using the scrollbars etc. There are 2 useful techniques available to make this easier, both using the Outline View.

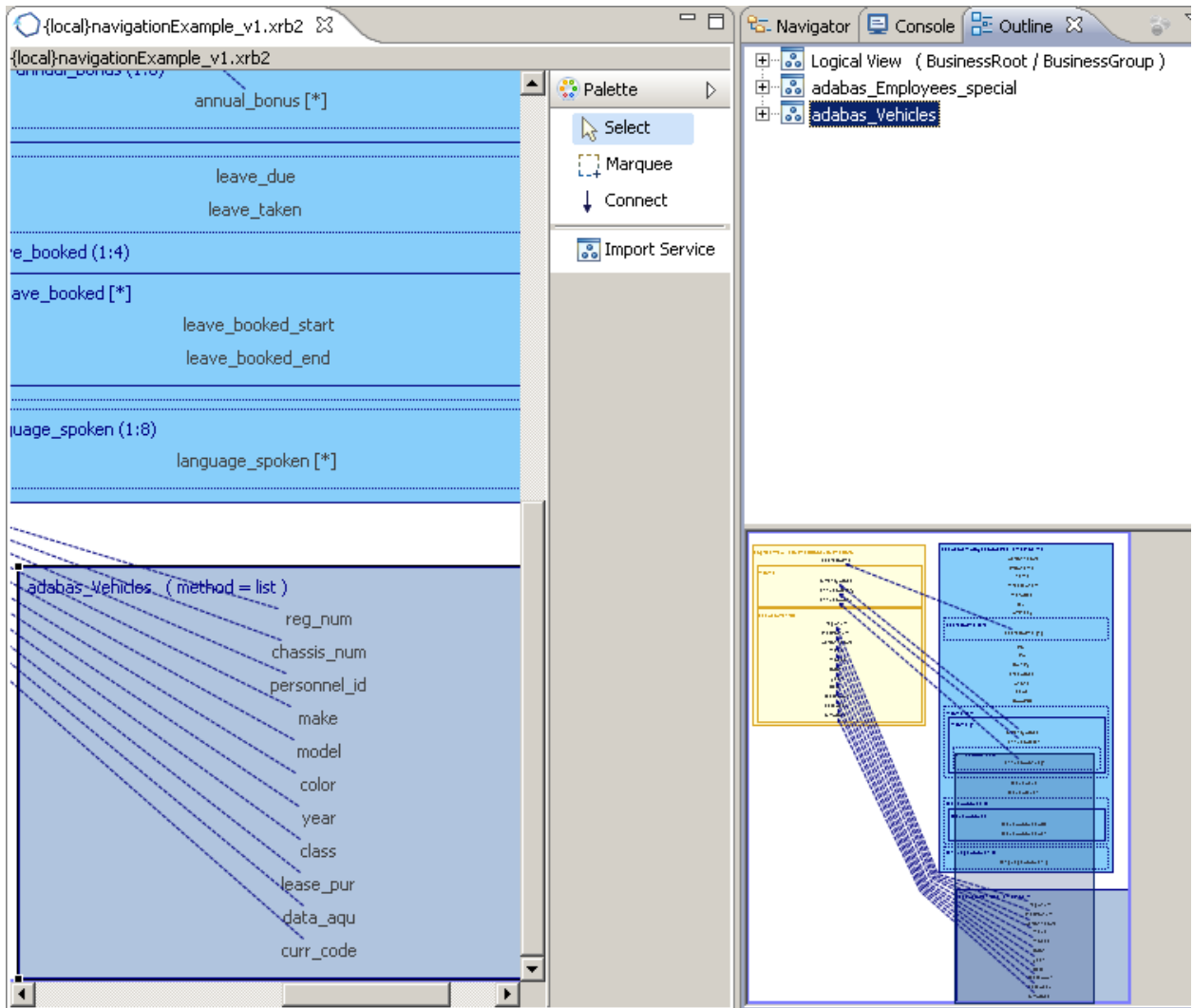
If the Outline view is not visible, from the menu select Window -> Show View -> Other..., select Outline in the General folder and hit OK.



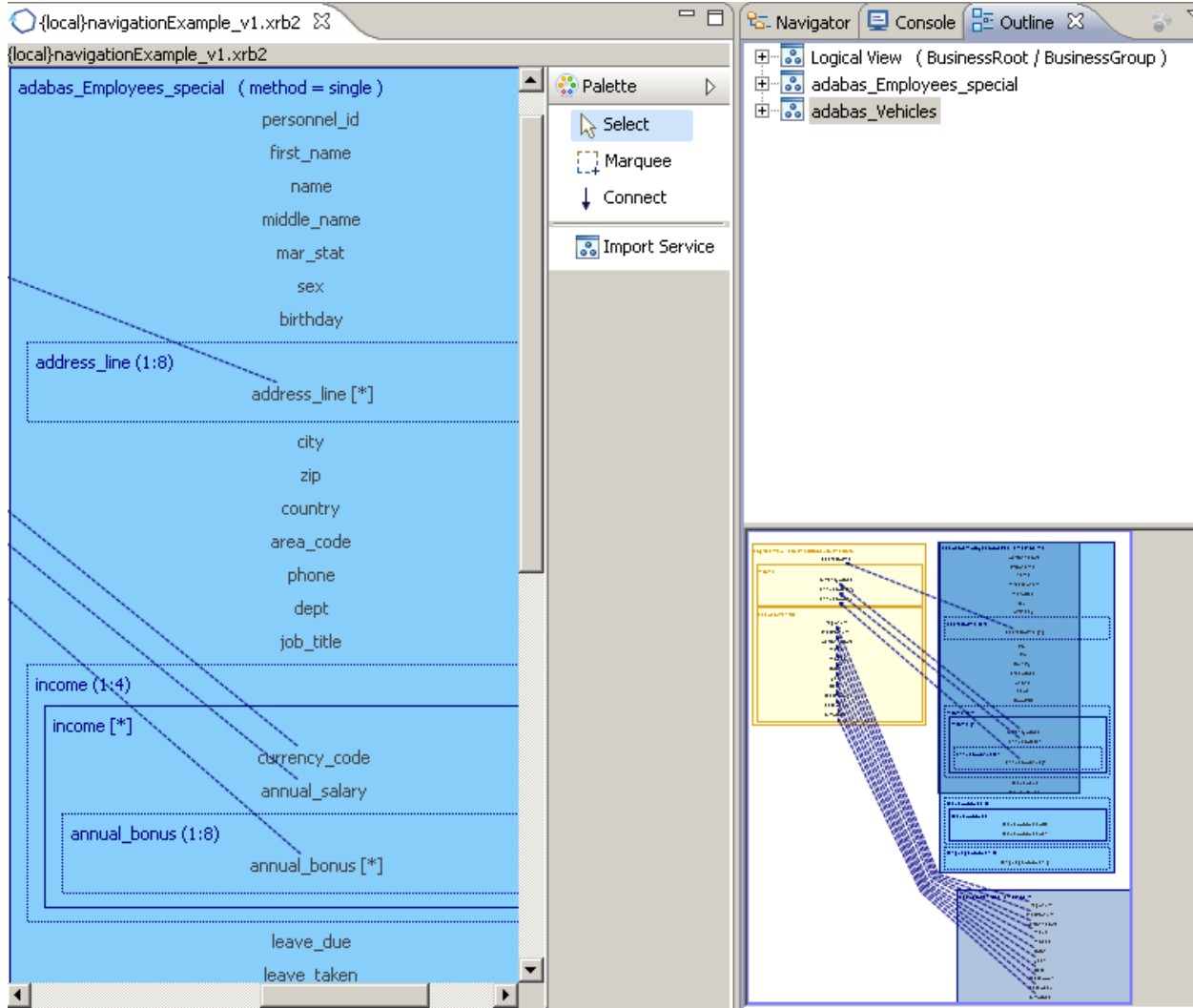
In this example we have a populated LV plus 2 PRs, with the employees PR partially visible. To make the vehicles, select the entry in the Outline view as



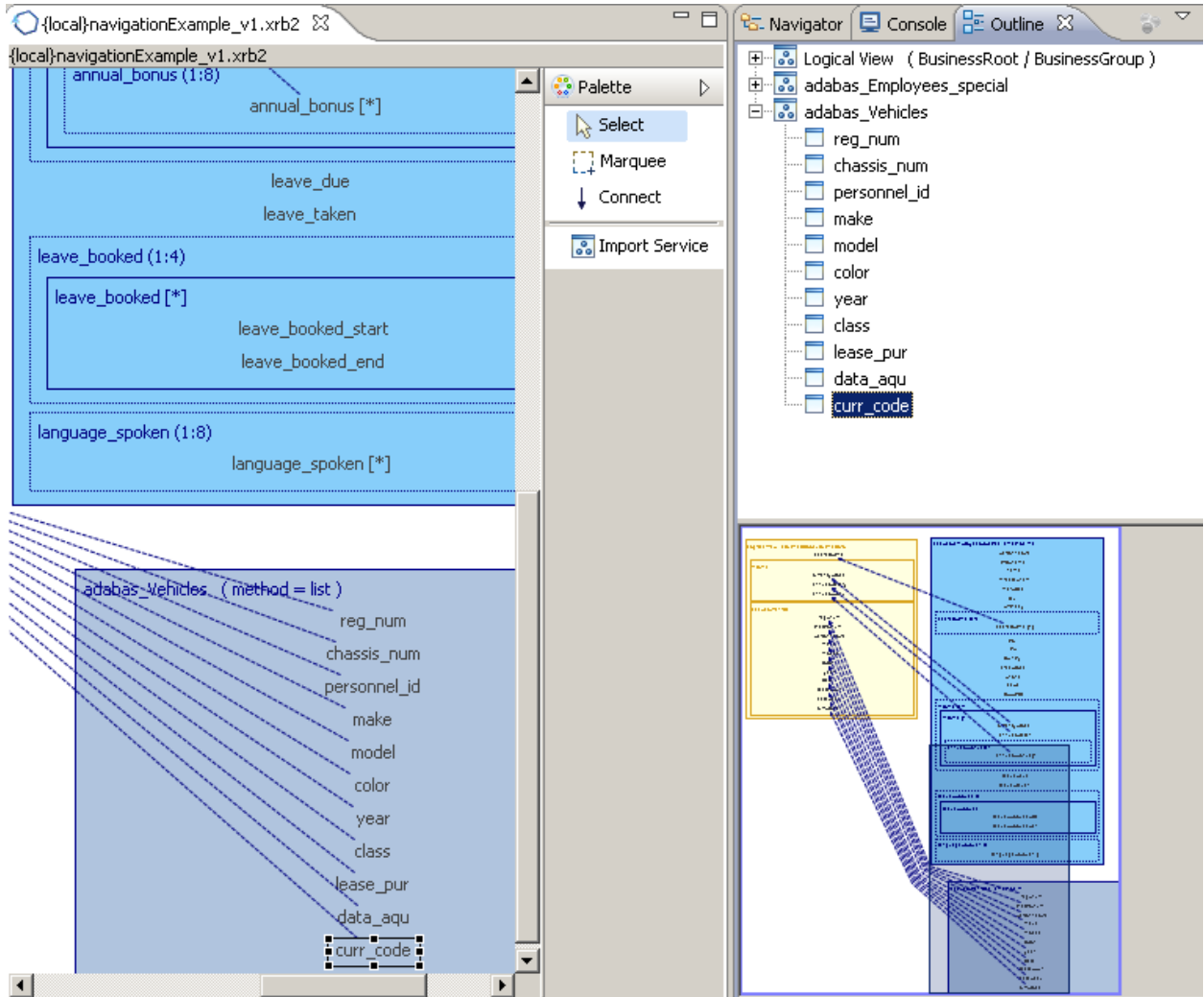
To make the vehicles PR visible, select the entry (adabas_Vehicles in this instance) in the Outline view as shown below. The view will align itself with the PR selected. Notice also how the aerial view in the bottom right-hand corner has also moved the focus to the PR, with the darker blue rectangle reflecting the view in the BDV editor.



The latter is the other way focus can be shifted around the BDV editor. Select the darker rectangle and move it around its area. The moves will again be reflected in the editor.



Finally the view can be more fine grained. Open a tree view in the Outline view and select an individual field in the PRs or LV. The focus will shift to that selected.



When a Physical View changes

Or more precisely, when the DataView associated with an imported WebService changes.

Currently there is no way of automatically reacting to changes in underlying WebService DataViews, so when new fields are added which are to be used in the Logical View, there is no way of updating the imported Physical View within the BusinessDataView.

The only way to deal with such a situation is to remove the Physical View from the BDV, re-import it, copy the required fields to the Logical view and recreate the links between the Physical Views.

Following versions of the Portus Control Center will provide functions to capture changes to the underlying DataView(s) without having to recreate the Physical View and the links from scratch.

16 The Portus Control Center Action Log

- Working with the Action Log 140

The Action Log View displays information and error messages related to the 'client' actions carried out within your Control Center session, it does NOT show any server messages, these are available through the Apache log-files on your server(s).

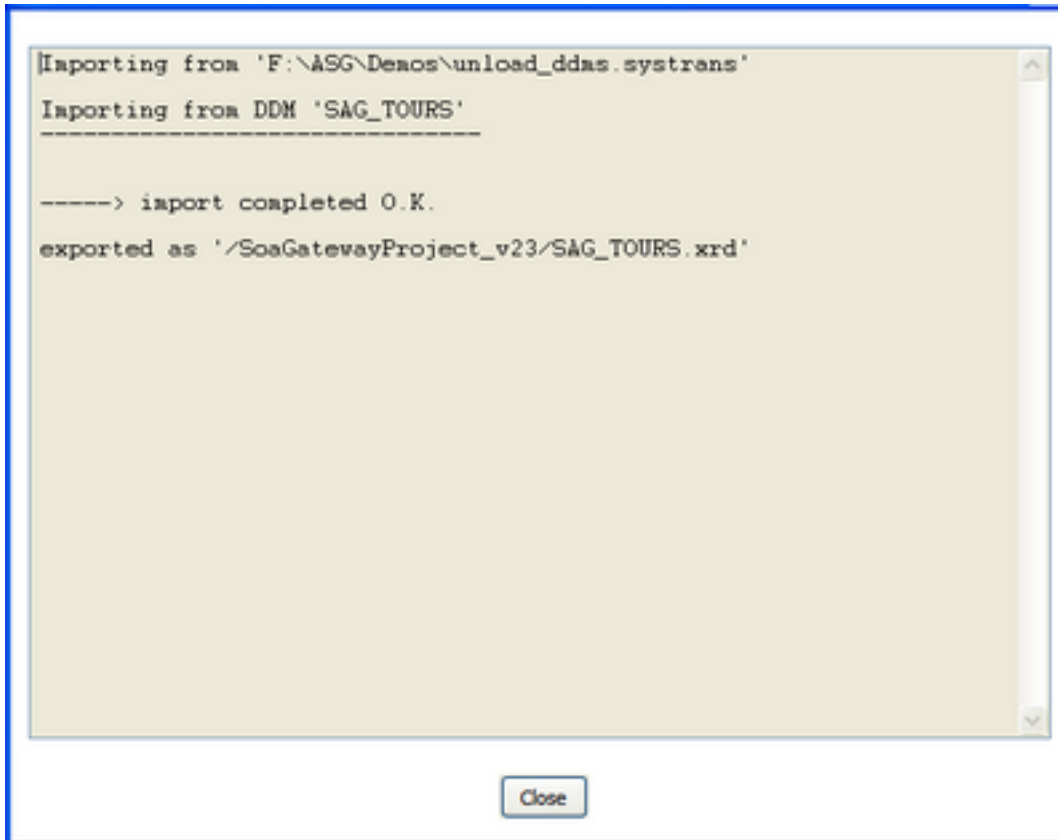
Working with the Action Log

These are the columns visible in the Action Log View

- The first column displays an icon regarding the message severity (information, warning, error).
- Message directly related to an action targeted at a specific server display the server name.
- The message text
- The unnamed column between 'Message' and 'Timestamp' indicates if more information is available for a given message. If this column shows a '+' (plus sign), right-click the message entry in question, then select "Show Log Entry Details" from the context menu.

Server name	Message	Timestamp
localhost	Connection lost - server unreachable !	Wed Aug 08 15:33:46 BST 2012
localhost	Configuration refreshed	Wed Aug 08 15:32:43 BST 2012

- This will display the detailed information or error description in a popup-window.



Click the 'Close' button to dismiss the window.

17

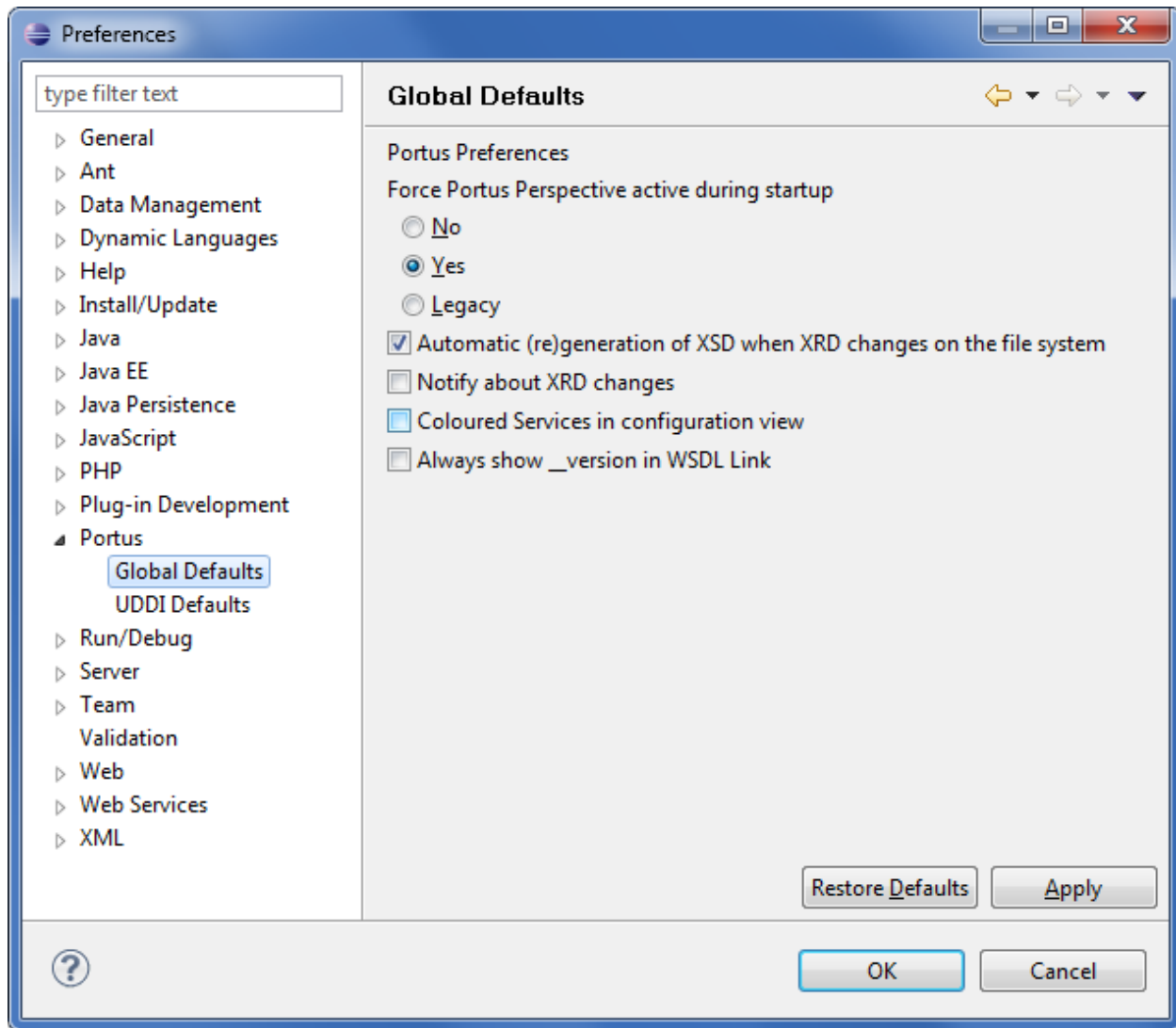
Setting Preferences for the Portus Control Centre

- Working with the Preferences 144

Preference switches are available to tailor the Portus Control Centre to your needs.

Working with the Preferences

Invoke the preferences via Window->Preferences...



- "Force Portus Perspective active during startup" can be either
 - *No*: Eclipse will be started with the state restored to what it was when last exited.
 - *Yes*: The Portus Administration perspective will be activated when Eclipse is started
 - *Legacy*: The Portus "Legacy" perspective will be activated when Eclipse is started
- "Automatic (re)generation of XSD when XRD changes on the file system.

- "Notify about XRD changes" informs you whenever an XRD is added to or changed within your Eclipse workspace.
- "Coloured Services in configuration view" assigns different colours to the various Service types (Adabas, Natural, ODBC, ..) for better distinction
- "Always show _version in WSDL Link" shows the Service version number for better distinction

Click "Apply" and/or "OK" to save your preference changes.

18

The Portus (legacy) Configuration View

▪ Working with the Configuration View	148
▪ Filter displayed Services	150
▪ Edit a Service Definition	151
▪ Add a Service	151
▪ Duplicate a Service Definition	153
▪ Duplicate a Service Definition to other server	153
▪ Remove a Service from the server configuration	254
▪ Import a DataView	155
▪ Export a DataView	156
▪ Generate Java Wrapper for a Service	157
▪ Register a Service with an UDDI Business Registry (UBR)	157
▪ Create a Resource	158
▪ Delete a Resource	159
▪ Replicate a Service	160
▪ Refresh a Service	263
▪ Import Service Definition files	163

The Configuration View provides context based interfaces to

All Service related functions. A Service is the Portus definition exposing an Adabas file, ODBC table etc. to the world..
The DataViews (XRDs) , XML schema files (XSDs) and Stylesheets (XSLs) available on the target server.
The BusinessDataViews (BDVs) available on the target server.



Note: The Configuration view is part of the Portus "legacy" perspective. This perspective is superseded by the new "Portus perspective", you are strongly encouraged to familiarise yourself with, and start using the new perspective as early as possible, only the latter will be enhanced further some features are available in the Portus perspective *only*.

Working with the Configuration View

Right-click on any Service name in the Server Configuration View's 'Services' tab to bring up the context menu.

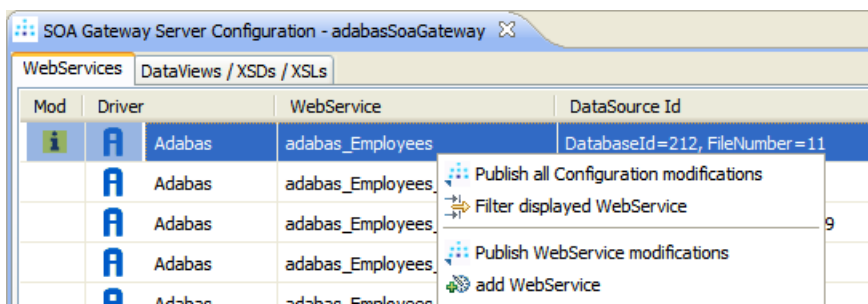
In addition to the context menu, a number of functions are also available from the Views toolbar.

Note: All functions will make 'published' changes to Service(s) available within a running Portus server immediately, but these changes are only preserved over a server restart when the configuration is either saved manually, or the server is defined with the "autosave" option.

Function	Description
Refresh Service	'Refresh' a Service to purge its cache entry from the server.
Edit DataView	Edit the DataView associated with the selected Service.
Import DataView	Import a DataView (Service data layout description)
Export DataView to server	Export a DataView (Service data layout description)
Filter displayed Service	Set filter criteria for the Configuration View Service display.
Add Service	Add a new Service
Duplicate Service	Duplicate a Service, use definitions of an existing Service as the basis for one to be created.
Rename Service	Rename a Service..
Copy Service definitions to other server(s)	Copy selected Service(s), including all associated resource files (XRD / XSD / XSL), to one or more other server(s).
Remove Service	Remove a Service from a Portus Server Configuration
Generate wrapper code	Generate a Java wrapper, based on the Axis2 framework, for a Service.
UDDI Register Service (via JAXR or SOAP)	Register a datasource - a Service - with a UDDI Business Registry

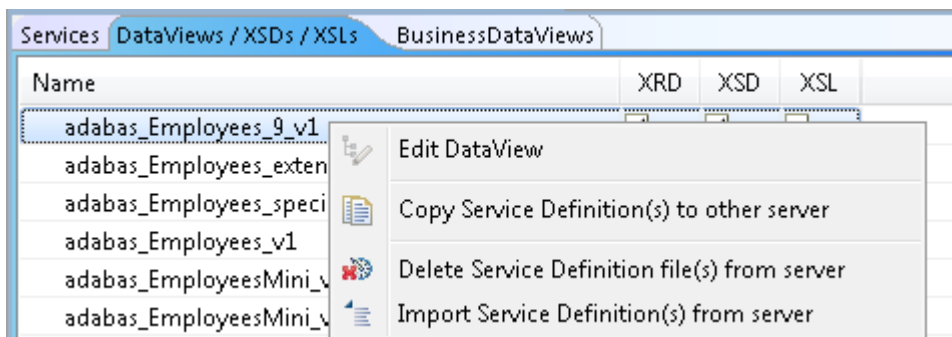
Function	Description
Show WSDL	Opens the WSDL for the selected Service in a browser window.
Create PHYSICAL Resource	Create a Resource (Adabas file, ODBC table etc.), physically, based on the Portus Definitions.
Delete PHYSICAL Resource	Physically delete a Resource (Adabas file etc.) pointed to by a Portus Service.
Replicate Service	Duplicate the Service definition(s) and (optionally) create a physical copy of the original Service's resources.

When parameters of a Service definition have been changed, the context menu will change as follows



Function	Description
Publish all configuration modifications	Send ALL modifications for all changed Services to the server at once
Publish Service modifications	Only modifications for the selected Service are published, i.e. sent to the server.

The second tab, 'DataViews / XSDs / XSLs' provides an import function for these Service Definition file types, more information is given [here](#).



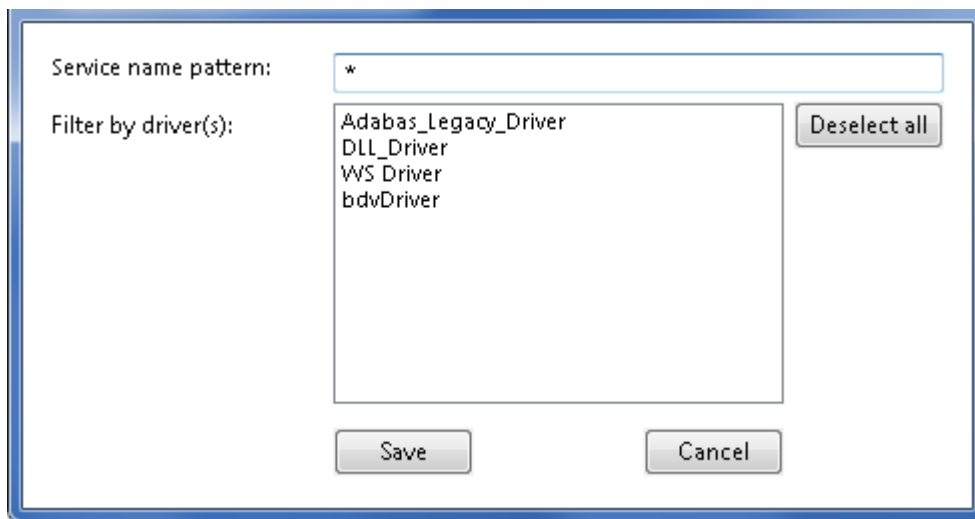
Filter displayed Services

The Service filtering function allows the display to be limited by Service name and/or Service type.

1. Select the filter function from either the context menu or View toolbar



2. Enter a Service name pattern and/or select one or more drivers from the displayed list to limit the Service display to those matching the criteria.



Valid Service name patterns:

- A fully specified name to limit to a single Service
 - *<Service name pattern> for an 'ends with' condition (e.g. *ees will match 'Employees', 'Trainees' etc.)
 - <Service name pattern>* for a 'starts with' condition (e.g. ada* will match 'adabas_Employees', 'adatest' etc.)
 - *<Service name pattern>* for a 'contains' condition (e.g. *lo* will match 'Employees', 'globals' etc.)
3. Click the 'Save' button to apply your selection, or 'Cancel' to leave without changing the previously selected criteria.

Edit a Service Definition

Click on the Service name in the Configuration View, modify the Service definition in the Properties View. Modified Services will be indicated by an information icon in the 'Mod'(ified) column. Modifications stay local until one of the **publishing functions** is executed.



Important: Changing the Service Name will rename the Service.

Add a Service

Select the **Add Service** function from the context menu.

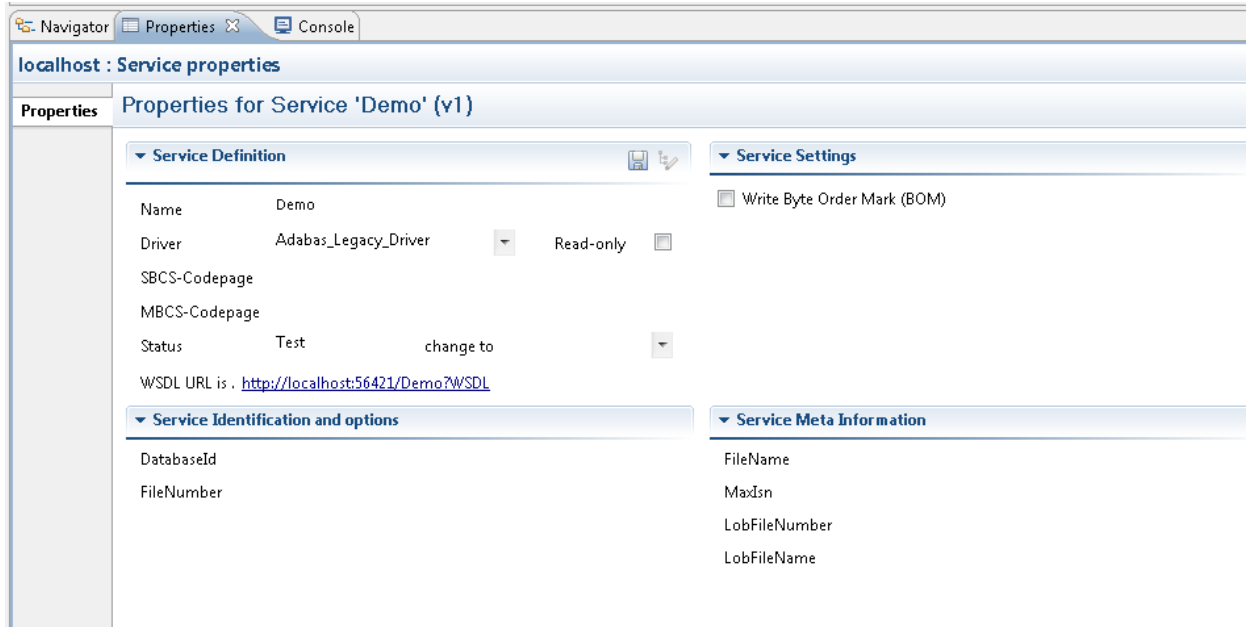
You will be asked to enter a name for the Service and to select a driver. Select Create.

Enter name for new Service

Use driver: DLL_Driver

Create Cancel

A new Service will be added and the properties can be modified as appropriate. In this example the properties are those which apply to a Service based on an Adabas driver. These properties will differ according to the driver used.



The following definition elements are required to describe an Adabas file to Portus:

1. Service Definition:

Name: The Service name which is used to identify the Adabas file to the outside world.

A Service is accessible by its URI `http://<your_server>/<Service_name>` When retrieving the WSDL for a Portus URI, append the string `"?WSDL"` to the above URI.

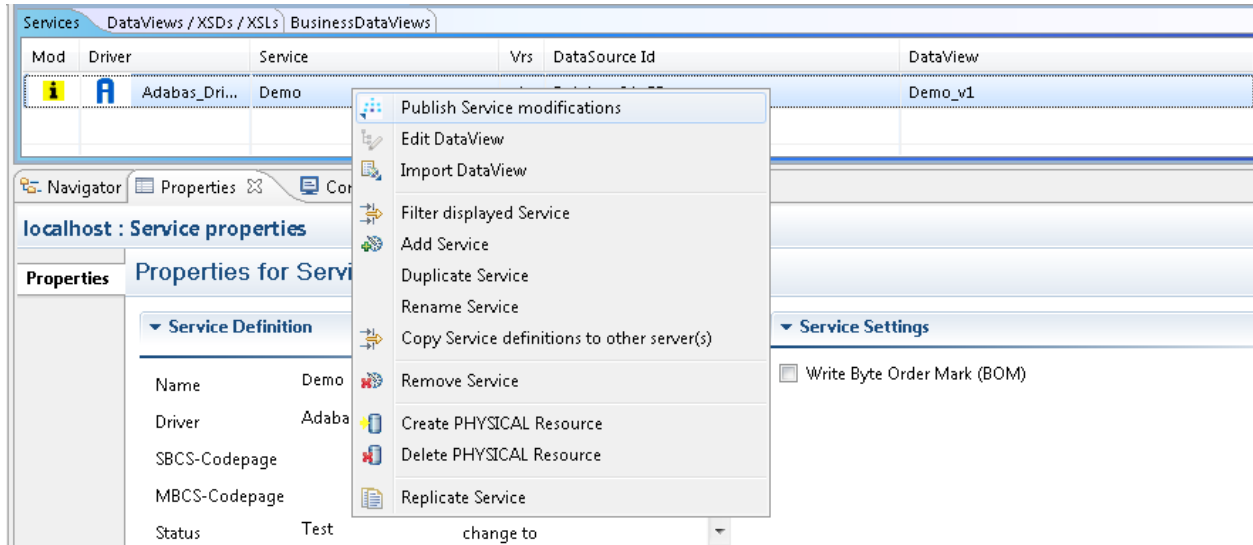
The URI is preformatted in the 'WSDL URL is' field, either select it, then right-click and copy it, or double-click it to open a browser window and display the WSDL.

2. Service Identification and options:

provide the actual pointer to the Adabas file being accessed through the "Database Id" and "File Number" parameters.

3. Service Meta Information:

The Adabas "FileName" and "MaxIsn" fields in the 'Service Meta Information' section are used for the "Create Service" function which physically creates the Adabas file, based on the Portus DataView definitions, on the Dbid / File number specified for the Service.



When modifications are made an icon will appear as shown. Right-click on the Service and select Publish service modifications to save the changes made.

Duplicate a Service Definition

Select the **duplicate Service** function, this will create a Service named "Copy_of_<original_Service_name>" and bring up a the Properties for the newly created item, prefilled with the definitions of the Service specified as the input for duplication. The Service name will be preset to copyOf_<original_Service_name>. Make changes as required and **publish** the duplicated Service.

Duplicate a Service Definition to other server

Select the **Copy Service definitions to other server(s)** to transfer entire Service definitions, including their resource files (DataView / XSD / XSL), to one or more other servers.

Select the target server(s) and click **OK**.

Copy from: localhost (localhost:56421) - v4.2.1.002000 (Jul 12 2012 / 12:50:42)

Copy to: cloud (46.4.117.234:56421) - v4.2.1.002000 (May 31 2012 / 11:05:52)

Overwrite existing definitions on target server(s) unconditionally
 Do not execute copy function, generate batch command file instead

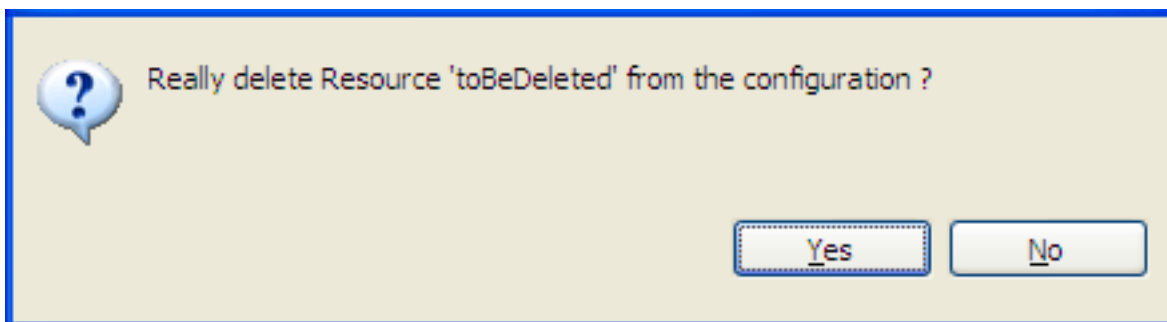
Successful execution of the copy operation will be indicated by a message in the Portus Action Log


Double-click the log entry to see the details

When a copy operation is initiated for a Service using a driver name not known on the target system, a prompt will offer a list of drivers of the same type, select one and click **Use**, when **Cancel** is selected the current Service will not be copied, this fact is also documented in a Portus Action Log entry.

Remove a Service from the server configuration

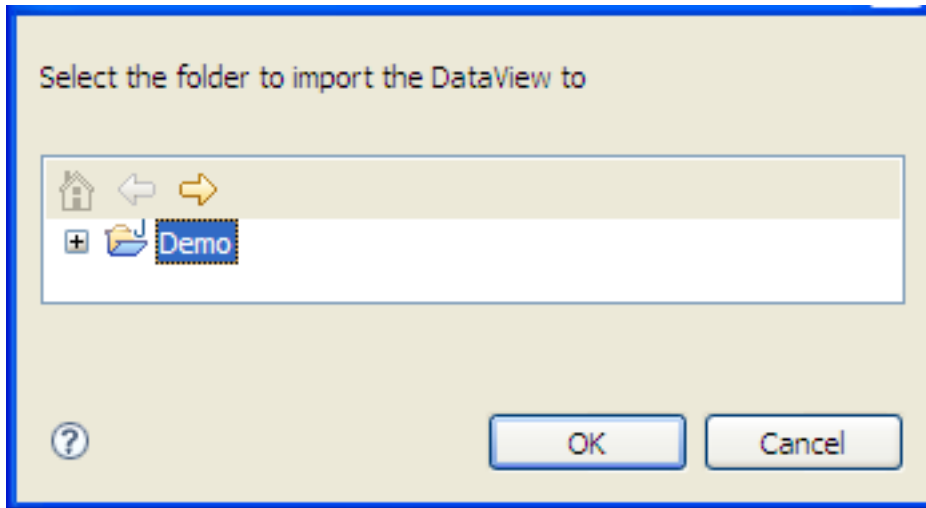
Select the **Remove Service** function from the context menu to delete a Service from the server configuration. You will be requested to confirm the deletion:



 **Important:** The Service is only deleted from the configuration currently active ("live") within the Portus server, to remove it permanently the configuration has to be saved (written to disc).

Import a DataView

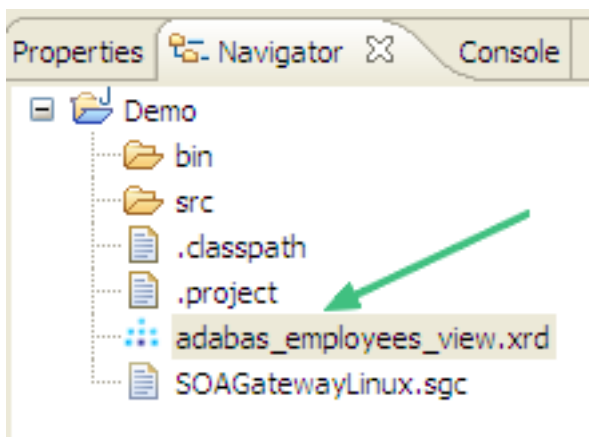
Select the **Import DataView** function from the context menu to import the physical layout description (DataView) into the Eclipse workspace for editing, archiving etc.



Select the workspace folder where the imported DataView is to be stored, click **OK**.

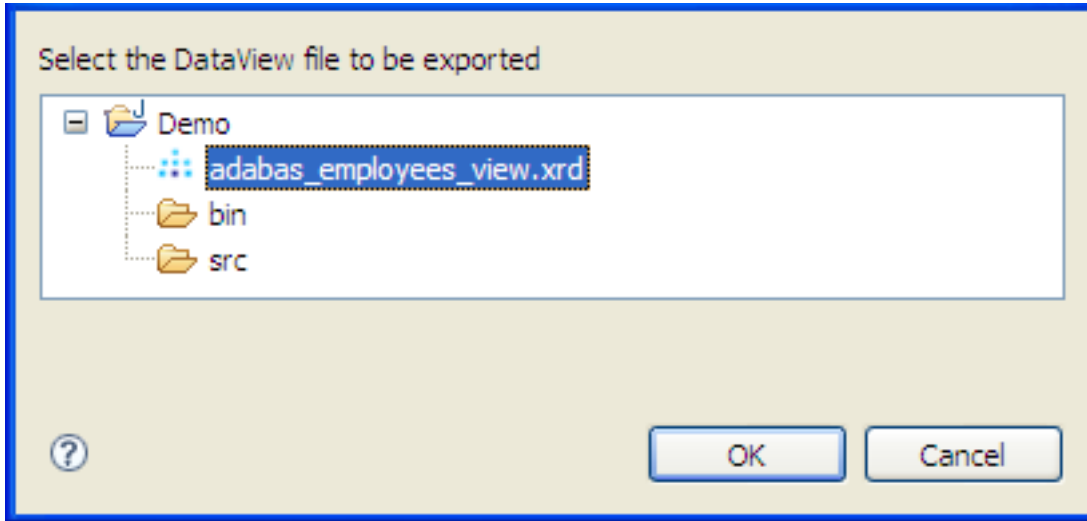
The import will be confirmed by a message in the Status pane.

The imported DataView appears, with an extension of ".xrd", at the selected location:



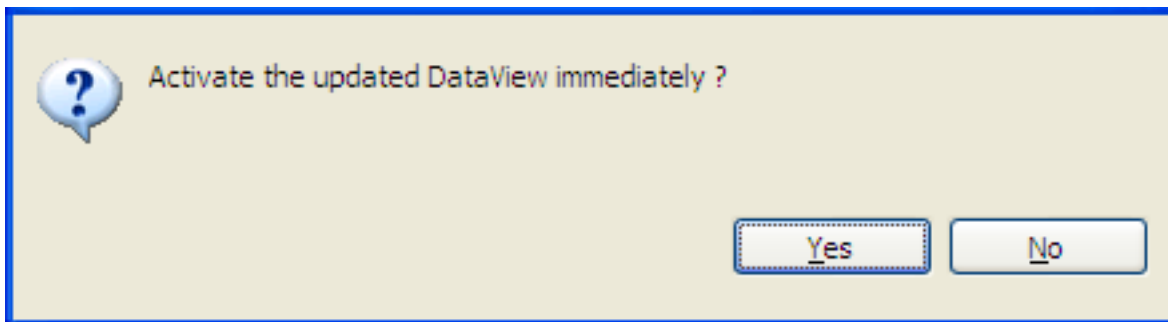
Export a DataView

Select the **export DataView** function from the context menu to export the physical layout description (DataView) to the Portus server.



Select the DataView (.xrd) file to be exported, click **OK**.

You are now asked if the exported DataView is to be activated immediately (answer **Yes**) or after the Portus server is restarted (answer **No**)



The export will be confirmed by a message in the Status pane:

Resource 'adabas_Employees' DataView exported from /Demo/adabas_employees_view.xrd

Generate Java Wrapper for a Service

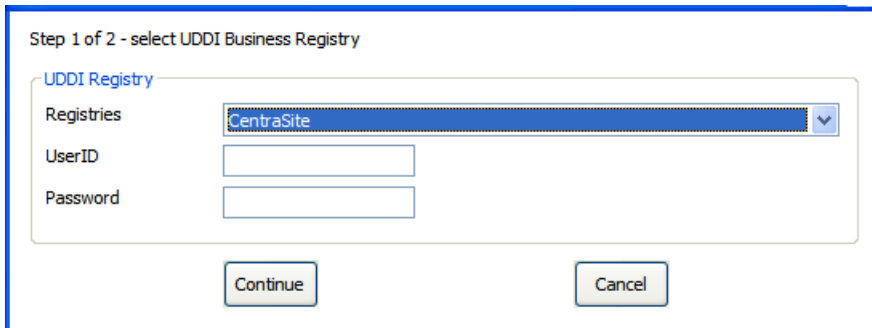
Java wrapper/stub classes are generated using the [Apache Axis2](#) feature [WSDL2Java](#).

Register a Service with an UDDI Business Registry (UBR)

Portus Services, which are essentially "Services", can be registered with an UDDI Business Registry, for example Software AG's Centrasite, so that any UDDI enabled client application can find an exposed service, retrieve its signature, issue requests against it etc.

These are the steps required to register a Service with an UBR:

1. Define a UDDI Business Registry to the Portus Control Centre (from the Portus Eclipse Preferences dialog)
2. Select the **UDDI register Service** function from the context menu
3. Select one of the predefined UDDI Business Registries, enter the user credentials required to be able to publish



Step 1 of 2 - select UDDI Business Registry

UDDI Registry

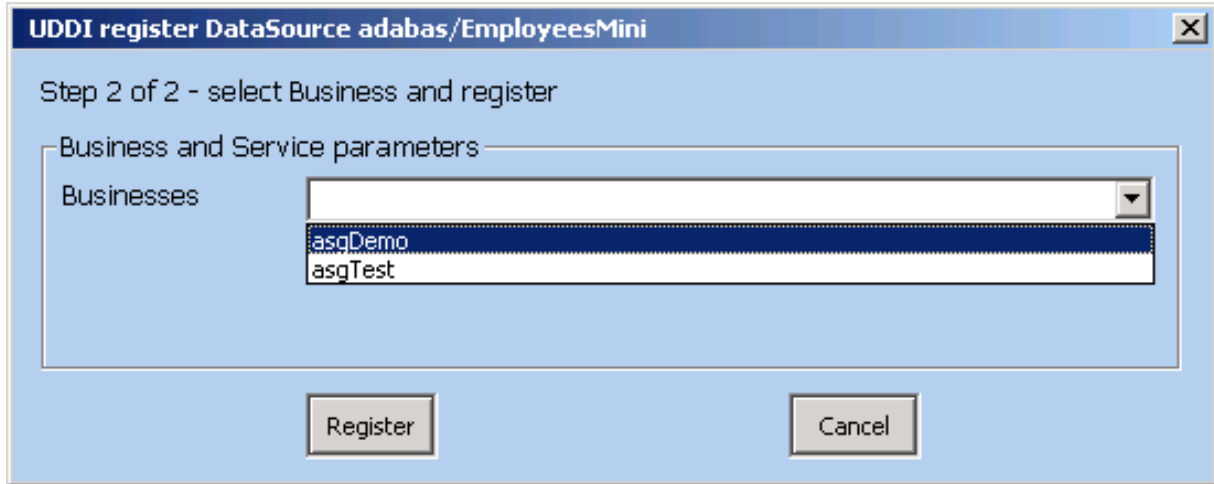
Registries: CentraSite

UserID:

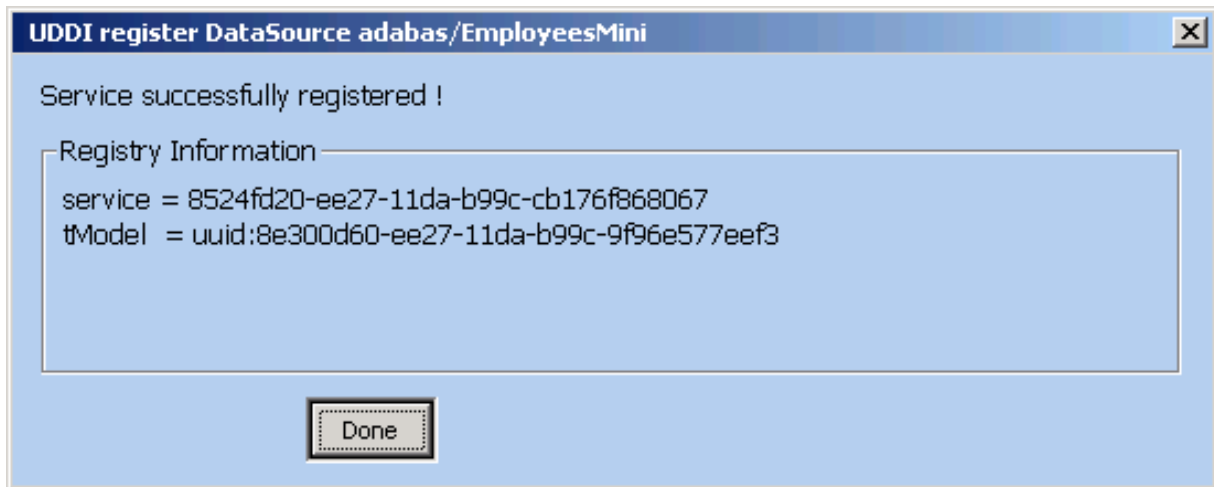
Password:

Continue Cancel

4. Select a Business registered with the UDDI Registry for the selected account, click **Register** to continue



5. After successful completion of the registration process the service and tModel registry keys will be shown



6. Click **Done**

Create a Resource

A physical Adabas file can be created from a Portus Service definition.

For this to work, the Service in question must have the "File-name" and "Max. ISN" parameters set, the "DbId" and "FileNr" parameters will be used.

WebService Meta Information	
File-name	myEmployees
Max. ISN	1000

Select the **create Resource** function from the context menu, successful creation will be indicated by a message in the status line

Resource 'copyOf_adabas/Employees' created

The ADABAS file has been created on the target database.

```

%ADAREP-I-STARTED,      03-JUN-2006 18:58:18, Version 5.1.1.03(03) (Linux 32Bit)
%ADAREP-I-DBON, database 5 accessed online
Database 5, File   99 (Employees      )           3-JUN-2006 18:58:18

Highest Index Level:      3      Padding Factors:      ASSO  5%, DATA  5%
Top ISN:                  0      Maximum ISN expected:      2,047
Records loaded:          0
  
```

Delete a Resource

A physical Adabas file linked to a Portus Service can be deleted from a Portus Control Centre.

Select the **delete Resource** function from the context menu, you will be asked to confirm the deletion

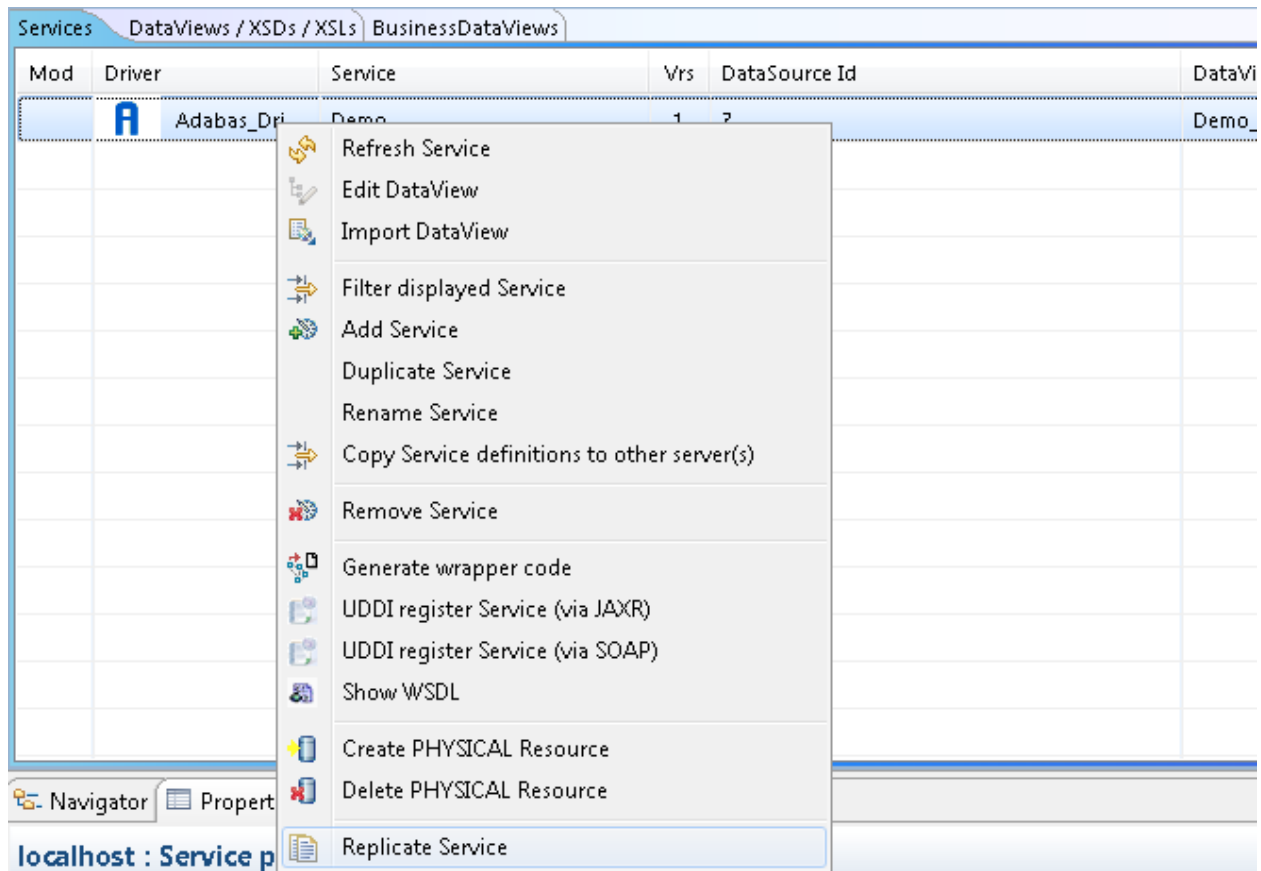
As the Resource (the Adabas file) will be deleted physically from the database, you will be asked to confirm a second time.

Successful deletion will be confirmed by a message in the Status pane.

Replicate a Service

A physical copy of a Service's resources (for example an Adabas file) can be created based on the Portus definitions.

Select the **Replicate Service** function from the context menu



You will be asked on which server the copy will be created, and which driver is to be used to create the copy. Click **Next** after having selected the required information.

Step 1 - Select target Server and Driver

Copy to server: [Dropdown menu]

And use Driver: [Dropdown menu]

Replication commencing for service 'Demo'
... DataView found - will be copied

Next > Cancel

Specify all required parameters required to define the Service on the target server. Check the "Replicate data as well ?" box in case you want the Service's data to be copied in addition to the definition. Click **Next**

Step 2 - Specify target Service options

Copy to server: localhost (localhost:56421) - v4.2.1.002000 (Jul 12 2012 / 12:50:42)

And use Driver: Adabas_Driver

Name of replicated Service: Demo_replicated

Replicate data as well ?

Replicated Service Identification:

DatabaseId: [Text box]

FileNumber: [Text box]

Replicated Service Meta Information:

FileName: [Text box]

Maxdsn: [Text box]

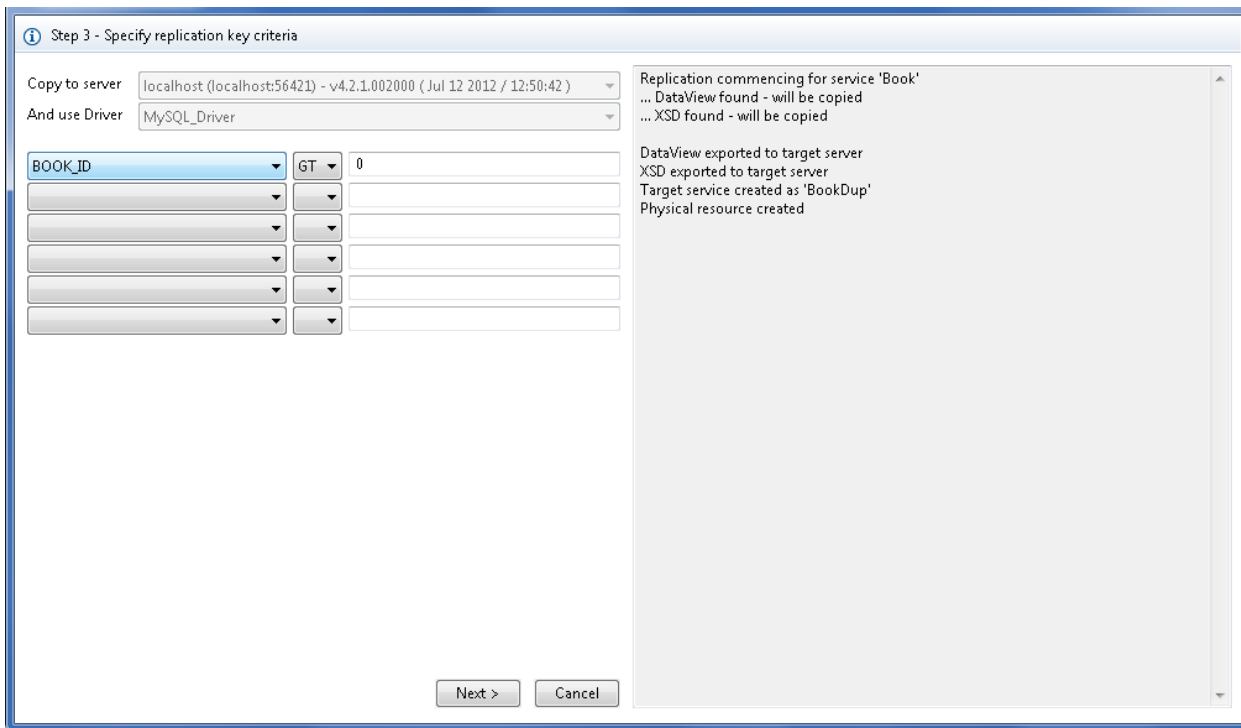
LobFileNumber: [Text box]

LobFileName: [Text box]

Replication commencing for service 'Demo'
... DataView found - will be copied


Next > Cancel

Specify key criteria, this allows you to specify if all, or just a subset, of the data on the "source file" is to be copied. In this example, to copy all with BOOK_ID > 0:



Click **Next**

The next dialog indicates the amount of data to be copied. Start the copy process by clicking **Replicate**, or click **Cancel** and either refine the key criteria, or abandon the copy altogether.

 **Note:** At that point the physical Adabas file has already been allocated on the target database, and will not be deleted automatically, even if the actual data replication is not carried out.

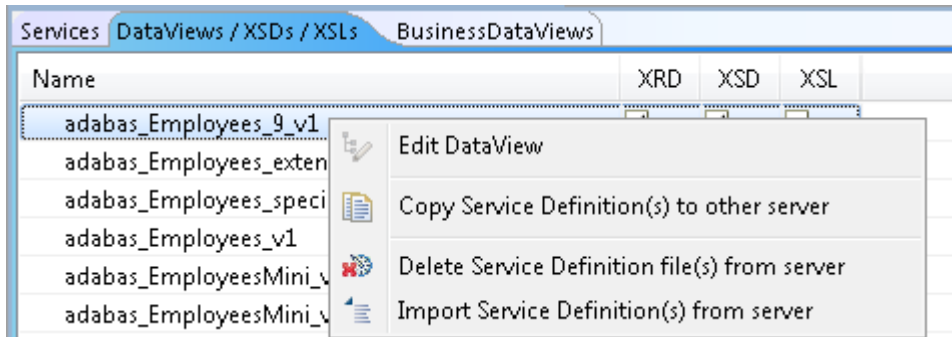
At each stage messages are displayed in the right-hand pane. Click **Done** to dismiss the Replication dialog.

Refresh a Service

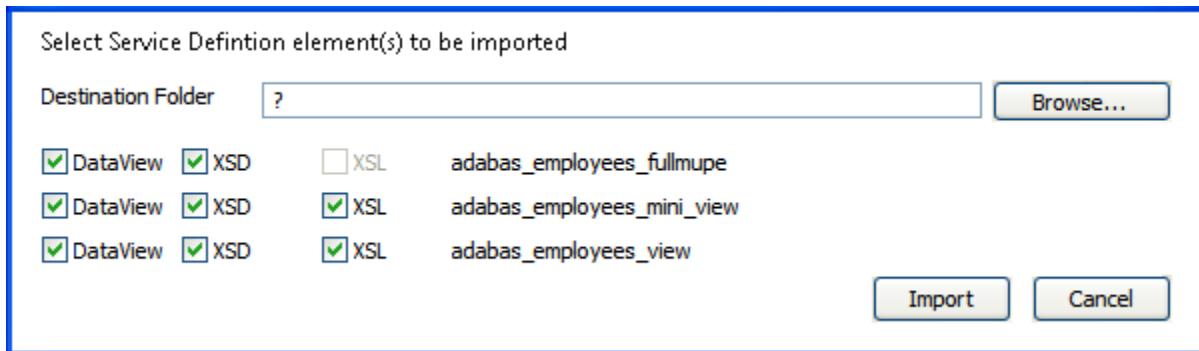
When modified Service definitions files (DataView or XSD) are exported to the Portus server, Service(s) using these file(s) may need to be 'refreshed' on the server to pick up the modifications.

Select the **Refresh a Service** function from the context menu.

Import Service Definition files



When one or more Service Definition elements are selected on the 'DataViews / XSDs / XSLs' tab, the 'Import Service Definition(s) from server' context function, the following import detail dialog will be presented:



- Specify the destination folder, or click the 'Browse' button to select it from a file-chooser.
- Select the DataView(s), XSD(s), XSL(s) to be imported.
- Click 'Import' to execute the import.

19 Service Configuration

▪ Service Properties	167
▪ Service context menu	168
▪ Lifecycle Governance	171

When a driver entry is expanded, the list of services associated with this driver is also displayed.

Left-clicking the web service name brings up the [Service Properties](#) in the Properties view.

Right-clicking the web service name brings up the [Service Context Menu](#)

This section also provides information about [Lifecycle Governance](#) (LG)

Service Properties

local : Service properties

Properties Properties for Service 'adabas_Employees_extended' (v1)

Service Definition

Name: adabas_Employees_extended

Driver: Adabas_Driver Read-only:

SBCS-Codepage:

MBCS-Codepage:

Status: Test change to:

WSDL URL is .. http://localhost:56005/adabas_Employees_extended?WSDL

Service Settings

Write Byte Order Mark (BOM)

Service Identification and options

DatabaseId: 1

FileNumber: 9

Service Meta Information

FileName:

MaxIsn:

LobFileNumber:

LobFileName:

Usage Governance and Statistics

Service Statistics

Use service overrides Use global settings

Information Input Data Output Data

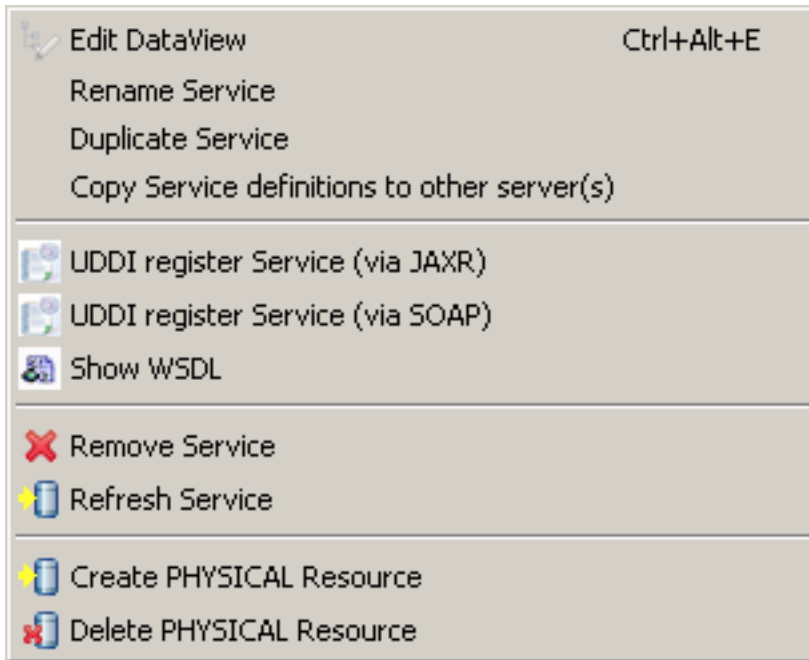
When a web service is selected, the following information is displayed.

- Name : The name of the web service. This will determine the WSDL URL and service endpoint.
- Driver: The driver in use for this web service.
- SBCS/MBCS : The SingleByte/MultiByte codepages used for data conversion with this service.
- Service Status: For more details see [here](#)
- Service Identification. These fields vary on the type of service. They define the resource-specific parameters for this service.
- Service Meta Information: These fields vary on the type of service. They define the resource-specific meta parameters for this service, often used when creating new resources. For example the "MaxIsn" value (setting the maximum amount of records which can be stored on the file) to be used when creating a new Adabas file.
- Usage Governance Information. For more details see [here](#)



Important: The **Save** button must be clicked to publish the changes to the server. The button is located in the top-right corner of the properties view.

Service context menu



- *Edit DataView*: Edit the DataView associated with the selected Service. The dataview will open in the Editor View, and the details of each field will appear in the Properties View. Changes will not be reflected on the web service (WSDL) until the file is saved.
- *Rename Service*: Rename the service. This will have consequences on the URL of the WSDL, and the endpoint of the web service.
- *Duplicate Service*: Create a new web service that is a copy of the existing. This service needs to have a different name from the original.
- *Refresh Service*: clears the internal cache on the Portus server to pick up changes to the (Business)DataView, XSD or XSL
- *Copy Service definitions*:
- *UDDI Register Service via JAXR or SOAP*
- *Show WSDL* : Display the WSDL for this service. This is the starting point of for any web service clients that wish to consume this web service.
- *Remove Service*: Remove this service from the Portus Server
 - ⚠ **Important:** If the *Remove orphaned dataviews* box is checked, the internal configuration files (XRD, XSD and XSL) will also be deleted from the server.

- *Create Physical Resource*: Create a back-end resource (E.g Adabas file/MySQL Table/etc), physically, based on the Portus service.
- *Delete Physical Resource*: Physically delete a back-end resource described by the web service.

Rename Service

- Select "Rename Service" from the context menu
- This will bring up a dialog and allow you to choose a new name for the service. By default the original service name will be appended with the string "_Rename".

Enter the new name of your choice and click *OK*.

- The service will appear in the list of services with its new name immediately

Duplicate Service

This function will create an initially exact copy of a service. You will be prompted to enter a name for the copy.

The copy will appear in the list of services immediately

Modify the Dataview and/or service parameters as needed.

Copy Service definitions to other server(s)

Copying service definitions from one server to another is possible in two ways

- using the [Copy Wizard](#)
- using drag-and-drop

Copy Service(s) using the Copy Wizard

- Right-click the service you wish to copy, select the *Copy Service definitions ...* function from the context menu.
- Assumptions:
 - Both Portus Servers are accessible from the Control Centre.
 - If migrating services from a previous version of Portus, the utility is invoked from the most recent Control Centre.
 - Both Portus servers have the same drivers enabled and configured. See here for more info.



Note: "Same drivers" refers to the *type* of drivers, they need not necessarily be of the same name.

Process:

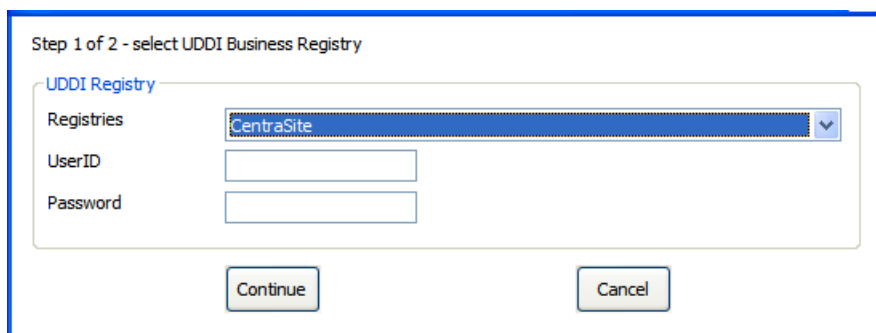
1. In the Portus you wish to *copy from*, choose the service you wish to copy by right-clicking and choosing **Copy Service Definitions**
2. Choose the server you wish to *copy the service to*.
3. If no driver with the same name exists on the target server you will be given the option to choose another driver of the same type as the source.
4. If you wish to overwrite the service on the target server, check the **Overwrite existing....** box.
5. The status of the copy will be written to the Action Log, and this you can double-click this to get more information.

Register a Service with an UDDI Business Registry (UBR)

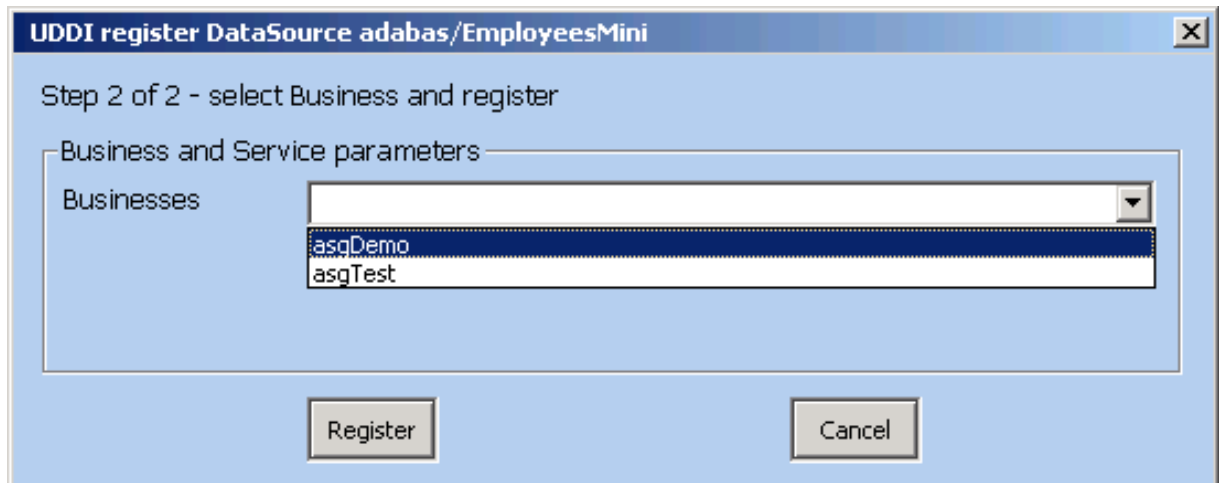
Portus Services, which are essentially "Services", can be registered with an UDDI Business Registry, for example Software AG's Centrasite, so that any UDDI enabled client application can find an exposed service, retrieve its signature, issue requests against it etc.

These are the steps required to register a Service with an UBR:

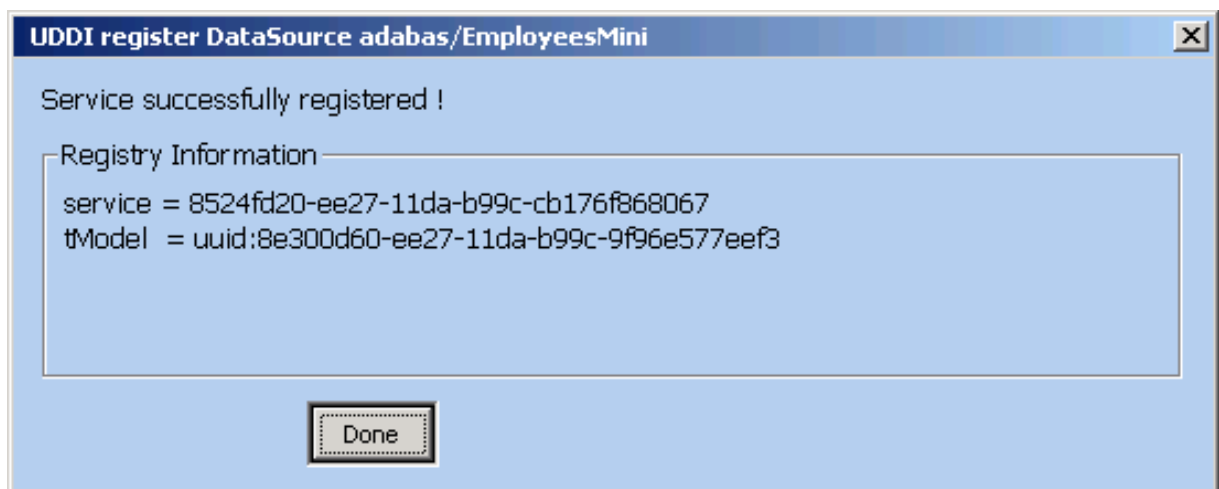
1. Define a UDDI Business Registry to the Portus Control Centre (from the Portus Eclipse Preferences dialog)
2. Select the **UDDI register Service** function from the context menu
3. Select one of the predefined UDDI Business Registries, enter the user credentials required to be able to publish



4. Select a Business registered with the UDDI Registry for the selected account, click **Register** to continue



5. After successful completion of the registration process the service and tModel registry keys will be shown



6. Click **Done**

Lifecycle Governance

The term *Lifecycle Governance* (LG) is used to define the Portus handling of distinct versions and states of web services. States allow the administrator to define services in Test mode, move them to frozen production environments, and, in time, deprecate them from use. Versioning allows the administrator to define multiple versions of a service.

Service Version

The WSDL is the main starting point for web service clients who wish to consume Portus web services. As part of the Portus LG support, it is possible to create multiple versions of the WSDL (and so the web service), thus allowing different clients to consume the different versions. It also allows modifications to be made to existing services, safe in the knowledge that existing clients will not be affected.

- When requesting the WSDL, Portus will always return the most recent version. To retrieve a specific version, the `__version` argument must be provided on the WSDL URL. E.g.

```
http://host:port/myService?WSDL&__version=5
```

The WSDL will contain a service endpoint which reflects the service version.

- When working with REST requests, the same `__version` argument can be used.

```
http://host:port/myService?LIST&__version=5&ID=4*
```

- Internally, a new `DataView` is created for every service version. The `DataView` name will have the version number appended on the end of the file name.

Service Status

There are 4 statuses a Portus service can be in:

1. *Test*: The service may be modified as often as is required to bring the service to a point where it is to be made available to others
2. *Frozen*: The `DataView` (metadata) for this service cannot be modified. If the service is being migrated from a *Test* status, it will be assigned a version number incremented by 1.

If a newer version of this service exists, clients will receive an informational message in the SOAP Header response.

When the Administrator edits the `dataView` of a frozen service, a new version of this service will be created, it's status set to *test*, and it's version number incremented by 1.

3. *Deprecated*: Clients will still be able to call a service with this status, but they will receive a warning message in the SOAP headers. A warning message will also be written to the Portus log, and the client should be urged to upgrade.
4. *Historical*: The service is no longer active, and clients who attempt to call this service will receive a SOAP fault.

When a service is in a specific status, the following transitions are possible

<i>Current Status</i>	<i>Next Status</i>
Test	Frozen
Frozen	Test, Deprecated
Deprecated	Test, Historical
Historical	Deprecated

20

Creating a SYSOBJH extract file

- Generating a SYSOBJH extract (Windows) 176
- Generating a SYSOBJH extract (Mainframe) 187

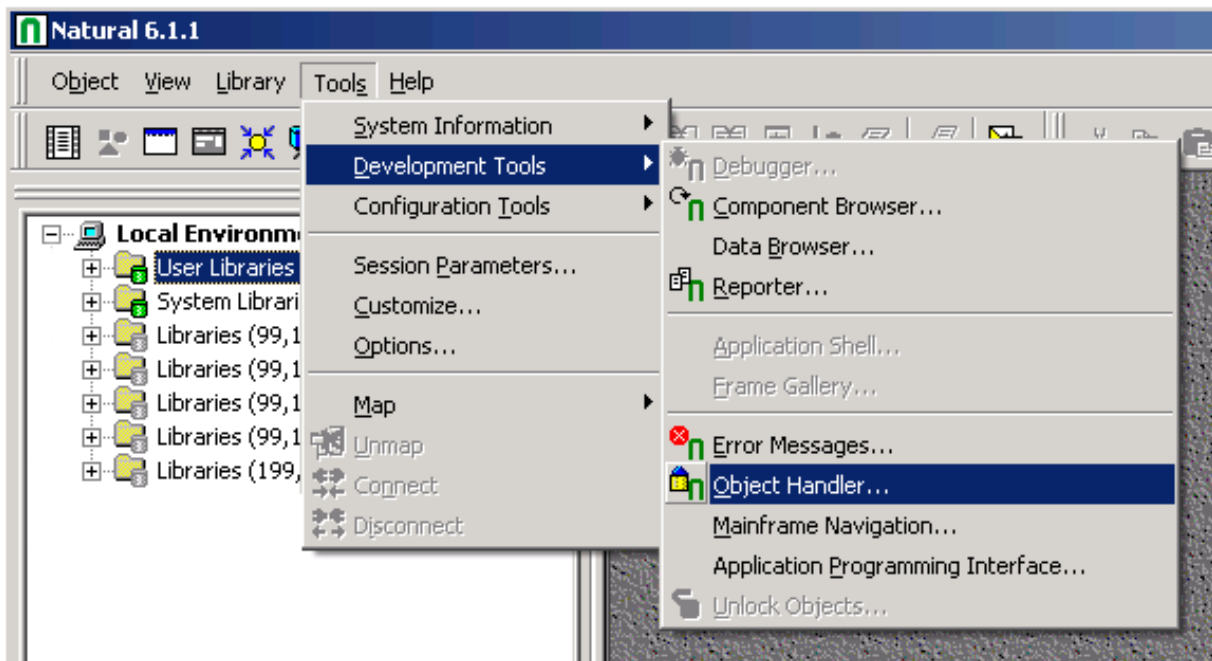
Adabas DataView(s) can be generated from DDM(s)

Natural DataView(s) from PDA(s) in a SYSOBJH extract file.

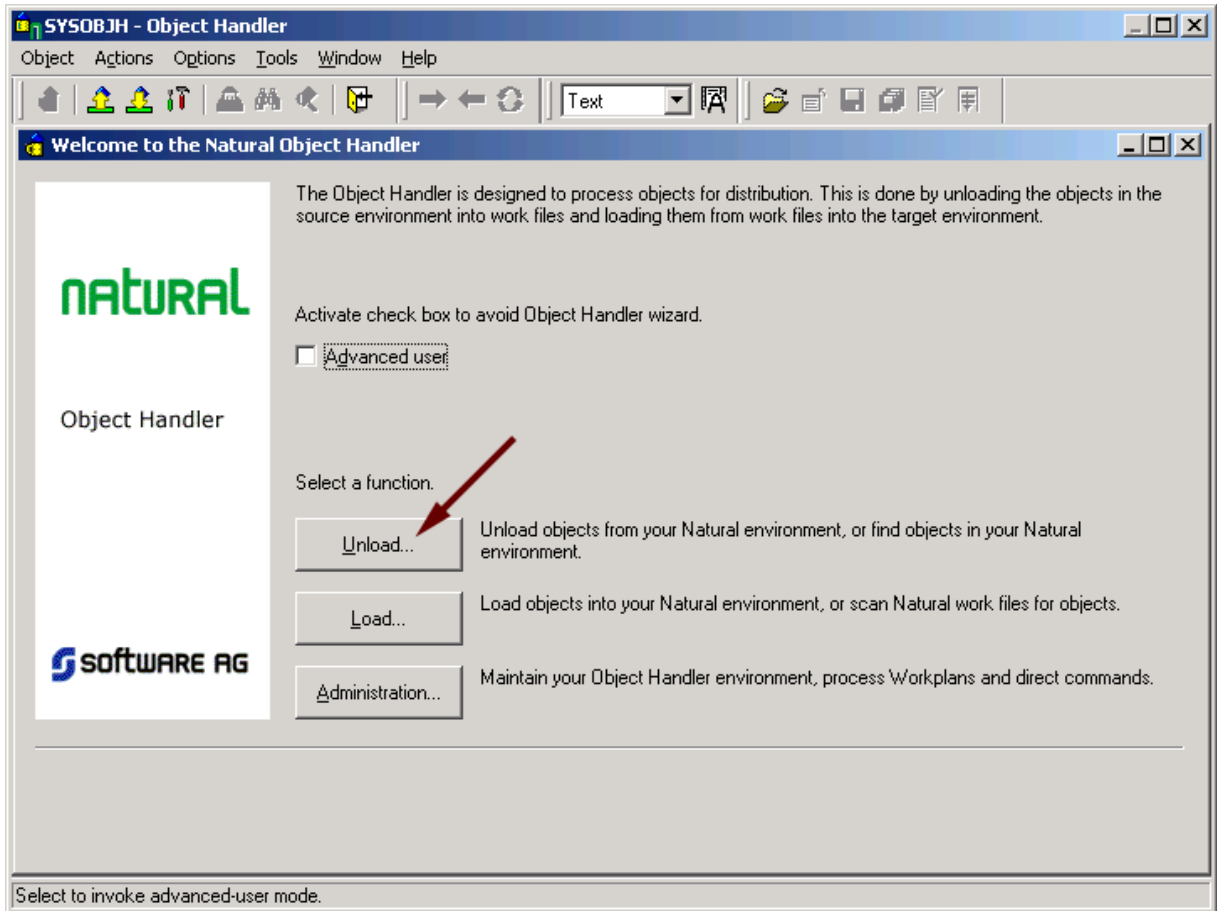
Step-by-step instructions on how to create a SYSOBJH extract file are given for **Windows** and **Mainframe** platforms.

Generating a SYSOBJH extract (Windows)

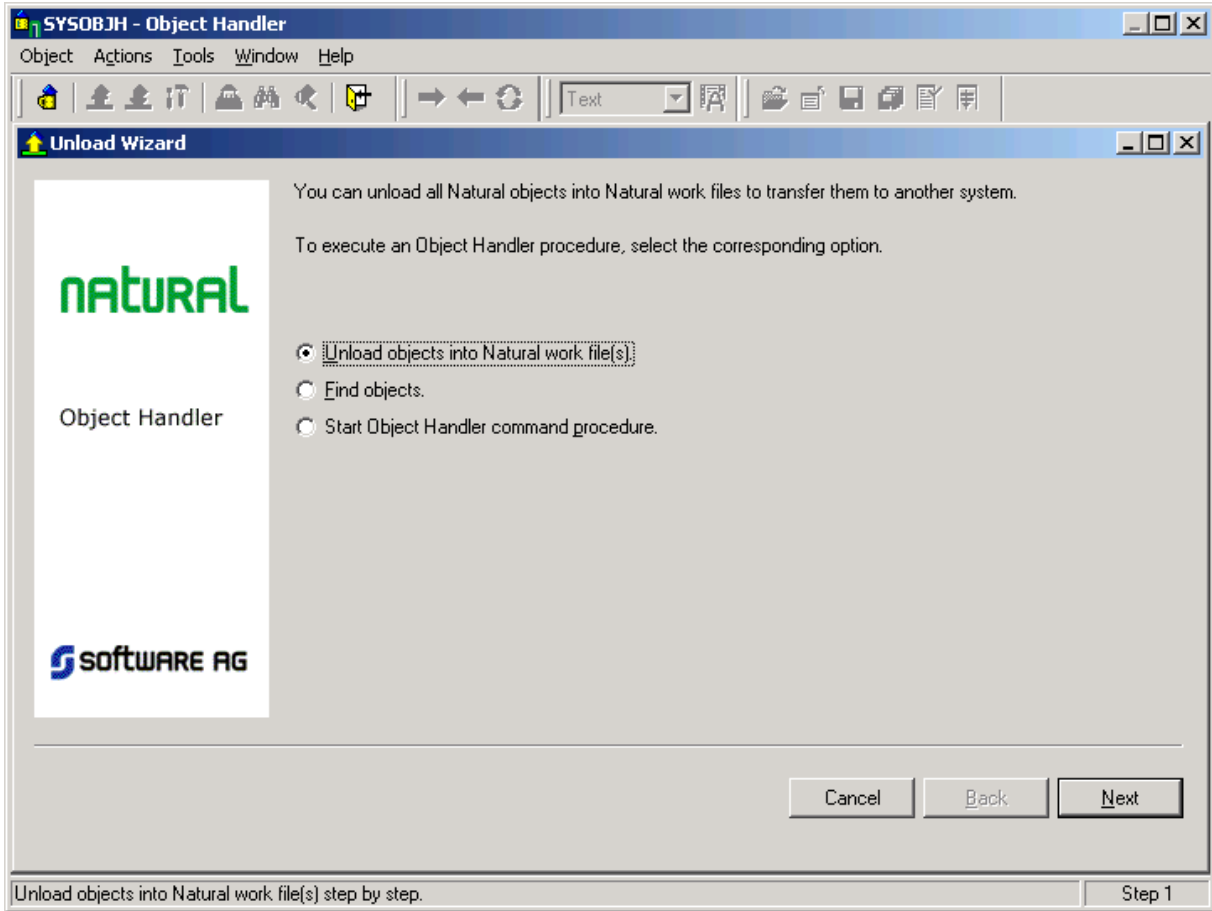
1. Start Natural, from the **Tools** menu bar select **Development Tools"**, then select **Object Handler**.



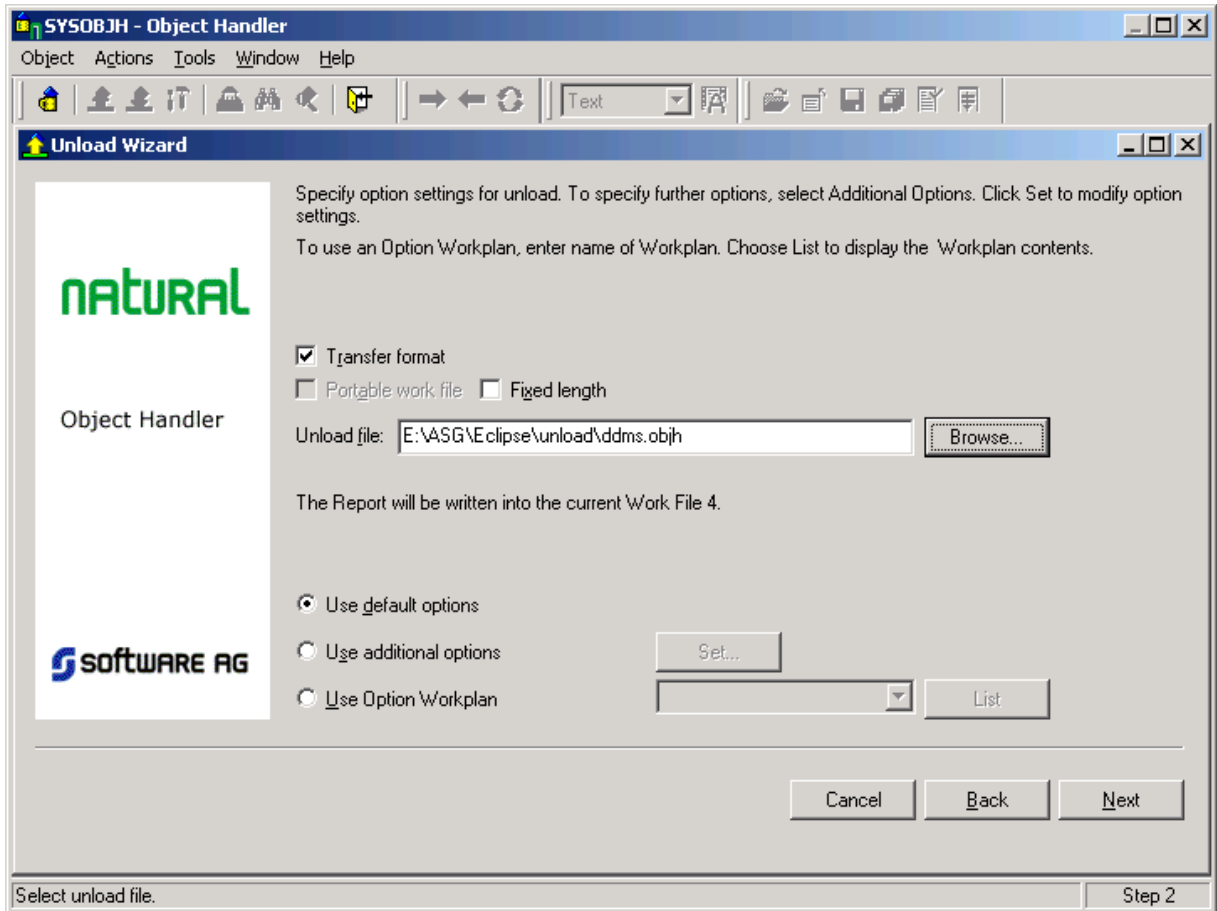
2. Click the **Unload** button to start the Unload Wizard.



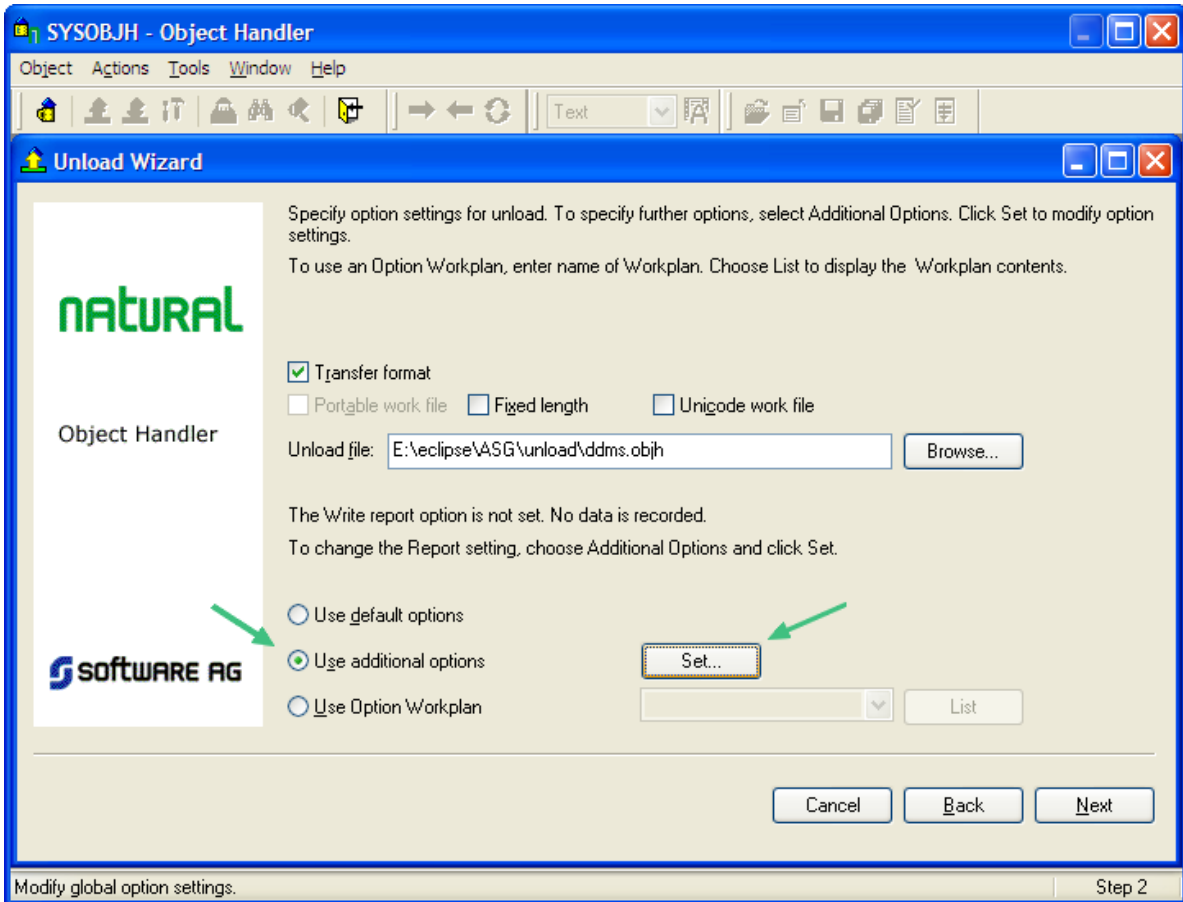
3. Check the **Unload objects into Natural work file(s)** radio button. Click **Next**.



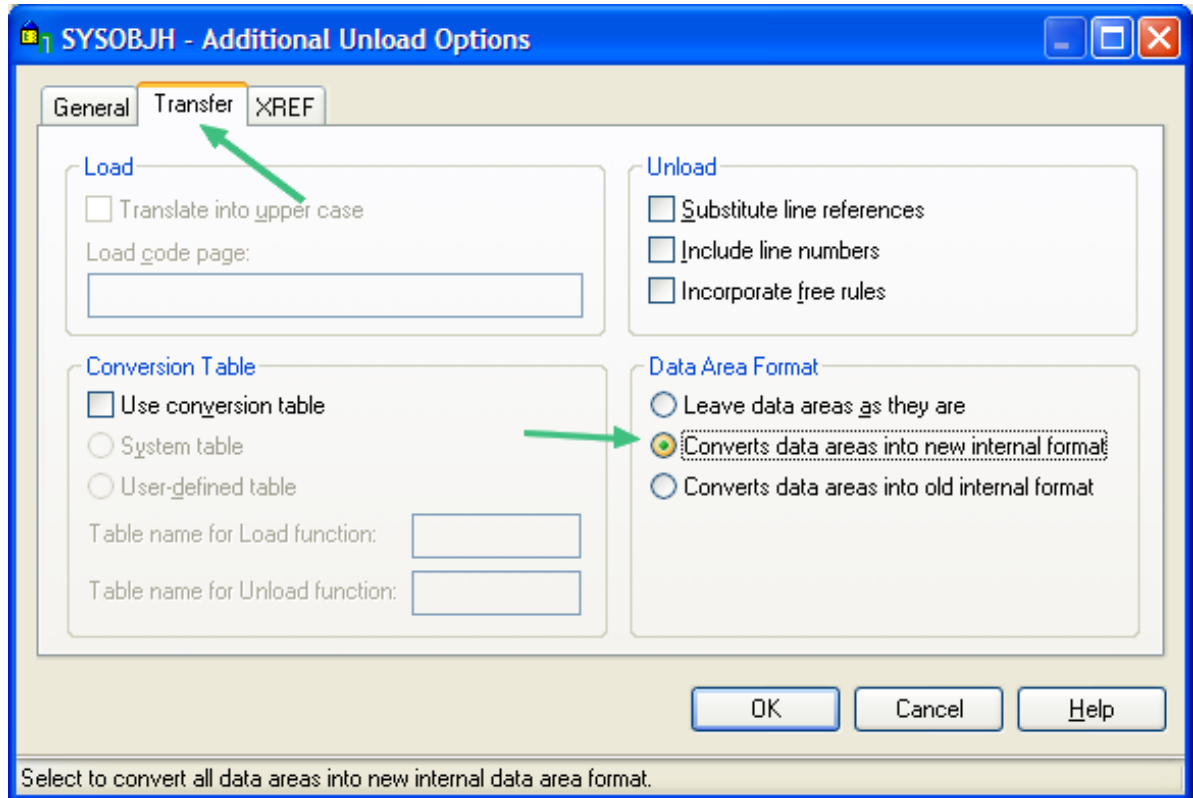
4. Select the **Transfer format** check box, enter the location to which you want the files to be unloaded in the Unload file text box (or click the **Browse** button to bring up a folder/file selection dialog). Click **Next**.



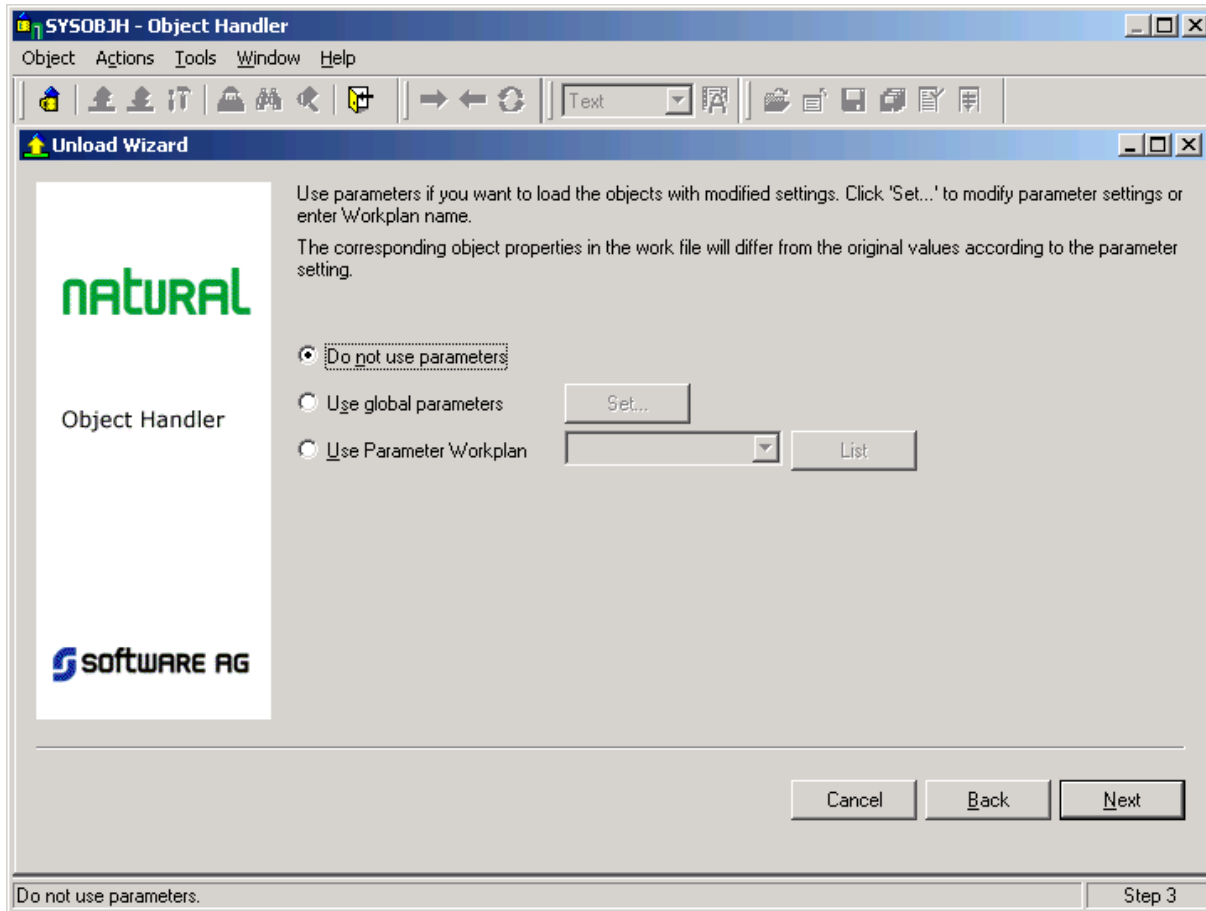
5. When generating an OBJH extract including PDAs, it is essential to set the following:
 - Check "Use additional options" and click the "Set..." button



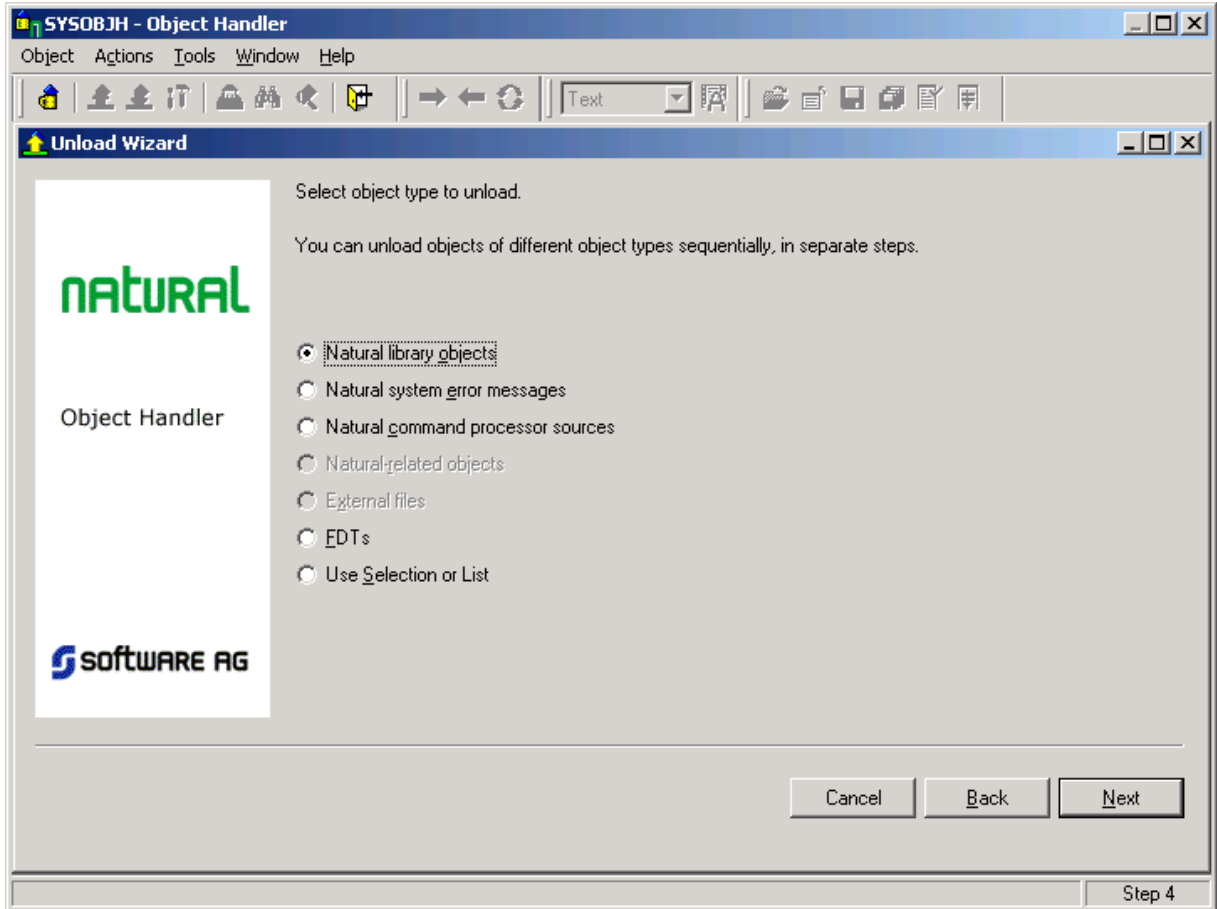
- Select the "transfer" tab and check the "Converts data areas into new internal format", then click "OK"



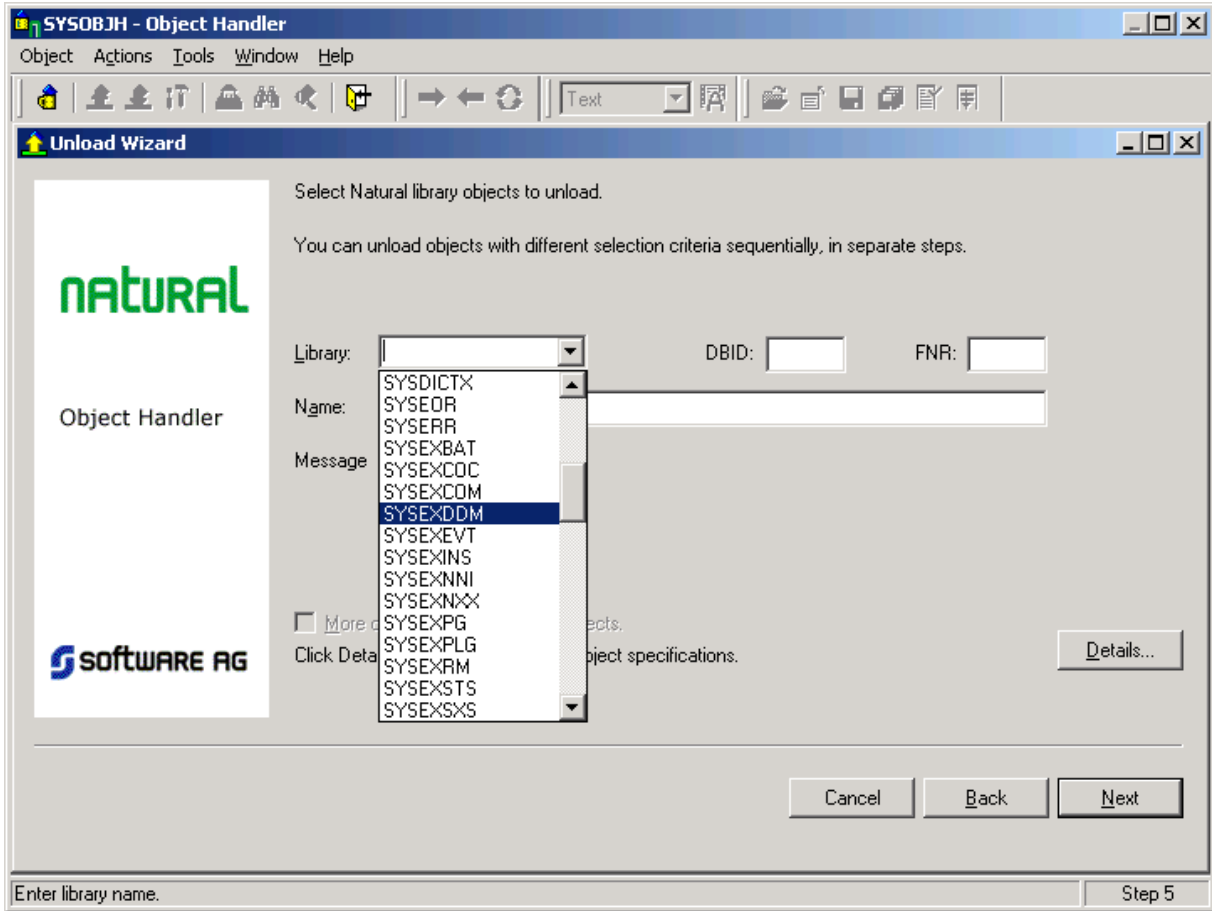
6. Select the **Do not use parameters** radio button, click **Next**.



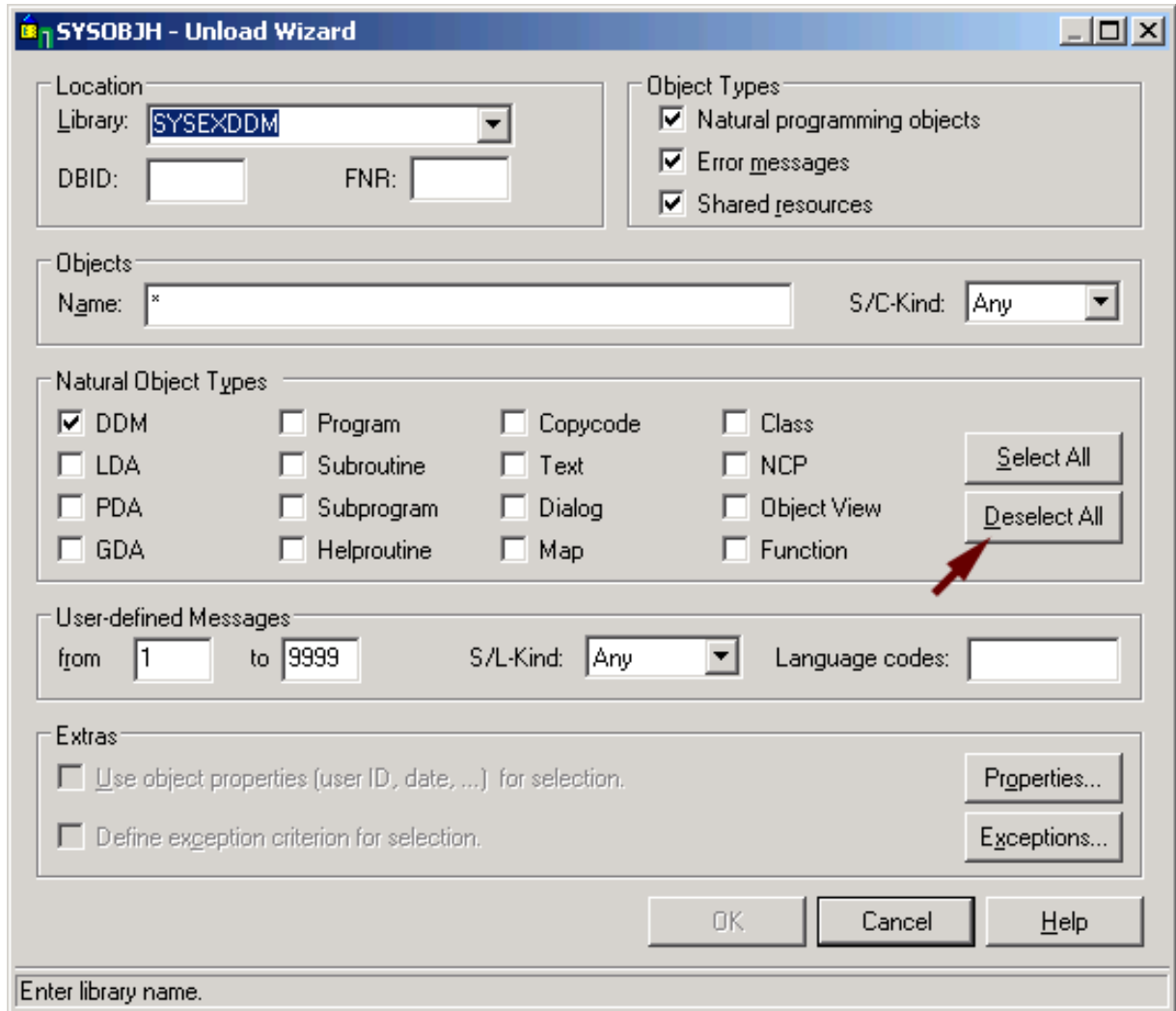
7. Select the **Natural library object** radio button, click **Next**.



8. Select the Library containing your DDM definitions (e.g. SYSEXDDM) from the Library list box, then click the **Details** button. If you also enter a specific DBID and/or FNR, only the relevant DDMs are selected, otherwise all valid DDMs are returned.



9. In the Unload Wizard details window, first click the **Deselect All** button to deselect all Natural Object Types, then select **DDM** and/or **PDA** . Click the **OK** button to return to the Unload Wizard.



SYSOBJH - Unload Wizard

Location
 Library: SYSEXDDM
 DBID: [] FNR: []

Object Types
 Natural programming objects
 Error messages
 Shared resources

Objects
 Name: * S/C-Kind: Any

Natural Object Types
 DDM Program Copycode Class
 LDA Subroutine Text NCP
 PDA Subprogram Dialog Object View
 GDA Helproutine Map Function

Select All
 Deselect All

User-defined Messages
 from 1 to 9999 S/L-Kind: Any Language codes: []

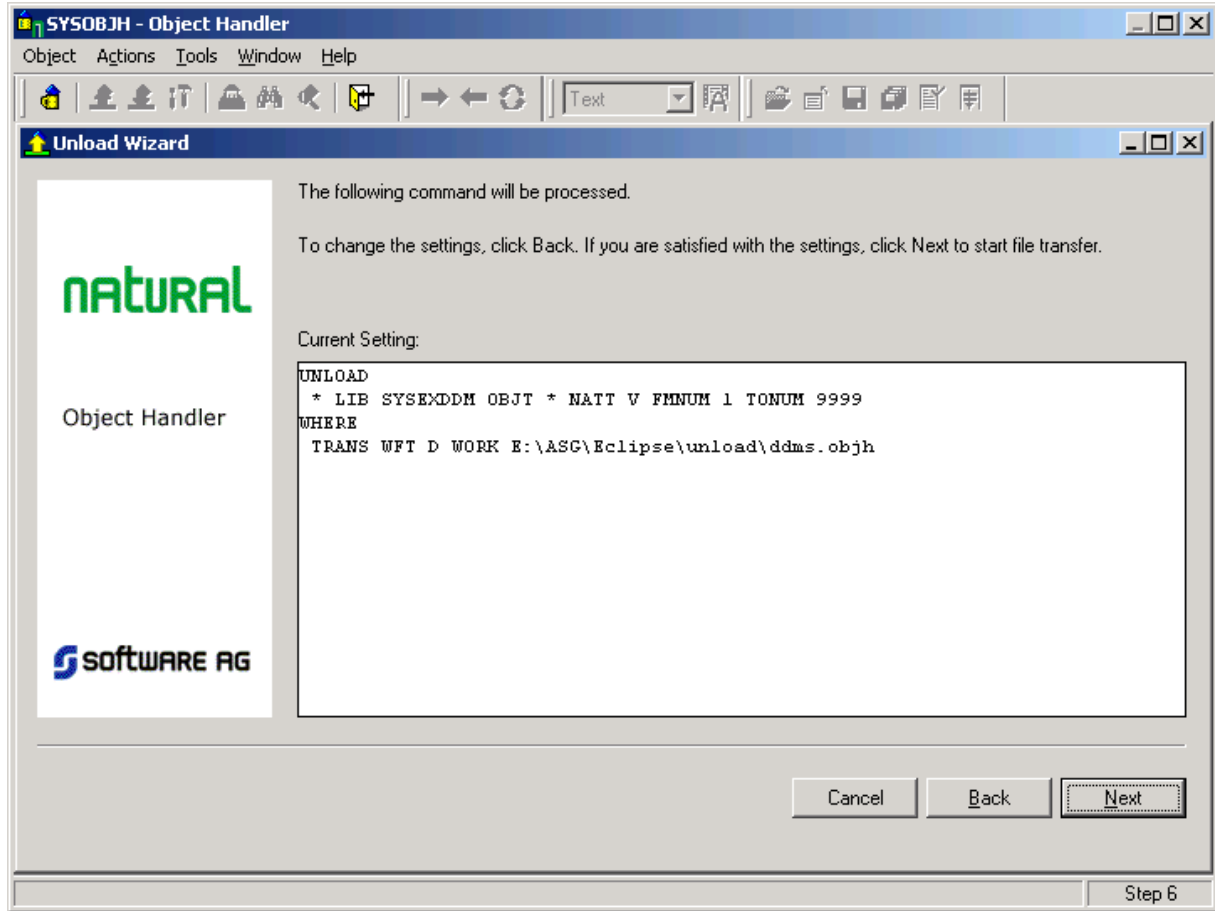
Extras
 Use object properties (user ID, date, ...) for selection.
 Define exception criterion for selection.

Properties...
 Exceptions...

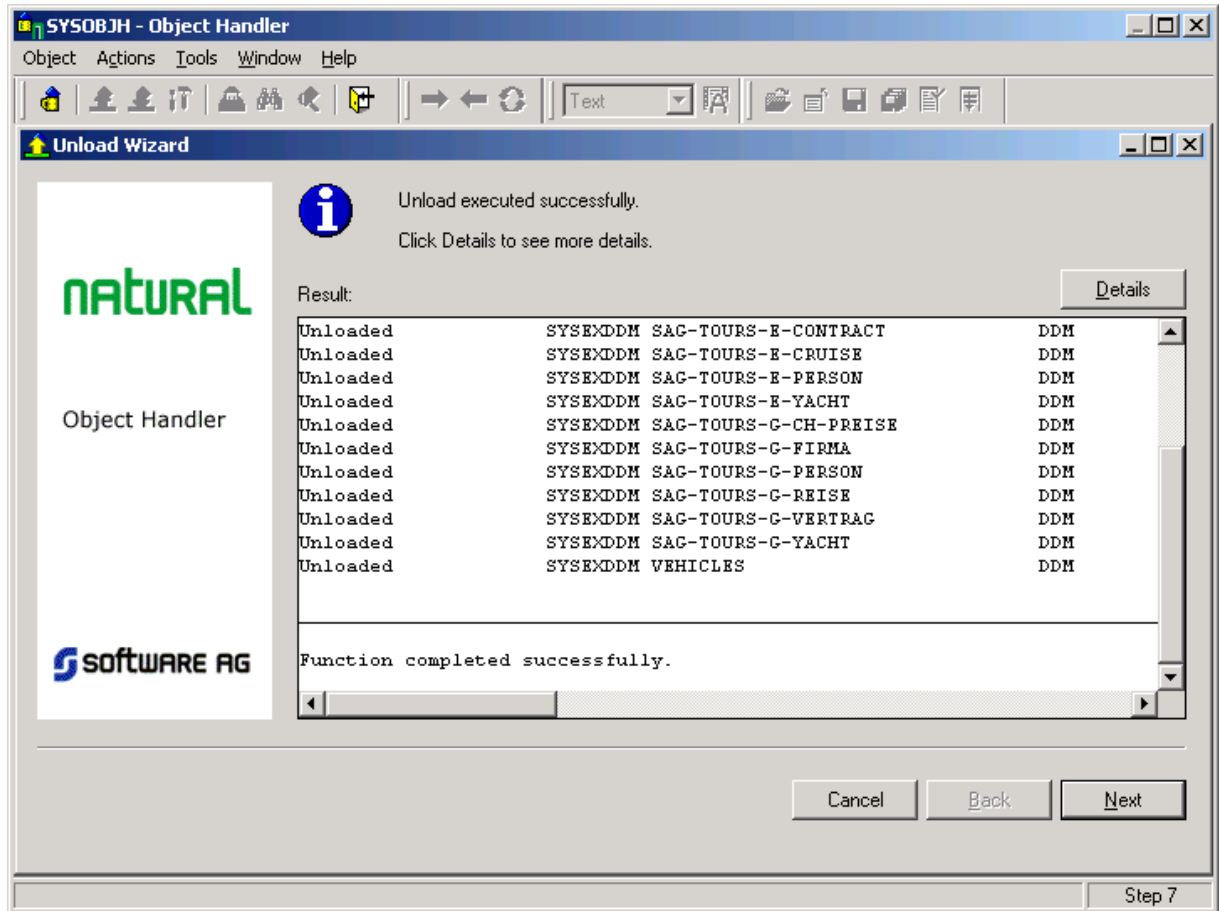
OK Cancel Help

Enter library name.

10. The generated Unload command is shown. Click the **Next** button to start the actual unload process.

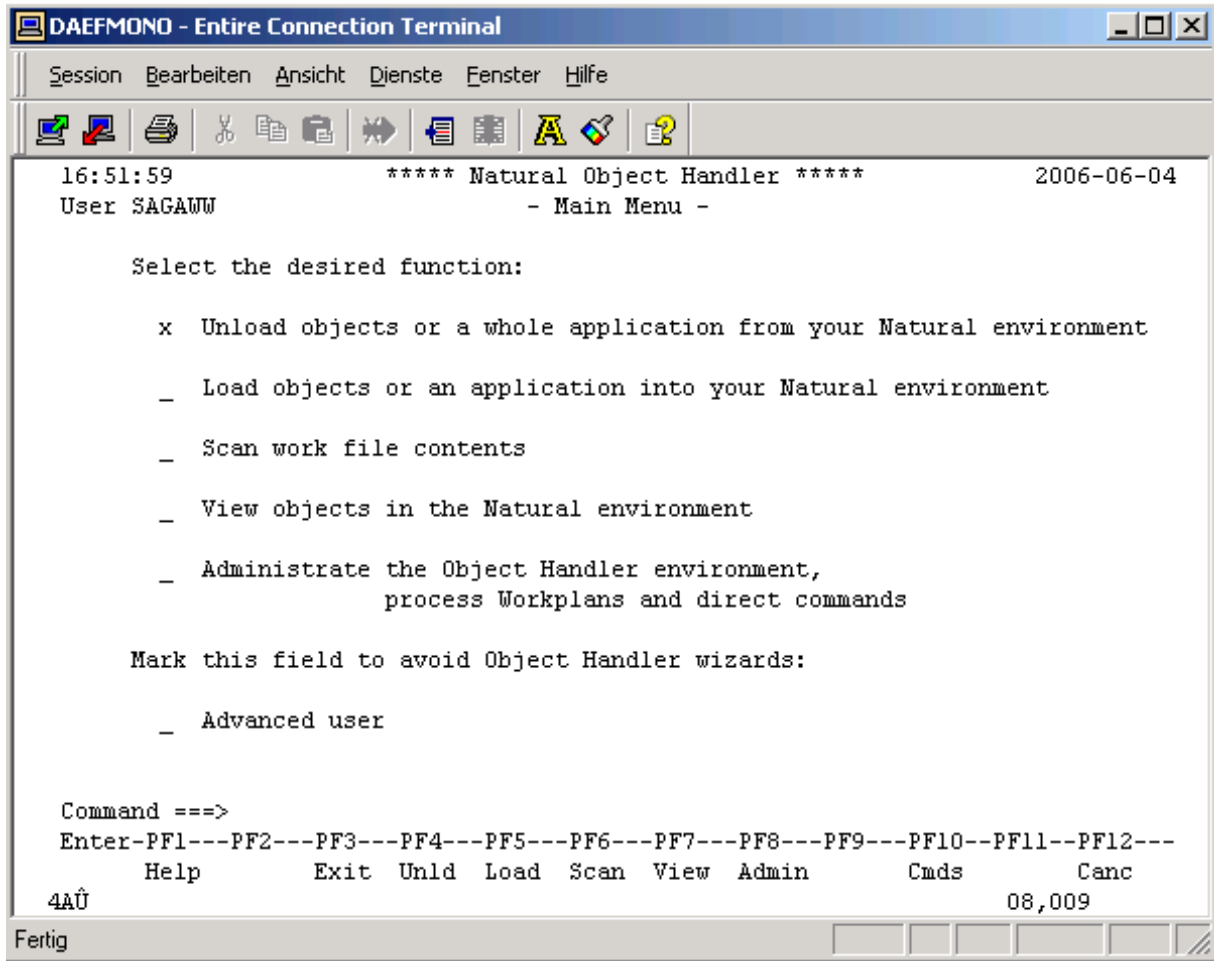


11. The status of the unload is shown. If it was successful, a file is created under the path and file name specified. Click the **Next** button to complete the unload, close the SYSOBJH wizard.

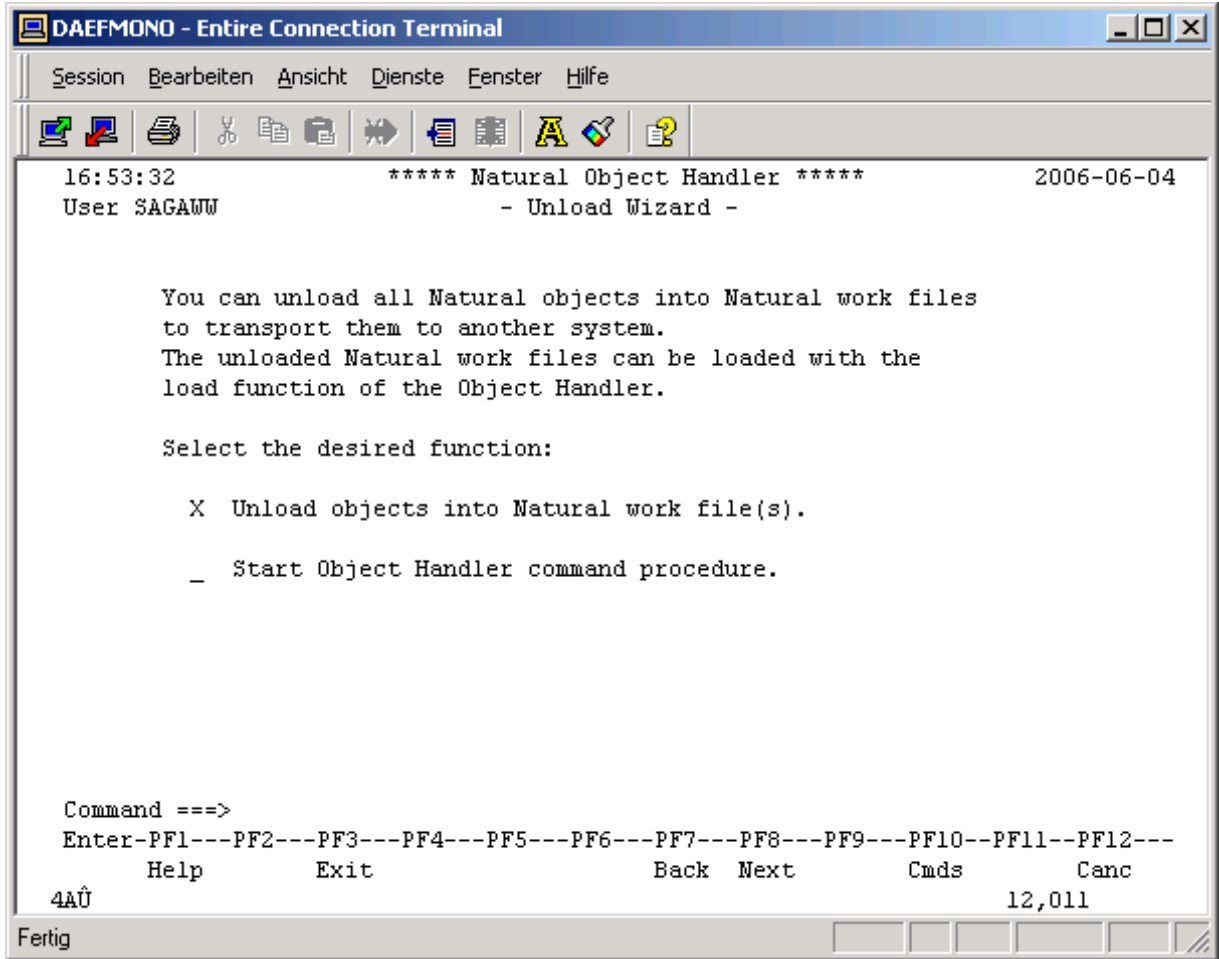


Generating a SYSOBJH extract (Mainframe)

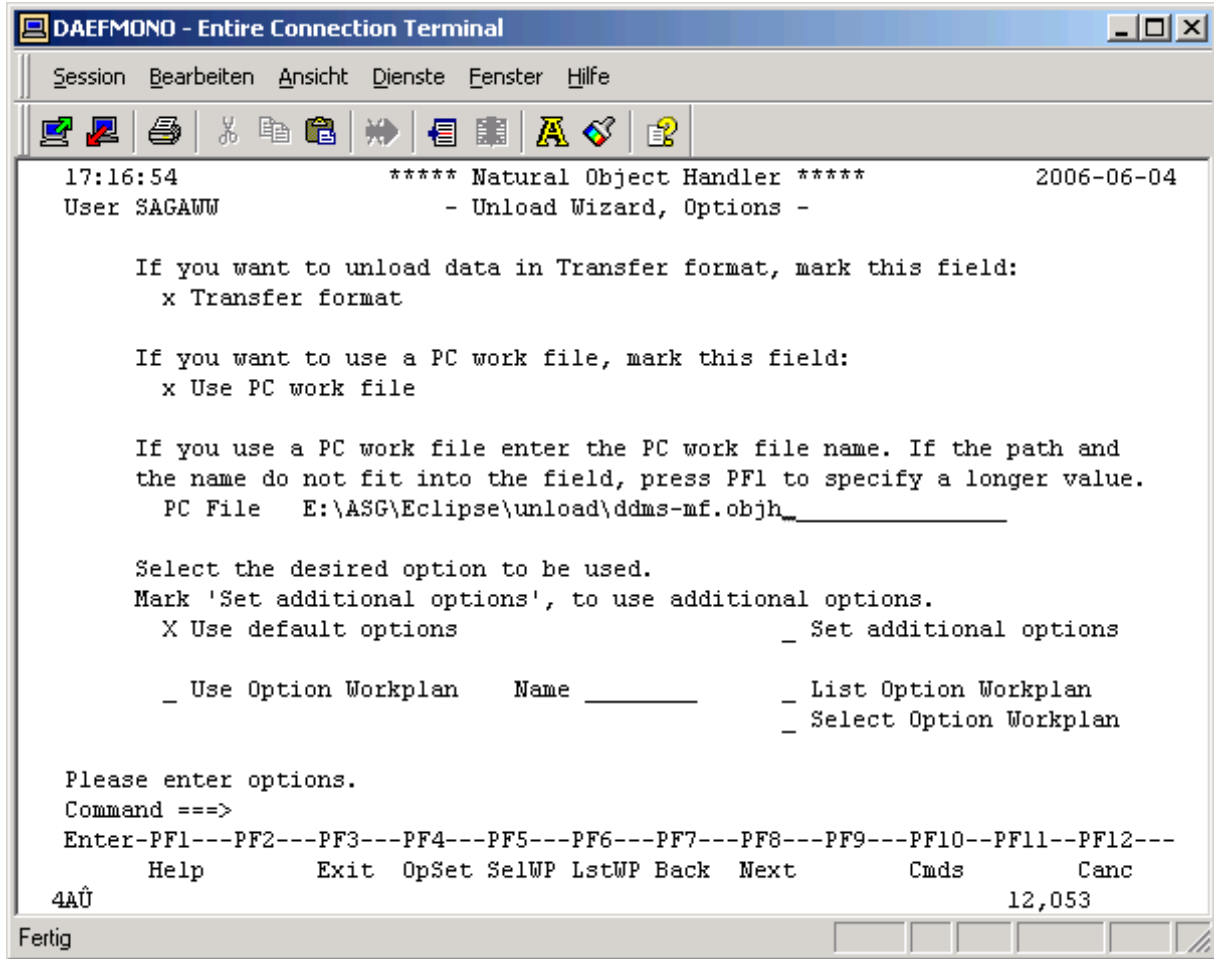
1. Start Natural, enter the command `SYSOBJH` to start the Natural Object Handler.
2. Select the **Unload** function to start the Unload Wizard.



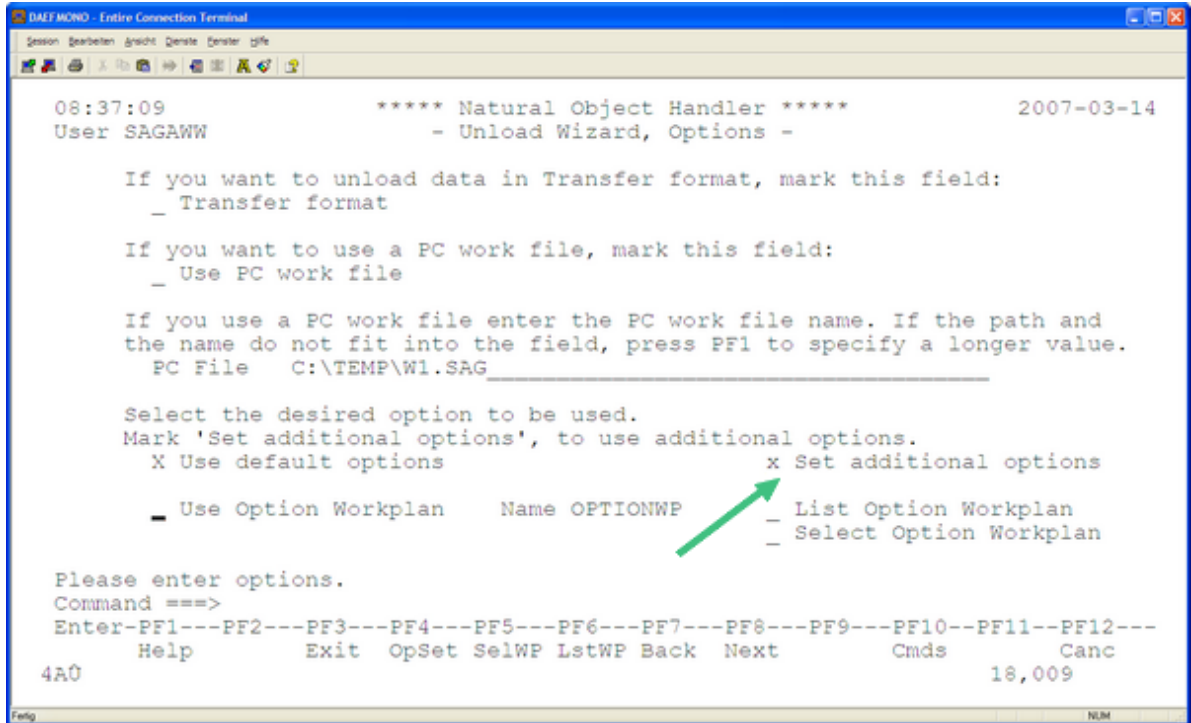
3. Check **Unload objects into Natural work file(s)** radio button.



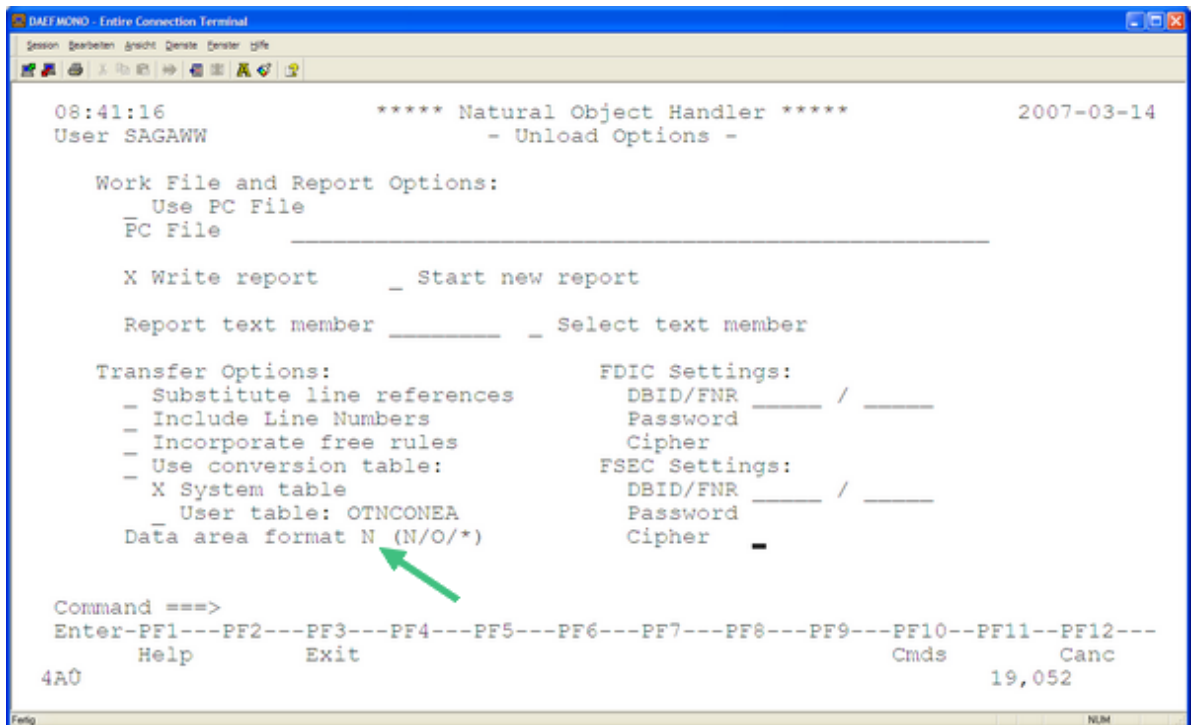
4. Select Transfer format, this example uses Entire Connection to transfer the unload file to your PC, enter the location to which you want the files to be unloaded in the PC File input field.



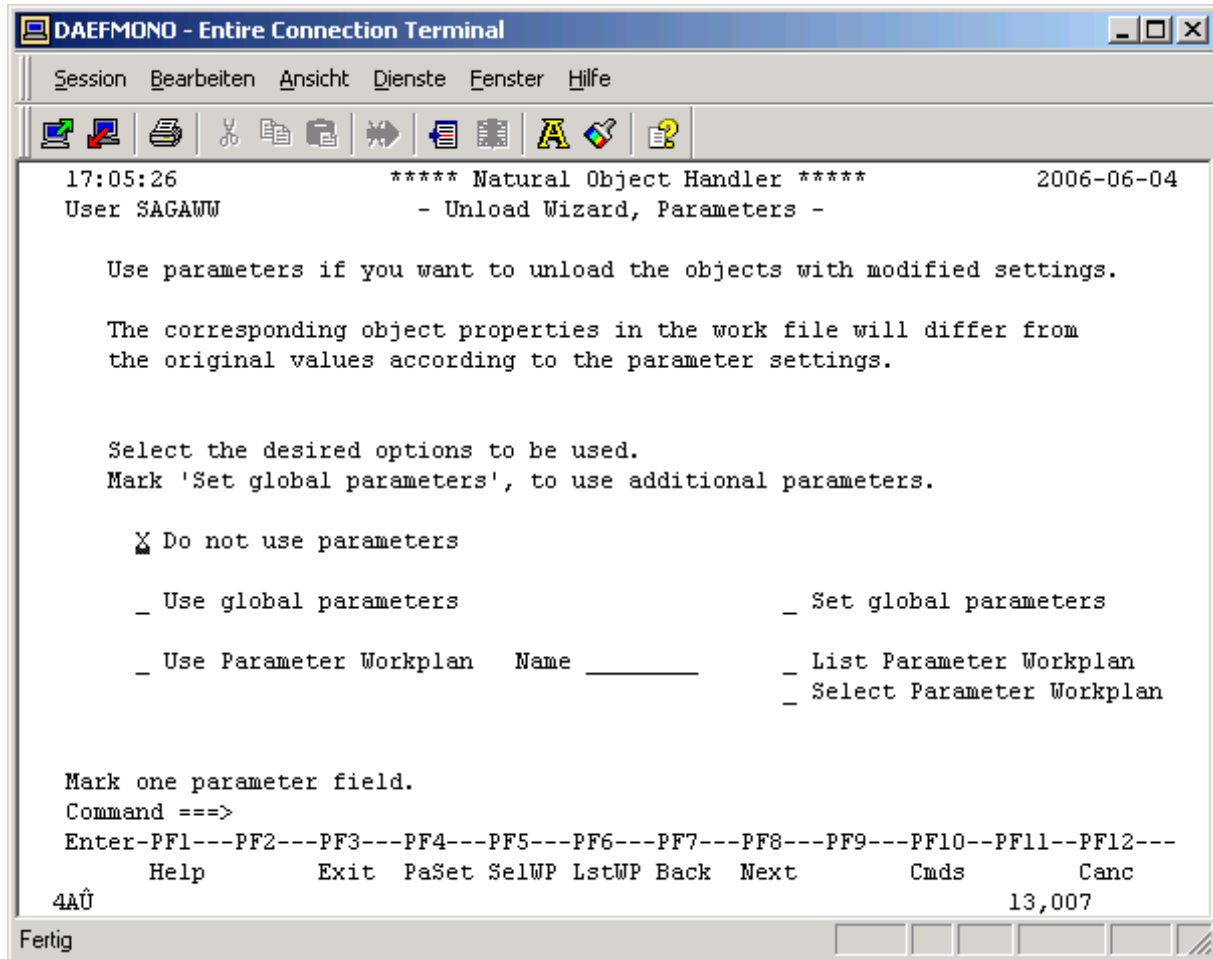
5. When generating an OBJH extract including PDAs, it is essential to set the following:
 - Check "Set additional options" and press <ENTER>



- Enter a "N" into the "Data area format", then press "ENTER"



6. Select **Do not use parameters**.



7. Select **DDMs only**.

```

DAEFMONO - Entire Connection Terminal
Session Bearbeiten Ansicht Dienste Fenster Hilfe
17:06:35          ***** Natural Object Handler *****          2006-06-04
User SAGAWW      - Unload Wizard, Select Unload Type -

Select the object type for unload.

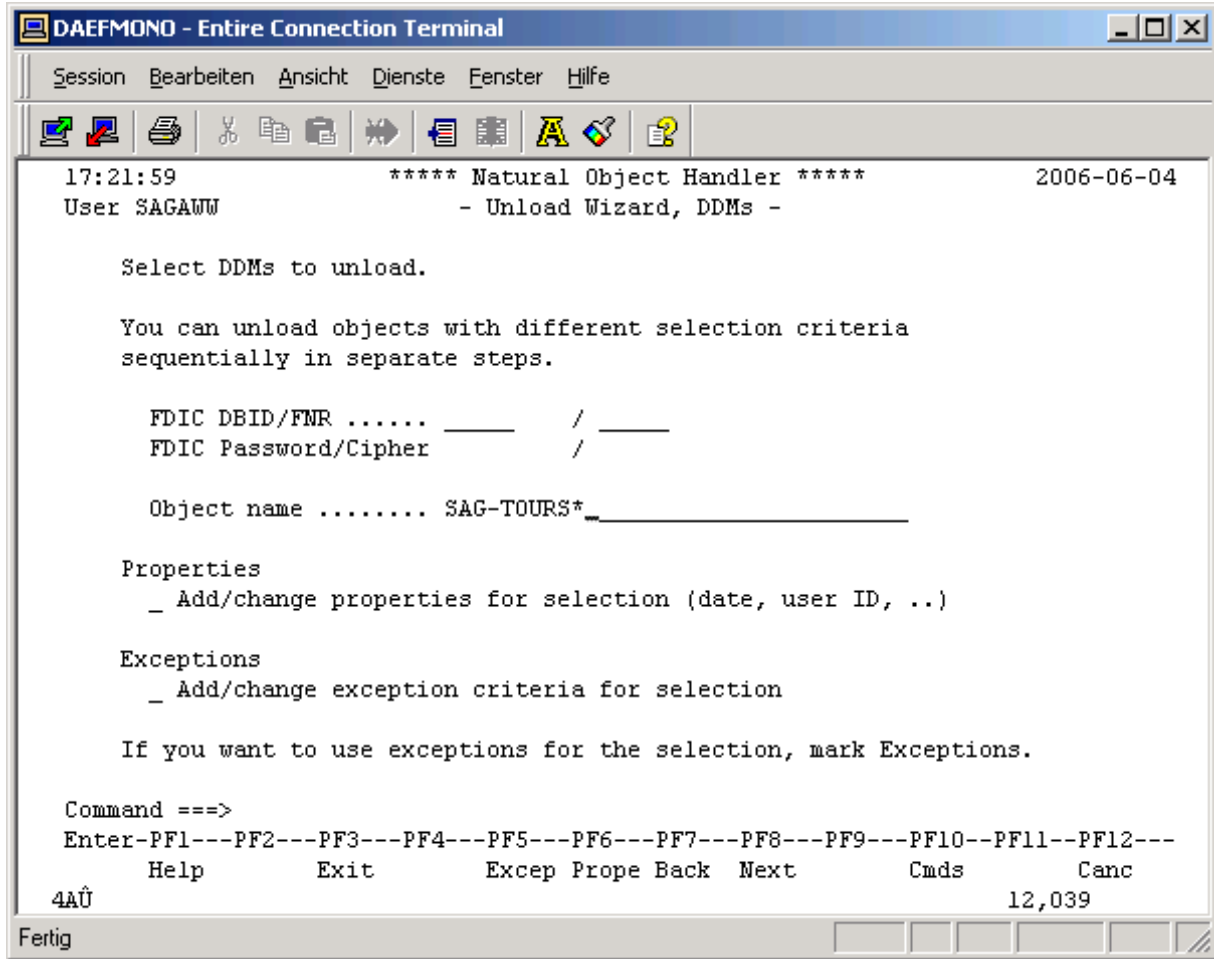
You can unload objects of different object types sequentially
in separate steps.

    Natural library objects only
  - Natural system error messages only
  - Natural command processor sources only
  - Natural-related objects only
x  DDMs only
  - FDTs only
  or
  - Use Selection or List Workplan

Mark one object type.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit          Back Next      Cnds      Canc
4AÛ
14,011
Fertig

```

8. Select the DDM(s) to be unloaded, optionally specify additional selection criteria to narrow the search.



- The generated Unload command is shown. Hit the Enter or PF8 key to start the actual unload process.

The screenshot shows a terminal window titled "DAEMON - Entire Connection Terminal". The window has a menu bar with "Session", "Bearbeiten", "Ansicht", "Dienste", "Fenster", and "Hilfe". Below the menu bar is a toolbar with various icons. The main terminal area displays the following text:

```

17:22:46          ***** Natural Object Handler *****          2006-06-04
User SAGANW          - Unload Wizard -

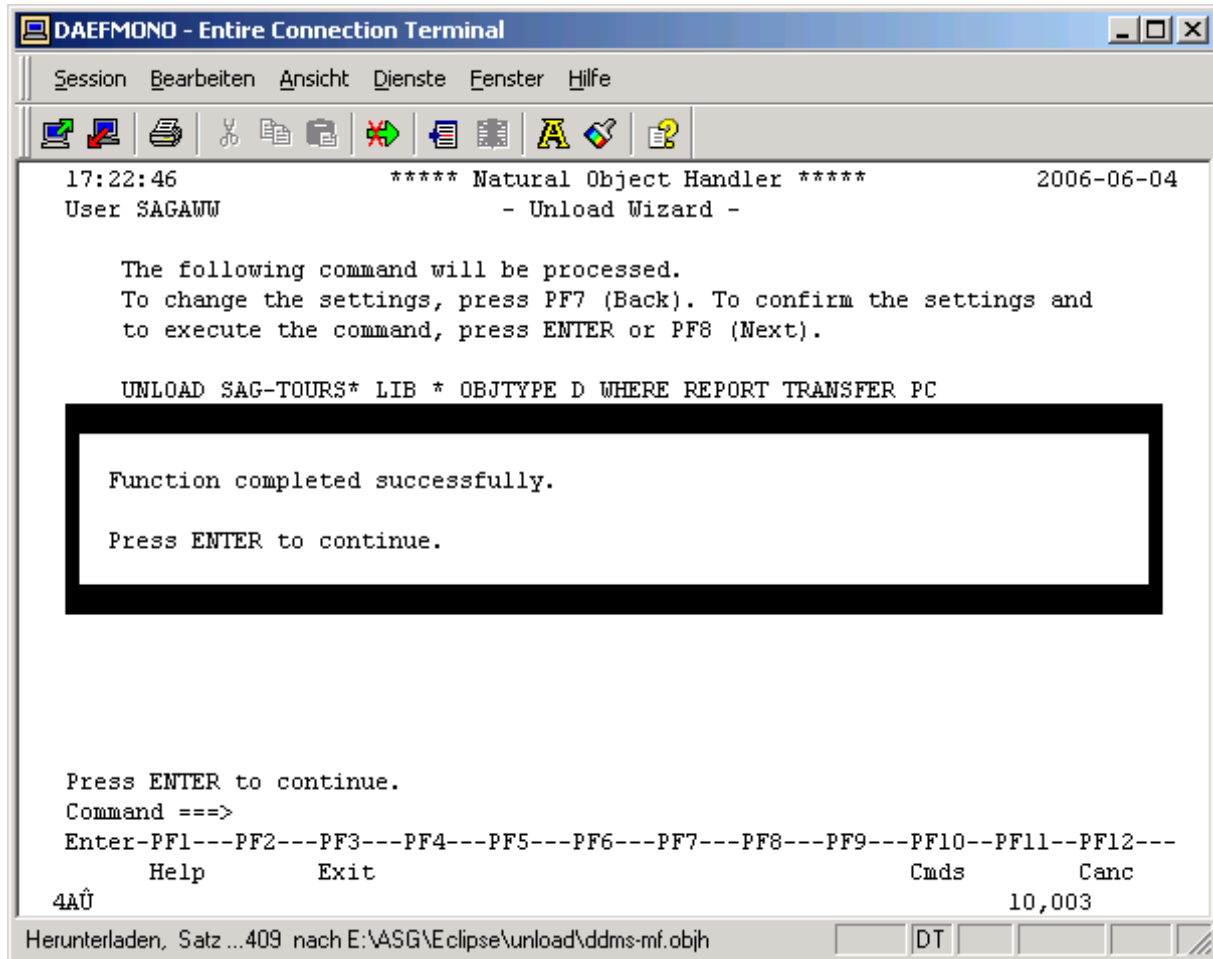
The following command will be processed.
To change the settings, press PF7 (Back). To confirm the settings and
to execute the command, press ENTER or PF8 (Next).

UNLOAD SAG-TOURS* LIB * OBJTYPE D WHERE REPORT TRANSFER PC
E:\ASG\Eclipse\unload\ddms-mf.objh

Please press PF7 (Back) or ENTER/PF8 (Next).
Command ==> _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit      Save      Back Next      Cnds      Canc
4AÛ                                                    22,015
Fertig

```

10. The status of the unload is shown. If it was successful, a file is created under the path and file name specified. Hit the **Enter** key to complete the unload, terminate the SYSOBJH wizard.



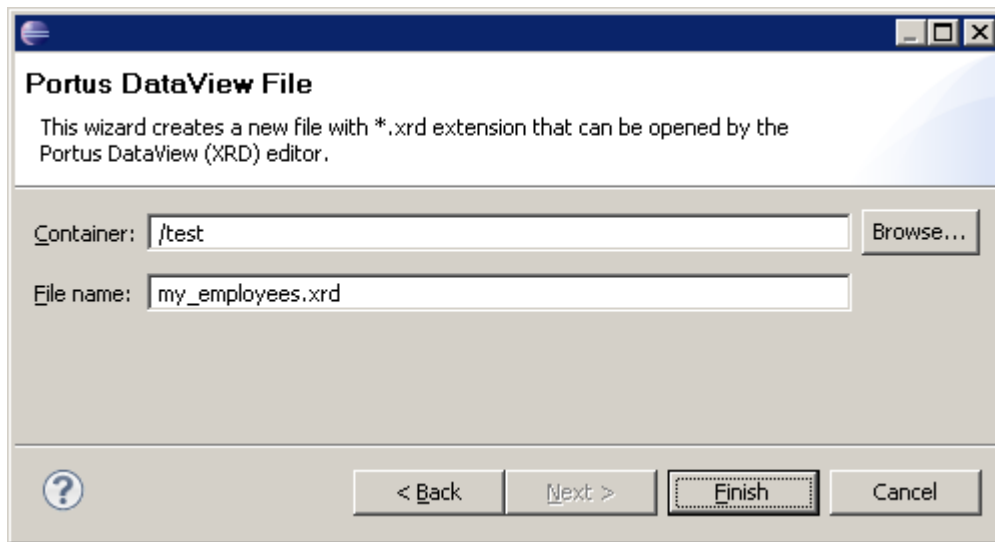
21 Portus DataViews

▪ Create a DataView from scratch	198
▪ Opening a Portus DataView for editing	199
▪ Editing a Portus DataView	200
▪ Editing "special" fields	205
▪ Enhanced Type Conversion	322
▪ Exporting a DataView to a Portus server	325
▪ Importing an existing Portus DataView	211
▪ Creating a XML Schema for a DataView	211

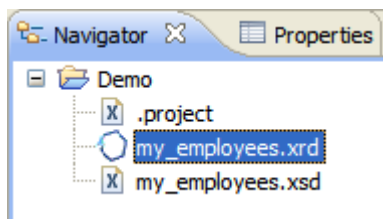
- **Create a DataView from scratch**
 - adding MU (multiple value) fields
 - adding PE (periodic group) fields
 - adding "special" fields (Adabas Super-Descriptors etc.)
- **Opening a Portus DataView for editing**
- **Editing a Portus DataView**
- **Export a DataView to a Portus server**
- **Importing an existing Portus DataView**
- **Creating a XML Schema (XSD) for a DataView**

Create a DataView from scratch

1. A vanilla Portus DataView is created with an Eclipse "New Wizard", start it with **File -> New -> Other** (or use the shortcut Ctrl+N) to bring up the list of available wizards. Select **Other-> Portus DataView** from this list. Click **Next**.
2. Enter or select a Destination folder, specify a name for the DataView file, click **Finish** to create it.



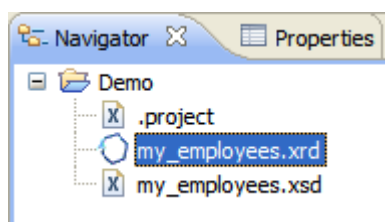
3. You have now created an "empty" (i.e. no fields defined yet) Portus DataView file in the selected folder.



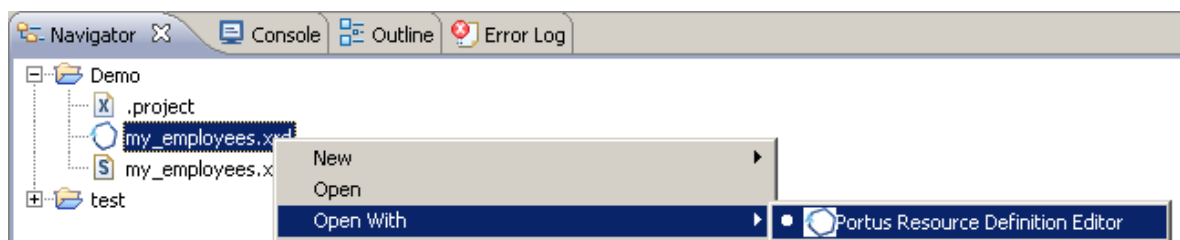
4. For information on actually editing the DataView, adding fields etc., please continue reading at [Edit a Portus DataView](#)

Opening a Portus DataView for editing

1. Open a "local" file, contained in an Eclipse project within the active workspace...
 - by double-clicking on it's name in the Package Explorer or Navigator



- by right-clicking the DataView (.xrd) file, Open With -> Portus Resource Definition Editor



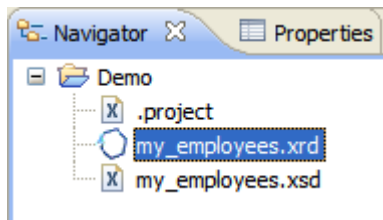
2. Open a "remote" DataView file, directly on the server, without importing it into a workspace / project first, by right-clicking the DataView name in the Configuration View to bring up the context menu, then select "edit DataView"

Name	DataView	-XSD-	-XSL-
adabas_photoblobs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
adabas_vehicles_view	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
my_employees	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Editing a Portus DataView

1. For the purpose we will open an empty DataView file and populate it with all information required to be able to start issuing requests against an Adabas Resource (= Adabas file on an Adabas database). This tutorial will be based on the Adabas "Employees" demo file.

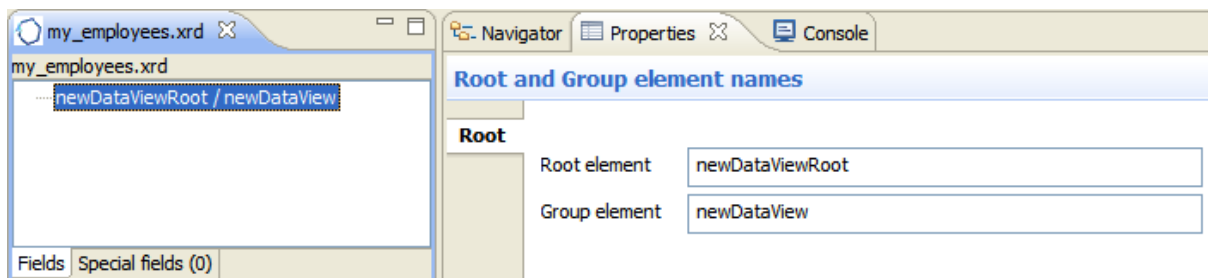
DataViews are not tied to a specific "resource type", the same mapping can be used to access an Adabas file or a SQL table. There are, however, elements of a DataView which are only meaningful in the context of a specific resource type, for example "special descriptors" (super-, sub-, ...) for Adabas.



2. The display areas relevant for editing the DataView file are
 - the actual editing area tab, showing the DataView file name in it's tab header
 - the Properties area

In case the "Properties" view somehow got hidden, right-click into the edit window and select "Show Properties View" from the context menu.

The result should look like this:

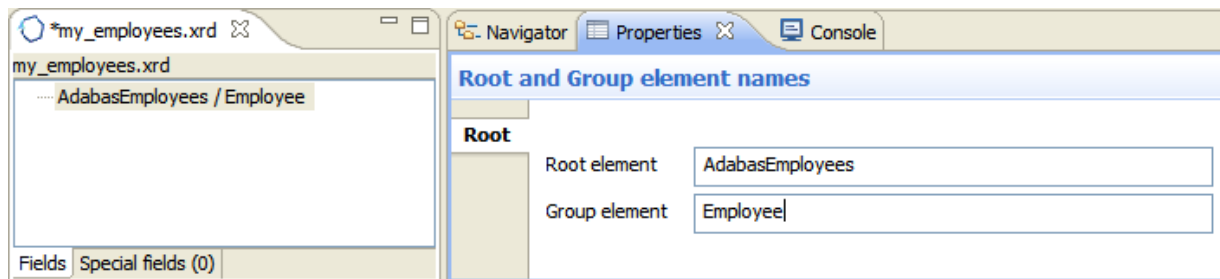


- First of all, enter the Root element name: The (XML) "structure" or "set" name under which items (records) for a Resource linked to this DataView will be referred to. E.g.: AdabasEmployees

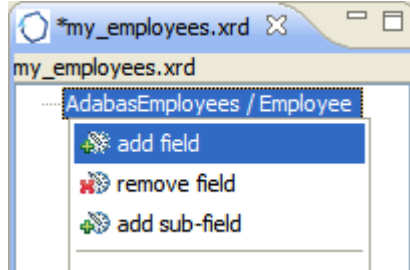
Enter the Group element name, this is the Portus "record name". (E.g.: Employee).

Changes applied to name fields in the Properties view are reflected in the editor's tree view immediately.

- ⚠ Important:** The 'Root' and 'Group' element names may NOT be identical, as this would lead to duplicate XML element names and thus 'loops' in the schema.

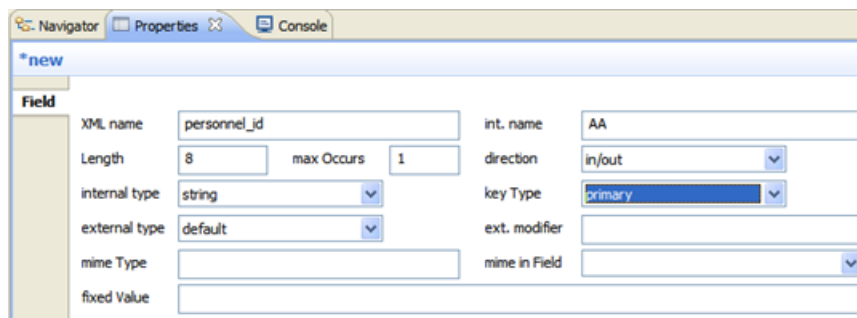


- We can start adding the actual DataView Definition elements now, right-click on the root element in the editor window, from the context menu appearing now select **add field**.



- A new element ("Field") has been added with all properties set to initial values.

The first field we are going to add is the "personnel Id", set the properties as follows:



6. Add a few more fields with the following attributes:

first_name

Field	XML name	first_name	int. name	AC
	Length	20	max Occurs	1
	direction	in/out		
	internal type	string	key Type	none
	external type	default	ext. modifier	
	mime Type		mime in Field	
	fixed Value			

name

Field	XML name	name	int. name	AE
	Length	20	max Occurs	1
	direction	in/out		
	internal type	string	key Type	secondary
	external type	default	ext. modifier	
	mime Type		mime in Field	
	fixed Value			

city

Field	XML name	city	int. name	AJ
	Length	20	max Occurs	1
	direction	in/out		
	internal type	string	key Type	secondary
	external type	default	ext. modifier	
	mime Type		mime in Field	
	fixed Value			

Next we are going to define the "address line" field, which is a MU (multiple value) field. MU fields are defined like a "flat" field, with the exception of the max Occurs Property being set to a value >0

address_line

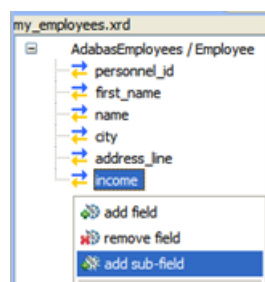
Field	XML name	address_line	int. name	AI
	Length	20	max Occurs	4
	direction	in/out		
	internal type	string	key Type	none
	external type	default	ext. modifier	
	mime Type		mime in Field	
	fixed Value			

7. Lastly, we will define a PE (periodic group), the structure of the "income" group of the "Employees" file:

- currency Code (simple field)
- annual Salary (simple field)
- annual Bonus (MU field)

So we first add the "group field" income. Only the xml Name, int. name (internal name = Adabas field name) and max Occurs properties are relevant here.

To actually turn the income field to a PE group field, right-click it, select **add subfield**



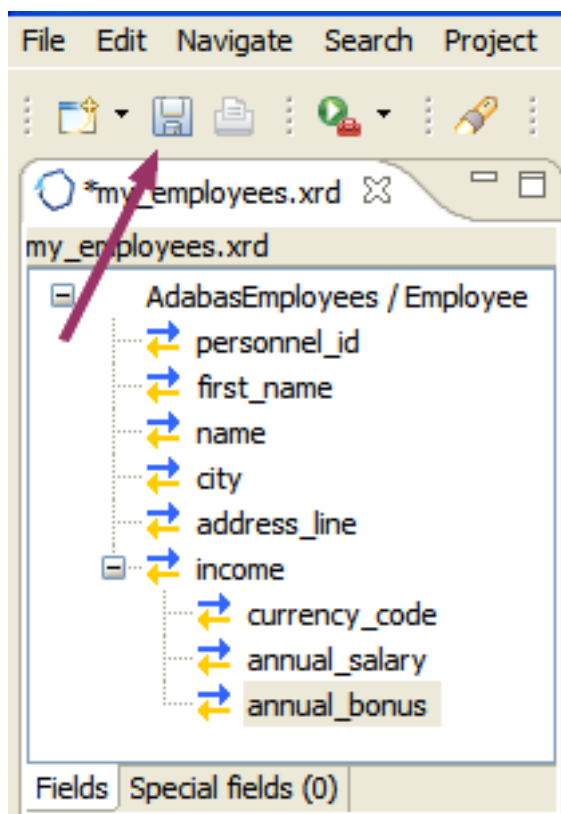
PE Sub Field properties are equivalent to those of "regular" fields. Define the three PE-fields as follows:

The screenshot shows the configuration for the 'annual_salary' field. On the left, a tree view shows the file structure: 'my_employees.xrd' containing 'AdabasEmployees / Employee' with fields 'personnel_id', 'first_name', 'name', 'city', 'address_line', 'income', 'currency_code', and 'annual_salary'. The 'annual_salary' field is selected. The main configuration area for 'annual_salary' includes: XML name: 'annual_salary', int. name: 'AS', Length: '5', max Occurs: '1', direction: 'in/out', internal type: 'packed decimal', key Type: 'none', external type: 'default', ext. modifier: (empty), mime Type: (empty), mime in Field: (empty), and fixed Value: (empty).

max Occurs >0 on the PE-field level denotes a MU within a PE.

The screenshot shows the configuration for the 'annual_bonus' field. On the left, the tree view is similar to the previous screenshot, but 'annual_bonus' is selected. The main configuration area for 'annual_bonus' includes: XML name: 'annual_bonus', int. name: 'AT', Length: '5', max Occurs: '8' (with a red arrow pointing to it), direction: 'in/out', internal type: 'packed decimal', key Type: 'none', external type: 'default', ext. modifier: (empty), mime Type: (empty), mime in Field: (empty), and fixed Value: (empty).

8. The "view" to the "Employees" file is now complete. Save the DataView by selecting the Save button or Ctrl+S.



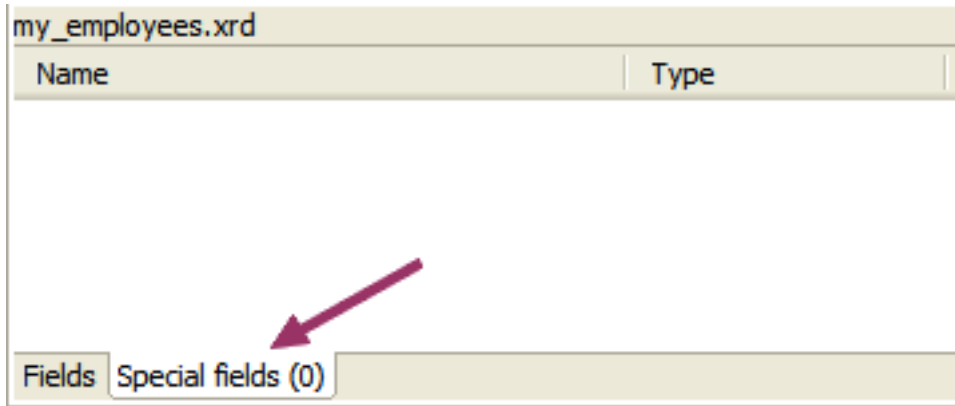
9. **Export** the DataView to the target Portus server.

Editing "special" fields

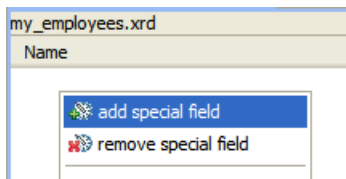
The following "special" fields can be defined for a DataView

- Adabas SuperDescriptor
- Adabas SubDescriptor
- Adabas HyperDescriptor
- Adabas Phonetic Descriptor

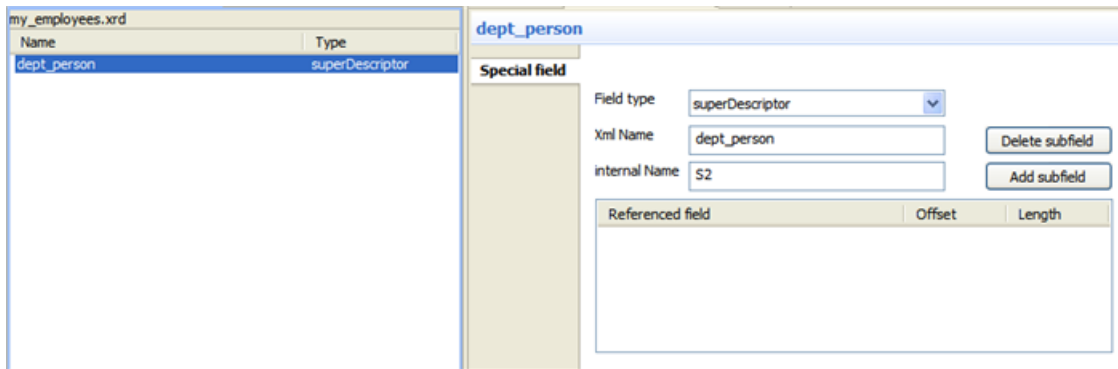
1. In the DataView editor window, click on the "Special fields" tab to add special fields



2. Right-click into the empty editor area, select "add special field"



3. For example we define a special field called "dept_person" with an "internal name" (the Adabas short name) of "S2", being of type "superDescriptor"



4. Click on the "Add subfield" button, a new field element will appear in the subfield table

dept_person

Field type

Xml Name

internal Name

Referenced field	Offset	Length
?	?	?

5. Click on the field value under the "Referenced field" heading, select the field to be added from the list of fields in the dropdown-box. Here we select the "dept" field

dept_person

Field type

Xml Name

internal Name

Referenced field	Offset	Length
dept	?	?
name		
city		
address_line		
income		
dept		

Initial "Offset" and "Length" values will be derieved from the selected field's definition

6. Select the "name" field as the second subfield just like the "dept" field, the result should look like this

dept_person

Field type superDescriptor

Xml Name dept_person Delete subfield

internal Name 52 Add subfield

Referenced field	Offset	Length
dept	0	6
name	0	20

Enhanced Type Conversion

This facility allows the conversion of a string value to an integer equivalent and visa versa. This is analogous to an enumeration i.e. for Jan substitute 1, Feb substitute 2 etc.

In the DataView editor window, select the field.

In the Properties View 2 items need to be changed

1. Open the **Format** dropdown list and select *substitution*.
2. Set the **Format Mask** field to the value of the enumeration string. This should be in the format `strvalue1=num1, strvalue=num2, strvalue3=num3` e.g.

Jan=1, Feb=2, Mar=3, Apr=4, May=5, Jun=6, Jul=7, Aug=8, Sep=9, Oct=10, Nov=11, Dec=12

The screenshot shows the 'Properties' dialog for a field named 'Month'. The dialog is organized into several sections:

- Field Section:** Contains fields for XML Name (Month), Int. Name (Month), Length (10), Max Occurs (1), Direction (input/output), Internal Type (string), Key Type (none), External Type (default), Ext. Modifier, Mime Type, Mime in Field, Fixed Value, Format (substitution), and Format Mask (Jan=1, Feb=2, Mar=3, Apr=4, May=5, Jun=).
- Table Section:** A table with three columns: Scope, Native Attribute, and Value. To the right of the table are 'Add' and 'Remove' buttons.

To save select Ctrl+S or close the DataView editor.

Right-click on the Service and select 'Refresh Service'.

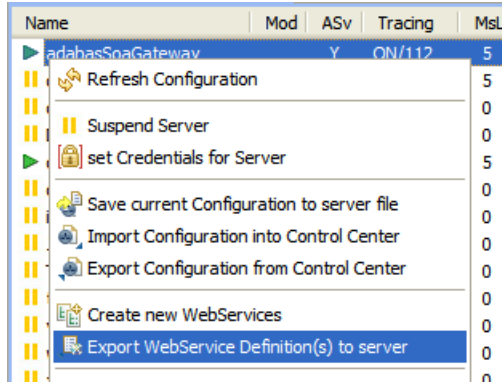
Exporting a DataView to a Portus server

Whether you created a new one or imported and edited an existing DataView, you will now need to export it to the Portus server. DataViews are stored in the "xrd" subdirectory of the server configuration directory.

To export a Portus DataView to a Portus server execute one of the the following procedures:

Using the server based export function

1. Select **Export Resource Definitions** from the context menu of the server you wish to export to.



2. Select the DataView to be exported from the file selection dialog, click **OK**
3. The newly added DataView appears in the list

Using the configuration based export function

1. On the configuration view select **export DataView to server** from the context menu of a Resource definition pointing to the DataView to be exported.

T.	Resource	DataSource Id	DataView
A	adabas_my_employees	Dbid=6, Fnr=11	my_employees
A	adabas_p	90	adabas_photoblobs_f
A	adabas_p	90	adabas_photoblobs
A	adabas_C	11	qe_adabas_employee
A	adabas_C	nr=20009	qe_adabas_employee
A	adabas_v	12	adabas_vehides_view
O	adabas-d	da, Table=city	adabasd city

2. Select the DataView to be exported from the file selection dialog, click **OK**
3. You are now asked if the DataView is to be activated immediately.

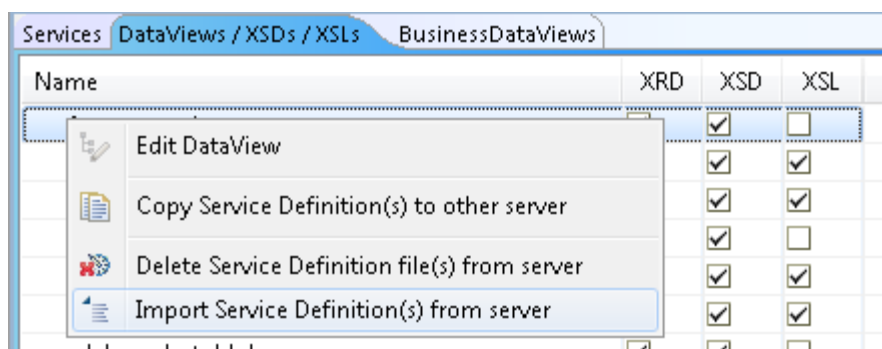
Reply **Yes** if you want to use the newly exported DataView immediately, that is as soon as all requests currently using that DataView have completed, the new copy will be used for all subsequent requests.

A successful export will be indicated by a message in the status line.

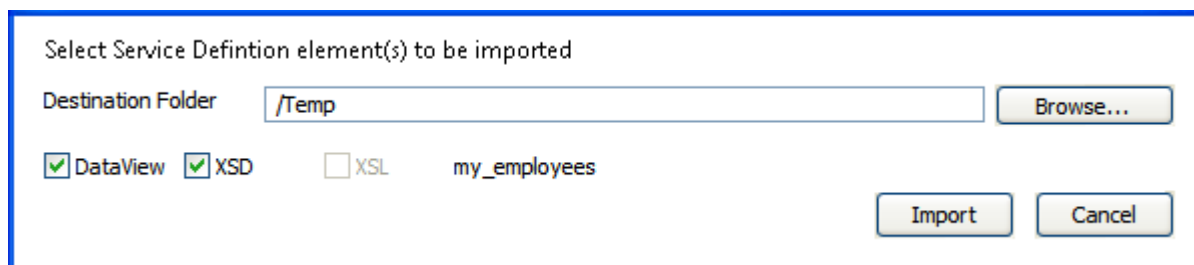
Importing an existing Portus DataView

To import an existing DataView:

1. From the Portus Configuration View's "DataViews / ..." tab, select the elements and right click. Select "Import Service Definition(s) from server"



2. Select the destination folder and click "Import"



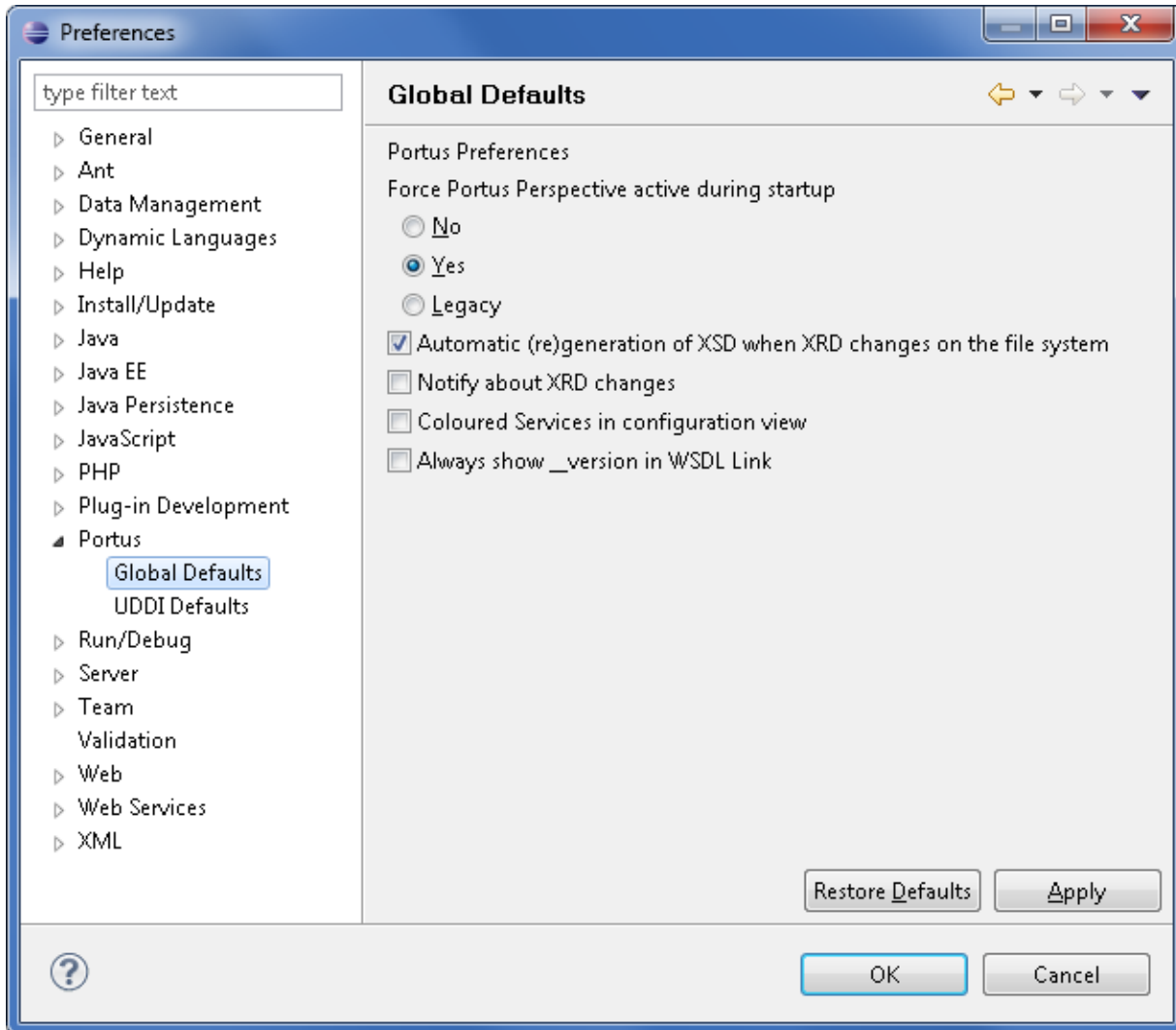
Creating a XML Schema for a DataView

An XML Schema (XSD) can be used to express a schema: a set of rules to which an XML document must conform in order to be considered 'valid' according to that schema. In Portus an XML Schema can be used to validate the input coming from the user. This validation occurs at a very early stage of the processing, so this can be a useful method of enforcing data rules in Portus in a fast and efficient manner. For more information about the structure, rules and possibilities refer to the [W3C XML Schema Specification](#)

There is a one-to-one relationship between the DataView and the XML Schema. For example, if a personnel_id field is present in the DataView, the XML Schema can be used to enforce rules that this field must be an integer of at least, and not more than, 8 digits.

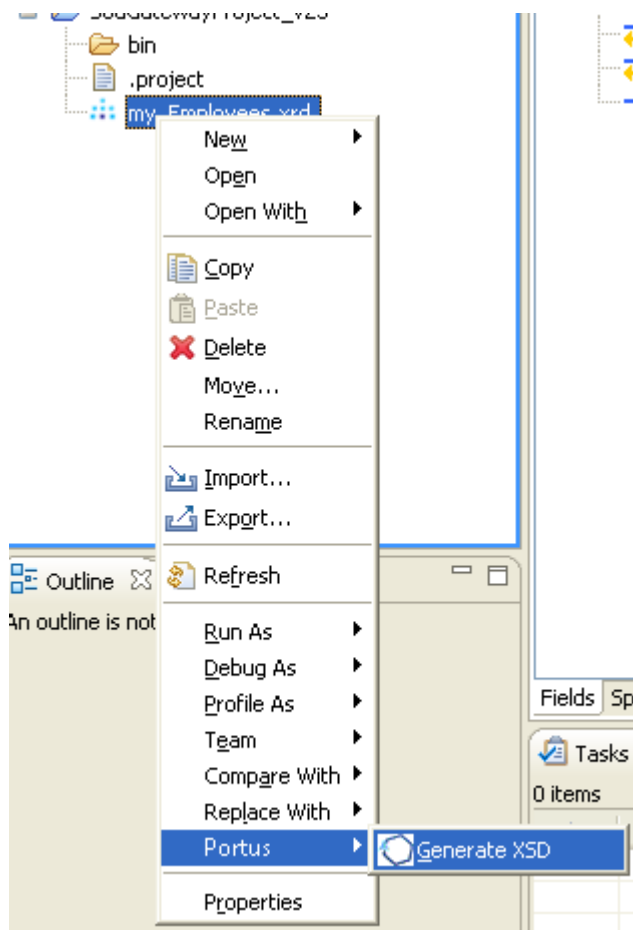
A XML Schema for a DataView can be created

- Automagically when the "Automatic (re)generation of XSD when XRD changes on the file system" preference is enabled. This option takes effect for BOTH local and remote edition of DataViews. See Window -> Preferences -> Portus -> Global Defaults.



- By right-clicking on the DataView in the Package Explorer.

Select **Portus** and then **Generate XSD**



An XML Schema will be created at the same level as the DataView.

⚠ Important: The filenames of the DataView and XML Schema must be identical, the only difference being the file extension (xrd versus xsd)

You may now **export** this XML Schema and/or the DataView to the server.

22

Using Portus with Adabas

- Creating a Portus DataView from an Adabas FDT 217
- Create DataView(s) from a SYSOBJH unload file 217

Portus interfaces with Adabas in a smooth, seamless way, implementing almost all features of both the Mainframe and OpenSystems version of Adabas:

- Sequential / direct access with various positioning options
- Transactionality
- LOB (large objects) access

All Portus database access operations (see the SOAP reference for more information) are supported:

Operation	Function
get	Retrieve a single record, access via descriptor (defined as primary key in the DataView) or ISN
list	Return 1 to n records in a single result set, access by simple key criteria.
select	Return 1 to n records in either a single result set (non-conversational mode) or the first set ("chunk") of records (in conversational mode), access by complex key criteria (including from/to ranges, OR conditions, etc.)
selectNext	Return the next n records for a conversation initiated by a "select" operation.
selectEnd	Terminate a conversation initiated by a "select".
selectCount	Return the number of records matching the specified key criteria.
add	Add a new record to an Adabas file.
update	Update a record accessed by key or ISN.
delete	Delete a record accessed by key or ISN.

Requests can be fine tuned with SOAP header options. In addition to the general options the following Adabas specific headers are available:

SOAGateway_Internal_Adabas_ISN	On certain operations an ISN (Adabas Internal Sequence Number) can be used in lieu of a primary key to retrieve a record.
SOAGateway_Internal_Adabas_StartAtISN	In addition to the specification of key data this header option can be used to further limit the result set by setting a starting ISN for LIST and SELECT operations.
SOAGateway_Internal_Adabas_Read_Direction	Specify an "A" (the default) for ascending retrieval, a "D" for descending.
SOAGateway_Internal_Adabas_Sort_Order	The result set can be sorted by up to three fields. Specify their XML names, comma separated, with this header option
SOAGateway_Internal_Adabas_Password	The password required to access an Adabas file secured with ADASCR (Adabas Security). This password is different from that in the global "soap security" section.


SOAGateway_Internal_Adabas_Multifetch_Limit	Limit the number of records to be retrieved by a "multifetched" operation (LIST or SELECT / SELECTNEXT) with a single call to Adabas.
---	---

Before a Portus Service can be used, the mapping between the physical layout (the Adabas FDT) and the Portus (XML) representation must be in place. This mapping is called the Portus 'DataView'. In addition to the manual approach described in detail in the Data Views section, there are a number of semi-automated methods aiding in the generation of Portus DataViews for Adabas resources

1. [Dynamically create a DataView from an Adabas FDT](#)
2. [Create DataView\(s\) from a SYSOBJH unload file](#)

Creating a Portus DataView from an Adabas FDT

A new DataView can be generated from an Adabas file using the 'Service Discovery wizard', which will not only generate the DataView (XRD), but also a XSD, plus the Portus Resource definition itself.

1. Use the 'Service creation Wizard' to generate the Service(s)
2.  **Important:** As Adabas only knows about the 2-character "internal field names", you will have to edit the DataView and assign "friendly names" to all fields.

Create DataView(s) from a SYSOBJH unload file

DataView(s) for Portus Adabas resources can be generated/imported from SYSOBJH extracts.

- Use the 'Service creation Wizard' to generate the Service(s)

23

Using Portus with Natural

- Natural Service generation 220
- Other ways of defining a Service 220
- Preparing the Natural environment 222
- Type mapping 246

Natural subprograms can be exposed as (Web) Services through , a DataView (XRD) maps the parameters passed to the called subprogram.

The following steps are required:

- Define a driver and prepare the environment, described in separate sections for [Mainframe](#), [Unix / Linux](#) and [Windows](#) platforms.

For prerequisites and compatibility information please read [here](#)

- Create the mapping *PDA -> DataView*, the following sections outline the required steps.

Natural Service generation

The preferred way for creating DataView(s), the mapping from a Natural Parameter Data Area (PDA) to the Portus structures exposing the Service via a WSDL, is through an automatic "discovery" process as described [here](#)

Other ways of defining a Service

In addition to the automatic generation process outlined above, Services can be created / defined as follows.


Define a Natural Service manually

1. Add a new Portus Service, specifying the following:

The screenshot shows two sections of a configuration interface:

- Service Definition:**
 - Name: Natural_Service
 - Driver: Natural_Driver (dropdown menu)
 - Read-only:
 - SBCS-Codepage: [empty text box]
 - MBCS-Codepage: [empty text box]
 - Status: Test (text box) change to [empty dropdown menu]
 - WSDL URL is .. http://localhost:56005/Natural_Service?WSDL
- Service Identification and options:**
 - Library: SYSSOAEX
 - Program: ENVIRON

- A Service name
- Select a driver of type "Natural" from the dropdown list
- For "Service Identification and Options":
 - The Natural Library name in the "Library" field
 - The Natural subprogram name in the "Program" field

 **Note:** For Mainframe Natural only the Program name is to be entered, because the library is specified on the driver, not the individual Service

Create Service(s) and DataView(s) from Natural sources or a SYSOBJH unload file

In addition to the manual approach described in detail in the Data Views section, Services and DataView(s) for Portus can be generated/imported from SYSOBJH extracts or directly from Natural sources.

For a detailed description of the process refer to the Service Creation section

Preparing the Natural environment

Certain steps are required to enable the Portus server to call Natural subprograms, this section describes them.


- [Prepare the Natural environment on Linux / UNIX](#)
- [Prepare the Natural environment on Windows](#)
- [Prepare the Natural environment on the Mainframe](#)

Prepare the Natural environment on Linux / UNIX

Using Portus in an environment protected by Natural Security (NSC)


- Specify a valid *LOGON* *<library>,<userid>,<password>* in the *STACK* parameter of the parameter module used with the Portus Natural driver.
- Library *SYSEXT* must be defined as a *STEPLIB* to library *SYSSOA*.

Defining the driver

 **Important:** The Portus interface to Natural requires the interface library *libnatural.so* in the *LD_LIBRARY_PATH* (*LIBPATH* on AIX). However, this shared library is contained in *\$NATDIR/\$NATVERS/bin*, and not in the Natural */lib* directory. Due to this, either the Natural */bin* directory must be added to search path variable, or *libnatural.so* copied / moved to a directory contained in the search path variable.

In the Portus Servers View, select the server you want to define a Natural driver for, in the Properties View select the *Drivers* tab, click the *Add...* button.

From the Driver Configuration Dialog select "Natural_Driver"

 **Important:** When defining a Natural driver for the first time, make sure to check the "Show (post-setup) Driver Information" box because only this will allow you to extract the *INPL* file containing the Portus Natural interface- and example programs.

The following values are preset, change as required:

Driver Title	<input type="text" value="NATURAL"/>	<input type="button" value="Save"/>
Driver Name	<input type="text" value="Natural_Driver"/>	
Driver DLL	<input type="text" value="naturalDriver.so"/>	<input type="button" value="Dismiss"/>
SBCS codepage	<input type="text" value="ascii"/>	
MBCS codepage	<input type="text" value="utf16"/>	

Only the *Driver DLL* parameter is to be entered as is, you are free to choose any *Driver Name* you wish, set the *SBCS codepage* and *MBCS codepage* parameters according to your local or internationalization requirements.

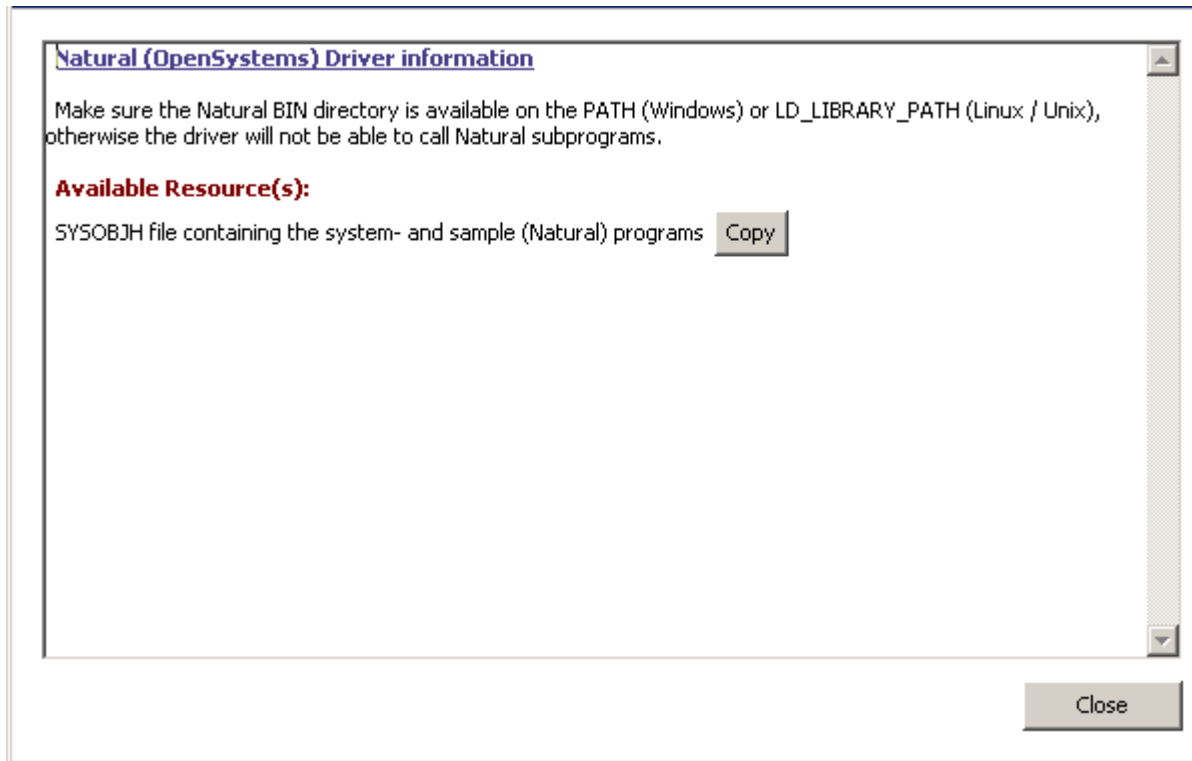
Click the *Save* button, you will now be asked to (optionally) enter Natural initialization parameters, for example *PARAM=MYPARM* to use a Natural parameter module other than NATPARAM.

Driver has options (which may be mandatory), please specify and/or click 'Save'

Driver Title	<input type="text" value="NATURAL"/>	<input type="button" value="Save"/>
Driver Name	<input type="text" value="Natural_Driver"/>	
Driver DLL	<input type="text" value="naturalDriver.so"/>	<input type="button" value="Dismiss"/>
SBCS codepage	<input type="text" value="ascii"/>	
MBCS codepage	<input type="text" value="utf16"/>	
InitParms	<input type="text"/>	

Click the *Save* button again, the newly defined driver will appear in the Driver Properties section.

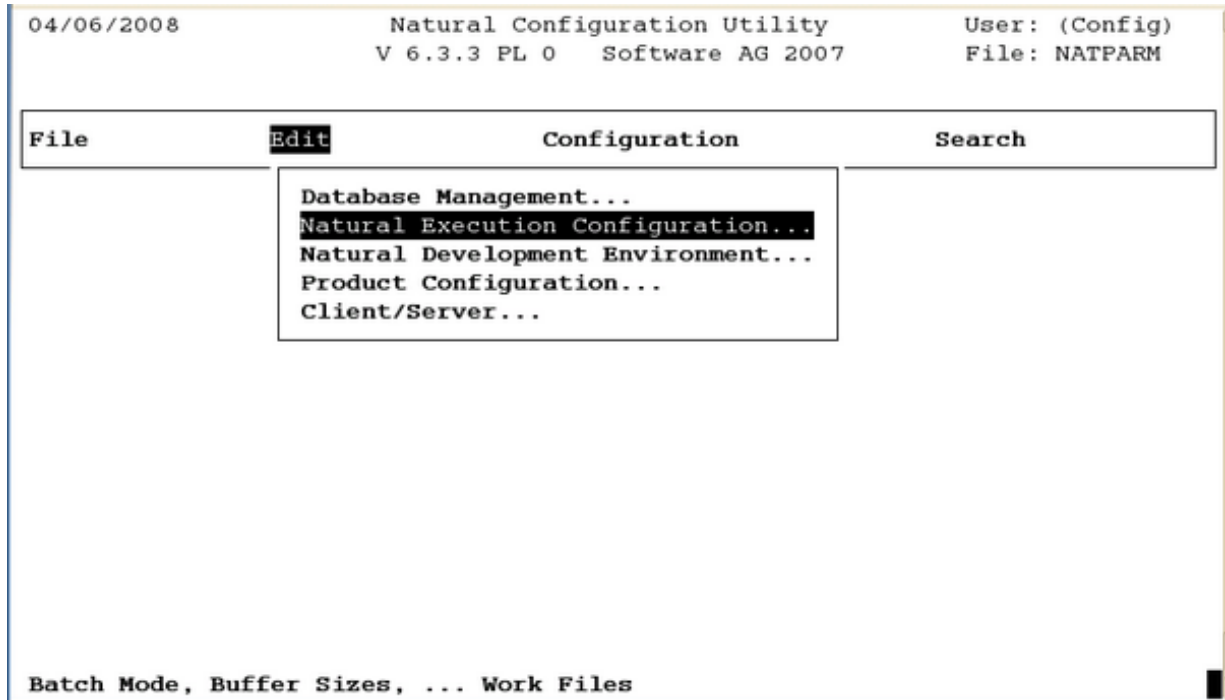
When "Show (post-setup) Driver Information" has been selected, the following information box will be shown, and allow the extraction of the SYSOBJH file containing the Portus Natural interface- and example programs.



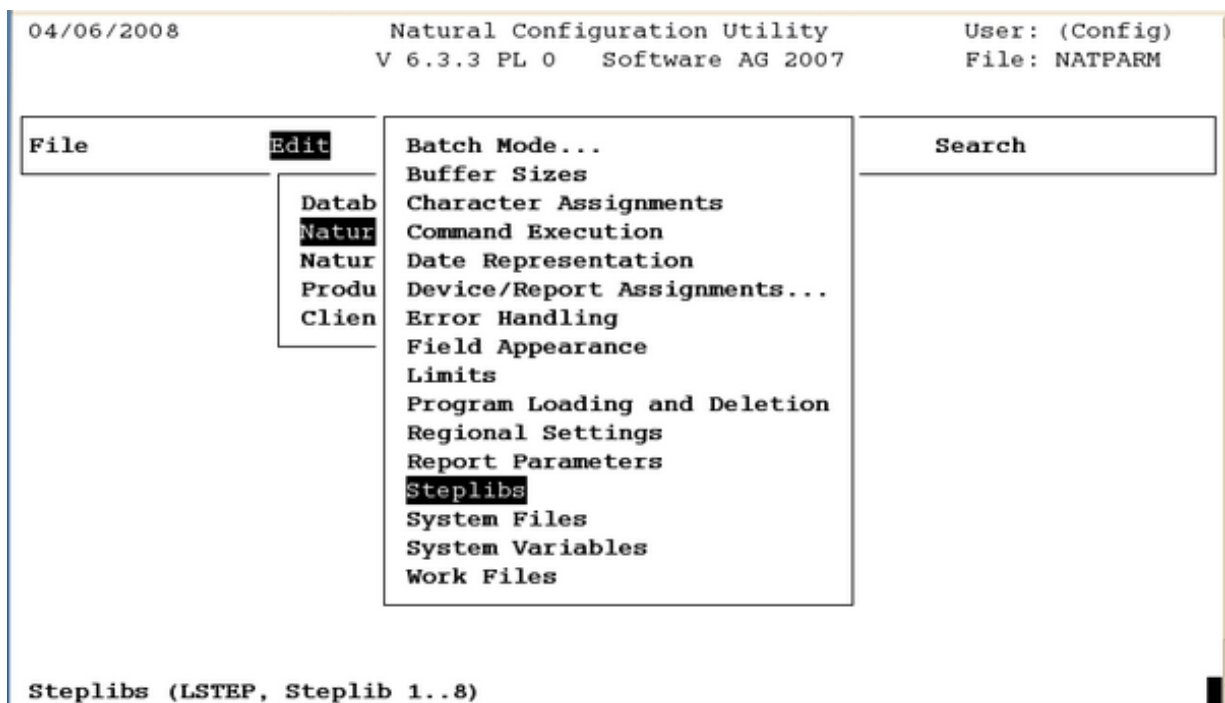
Loading the Portus Natural system- and demo programs on Linux / UNIX

The following steps are required to load the Portus Service creation and demo programs in a Linux / Unix environment

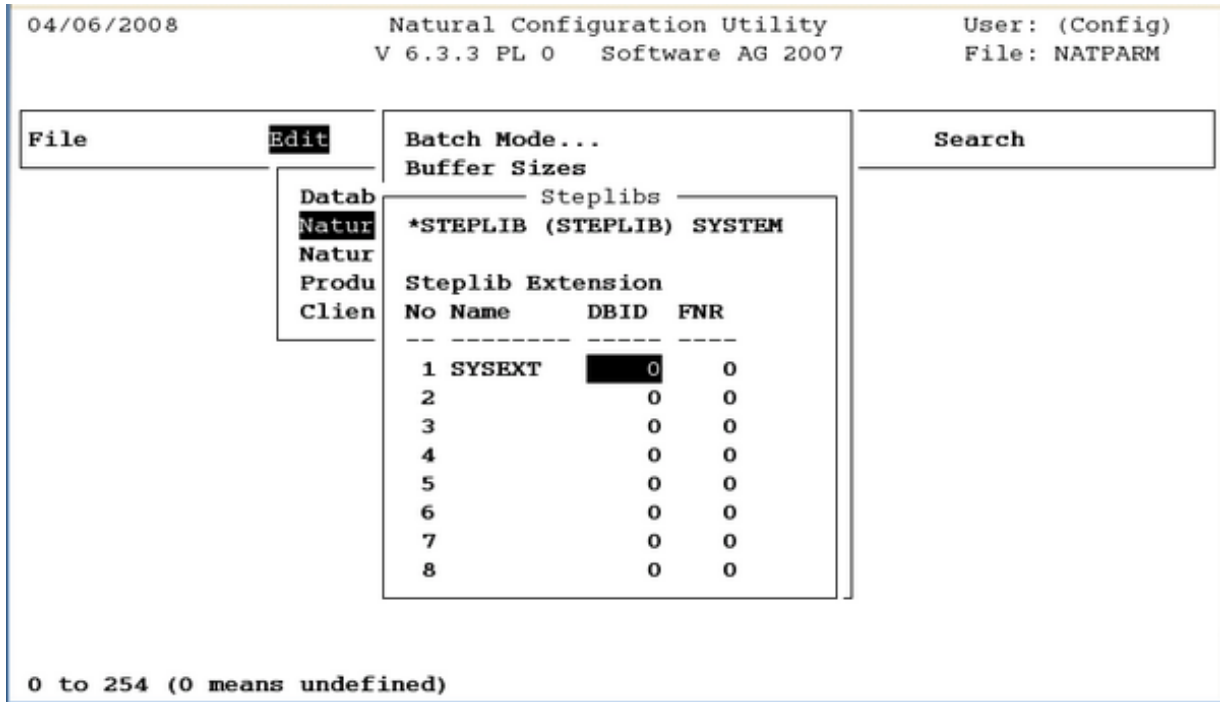
- Start the Natural Configuration Utility (*natparm*)
- For the specific Natural Parameter file to be used when accessing Natural from Portus (default: "NATPARAM"), select "Edit" -> "Natural Execution Configuration"



- Select "Steplibs"

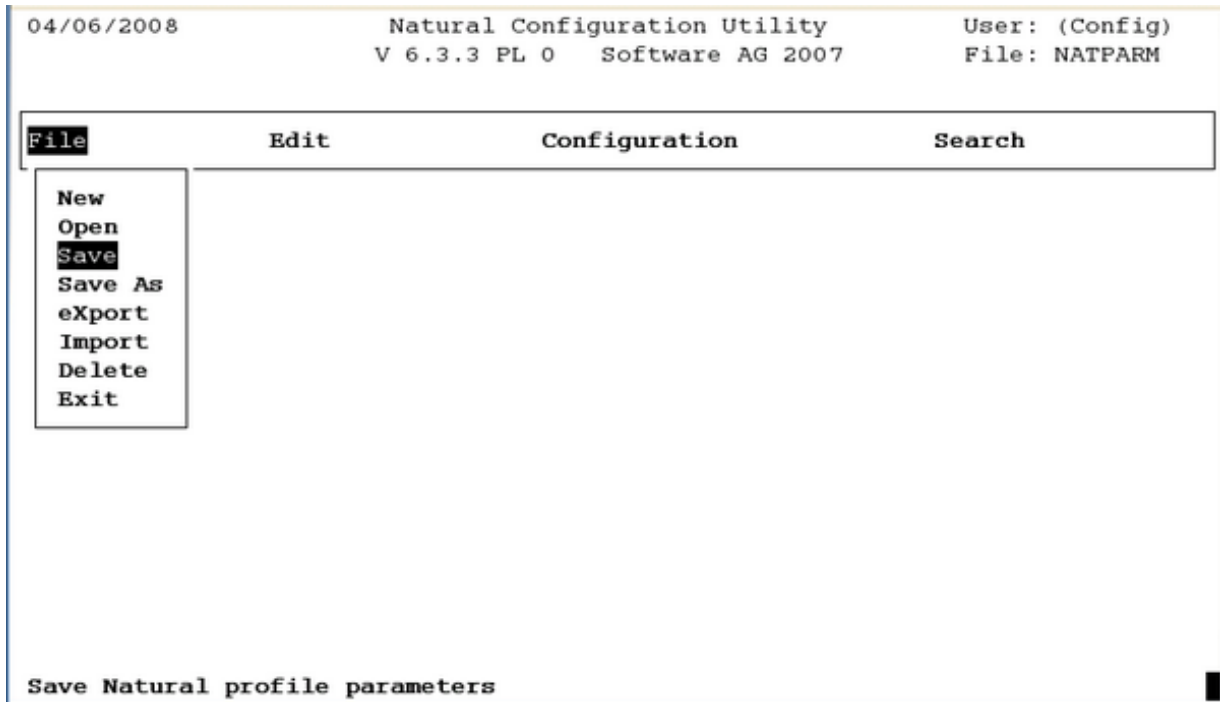


- Define library "SYSEXT" as a "Steplib Extension"



- Press the "return" key to save the Steplib modification(s), then use the "Esc" key to close all windows.

Select "File" -> "Save" to preserve the adjusted Steplib setting



- Quit the Natural Configuration Utility with “File” -> “Exit”.
- Before entering Natural to load the Portus system- and demo-programs the environment variable WRKF3 needs to be set to allocate work file 3 for the “Natural Object Handler”

```
/sag/tmp> export WRKF3=network3.sag
```

Here, as an example, the work file name is defined as “network3.sag” in the current working directory.

- Start Natural, tab to the “Direct” entry on the main menu, enter “SYSOBJH” to start the Natural Object Handler:

```
2008-06-04          NATURAL          Library: SYSTEM
22:56:25           V 6.3.3 PL 0   Software AG 2007   Mode  : REPORT
User: SAG                               Work Area : empty

Library   Direct   Services   OS         Fin
-----
          Direct Command
          SYSOBJH

Enter Command or NATURAL Program Name
```

- Select the Object Handler “Load” function

```

23:01:04          ***** Natural Object Handler *****          2008-06-04
User SAG              - Main Menu -

  Select the desired function:

  _  Unload objects or a whole application from your Natural environment
  X  Load objects or an application into your Natural environment
  █  Scan work file contents
  _  View objects in the Natural environment
  _  Administrate the Object Handler environment,
      process Workplans and direct commands

  Mark this field to avoid Object Handler wizards:

  _  Advanced user

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      Exit  Unld  Load  Scan  View  Admin      Cnds      Canc

```

- Select “Load objects from Natural work file(s)”

```

23:04:55          ***** Natural Object Handler *****          2008-06-04
User SAG              - Load Wizard -

  You can load objects from transfer and non-transfer work files
  into the Natural environment or execute a command procedure.

  Select the desired function:

  █  Load objects from Natural work file(s) .
  _  Start Object Handler command procedure.

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      Exit      Back  Next      Cnds      Canc

```

- Enter the Portus OBJH “Work file” name and select “Portable work file” and “Set additional options”

```

23:09:50          ***** Natural Object Handler *****          2008-06-04
User SAG          - Load Wizard, Options -

  If the work file contains data in Transfer format, mark this field:
    _ Transfer format

  If you want to use a portable work file, mark this field:
    X Portable work file

  Enter the name of the load file. If the path and the name
  do not fit into the field, press PF11 to specify a longer value.
  Work file /tmp/ASG411_000003_OBJH.sag_____

  Select the desired option to be used.
  Mark 'Set additional options', to use additional options.
    X Use default options                      X Set additional options

    █ Use Option Workplan   Name _____   _ List Option Workplan
                                     _ Select Option Workplan

Please enter options.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit OpSet SelWP LstWP Back Next      Cnds WorkF Canc

```

- █ Either enter a "Report file" name, or deselect the "Write report" option

```

23:14:36          ***** Natural Object Handler *****          2008-06-04
User SAG          - Load Options -

  Work File Options:
  Work file:      /tmp/ASG411_000003_OBJH.sag_____

  Write report    █ Start new report
  Report file:   $WRKF4_____
  _ Write restart information
  Restart file:  _____

  Replace Options:          XREF Options:          FDIC Settings:
  X Do not replace          _ Yes          DBID/FNR _____ / _____
  _ Replace all            X No          Password
  _ Replace obsolete       _ Force       Cipher
  _ Replace except newer   _ Doc         FSEC Settings:
                          _ Special          DBID/FNR _____ / _____
                          Password
                          Cipher

  Number to process:
  _____

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit WorkF RepoF RestF          Cnds      Canc

```

- █ Hit the "return" key a number of times, the following screen will finally indicate what Object handler operation will be carried out:

```

23:18:41          ***** Natural Object Handler *****          2008-06-04
User SAG              - Load Wizard -

The following command will be processed.
To change the settings, press PF7 (Back). To confirm the settings and
to execute the command, press ENTER or PF8 (Next).

LOADALL WHERE NOREPORT WORKFILE /tmp/ASG411_000003_OBJH.sag

Please press PF7 (Back) or ENTER/PF8 (Next).
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      Exit      Save      Back Next      Cnds      Canc
    
```

- Hit the “return” key once again, the Portus Natural system- and demonstration objects will be loaded, the following message indicates the completion of the load operation.

```

23:18:41          ***** Natural Object Handler *****          2008-06-04
User SAG              - Load Wizard -

The following command will be processed.
To change the settings, press PF7 (Back). To confirm the settings and
to execute the command, press ENTER or PF8 (Next).

LOADALL WHERE NOREPORT WORKFILE /tmp/ASG411_000003_OBJH.sag

Function completed successfully.

Press ENTER to continue.

Press ENTER to continue.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      Exit      Cnds      Canc
    
```

- Press the Enter (= return) key again, and one more time to quit the load function, followed by the “Esc” key to terminate the Natural Object Handler.


```

23:18:41          ***** Natural Object Handler *****          2008-06-04
User SAG          - Load Wizard -

The following command will be processed.
To change the settings, press PF7 (Back). To confirm the settings and
to execute the command, press ENTER or PF8 (Next).

LOADALL
Function
Press ENT


Do you want to continue the load function?
 Yes   X No

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help           Exit                               Cnds           Canc

```

Prepare the Natural environment on Windows

The Portus interface to Natural requires the interface library *natni.dll*.

 **Important:** Make sure the `%NATDIR%/NATVERS/bin` directory, which contains *natni.dll*, is in the path.


Using the Portus in an environment protected by Natural Security (NSC)

- Specify a valid `LOGON <library>,<userid>,<password>` in the `STACK` parameter of the parameter module used with the Portus Natural driver.
- Library `SYSEXT` must be defined as a `STEPLIB` to library `SYSSOA`.

Defining the driver

In the Portus Servers View, select the server you want to define a Natural driver for, in the Properties View select the *Drivers* tab, click the *Add...* button.

From the Driver Configuration Dialog select "Natural_Driver"

 **Important:** When defining a Natural driver for the first time, make sure to check the "Show (post-setup) Driver Information" box because only this will allow you to extract the `INPL` file containing the Portus Natural interface- and example programs.

The following values are preset, change as required:

Driver Title	<input type="text" value="NATURAL"/>	<input type="button" value="Save"/>
Driver Name	<input type="text" value="Natural_Driver"/>	
Driver DLL	<input type="text" value="naturalDriver.so"/>	<input type="button" value="Dismiss"/>
SBCS codepage	<input type="text" value="ascii"/>	
MBCS codepage	<input type="text" value="utf16"/>	

Only the *Driver DLL* parameter is to be entered as is, you are free to choose any *Driver Name* you wish, set the *SBCS codepage* and *MBCS codepage* parameters according to your local or internationalization requirements.

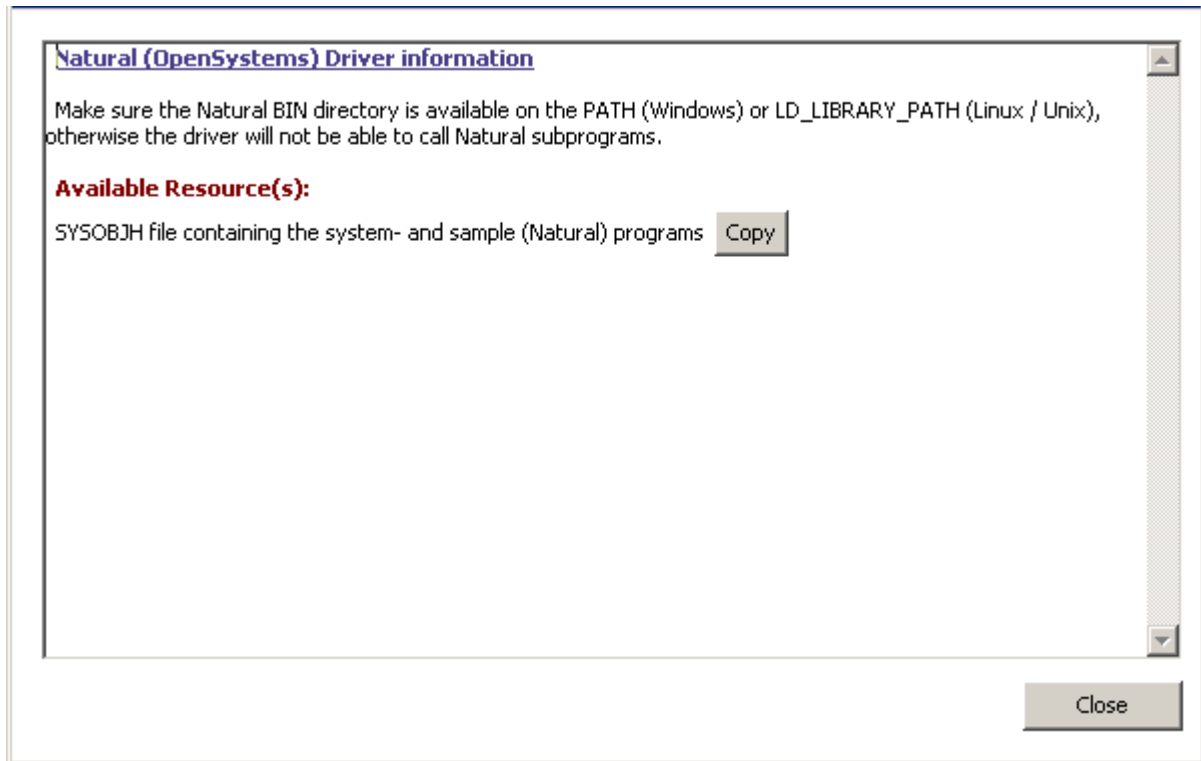
Click the *Save* button, you will now be asked to (optionally) enter Natural initialization parameters, for example *PARAM=MYPARM* to use a Natural parameter module other than NATPARAM.

Driver has options (which may be mandatory), please specify and/or click 'Save'

Driver Title	<input type="text" value="NATURAL"/>	<input type="button" value="Save"/>
Driver Name	<input type="text" value="Natural_Driver"/>	
Driver DLL	<input type="text" value="naturalDriver.so"/>	<input type="button" value="Dismiss"/>
SBCS codepage	<input type="text" value="ascii"/>	
MBCS codepage	<input type="text" value="utf16"/>	
InitParms	<input type="text"/>	

Click the *Save* button again, the newly defined driver will appear in the Driver Properties section:

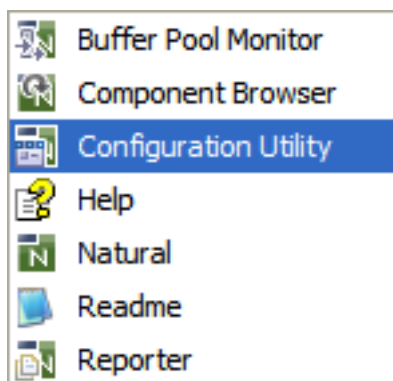
When "Show (post-setup) Driver Information" has been selected, the following information box will be shown, and allow the extraction of the SYSOBJH file containing the Portus Natural interface- and example programs.



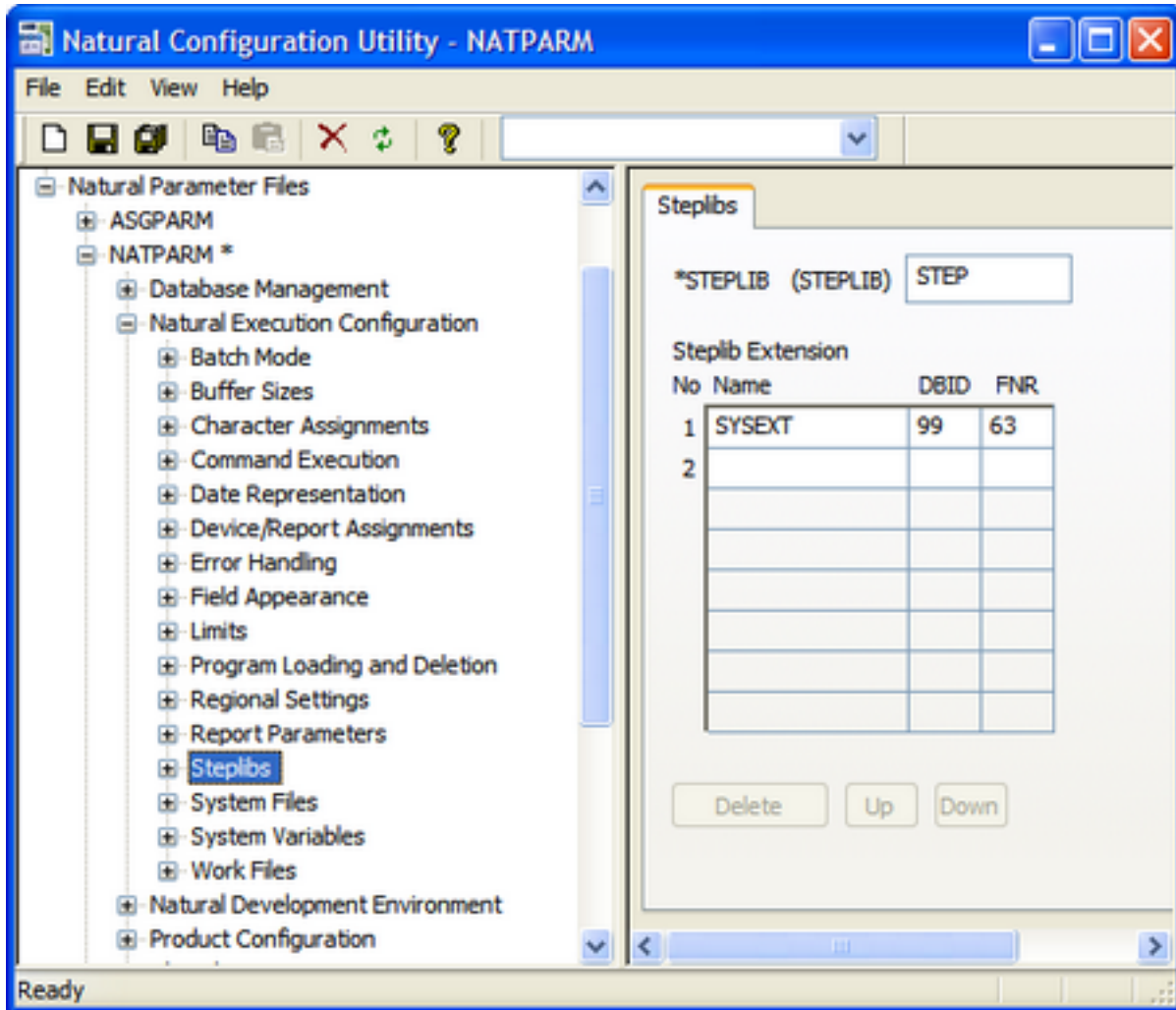
Loading the Portus Natural system- and demo programs on Windows

The following steps are required to load the Portus Service creation and demo programs in a Windows environment

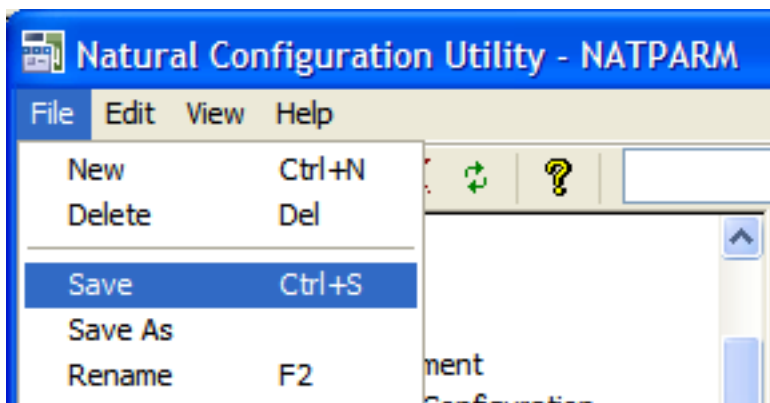
- Start the Natural Configuration utility



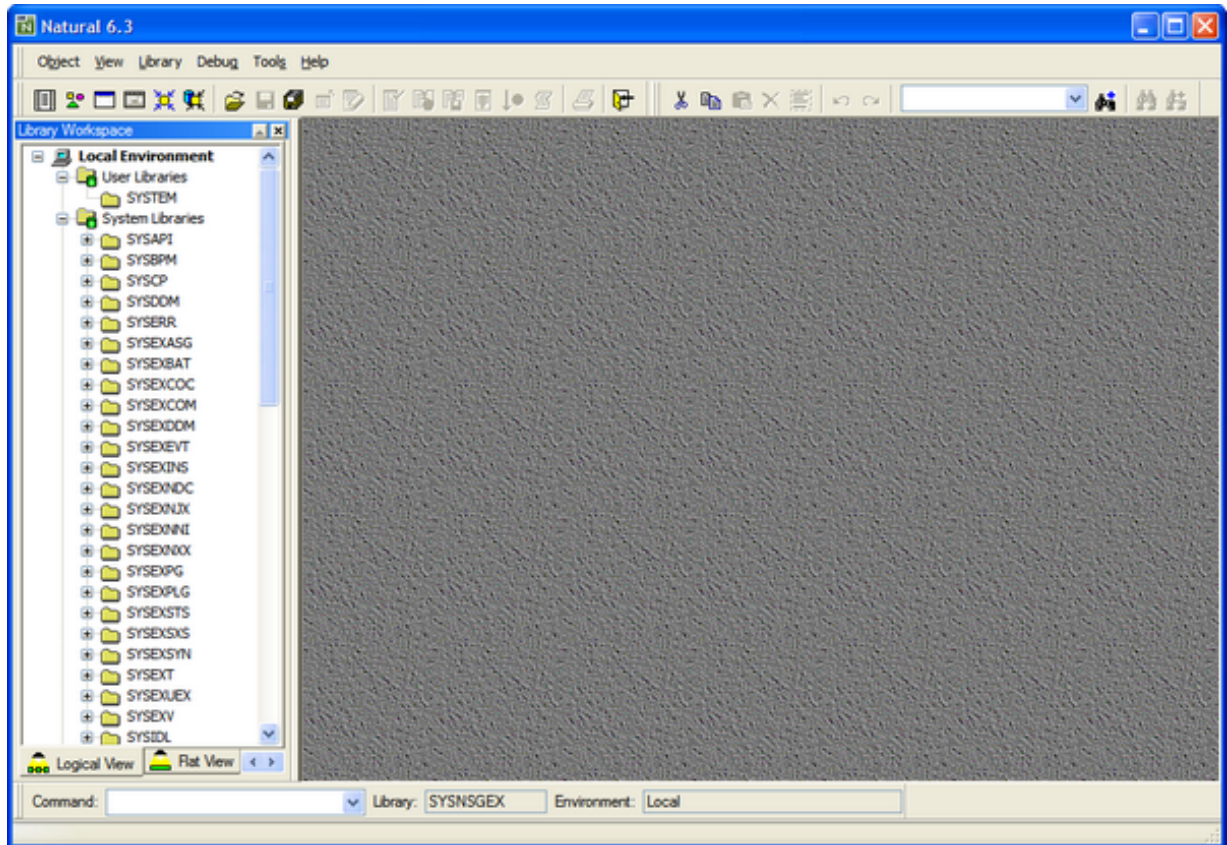
- For the specific Natural Parameter file to be used when accessing Natural from Portus (default: "NATPARAM"), under "Natural Execution Configuration" -> "Steplibs", add library "SYSEXT" as a "Steplib Extension"



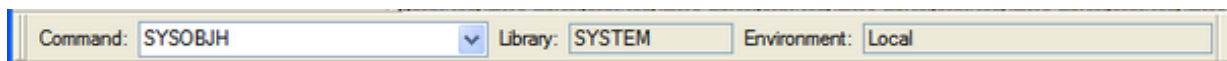
- Click the "Save" button, or select File -> Save



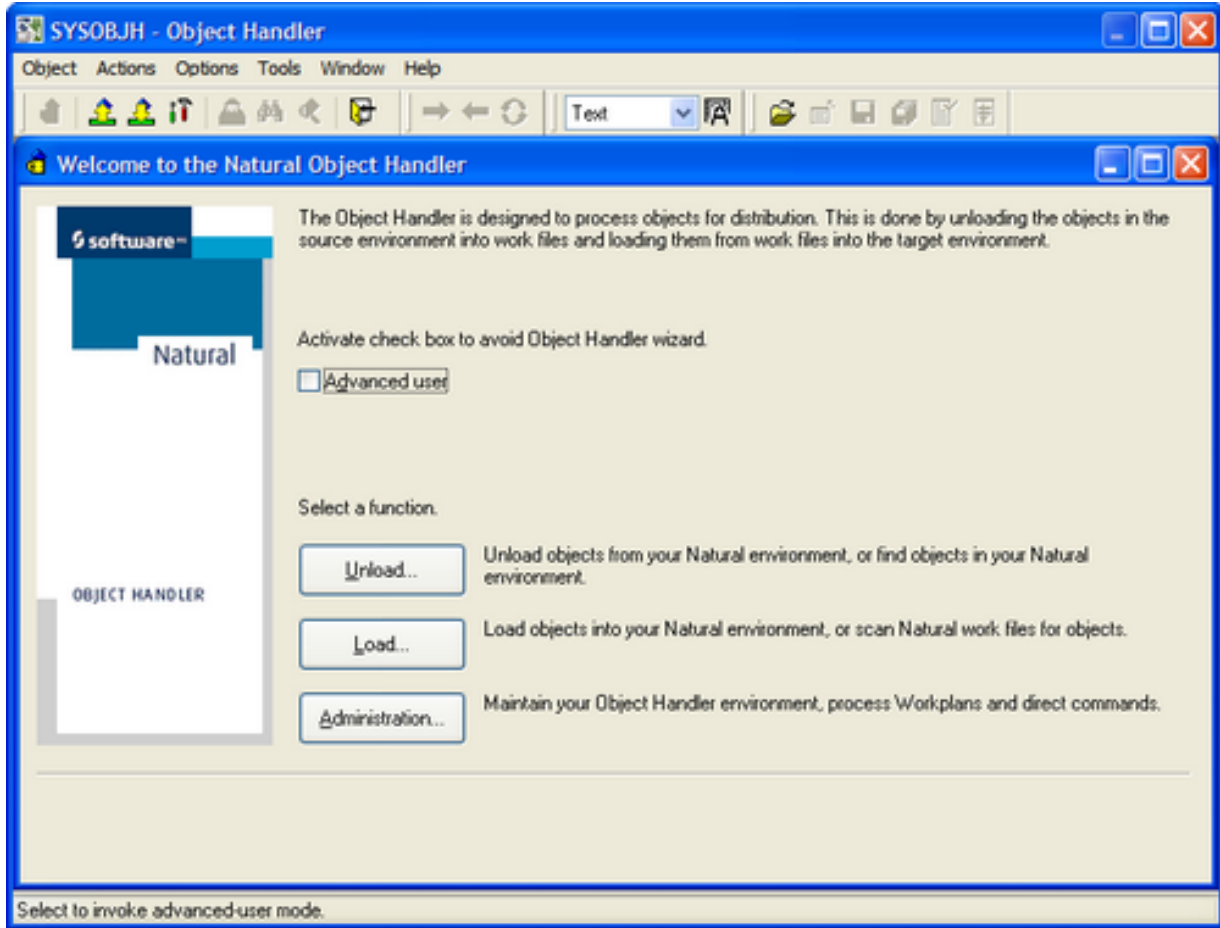
- Start Natural Studio, the initial display will be something like this



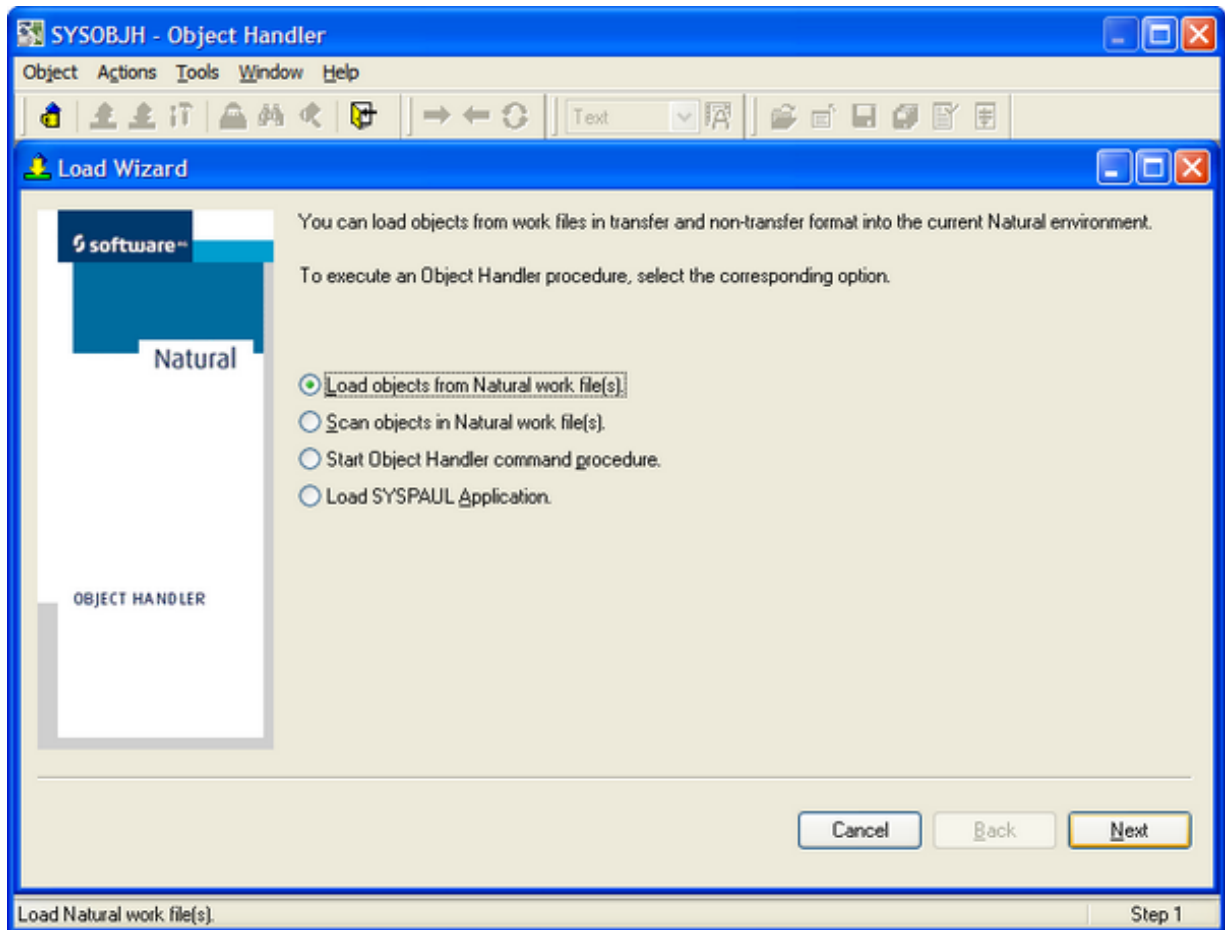
- Import the Portus Natural interface and the demo programs. For this, start the Natural Object Handler – enter the command “SYSOBJH” in Natural Studio's Command line



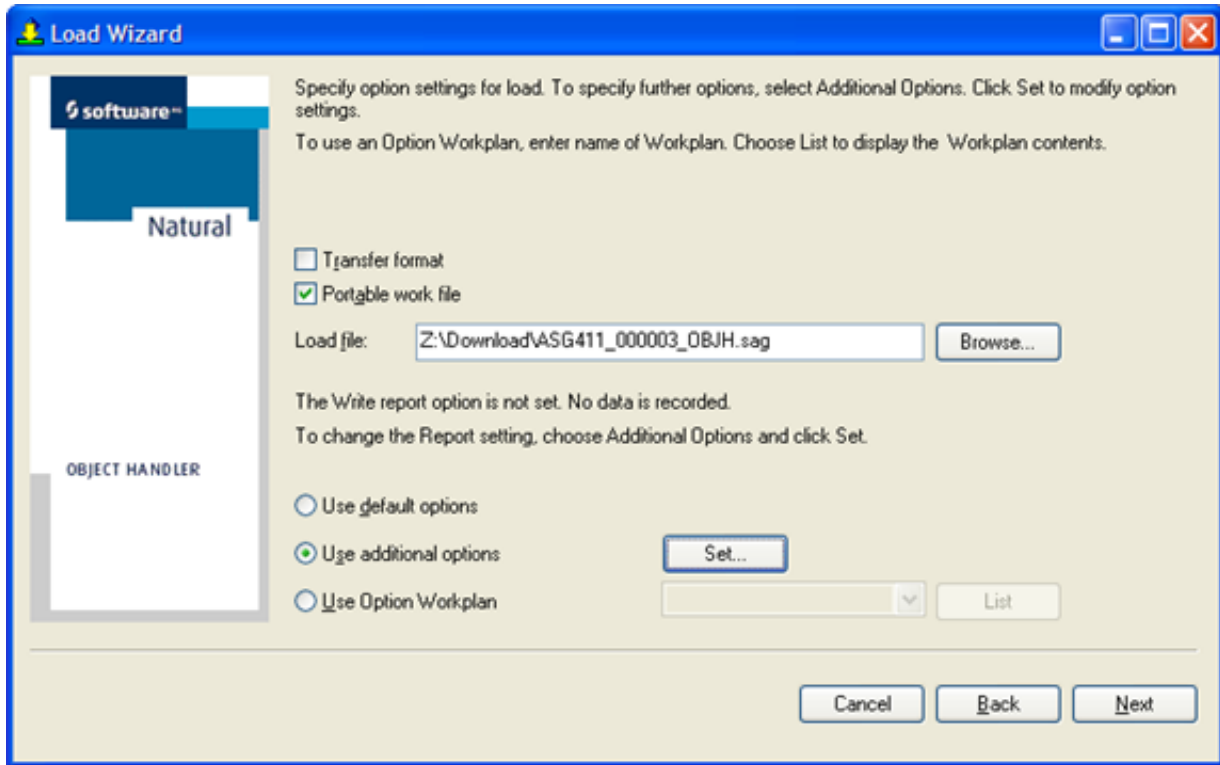
- Select the *Load* function



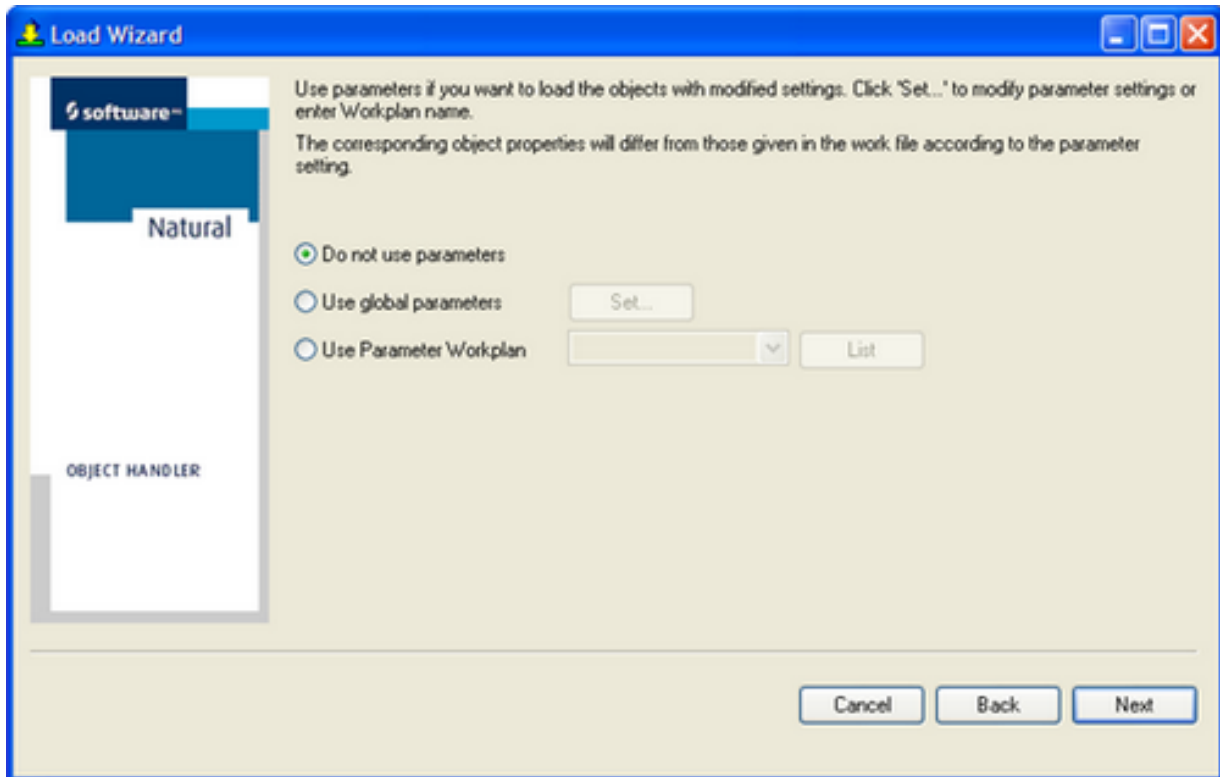
- Select “Load objects from Natural work file(s)”



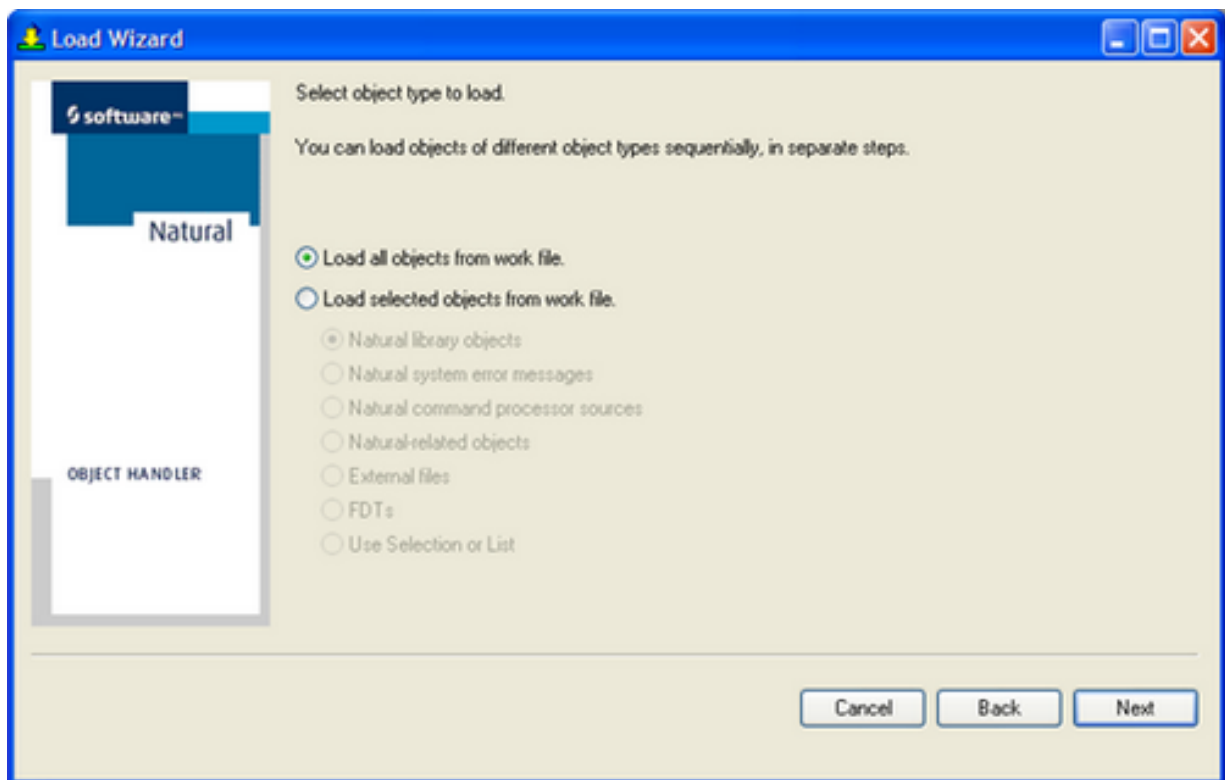
- Specify (or browse for) the Portus OBJH unload file, check *“Portable work file”*, select *“Use additional options”*, click *“Next”*



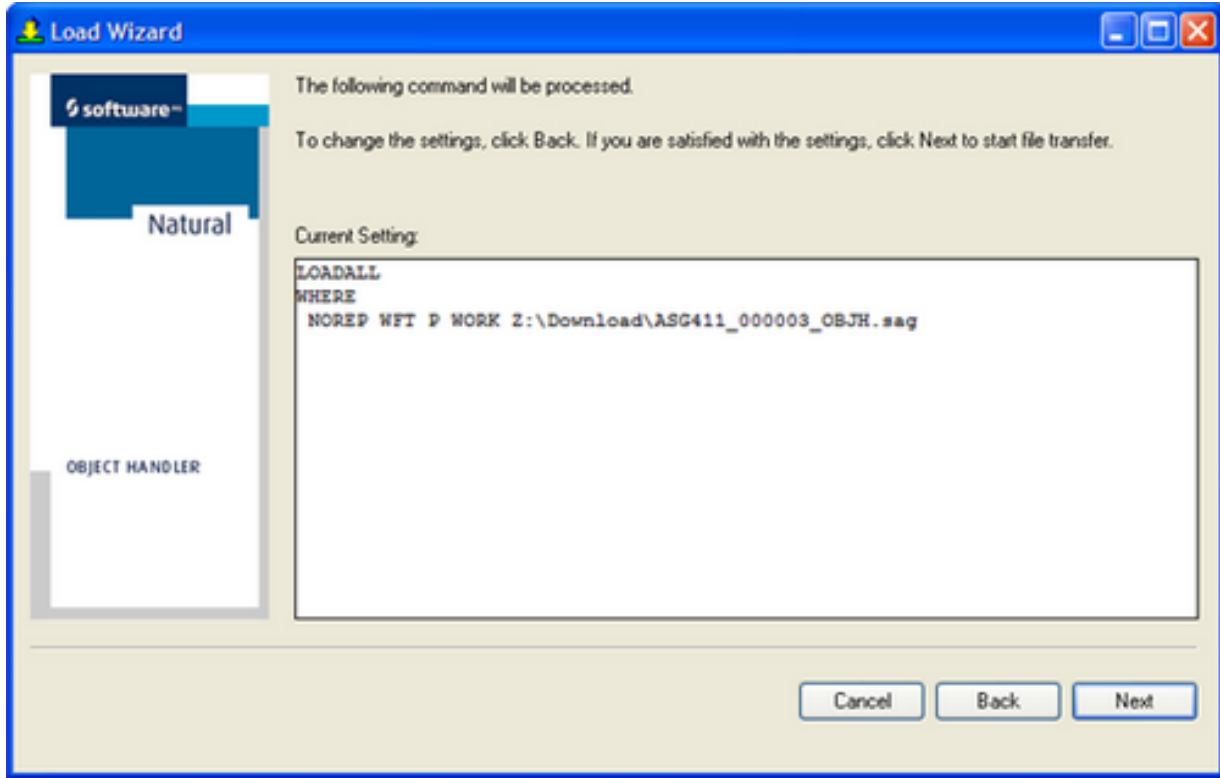
- click "Next"



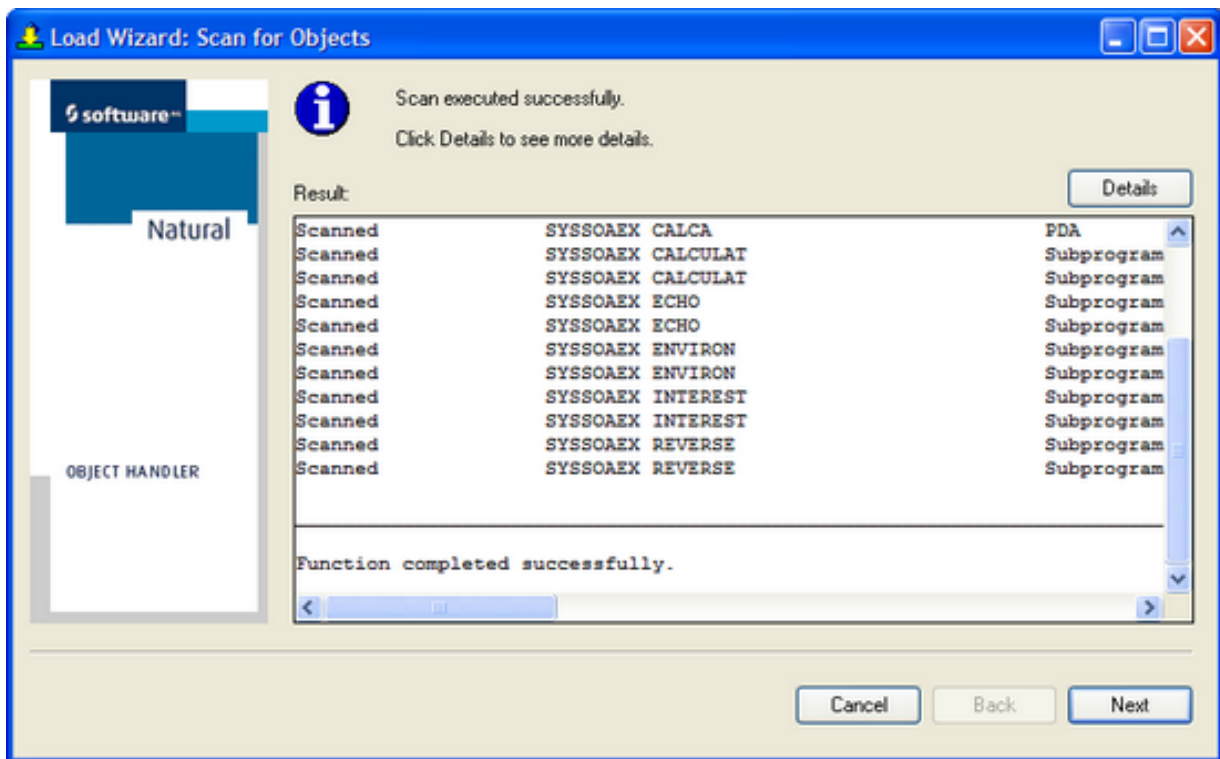
- Click “Next” to “Load all objects from work file”



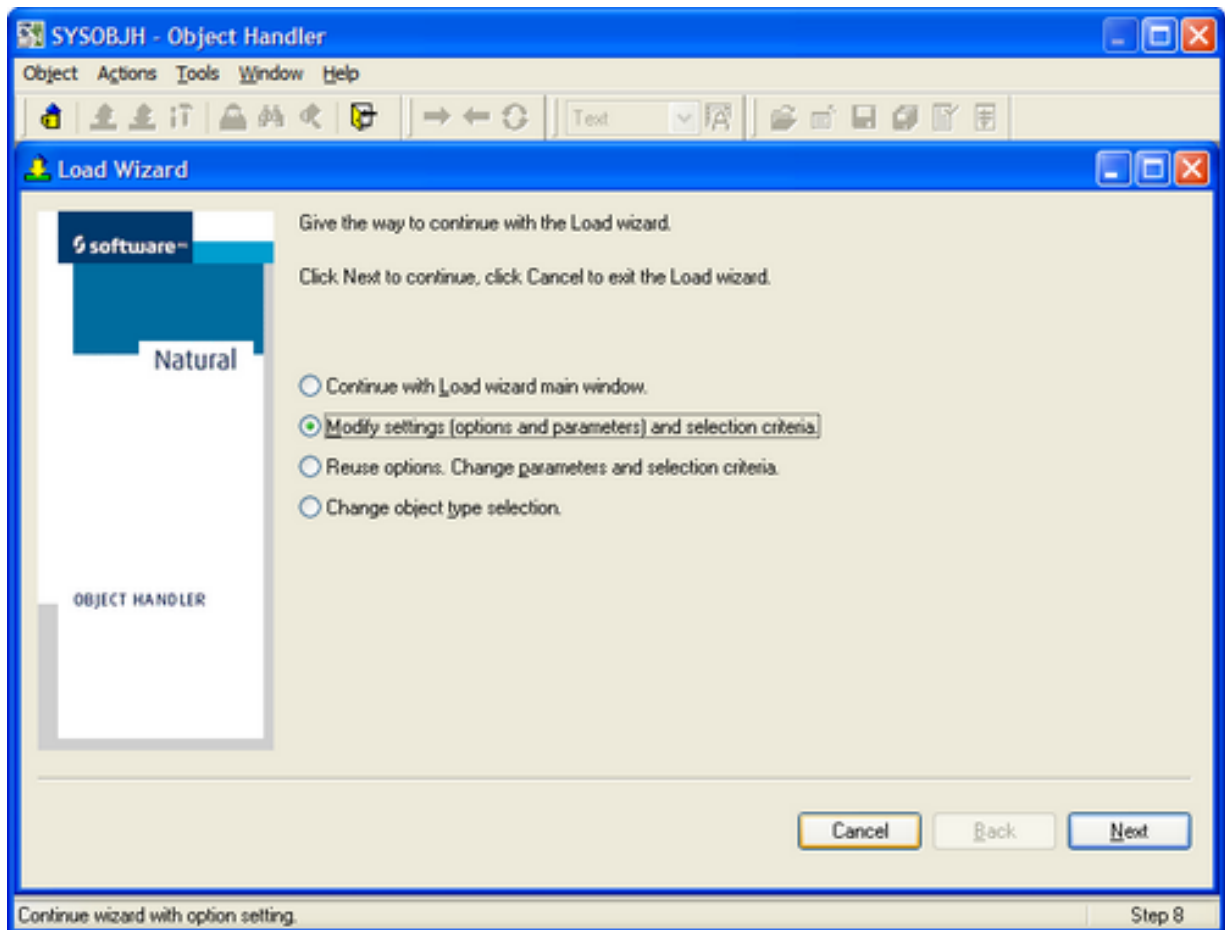
- Click “Next” again to start the actual load process.



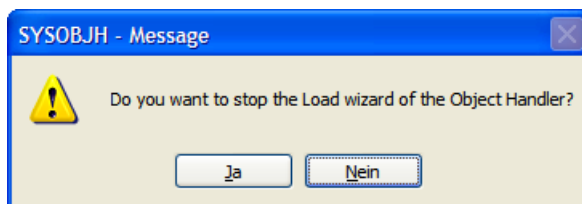
- The Portus System- and Demo-programs have been loaded. Click “Next”.



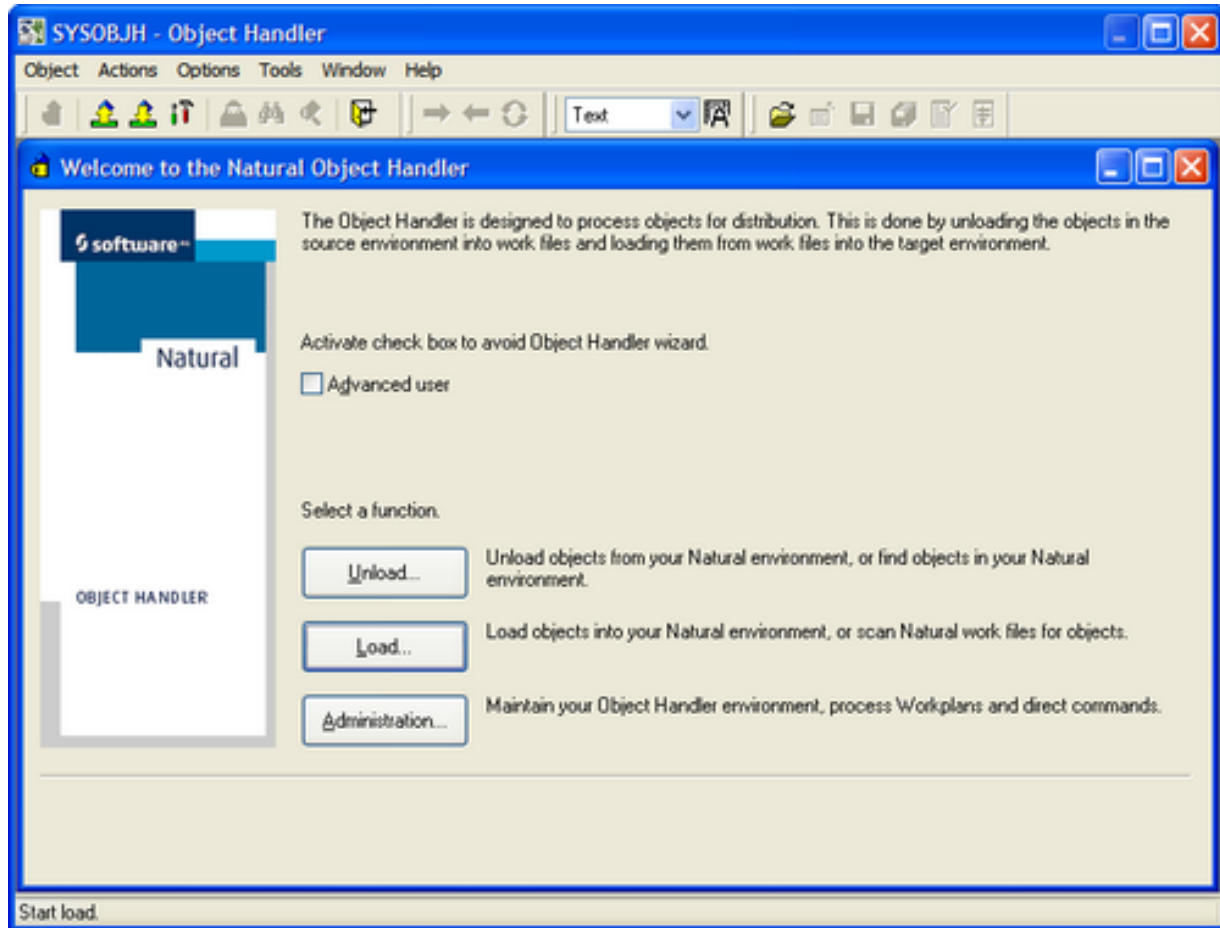
- Click “Cancel” to terminate the Object Handler load function



- Confirm termination of the Object Handler load function by clicking “Yes”



- Dismiss the Object Handler “the Windows way” by clicking the “close” icon in the upper right hand corner.



Prepare the Natural environment on the Mainframe

Defining the driver

In the Portus Servers View, select the server you want to define a Natural driver for, in the Properties View select the *Drivers* tab, click the *Add...* button.

Select the *Natural Driver*, click the **Configure** button

! **Important:** When defining a Natural driver for the first time, make sure to check the "Show (post-setup) Driver Information" box because only this will allow you to extract the INPL file containing the Portus Natural interface- and example programs.

The Driver Definition Dialog will be pre-set with the following values, change according to your needs and preferences, click the **Save** button

Driver Name	<input type="text" value="Natural Driver"/>	<input type="button" value="Save"/>
Driver DLL	<input type="text" value="NATDRVR"/>	
SBCS codepage	<input type="text" value="CP1141"/>	<input type="button" value="Dismiss"/>
MBCS codepage	<input type="text" value="utf16"/>	

As the Natural driver requires a number of additional parameters, the dialog will expand and prompt for more input. Again, fill these parameters as required, you are free to choose any *Driver Name* you wish, set the *SBCS codepage* and *MBCS codepage* parameters according to your local or internationalization requirements. Click **Save** again.

Driver has options (which may be mandatory), please specify and/or click 'Save'

Driver Name	<input type="text" value="Natural"/>	<input type="button" value="Save"/>
Driver DLL	<input type="text" value="NATDRVR"/>	
SBCS codepage	<input type="text" value="CP1141"/>	<input type="button" value="Dismiss"/>
MBCS codepage	<input type="text" value="utf16"/>	
NaturalBatchPgm	<input type="text" value="NATBATCH"/>	
InitParms	<input type="text" value="ETID=OFF"/>	
PreInitSessions	<input type="text" value="2"/>	
MaxSessions	<input type="text" value="8"/>	
NaturalLibrary	<input type="text" value="SYSSOA"/>	
NaturalSteplibs	<input type="text" value="SYSSOAEX,SYSEXT"/>	
NaturalSecurity	<input type="text" value="No"/>	
naturalUserId	<input type="text"/>	
naturalPassword	<input type="text"/>	

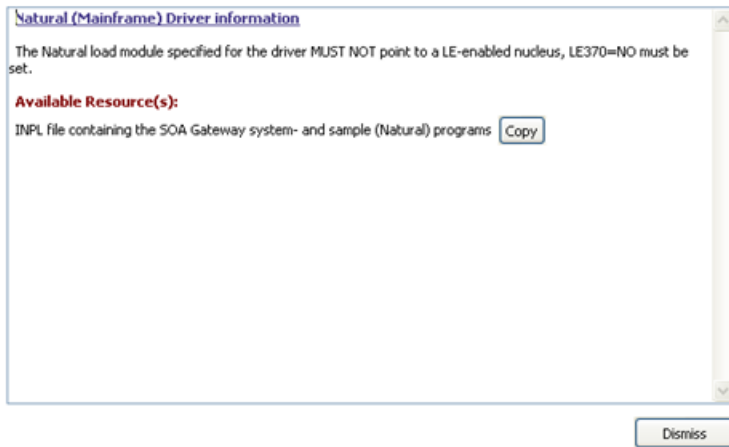
The parameters are as follows

Parameter	Value
NaturalBatchPgm	The name of the Natural Batch nucleus load module. A standard Batch-Natural is used, no modifications or additions to the Natural nucleus are required. Important: This MUST refer to a Natural batch nucleus with NATOS compiled with LE370=NO.
InitParms	Natural initialization parameters. Specify any required parameters overriding the Natural parameter module settings, all Natural profile parameters are allowed here.
PreInitSessions	The number of sessions in the session pool to be pre-initialized.
MaxSessions	The overall size of the session pool, this parameter should be set to a value high enough to accommodate the peak number of parallel sessions expected.
NaturalLibrary	The Natural Library to log on.
NaturalSteplibs	Specify up to 7 Natural library names, in a comma-separated list, to be set as STEPLIB(s). When the NaturalLibrary parameter is not specified as "SYSSOA", include it in the list

Parameter	Value
	<p>of Steplibs if you intend to use the "Service creation" feature to import PDA structures and expose them as Portus Service(s).</p> <p>Note: When running with Natural Security, the Steplib(s) specified here are ignored, they are set by Natural Security based on the logon library and logon user id.</p>
NaturalSecurity	<p>Specify <i>Yes</i> or <i>No</i>, depending on if your environment is protected by Natural Security or not.</p> <p>Important: A logon will be made to the specified NaturalLibrary, with the specified naturalUserId and naturalPassword. It is essential that library SYSEXT as well as any required application libraries are defined as STEPLIBS to that library in Natural Security</p>
naturalUserId	Specify a Natural userid defined to Natural Security
naturalPassword	Specify the password valid for the user defined on the naturalUserId parameter

Once saved, the newly defined driver will appear in the Driver Properties section:

When "Show (post-setup) Driver Information" has been selected, the following information box will be shown, and allow the extraction of the INPL file containing the Portus Natural interface- and example programs.



The Portus server Job / STC

1. Add the Natural load library, or more specifically the load library containing the Natural batch nucleus specified in the driver's *NaturalBatchPgm* parameter and the modules *NATXCAL* and *NATXCAL4* (contained in the Natural distribution load library), to the Portus server Job's / STC's STEPLIB.
2. You may want to add a CMPRINT DD statement to your server JCL, in case of an error during Natural session initialization or execution Natural will write its error messages to this dataset.
3. Ensure the Portus server region is sufficiently sized to cope for Natural's storage requirements, including the local bufferpool(s) if it is used.

Loading the Portus Natural system- and demo programs on the Mainframe

Any standard Natural INPL job can be used to load the Portus system- and demo-programs from the supplied dataset *ASGvrm.INPL*, for example (as usual, replace the library names, database parameters etc. to match your environment):

```
//SMAI061 JOB SAG,MSGLEVEL=1,
//          CLASS=0,MSGCLASS=X,REGION=8M
//*JOBPARM LINES=9999
//*
//* LOAD NATURAL PROGRAMS INTO ADABAS SYSTEMFILE
//* INPL Portus 411 SYSTEM FILE
//*      FROM FILE ASGvrm.INPL
//*
//NATB0100 EXEC PGM=NAT423BA,
//          TIME=1400,COND=(0,LT)
//STEPLIB  DD DSN=SAGLIB.SMALOAD,DISP=SHR
//          DD DSN=SAGLIB.ADA813.LOAD,DISP=SHR
//*
//CMPRMIN  DD *
STACK=INPL
IM=D,INTENS=1,XML=OFF,CFICU=OFF
NAFSIZE=0,DLISIZE=0,DB2SIZE=0,MADIO=0,MAXCL=0,MT=0
AUTO=ON
FUSER=(,8)
//*
//DDPRINT  DD SYSOUT=*
//DDDRUCK  DD SYSOUT=*
//MPMDUMP  DD DUMMY
//SYSUDUMP DD DUMMY
//DDKARTE  DD DUMMY
//DDCARD   DD *
ADARUN DB=001,DE=3390,SVC=249,MODE=MULTI
//CMPRINT  DD SYSOUT=*
//CMWKFO1  DD DSN=ASGvrm.INPL,DISP=OLD
B
FIN
/*
//
```

Type mapping

The Service Creation ("Discovery") process generates a DataView (XRD) and optionally a XML Schema (XSD) from either a Natural PDA (Parameter Data Area) or a parameter data definition (either an inline DEFINE DATA PARAMETER and/or referenced external PDA(s)) within a Natural subprogram.

These bits of mapping information are assembled into a WSDL *in real time* by the Portus server when a WSDL is requested through the

`http://<server>:<port>/<Service>?WSDL` URI for a specific Service.

When the signature (parameters) of a subprogram change, only the DataView needs to be recreated (via "discovery") and the Service "refreshed" (the DataView cleared out of the cache), with the next access to the WSDL it will automatically reflect the changed parameters.

NATURAL to WSDL XML Schema Data Type mapping

Natural Type	DataView	WSDL / XML Schema	Notes
A	sbcS (space padded)	xs:string	Dynamic variables allowed
W	mbcs (space padded)	xs:string	
B	hex.Binary	xs:hexBinary	Dynamic variables allowed
F4	float	xs:float	
F8	double	xs:double	
I1	byte	xs:byte	
I2	short	xs:short	
I4	int	xs:int	
L	Natural logical	xs:boolean	0, false / 1, true
N	zoned decimal	xs:decimal	
P	packed decimal	xs:decimal	
D	date	xs:date	YYYY-MM-DD
T	date + time	xs:dateTime	YYYY-MM-DDTHH:MM:SS(.s)

24

Using Portus to access MySQL database

- Introduction 248
- MySQL Connector 248
- Portus Configuration 250
- Accessing MySQL 252

The widely used MySQL database is used here to outline the steps required to expose a SQL table as a web service.

Introduction

MySQL is a multithreaded, multi-user SQL database management system (DBMS) which has, according to MySQL AB, more than 10 million installations. The ODBC interface is called "MySQL Connector/ODBC" (or also "MyODBC").

This HOWTO assumes that the Portus and MySQL database are both running on Windows.

It is assumed that MySQL is up and running on the local machine and is accessible.

There are many freely available demo databases available online. One of the best known ones is the the MySQL World database. It contains statistics about countries around the world.

See [here](#) for more information about the world database and more.

For the purpose of this HOWTO, the world database has been used.

MySQL Connector

If you already have the MySQL Connector driver installed, then skip this step.

You can check is it installed by selecting **Start Control Panel Add/Remove programs** and look for **MySQL Connector ODBC 3.xx / 5.xx** in the list of installed programs.

- To install MySQL Connector, go to <http://www.mysql.com/products/connector/odbc/> and download the MySQL ODBC Connector



Important: Both 64 and 32-bit versions of this software exist for most platforms. Ensure you download the architecture to match the installed Portus.

- Download the MSI installer
- Open the file you downloaded.
- Click **Next**
- Ensure the **Typical** radio button is selected and click **Next**
- Click **Install**
- Click **Finish**

The MySQL Connector is now installed. To configure it, follow these steps

- Click **Start, Control Panel, Administrative tools, DataSources(ODBC), System DSN**



Important: On 64-bit Windows 7, you need to start the 32-bit version of this program to add new DSNs. Run the following command directly : %WINDIR%\SysWOW64\odbcad32.exe

- Click **Add**
- From the list of drivers, select **MySQL ODBC 3.xx / 5.xx Driver**
- Click **Finish**
- Enter **world** as the **Data Source Name**
- Enter **localhost** in **Server**
- Entering a value in the Port box is optional. MySQL defaults to port 3306.
- Enter the username required to access the world database in **User**
- Enter the password required to access the world database in **Password**
- Select the world **Database** from the dropdown list.
- Click **Test** and ensure the server returns success

MySQL Connector/ODBC Data Source Configuration

MySQL Connector/ODBC

Connection Parameters

Data Source Name: world

Description:

Server: localhost Port: 3306

User: root

Password:

Database: world

Test

Details >> OK Cancel Help

- Click **OK**
- Click **OK**

Portus Configuration

Portus must now be configured to access and use this new MySQL DSN

- Start the Portus Control Center and add a Portus Server. See here for more information.
- If you do not have an MySQL driver, add one now. See here or here (legacy) for more info.
- See here for how to add/discover a Service.
- From the next dialog choose **MySQL Driver** and click **Next**
- The next dialog prompts you for the the
 1. **ODBC Dsn**: e.g. world
 2. **UserId / Password**. If in doubt, use root as the User Id, and the password will be the one you set up during MySQL installation.
 3. **Pattern**: This will be a pattern match value passed to the SQL Command. The search pattern characters are: an underscore (`_`), which represents any single character and a percent sign (`%`), which represents any sequence of zero or more characters.
 4. **Max. entries to list**: This is a hard-limit on the amount of tables Portus will attempt to discover.
 5. **Options for generated Service name**. By default the Service has the `odbc dsn` name prepended to the table name. By selecting this option you can change the name to suit you requirements.
 6. **Show all tables types**. If this is selected then all tables types are discovered. Usually this is best left unselected.

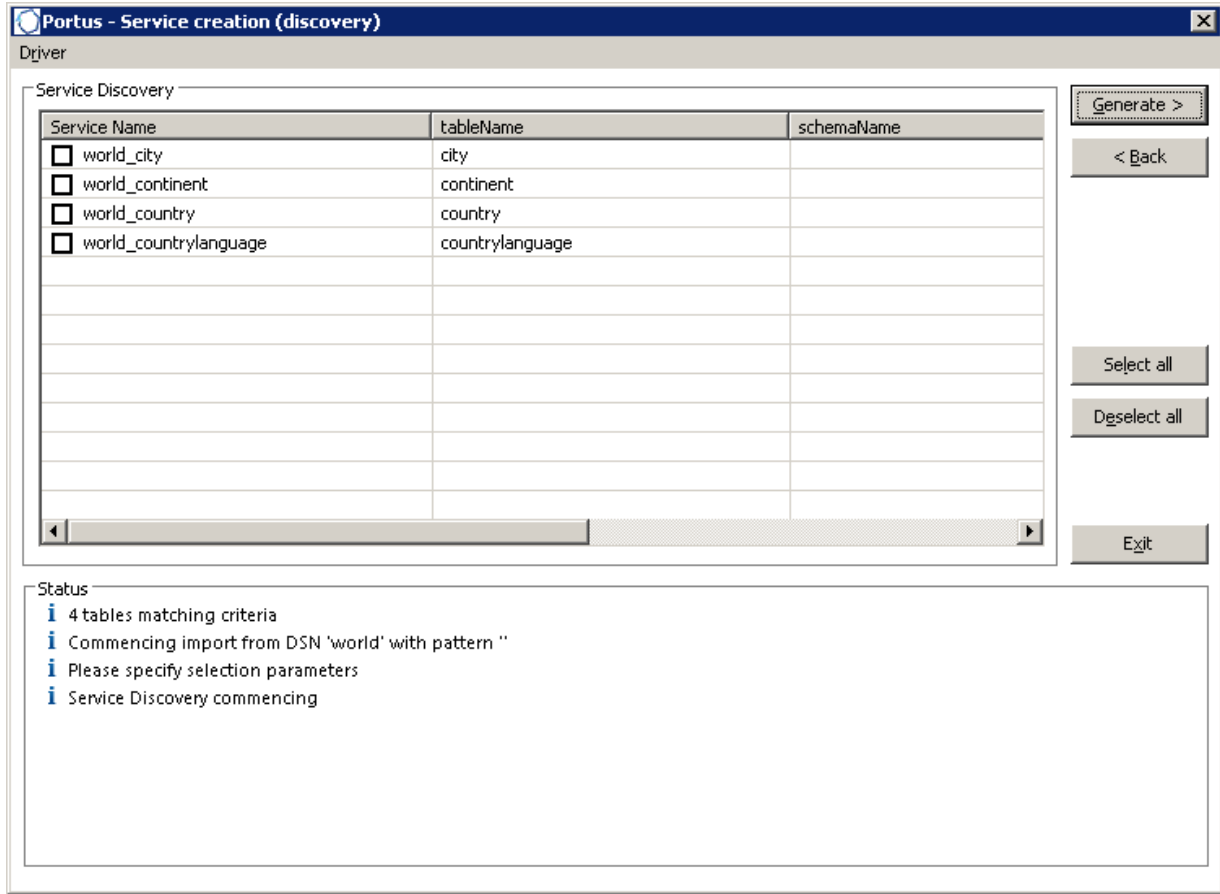
The screenshot shows a dialog box titled "Driver Options" with a "Service discovery" tab. The dialog is divided into three main sections:

- Specify ODBC descriptors & option(s):** This section contains four input fields: "ODBC Dsn", "UserId / Password" (split into two boxes), "Pattern", and "Max.entries to list".
- Advanced option(s):** This section contains two checkboxes: "Options for generated Service name" and "Show all table types", both of which are currently unchecked.
- Status:** This section displays two informational messages: "Please specify selection parameters" and "Service Discovery commencing".

Buttons for "Discover" and "Exit" are located on the right side of the dialog.

Now click **Discover**

- Portus will ask the MySQL database identified by the DSN to display all the tables which match the request. Select the required tables, and click **Generate**.



- The results of the import will be displayed in the Status section.
- Click "Exit" to finish the Service creation
- You have now created Services based on the "world" DSN!

Accessing MySQL

Now that the resources have been set up, you can access the Web Service Description Language (WSDL) by selecting the 'WSDL URL is ..' in the Properties of the Service:

Service Definition

Name: world_city

Driver: MySQL_Driver (dropdown) Read-only:

SBCS-Codepage: [empty text box]

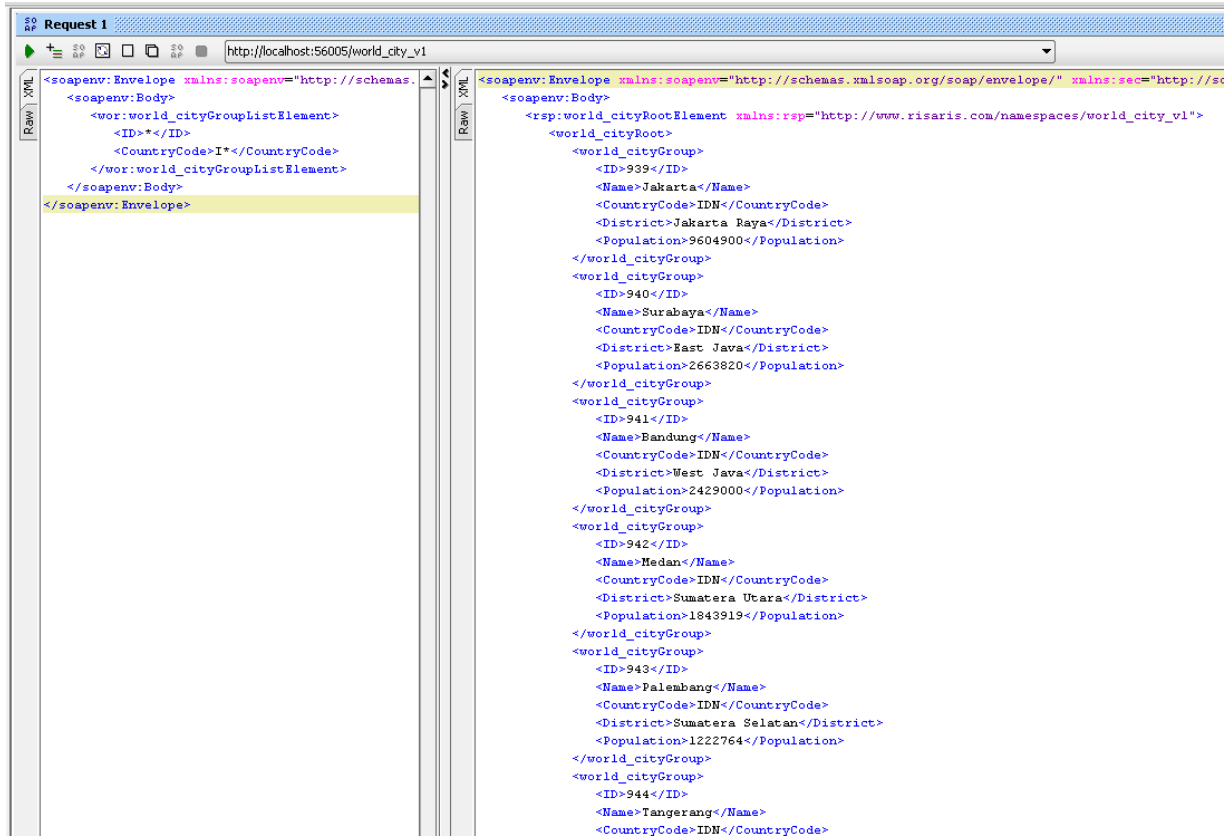
MBCS-Codepage: [empty text box]

Status: Test (dropdown) change to: [empty dropdown]

WSDL URL is .. http://localhost:56005/world_city?WSDL

There are many clients available to consume and use web services, for example soapUI, XMLSpy, and Infopath. Here we will use soapUI.

- Start soapUI and create a new WSDL project.
- Import using the WSDL
- Edit the list request
- Completely remove the `...world_cityGroupHeader>` element from the `<soap:Header>` element
- In the `<Security>` element, add the required Username and Password for accessing the world table or remove the `<Header>` element completely if not required (as shown below).
- Add `*` as the content of the `<ID>` element and `I*` as the content of `CountryCode`. This is equivalent to *select * from world where CountryCode like 'I%'*
- Hit the green arrow, and the results of the request should be displayed



Congratulations! You have now accessed MySQL using Portus!

25 Portus Service Creation

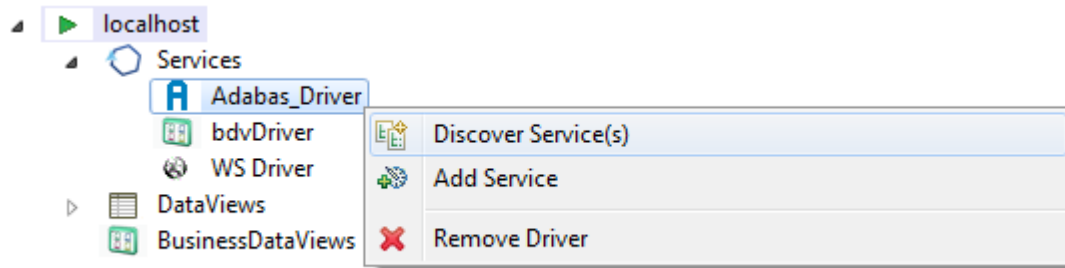
- Initiating Service creation 256
- Creating Service(s) from Adabas 258
- Assigning a different DataView to existing service 259
- Creating Services from SQL databases 261
- Creating Services from Stored Procedures 262
- Creating Services from a SYSOBJH extract file 263
- Creating Service(s) from Natural 266
- Creating Service(s) from 3rd party web services 270

Service definitions, and related mapping files (DataViews, XSDs) are created, in a semi-automatic to automatic process (also known as Service Discovery), from meta-data describing the characteristics of resources to be exposed as a Service through Portus, for example from an Adabas FDT, SQL Table description, Natural sources or even other Web Services.

Initiating Service creation

1. Administration Perspective

The Service creation process can be started by selecting the appropriate driver, right-clicking and selecting either Discover Service(s) or Add Service:

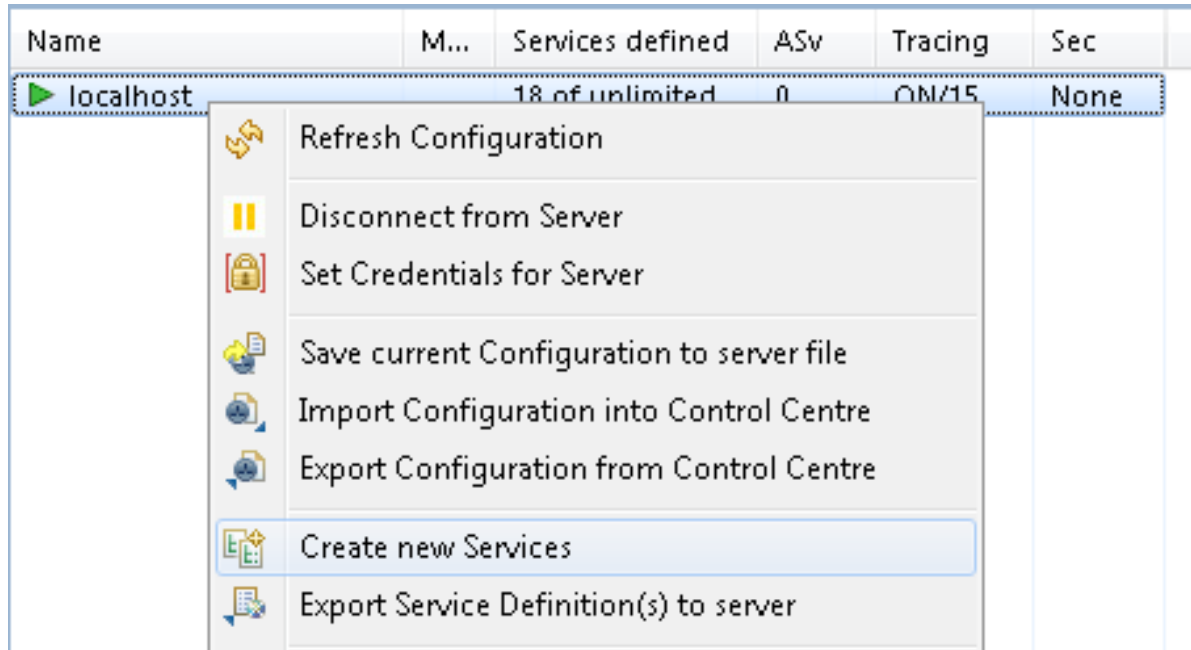


Discover Service(s) will start the discovery wizard and request the specific parameters required for the selected driver type.

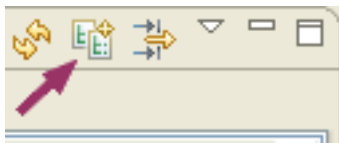
Add Service will ask you to name the service and you can then manually set the various parameters as per your particular system.

2. Legacy Perspective

- The Service creation process can either be started from the server's context menu by selecting 'Create new Services':



- or by clicking the 'Create new Services' action button in the title area of the Configuration View:

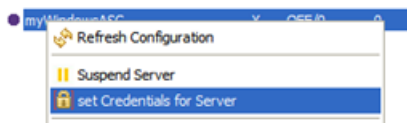


- The next step is to select the driver to be used as the source.

There will be a list of defined drivers available, plus a number of special entries, for example <SYSOBJH>, which can be used to create Service(s) from a SYSOBJH unload file containing DDMs, or <3GL> to create a DataView and XSD from a C- or COBOL program or copybook.

When a driver is chosen select Use Driver to display the Service creation (discovery) dialog.

It may be required to provide login credentials, for example when accessing a SQL database system, this can be done from the server's context menu as well:



Specify the credentials - User ID and Password - in the *SOAP Security* section, click *Set*

Status and error information can be retrieved later on from the Portus Action Log


Creating Service(s) from Adabas

- Specify the Database Id (range 1 - 255 on OpenSystems, 1 - 65535 on Mainframe platforms), plus 'File range from' and 'to' file numbers (1 - 5000) to limit the discovery to the selected range. Click the 'Discover' button:

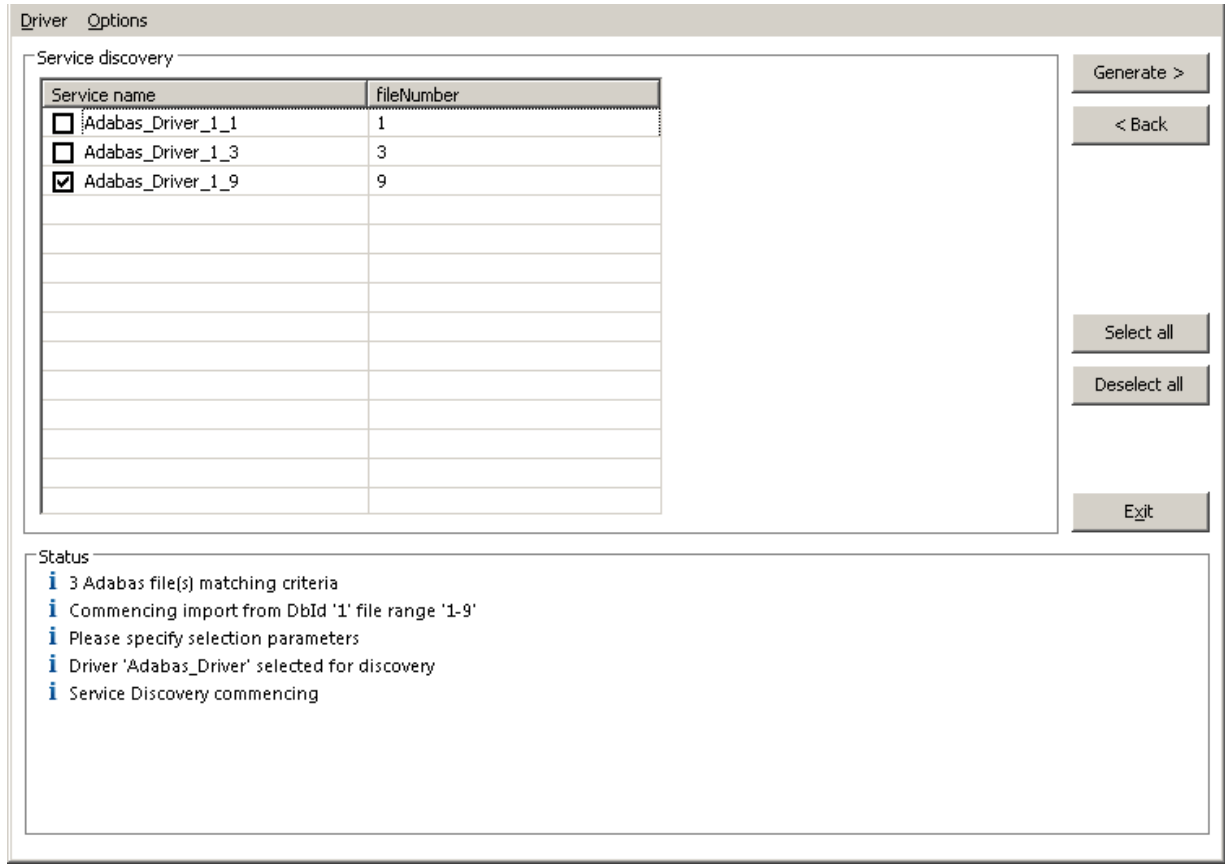
The screenshot shows a dialog box titled "Driver Options" with a "Q" icon. It is divided into three main sections: "Service discovery", "Status", and "Input & option(s)".

- Service discovery:** Contains a "Discover" button in the top right corner.
- Input & option(s):** A section with a light blue header. It includes:
 - "Database Id": A text input field.
 - "File range from": A text input field followed by "to" and another text input field.
 - "Password": A text input field.
 - "Generate ISN as DataView field": A checkbox.
- Advanced option(s):** A section with a light blue header. It includes:
 - "Influence generated Service name(s)": A checkbox.
- Status:** A section with a light blue header. It contains two informational messages:
 - "Please specify selection parameters"
 - "Service Discovery commencing"

At the bottom right of the dialog box, there is an "Exit" button.

 **Note:** Discovering the full range of possible file numbers (1 - 5000) may take substantial time and adversely affect performance of both the Portus server as well as the Adabas target database. Break the discovery process down to smaller range(s) of files.

- Select the file(s) to be turned into Services, click the 'Generate >' button:



Optionally, the Service name can be modified.

- The Status pane reports on the discovery process:
- The requested Services and Service definition files (DataView(s) and XSD(s)) have been added to your server configuration or, if not, a suitable error message will indicate the cause of failure.

Choose Exit to finish or Discover to generate more web services

Assigning a different DataView to existing service

Sometimes it is required to create a web service using the Discovery Wizard, and then assign a different DataView to this service. For example, an Adabas web service discovered with "short names" in the DataView, and a pre-created DataView with "long names" available.

Assumptions:

- The pre-created (Business)DataView has been imported into the local workspace, and is available in the Navigator View.

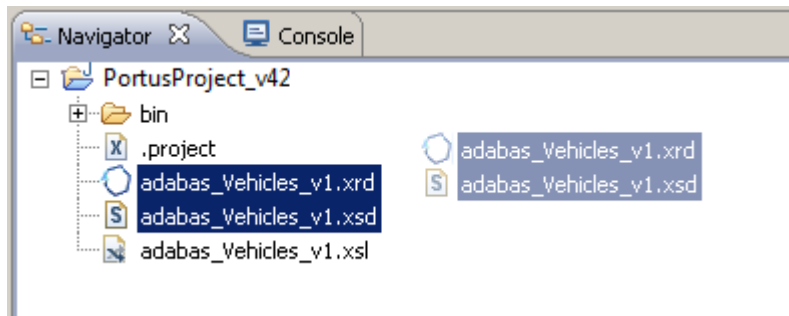


Notes:

1. The Adabas "long names" DataViews are available *on the Portus Server* where Adabas is licensed.
 2. These can be imported using the Legacy Perspective here
 3. These can be imported using the Portus Administration Perspective here.
- OR the pre-created (Business)DataView is available on the server already
 - The web service you wish to modify has already been created.

If the above assumptions are OK, follow these steps:

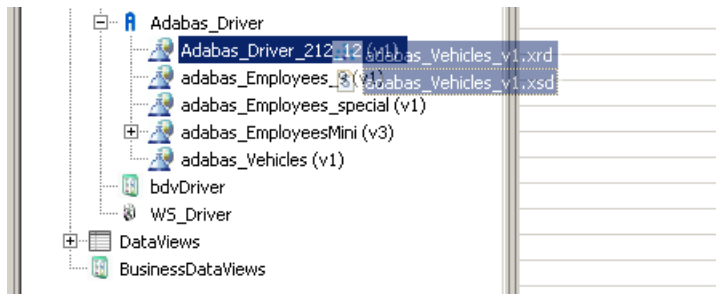
1. Drag the XRD and XSD (if available) files from the Navigator View.



2. In either perspective, drop the files onto the service you wish to modify

Services		DataViews / XSDs / XSLs	BusinessDataViews			
Mod	Driver	Service	Vrs	DataSource Id	DataView	
	Adabas_Driver	Adabas_Driver_212_12	1	DatabaseId=212, FileNumber=12	Adabas_Driver_212_12_v1	adabas_Vehicles_v1.xrd adabas_Vehicles_v1.xsd
	Adabas_Driver	adabas_Employees_9	1	DatabaseId=444, FileNumber=9	adabas_Employees_9_v1	
	Adabas_Driver	adabas_Employees_special	1	DatabaseId=212, FileNumber=11	adabas_Employees_special_v1	
	Adabas_Driver	adabas_EmployeesMini	4	DatabaseId=5, FileNumber=11	adabas_EmployeesMini_v4	

OR



In the Portus Perspective a new (Business)DataView can be assigned by dragging it from the respective Views section and drop it onto the service.

3. Right-click the service and select the *Refresh* function

4. The WSDL should now reflect the DataView.

Creating Services from SQL databases

- Select a defined driver pointing to the SQL database system of your choice from the list. This step-by-step guide will use MySQL to demonstrate the process.
- For this example the 'world database' demo has been loaded as per the guide published at the [MySQL Community website](#), and a ODBC Data Source been set up as 'WorldDS', pointing to the 'world' database.
- Click the 'Use Driver' button to start the process.
- Specify the ODBC Dsn to be browsed. Optionally enter a 'Pattern' limiting the operation, for example 'c%' will return a list of tables starting with an lowercase 'c'. Another option is to specify 'Max.entries to list' to further limit the number of tables to be listed. If appropriate for your database configuration enter a UserId and Password. Click the 'Discover' button:

The screenshot shows a dialog box titled "Driver Options" with two tabs: "Driver" and "Options". The "Options" tab is selected. The dialog is divided into several sections:

- Service discovery:** Contains a "Discover" button.
- Specify ODBC descriptors & option(s):** A highlighted section containing input fields for "ODBC Dsn", "UserId / Password", "Pattern", and "Max.entries to list".
- Advanced option(s):** A highlighted section containing two checkboxes: "Options for generated Service name" and "Show all table types".
- Status:** A section at the bottom containing two informational messages: "Please specify selection parameters" and "Service Discovery commencing".

An "Exit" button is located at the bottom right of the dialog.

- zOS and DB2

In the ODBC Dsn field above - enter the Location name returned from the DB2 command Display DDF

- in the example below the location name is S1D831

```
COMMAND INPUT ==> /-D831 DIS DDF
RESPONSE=POC1 DSNL080I -D831 DSNLTDDF DISPLAY DDF REPORT FOLLOWS: DSNL081I
STATUS=STARTD DSNL082I LOCATION LUNAME GENERICLU DSNL083I S1D831
DEIBMIPA.IPAA77D8 -NONE .....
```

Also ensure that the member CLINI in the JCL dataset has been modified as necessary. Refer to your DBA on this.

- Select the table(s) to be turned into Services. If appropriate the Service name(s) can be modified by selecting the 'Options for generated Service name' tick box and modifying the entries. Finally click the 'Generate' button:
- The Status pane indicates what has been generated.
- The requested Services and Service definition files (DataView(s) and XSD(s)) have been added to your server configuration
- The Wizard is still active and ready for further discovery, click 'Discover' to start the process again, or 'Exit' to dismiss the wizard.

Creating Services from Stored Procedures

- Details on how to create/modify/delete a stored procedure driver can be found [here](#).
- Select the stored procedure driver from the list This step-by-step guide will use MS SQL Server database AdventureWorks to demonstrate the discovery process.
- Click the 'Use Driver'.
- Specify the ODBC Dsn to be browsed. Optionally enter a 'Pattern' limiting the operation, for example 'uspGet%' will return a list of stored procedures starting with 'uspGet'. Optionally specify 'Max. entries to list' to further limit the number of stored procedures to be listed. Click the 'Discover' button.

Note that the generated web service name can also be changed from the default of prefixing with 'storedProcDriver_'

- Select the stored procedure(s) to be turned into Services, click the 'Generate' button:
- The Status pane will indicate what has been generated.
- The requested Services and Service definition files (DataView(s) and XSD(s)) have been added to your server configuration

- The Wizard is still active and ready for further discovery, click 'Discover' to start the process again, or 'Exit' to dismiss the wizard.

Creating Services from a SYSOBJH extract file

- Start Discovery by right-clicking the *Services* element on the server you want the DataView(s) to be created

Select <SYSOBJH> from the 'Select the Source' list and then select the ' Use Driver' button.

- A step-by-step guide on how to create a SYSOBJH extract file can be found here.
- Click the 'Generate' button to start the process
- Select the SYSOBJH file to be used as the input to the creation process, specify if XSD(s) are to be created for the generated Service(s) as well, in addition to the DataView(s).

The screenshot shows the 'Driver Options' dialog box. The 'Service discovery' section contains the following fields and options:

- SYSOBJH Input file:** E:\unload_ddms.systrans (with a 'Browse...' button)
- Use Adabas driver:** Adabas (dropdown menu)
- Also Generate and export XSD:**
- Adabas DbId for generated Service(s):** 6
- Default number of occurrences:** for MU fields: 8; for PE fields: 4
- Default size of Large Object (LOB) fields:** 2768
- Translate dashes in field names to underscores
- Flatten structures (create top-level elements only)
- Remove redefinitions when importing from PDAs
- Translate special characters in field names:** (Three pairs of 'to' fields with checkboxes)

The **Status** section at the bottom displays the following messages:

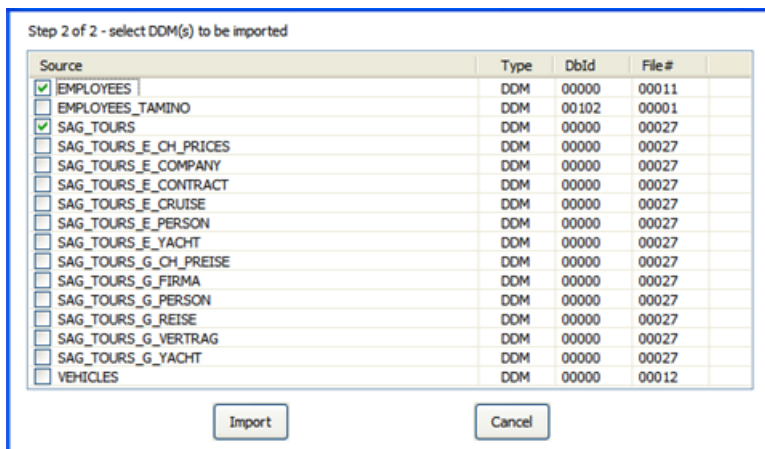
- i** Please specify selection parameters
- i** '<SYSOBJH>' selected for discovery
- i** Service Discovery commencing

Buttons for 'Discover' and 'Exit' are located on the right side of the dialog.

A number of options can be specified here

- defaults to be used when generating Service(s) and DataView(s) based on Adabas files
 - Database Id used when the DDM is defined with a DbId of 0 (zero)

- Default number of occurrences for MU(ltiple value) fields and PE(riodic) groups
- Default size of large object (LOB) fields
- if you wish dashes ('-') in file and element names to be translated to underscores ('_'). This may be desirable where the target programming language used to access Portus Resources do not accept dashes in field names, an example being PHP.
- When PDAs include redefinitions you will need to choose the layout that suits you for the DataView to be generated. If you are unsure at the moment, leave the "remove redefinitions" box unchecked, in which case ALL redefinitions will be included in the generated DataView, with a generated fgroup name of "*redefinition_of_<PDA_group_name>". Before the DataView is uploaded to the server, remove all definitions/redefinitions not required. Multiple DataViews (and thus resources) are required when requiring different layouts/set of parameters for the same subprogram.
- "Flatten structures" when the PDA includes nested groups but you want all fields to be generated at the "top level" without any group/field structure.
- You may need to translate special characters, like the # (hash) sign if they are used as part of field name(s), because they are not valid in a SOAP context.
- Click the *Discover* button
- Select the DDM(s) to be turned into Services, click the '*Generate >*' button:



- For each DDM selected, which contains MU and or PE fields, a dialog will request you to specify the maximum number of occurrences expected.

⚠ Important: The "MaxOcc" accepted by this dialog is 32767, depending on the ADABAS version the actual limit accepted by the database system may be lower (191).

Max...	M..	SN	Field
8	M	AI	ADDRESS_LINE
4	P	AQ	INCOME
8	M	AT	BONUS
4	P	AW	LEAVE_BOOKED
8	M	AZ	LANG
4	P	S3	CURRENCY_SALARY

OK Cancel

- The status pane indicates what has been generated, you can now start the process again or exit the wizard.

```
Importing from 'F:\ASG\Demos\unload_ddas.systrans'
Importing from DDM 'EMPLOYEES'
-----
* no referenced fields for PhonDe 'PHONETIC_NAME' - has been dropped
* no referenced fields for SuperDe 'LEAVE_LEFT' - has been dropped
* no referenced fields for SuperDe 'DEPARTMENT' - has been dropped
* no referenced fields for SuperDe 'DEPT_PERSON' - has been dropped
====>> inport completed WITH ERRORS <<<===
====>> please review the XRD !! <<<===
WebService exported as 'EMPLOYEES', XSD generated

Importing from DDM 'SAG TOURS'
-----

-----> inport completed O.K.
WebService exported as 'SAG TOURS', XSD generated
```

Close

- In case of a Natural Parameter Data Area (PDA), the type will be shown as "PDA"

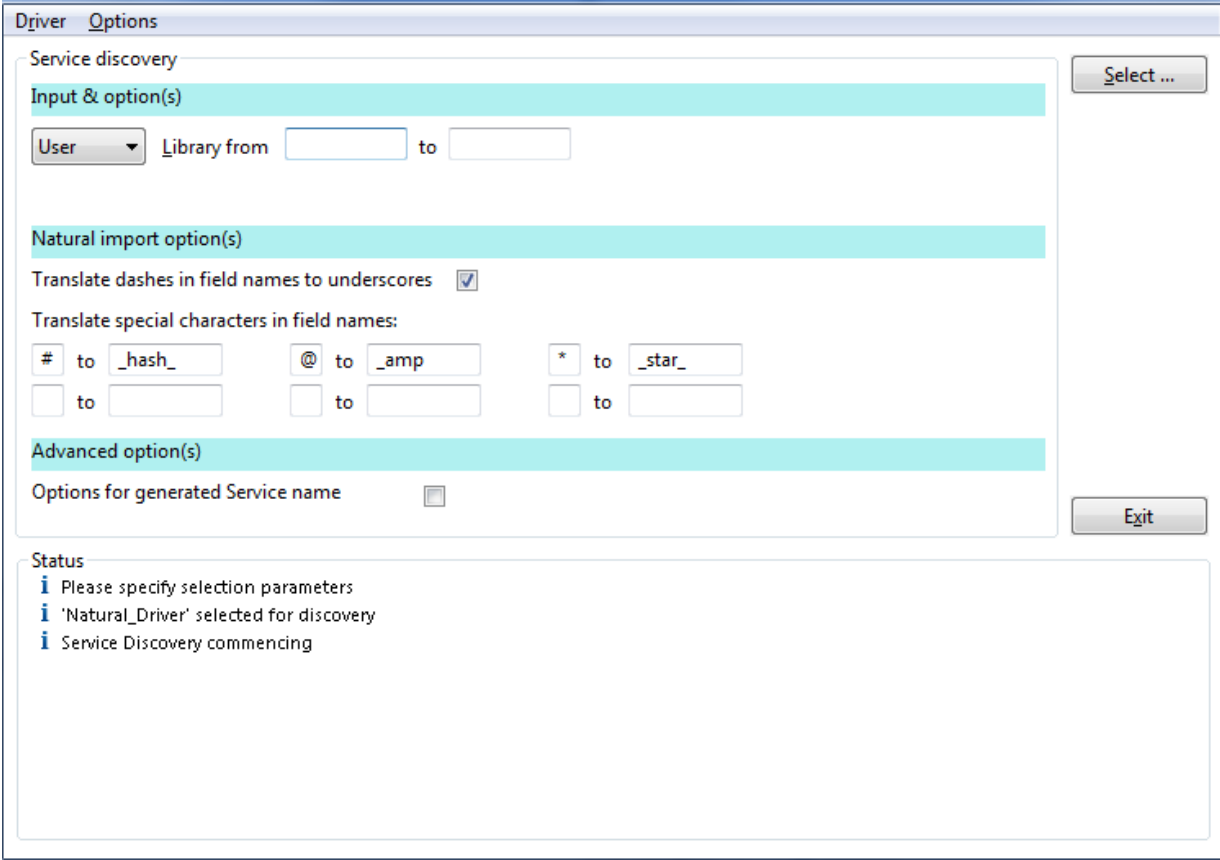
Step 2 of 2 - select Object(s) to be imported

Source	Type	DbId	File#
<input type="checkbox"/> iCALCA	PDA		

- The requested Services and Service definition files (DataView(s) and XSD(s)) have been added to your server configuration
- The Wizard is still active and ready for further discovery, click 'Discover' to start the process again, or 'Exit' to dismiss the wizard.

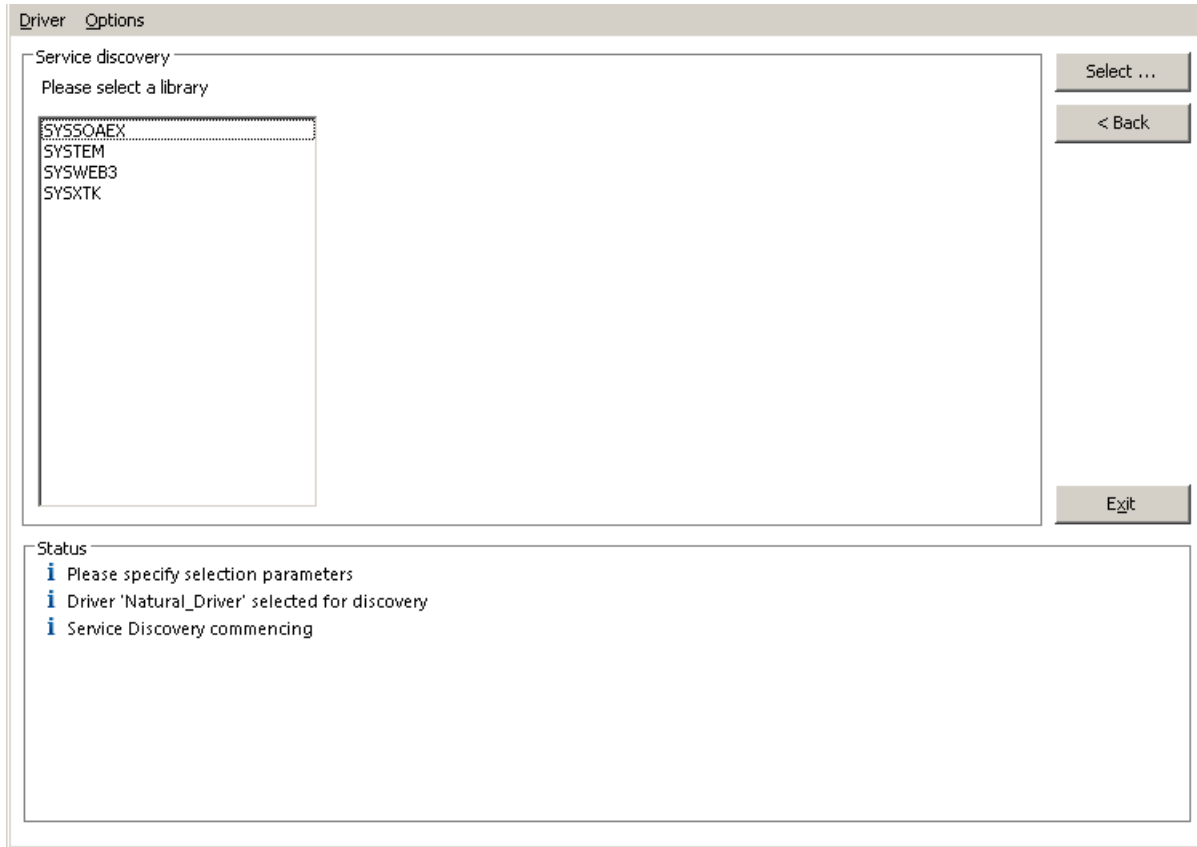
Creating Service(s) from Natural

- Select a defined driver of type 'Natural' from the list
- Click the 'Use Driver' button
- Specify the name of the Library to be scanned, select if it is a Natural "User" or a "System" library.



Alternatively, if the library name is unknown, specify a generic library name, or a from-to range and click the **Select...** button. For example, the above input

will result in this additional selection dialog to pop up



Select a library with a double-click, or select a library name and click **Select**, or stop the wizard by clicking the **Exit** button.

- Select the name(s) of the object(s) to be turned into Services, click the **Select...** button. Again, this can be a specific object name, a generic object name, or a range of objects. When a specific object name is entered in the Object "from" field, but the "to" Object name field is left empty, this will result in all objects *starting from the specified name* being listed

Driver Options

Service discovery

Input & option(s)

System Library from to

Object from to

Natural import option(s)

Translate dashes in field names to underscores

Translate special characters in field names:

to @ to * to

to to to

Advanced option(s)

Options for generated Service name

Status

- i** Please specify a Natural Object or a range and select
- i** Please specify selection parameters
- i** Driver 'Natural_Driver' selected for discovery
- i** Service Discovery commencing

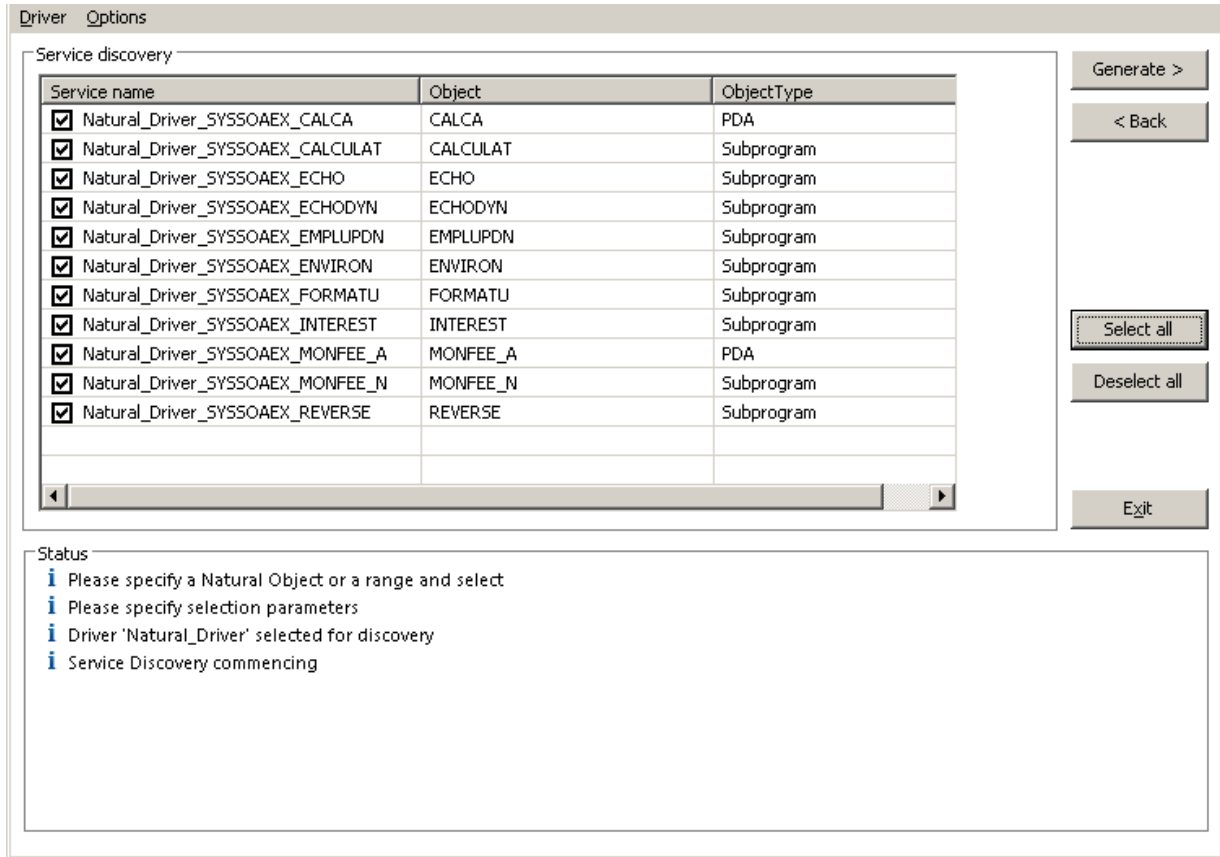
Select ...

< Back

Exit

You may return to Library selection mode by clicking the **Back** button.

- A list of objects matching the selection criteria is displayed



Either select all, or only the one(s) you are specifically interested in, click the **Generate** button.

- The Status pane will report on the progress of each item chosen.

The requested Services and Service definition files (DataView(s) and XSD(s)) have been added to your server configuration

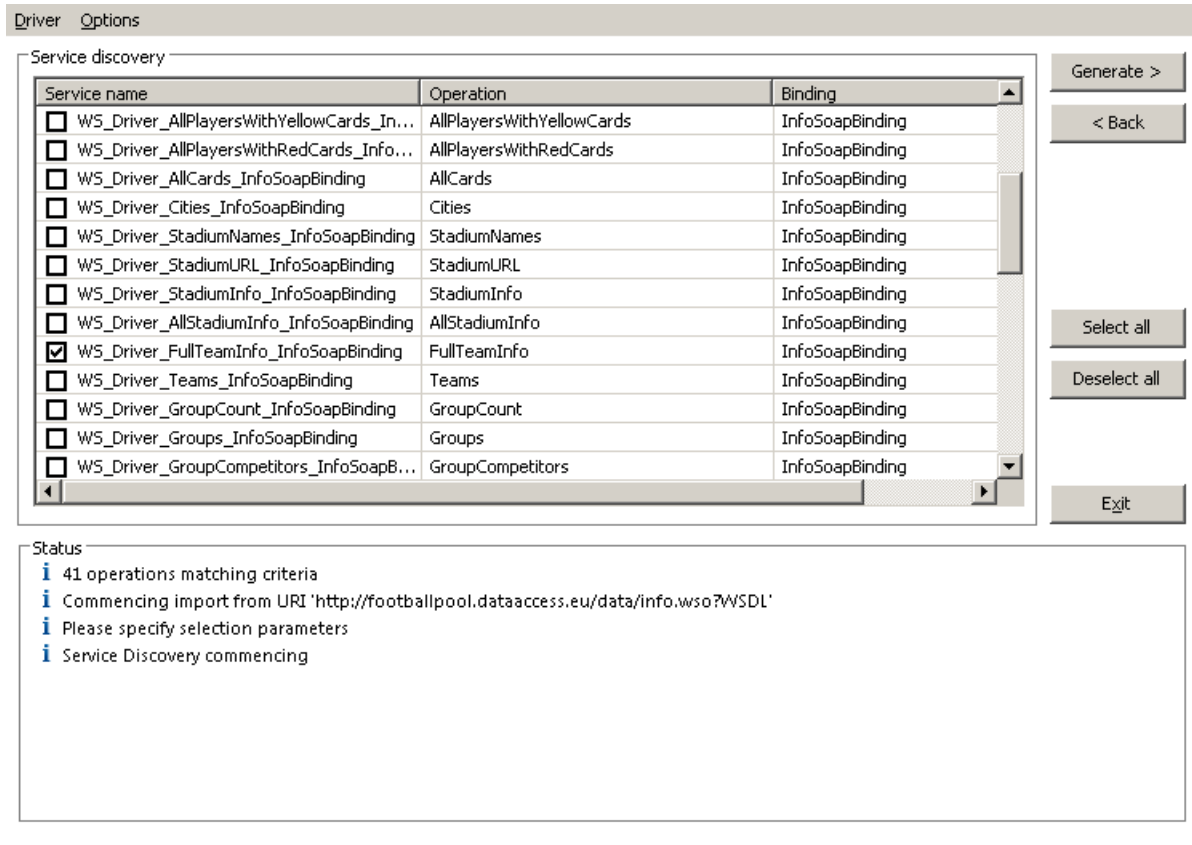
- The Wizard is still active and ready for further discovery, click 'Select...' to start the process again, or 'Exit' to dismiss the wizard.

Creating Service(s) from 3rd party web services

- Under Services select the Web Services Driver previously added. Right-click and select Discover Service(s).
- Specify the Web Service URI and click the 'Discover' button:

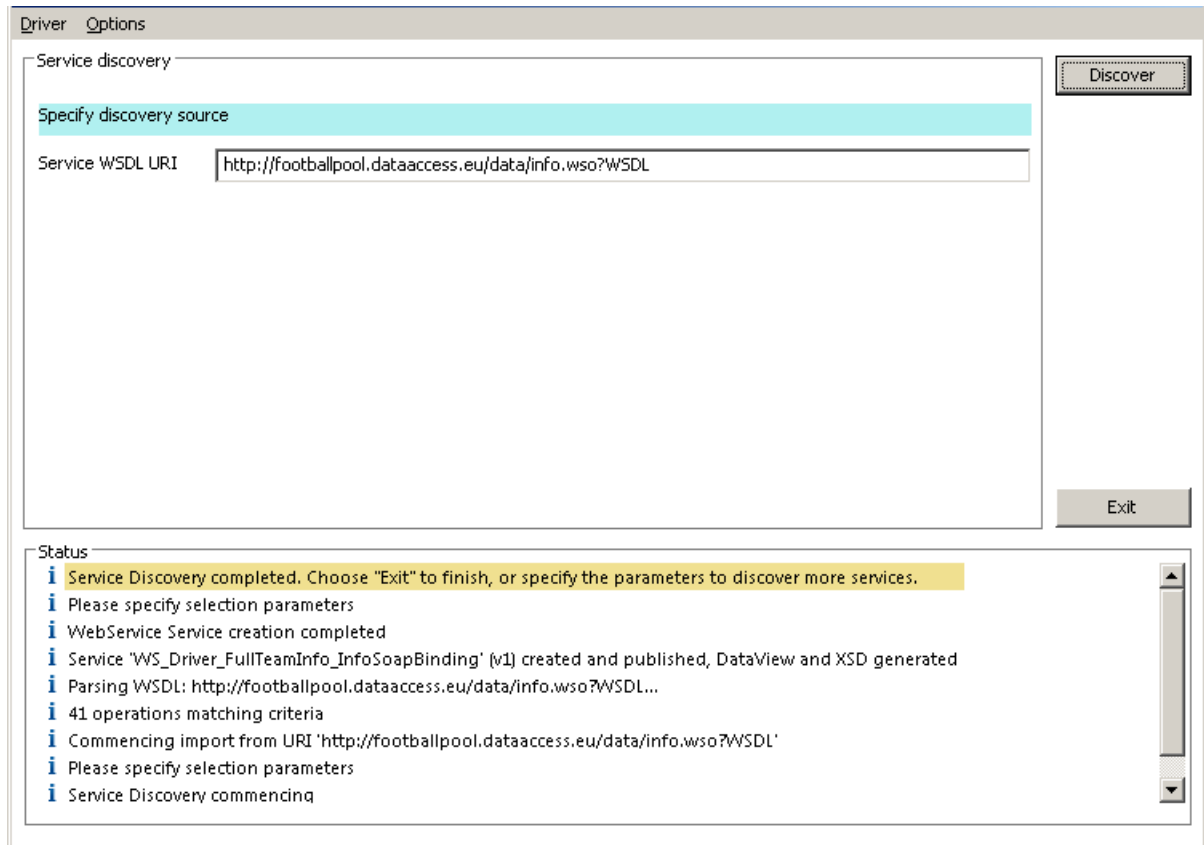
The screenshot shows a dialog box titled "Driver Options" with two tabs: "Driver" and "Options". The "Options" tab is active. It contains a "Service discovery" section with a "Specify discovery source" label and a text input field containing the URL "http://footballpool.dataaccess.eu/data/info.wso?WSDL". To the right of this section are "Discover" and "Exit" buttons. Below the input field is a "Status" section with two informational messages: "Please specify selection parameters" and "Service Discovery commencing".

- The Status section will display the progress of each stage of the web service discovery and the Service discovery section will display the following information retrieved from the service description:



Use Select all to choose all operations or select on an individual basis. When chosen click the **Generate >** button.

- The Status section will be updated with the progress of the generation process, listing the operations selected and the results for each one. For clarity we have chosen the TopGoalScorers operation.



- The Wizard is still active and ready for further discovery, click **Discover** to start the process again, or **Exit** to dismiss the wizard.
- The generated web service(s) will be listed under WS Driver.

26 Using Portus to access Microsoft SQL Server database

- Introduction 276
- SQL Server ODBC Setup 276
- Portus Configuration 279
- Accessing SQL Server using Portus 281

Using Portus to access Microsoft SQL Server database

Introduction

Microsoft SQL Server is a relational database management system (RDBMS) produced by Microsoft. Portus uses the ODBC interface to communicate to it.

This HOWTO assumes that the Portus and SQL Server database are both running on Windows.

It is assumed that the SQL Server has the ODBC Connector installed.

For the purpose of this HOWTO, we've installed the "AdventureWorks" sample that is provided with the SQL Server installation. See [here](#) for more information about AdventureWorks and other SQL Server sample databases.

SQL Server ODBC Setup

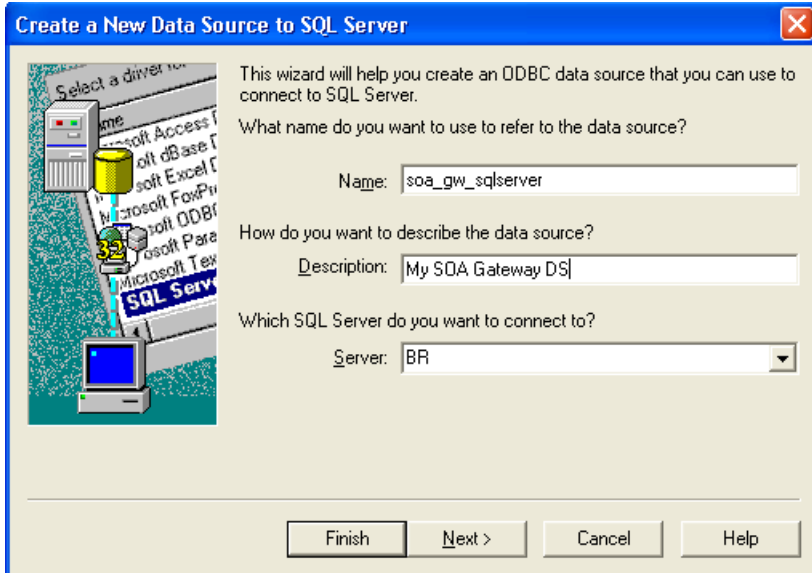
This section describes how to set up a system DSN used by Portus to connect to a SQL Server database.

To configure it, follow these steps

- Click **Start, Control Panel, Administrative tools, DataSources(ODBC), System DSN**
- Click **Add**
- From the list of drivers, select **SQL Server**

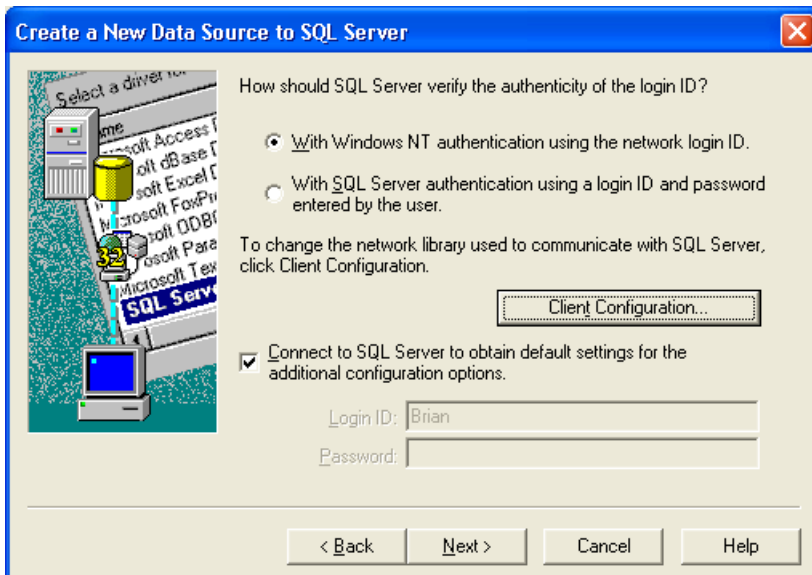
If you do not see SQL Server here, ensure that you have the ODBC components installed.

- Click **Finish**
- Enter `soa_gw_sqlserver` as the Name
- Enter `My Portus DS` as the Description.
- Enter the name of your SQL Server Instance. In this case it is "BR"



Click **Next**

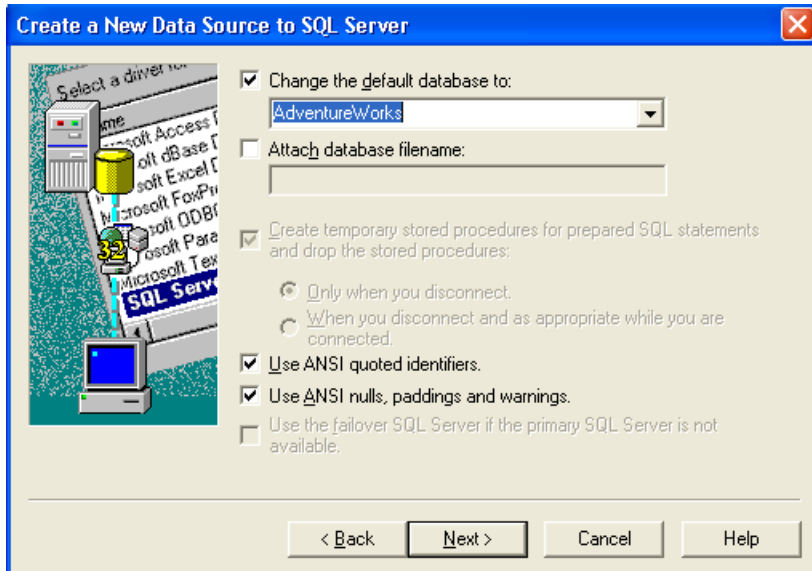
- Select how you wish to connect to the SQL Server Instance. These settings will be specific for the SQL Server installation. In our case we choose the following settings



Click **Next**

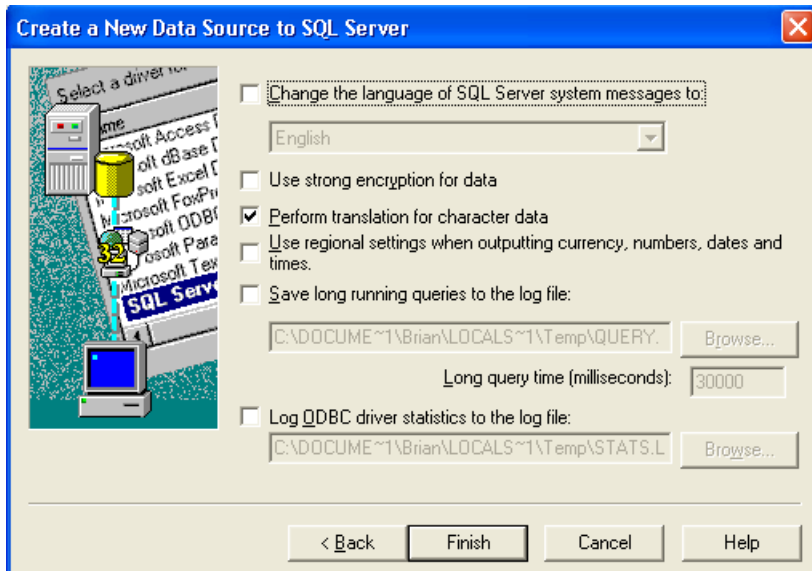
- Choose what the default should be. This will be the database that Portus will use to search for tables on.

If you are using the AdventureWorks sample database, this screen should be as follows



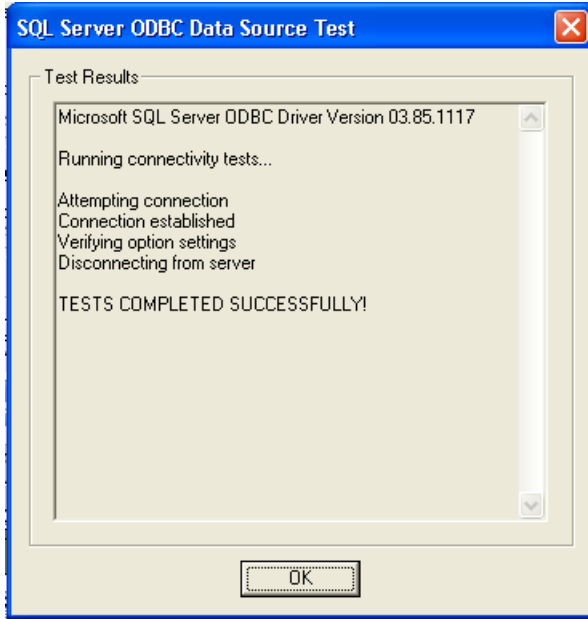
Click Next

- The final screen will again be specific to the SQL Server Installation. Change the necessary settings here, and click **Finish**



- It is recommended that you now test your new DSN using the SQL Server Wizard. Click **Test Data source** to begin

If your settings are correct, you should see the following report



Otherwise, re-check your settings and try again.

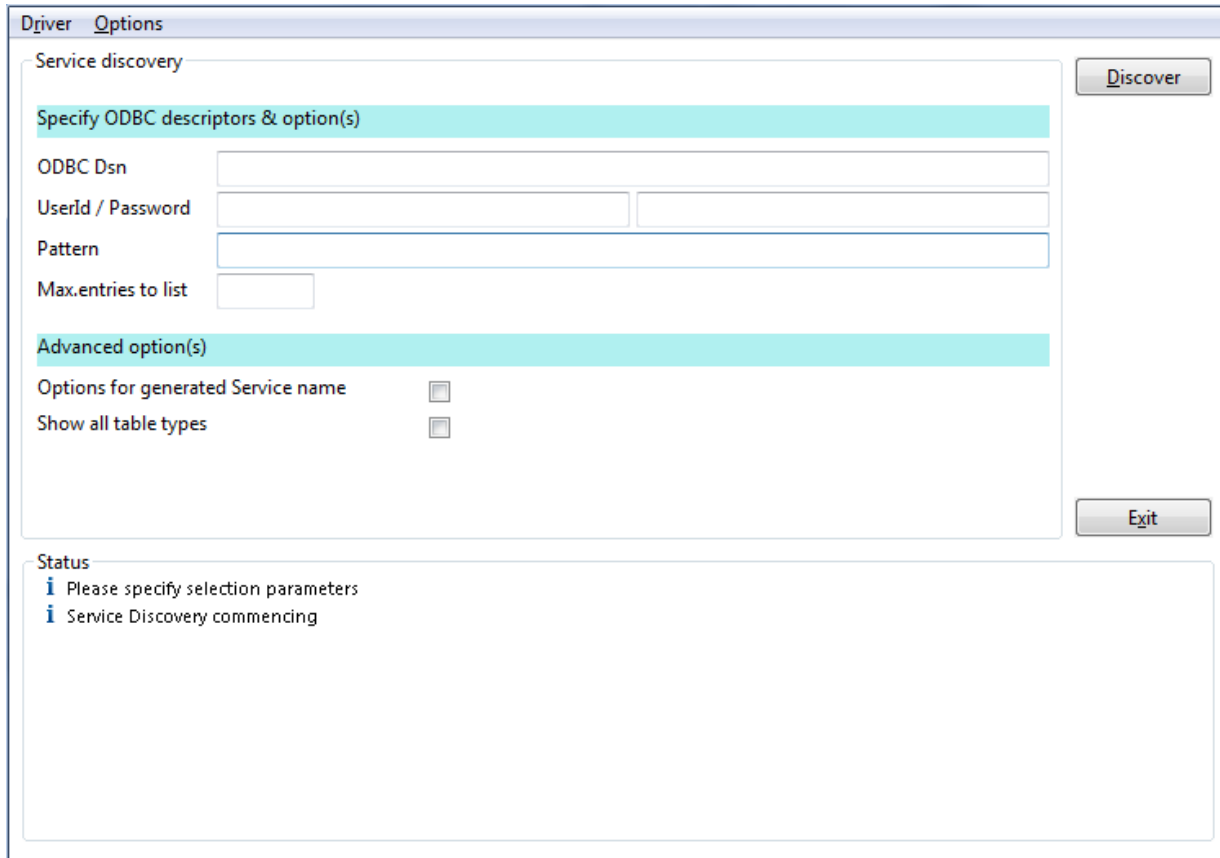
- Click **OK** to exit the wizard.
- Click **OK** to exit the ODBC Data Source Administrator.

Portus Configuration

Portus must now be configured to access and use this new SQL Server DSN

- Start the Portus Control Center and add a Portus Server. See here for more information.
- If you do not have an SQL Server driver, add one now. See here for more info
- See here for how to add/discover a Service.
- From the next dialog choose **Microsoft SQL Server Driver** and click **Discover**
- The next dialog prompts you for the the
 1. **ODBC Dsn:** e.g. soa_gw_sqlserver
 2. **UserId / Password.** If this has been set via the DSN then there is no need to supply any values here..
 3. **Pattern:** This will be a pattern match value passed to the SQL Command. The search pattern characters are: an underscore (`_`), which represents any single character and a percent sign (`%`), which represents any sequence of zero or more characters.
 4. **Max. entries to list:** This is a hard-limit on the amount of tables Portus will attempt to discover.

- Options for generated Service name. By default the Service has the odbc dsn name prepended to the table name. By selecting this option you can change the name to suit you requirements.
- Show all tables types. If this is selected then all tables types are discovered. Usually this is best left unselected.



Now click **Discover**

- Portus will ask the database (identified by the soa_gw_sqlserver DSN) to display all the tables which match the request. In our case, a number of tables were returned. Select these 4 tables, and click **Generate**
- The results of the import will be displayed in the Status pane.
- Click **Exit** to finish the Service Discovery
- You have now created a number of Web Services based on tables from the AdventureWorks database!

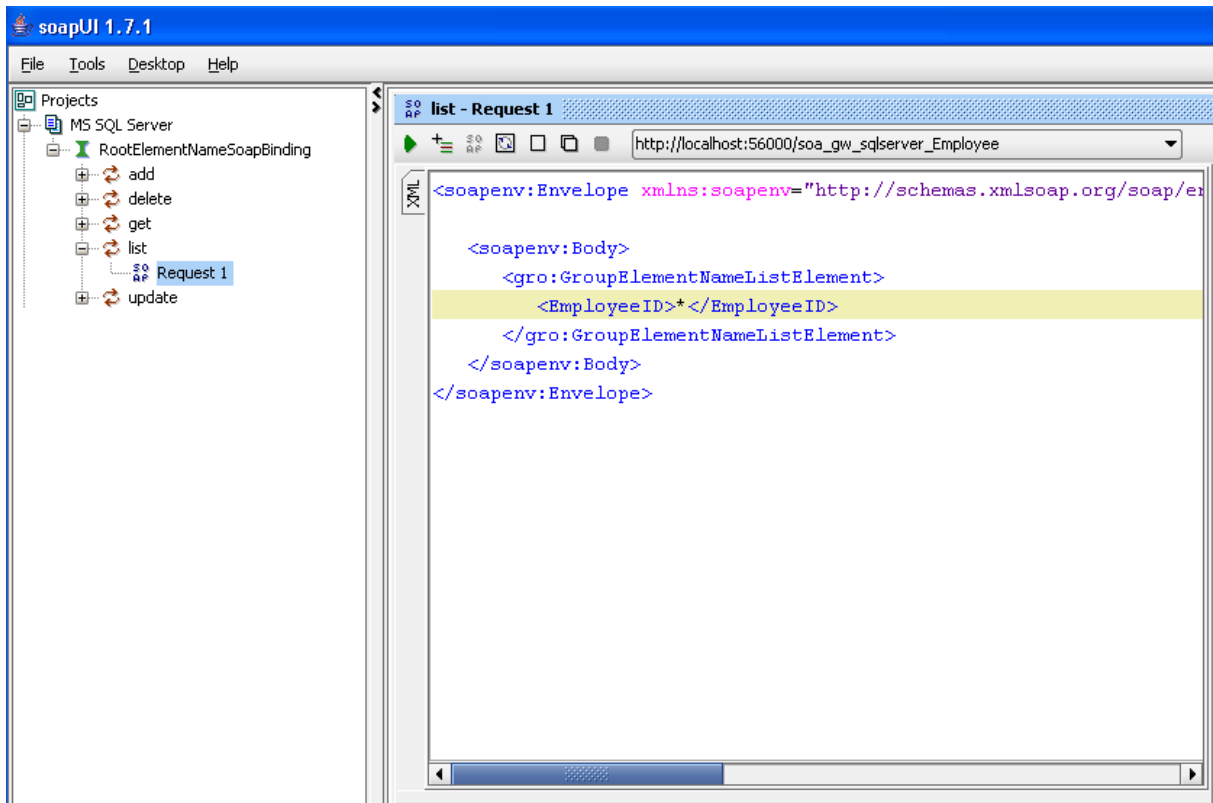
Accessing SQL Server using Portus

Now that the Web Services have been set up, you can access the Web Service Description Language (WSDL) by clicking in the WSDL URL link in the Control Centre.

This WSDL is the starting point to accessing the your tables and stored procedures using Portus. There are many clients available to consume and use web services, for example soapUi, XMLSpy, and Infopath.

A tutorial on how to access Adabas data through soapUI is available here. The follow example is based on that tutorial, and shows how to access the Employees table we have just enabled.

- Start soapUi and create a new WSDL project.
- Import the WSDL
- Edit the list request
- Completely remove the <GroupNameHeader> element from the <soap:Header> element
- In the <Security> element, add Username and Password for accessing the Employees table if required. Otherwise, the <Security> element, and indeed the rest of the <Header> element can be removed.
- Add "*" as the content of the <EmployeeID> element (same as `SELECT * from Employee;`)
- E.g
-



- Hit the green arrow, and the results of the request should be displayed

```

XML
<soapenv:Envelope xmlns:gro="com.SOAGateway/GroupElementName" xmlns:sec="http://schemas.xml
<soapenv:Body>
  <rsp:RootElementNameElement xmlns:rsp="com.SOAGateway/GroupElementName">
    <RootElementName>
      <GroupElementName>
        <EmployeeID>1</EmployeeID>
        <NationalIDNumber>14417807</NationalIDNumber>
        <ContactID>1209</ContactID>
        <LoginID>adventure-works\guy1</LoginID>
        <ManagerID>16</ManagerID>
        <Title>Production Technician - WC60</Title>
        <BirthDate>1972-05-15 00:00:00.000</BirthDate>
        <MaritalStatus>M</MaritalStatus>
        <Gender>M</Gender>
        <HireDate>1996-07-31 00:00:00.000</HireDate>
        <SalariedFlag>0</SalariedFlag>
        <VacationHours>21</VacationHours>
        <SickLeaveHours>30</SickLeaveHours>
        <CurrentFlag>1</CurrentFlag>
        <rowguid>AAE1D04A-C237-4974-B4D5-935247737718</rowguid>
        <ModifiedDate>2004-07-31 00:00:00.000</ModifiedDate>
      </GroupElementName>
      <GroupElementName>
        <EmployeeID>2</EmployeeID>
        <NationalIDNumber>253022876</NationalIDNumber>
        <ContactID>1030</ContactID>
        <LoginID>adventure-works\kevin0</LoginID>
        <ManagerID>6</ManagerID>
        <Title>Marketing Assistant</Title>
        <BirthDate>1977-06-03 00:00:00.000</BirthDate>
        <MaritalStatus>S</MaritalStatus>
        <Gender>M</Gender>
        <HireDate>1997-02-26 00:00:00.000</HireDate>
        <SalariedFlag>0</SalariedFlag>
        <VacationHours>42</VacationHours>
        <SickLeaveHours>41</SickLeaveHours>

```

SOAP Response Response Attachments (0) HTTP Headers (5) SSL Info (-)

28 : 18

Congratulations! You have now accessed an MS SQL Server table using Portus!

27

Using Portus to access the Sybase database

- Introduction 286
- ODBC Setup 286
- Portus Configuration 287
- Accessing SQL Server using Portus 289

Using Portus to access the Sybase EAServer database Adaptive Server Anywhere.

Introduction

The EAServer product set from Sybase includes Adaptive Server Anywhere. Adaptive Server Anywhere provides a full-featured SQL database for EAServer applications. It is designed to operate in environments with limited physical and database administration resources. Adaptive Server Anywhere is a transaction-processing relational database management system (RDBMS) with full recovery capabilities, online backup, referential integrity actions, stored procedures, triggers, row-level concurrency control, and a rich SQL language.

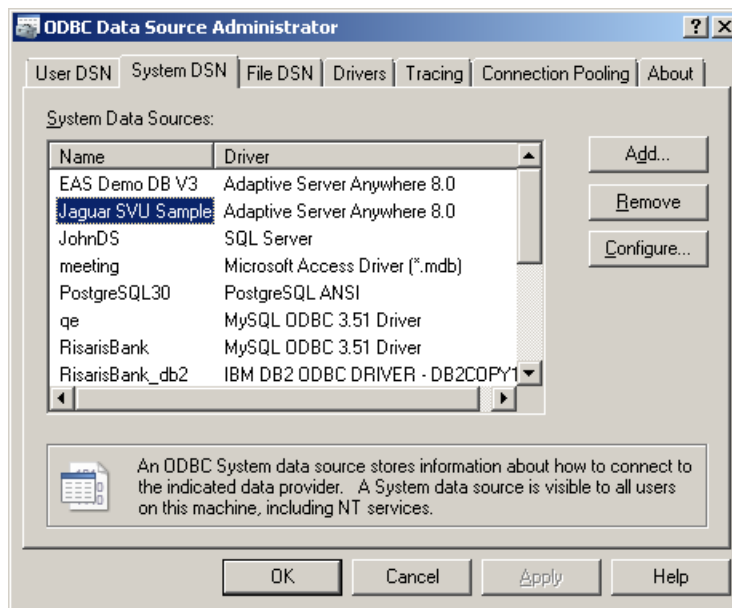
Go to [here](#) to download the Sybase EAServer Developer Edition which is used in this example.

This HOWTO assumes that the Portus and the Sybase database are both running on Windows.

For the purpose of this HOWTO, we will be using one of the sample databases provided by Sybase, "Jaguar SVU Sample".

ODBC Setup

By default, the Sybase installation sets the DSN information required for accessing the sample database:



Check that the login details are correct.

Select the Configure... button having first highlighted the sample database as shown above.

Select the Login tab:



If the User ID and Password is not set set them to dba and sql respectively.

Portus Configuration

Portus must now be configured to access and use this DSN

- Start the Portus Control Center and add a Portus Server (if necessary). See here for more information.
- If you do not have an Sybaser driver, add one now. See here for more information.
- See here for how to add/discover a Service.
- From the next dialog choose **Sybase_Driver** and click **Discover**

- The next dialog prompts you for the the
 1. **ODBC Dsn:** e.g. Jaguar SVU Sample
 2. **UserId / Password.** If this has been set via the DSN then there is no need to supply any values here..
 3. **Pattern:** This will be a pattern match value passed to the SQL Command. The search pattern characters are: an underscore (`_`), which represents any single character and a percent sign (`%`), which represents any sequence of zero or more characters.

Enter `a%` here to return tables starting with the letter `a`.

4. **Max. entries to list:** This is a hard-limit on the amount of tables Portus will attempt to discover.
5. **Options for generated Service name.** By default the Service has the `odbc dsn` name prepended to the table name. By selecting this option you can change the name to suit you requirements.
6. **Show all tables types.** If this is selected then all tables types are discovered. Usually this is best left unselected.

The screenshot shows a dialog box titled "Driver Options" with two tabs: "Driver" and "Options". The "Options" tab is active. The dialog is divided into several sections:

- Service discovery:** Contains a "Discover" button.
- Specify ODBC descriptors & option(s):** A section header with a light blue background. Below it are four input fields: "ODBC Dsn", "UserId / Password", "Pattern", and "Max. entries to list".
- Advanced option(s):** A section header with a light blue background. Below it are two checkboxes: "Options for generated Service name" and "Show all table types", both of which are currently unchecked.
- Status:** A section at the bottom containing two informational messages: "Please specify selection parameters" and "Service Discovery commencing".

Buttons for "Discover" and "Exit" are located on the right side of the dialog.

Now click **Discover**

- Portus will ask the Sybase Server database (identified by the Jaguar SVU Sample DSN) to display all the tables which match the request. In our case, the number of tables starting with 'a' were returned.

Select table allbooks, and click **Generate**

- The results of the import will be displayed in the Status pane.
- Click **Exit** to finish the Service creation
- You have now created a Service based on table allbooks from the Sybase database!

Accessing SQL Server using Portus

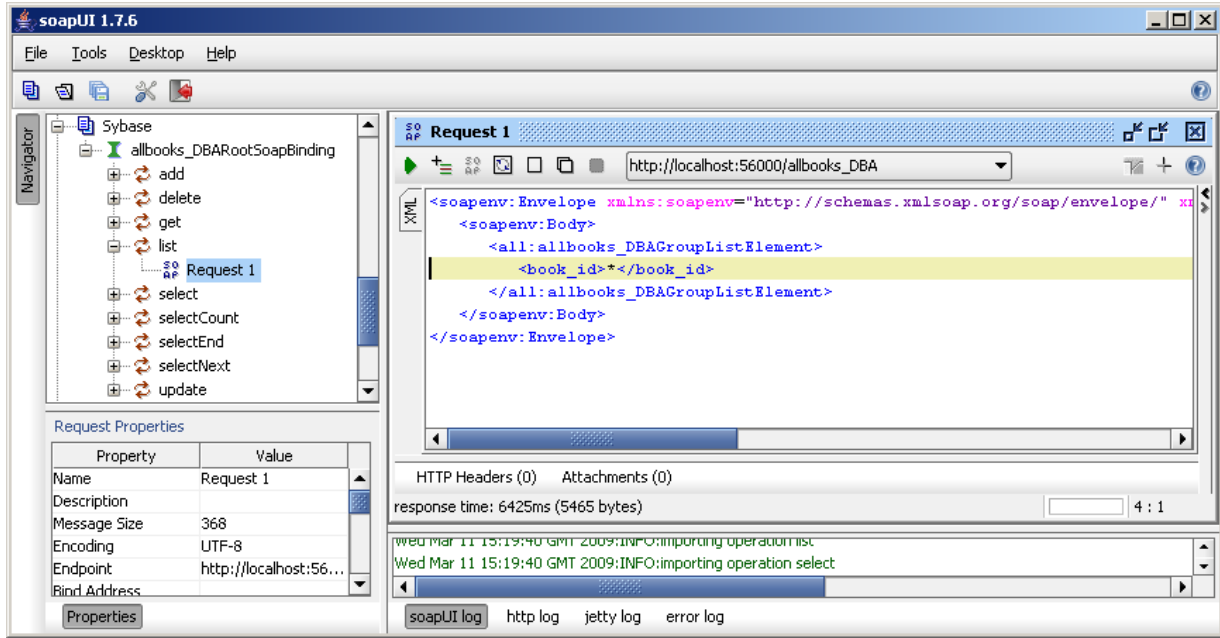
Now that the Web Services have been set up, you can access the Web Service Description Language (WSDL) by clicking in the WSDL URL link in the Control Centre.

This WSDL is the starting point to accessing the your tables and stored procedures using Portus. There are many clients available to consume and use web services, for example soapUI, XMLSpy, and InfoPath.

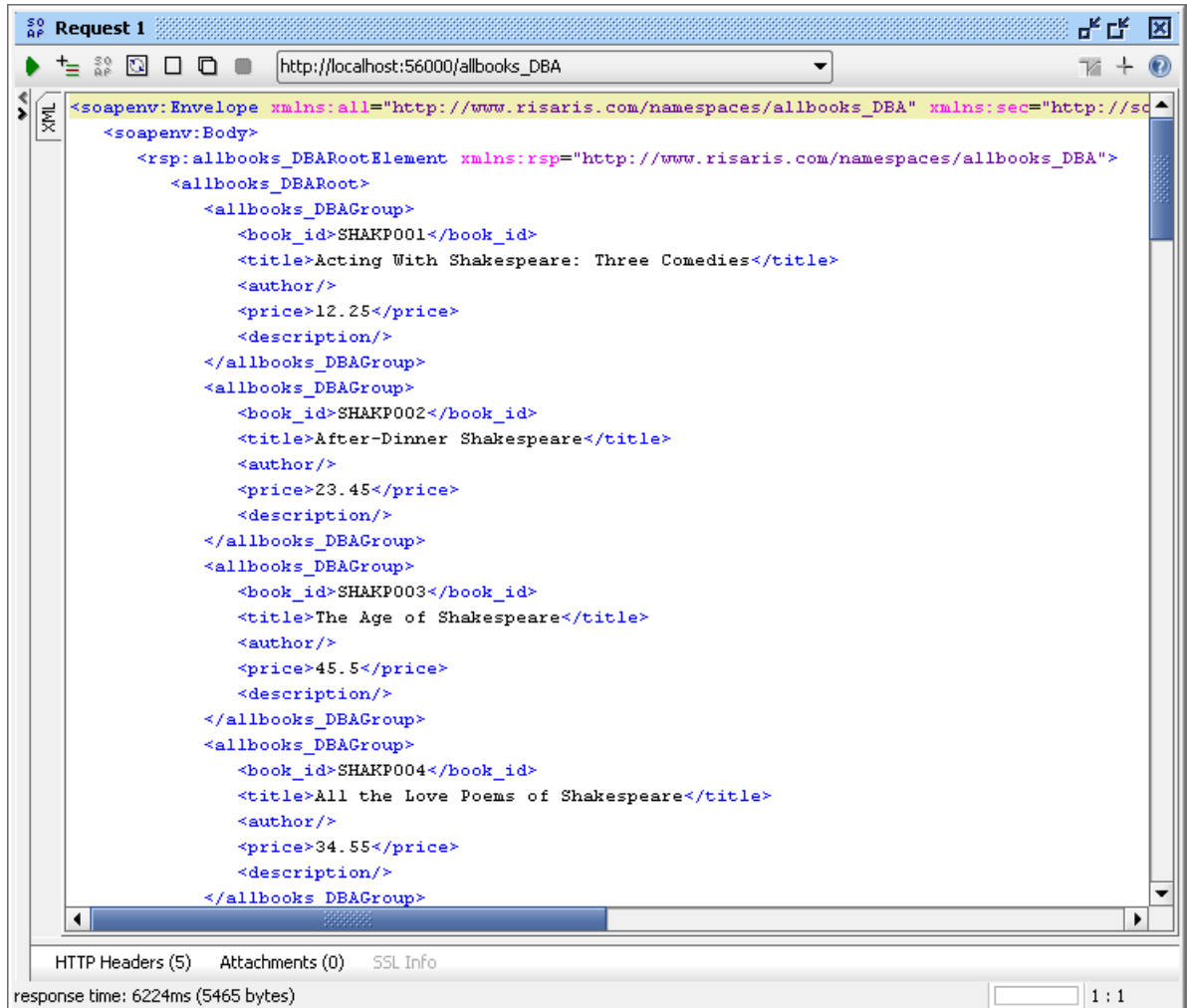
A tutorial on how to access Adabas data through soapUI is available [here](#). The follow example is based on that tutorial, and shows how to access the allbooks table we have just enabled.

- Start soapUI and create a new WSDL project.
- Import using the WSDL
- Edit the list request
- Completely remove the <all:allbooks_DBAGroupHeader> element from the <soap:Header> element
- In the <Security> element, add Username and Password for accessing the allbooks table if required. Otherwise, the <Security> element, and indeed the rest of the <Header> element can be removed as shown below.
- Add "*" as the content of the <book_id> element (same as `SELECT * from allbooks;`)

E.g



- Hit the green arrow, and the results of the request should be displayed



```
Request 1
http://localhost:56000/allbooks_DBA

<soapenv:Envelope xmlns:all="http://www.risaris.com/namespaces/allbooks_DBA" xmlns:sec="http://sc
  <soapenv:Body>
    <rsp:allbooks_DBARootElement xmlns:rsp="http://www.risaris.com/namespaces/allbooks_DBA">
      <allbooks_DBARoot>
        <allbooks_DBAGroup>
          <book_id>SHAKP001</book_id>
          <title>Acting With Shakespeare: Three Comedies</title>
          <author/>
          <price>12.25</price>
          <description/>
        </allbooks_DBAGroup>
        <allbooks_DBAGroup>
          <book_id>SHAKP002</book_id>
          <title>After-Dinner Shakespeare</title>
          <author/>
          <price>23.45</price>
          <description/>
        </allbooks_DBAGroup>
        <allbooks_DBAGroup>
          <book_id>SHAKP003</book_id>
          <title>The Age of Shakespeare</title>
          <author/>
          <price>45.5</price>
          <description/>
        </allbooks_DBAGroup>
        <allbooks_DBAGroup>
          <book_id>SHAKP004</book_id>
          <title>All the Love Poems of Shakespeare</title>
          <author/>
          <price>34.55</price>
          <description/>
        </allbooks_DBAGroup>
      </allbooks_DBARoot>
    </rsp:allbooks_DBARootElement>
  </soapenv:Body>
</soapenv:Envelope>
```

HTTP Headers (5) Attachments (0) SSL Info
response time: 6224ms (5465 bytes) 1:1

Congratulations! You have now accessed a Sybase table using Portus!

28

Using Portus to access Shared Libraries or DLLs

- Introduction 294
- Building the DLL 294
- Portus Configuration 295
- Accessing DLL using Portus 303

Using Portus to access Shared Libraries or DLLs

Introduction

You can also use Portus to access existing shared libraries or DLLs without any rebuilding or reverse-engineering.

You just need to know the name of the function you wish to call, and what the input and output parameters look like.

In this HOWTO, we'll build a piece of code into a DLL, and show you how to use Portus to expose a function within this as a Service.

Building the DLL

Consider this simple piece of C++ code.

```
#include <string.h>

#ifdef WIN32
#define PLATFORM_EXPORT __declspec( dllexport )
#else
#define PLATFORM_EXPORT
#endif

extern "C" PLATFORM_EXPORT int calc( char operation[20], int *operand1, int *operand2, int *result );

extern "C" int calc( char operation[20], int *operand1, int *operand2, int *result ){

    if( !strcmp( operation, "add" ) ){
        *result = (*operand1) + (*operand2) ;
    }
    else if( !strcmp( operation, "sub" ) ){
        *result = (*operand1) - (*operand2) ;
    }
    else if( !strcmp( operation, "mul" ) ){
        *result = (*operand1) * (*operand2) ;
    }
    else if( !strcmp( operation, "div" ) ){
        *result = (*operand1) / (*operand2) ;
    }
    else{
        *result = 0;
    }
}
```



```
return -1;
}

return 0;
}
```

This is simple calc method which will take a string, two integers, and return a integer based on the value of string.

For the purposes of this HOWTO, we are running Portus on Windows, and will build the above code into a Windows DLL using MS VC++.

If Portus is running on Linux, the DLL should be built there, and so on, depending on the platform.

A pre-built Windows (x86) DLL is available here

Portus Configuration

Portus must now be configured to load and run this DLL.

Ideally, our new DLL will be available in the system PATH. On Unix-based systems, you should change to the directory where the DLL is located, and run the command `export PATH=$PATH:$PWD`. Now restart Portus in this shell.

On z/OS, ensure the load library containing the DLL is in your STEPLIB concatenation.

As Windows requires a restart before picking up a new system PATH, the simplest thing to do is copy the built DLL into the bin directory of your SOA Gateway installation. Assuming the default installation was used, this will be `C:\Program Files\Risaris Limited\Portus {v.r.m}\Apache\bin\` where `{v.r.m}` denotes the actual Portus server version in use.

Now start your Portus Control Center and choose one of the following methods of creating a Service based on the "calc" example

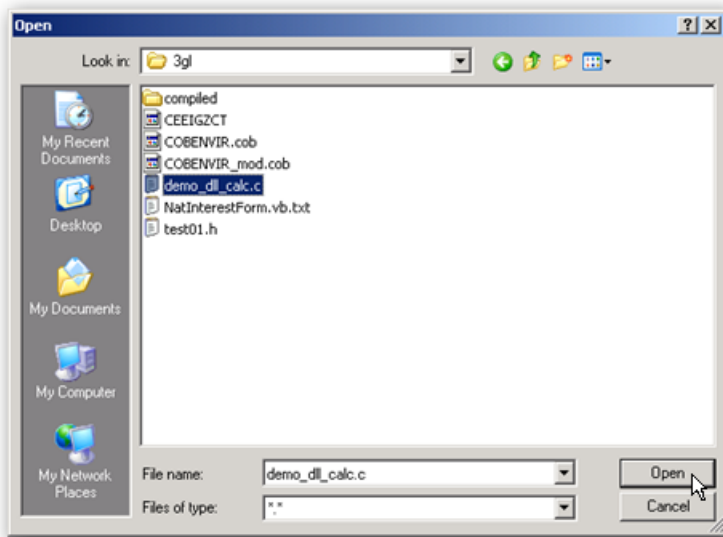
- **Create ("Discover") the Service automatically**
- **Define the Service manually**

Create ("Discover") the Webservice automatically

- Download the C-source to your local disk driver from here
- See here for how to add/discover a Service.
- From the next dialog choose **DLL_Driver** and click **Discover**
- Enter the Path where you downloaded (or compiled) the "Calc" DLL to. If it is located in a directory available on the search path, leave the Path field empty. Click **Discover**
- Select the *demo_dll_calc.dll*, click the **Generate** button
- You will now associate the DLL with the source, which will be used to generate the Portus DataView. From the *Connect to* column's dropdown box select *Source from File System*.

Advance to the *Connect with* column, click it, then click the push-button appearing at the right hand corner

- In the File Dialog popping up, navigate to the location where the C-source for the demo DLL has been downloaded to, select it



For files with an extension of *.c* or *.h* the *Language* will automatically be set to *C*. Click the **Generate** button.

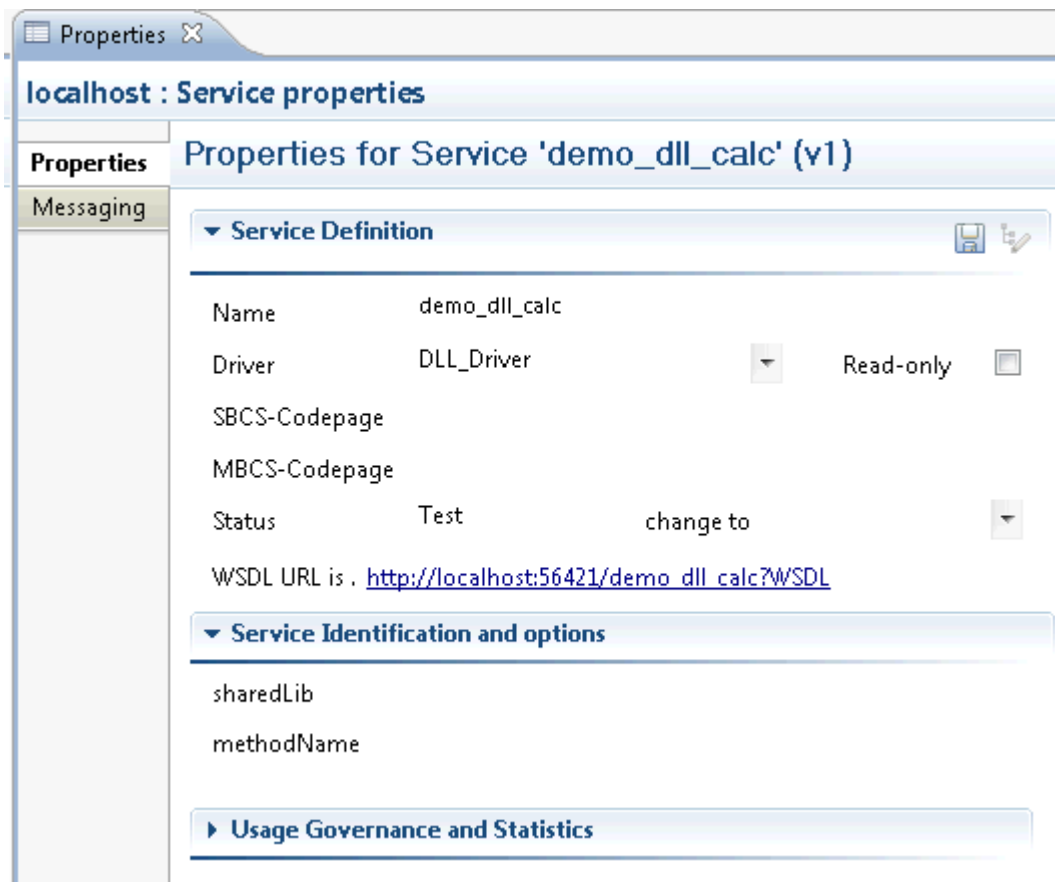
Progress and completion of the generation process will be shown in the Status pane.

- The Service has been created and is usable

Define the Service manually (Portus Perspective)

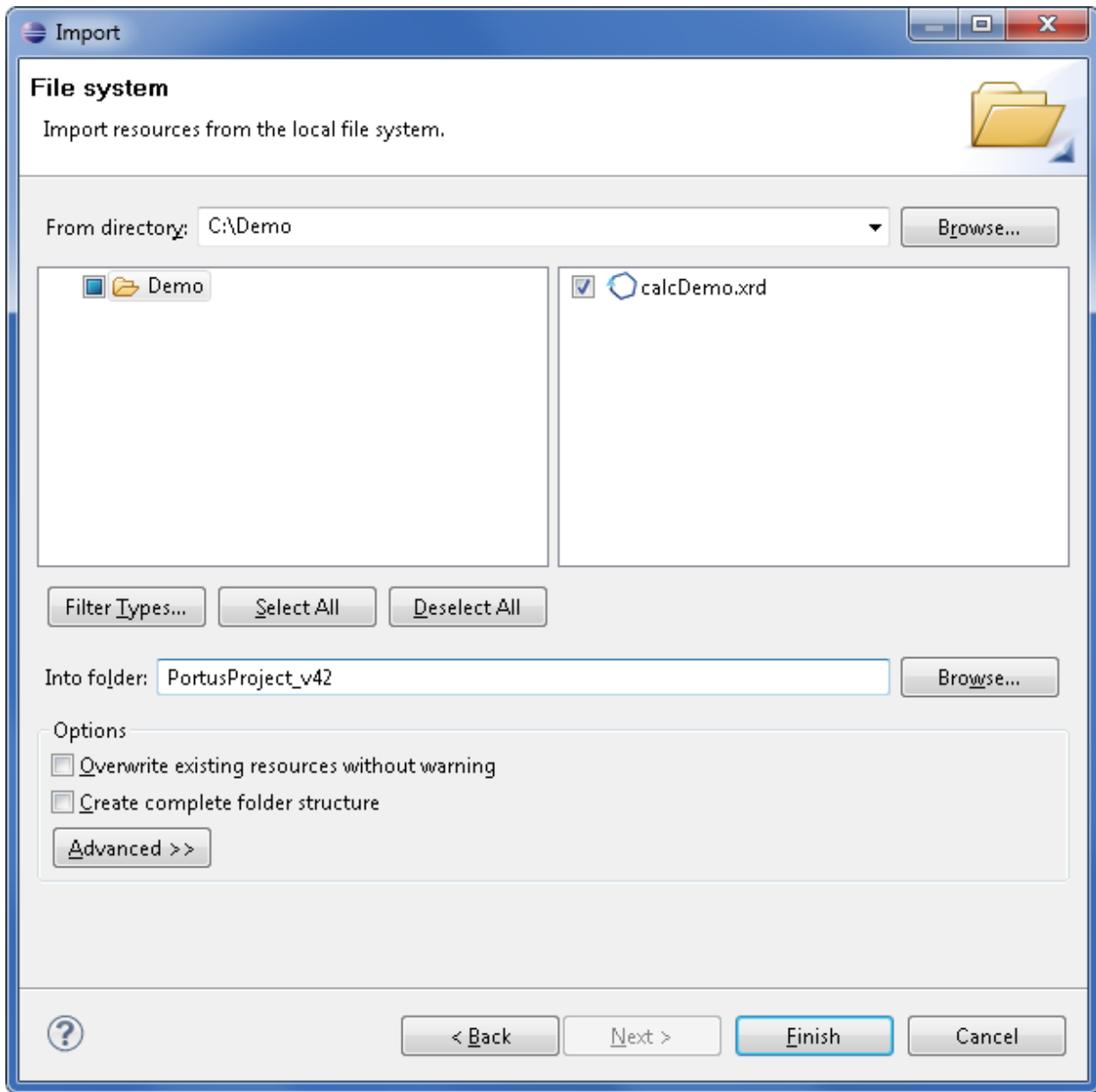
- In the Servers View, right-click on DLL_Driver (or whatever name you might have given it) under Services and select 'Add Service'
- In the Properties View add the following information
 - Name : demo_dll_calc_calc
 - Driver : DLL Driver (or equivalent on your system)
 - sharedLib : demo_dll_calc.dll (the name of the DLL you've built)
 - methodName : calc

Select the Save button.

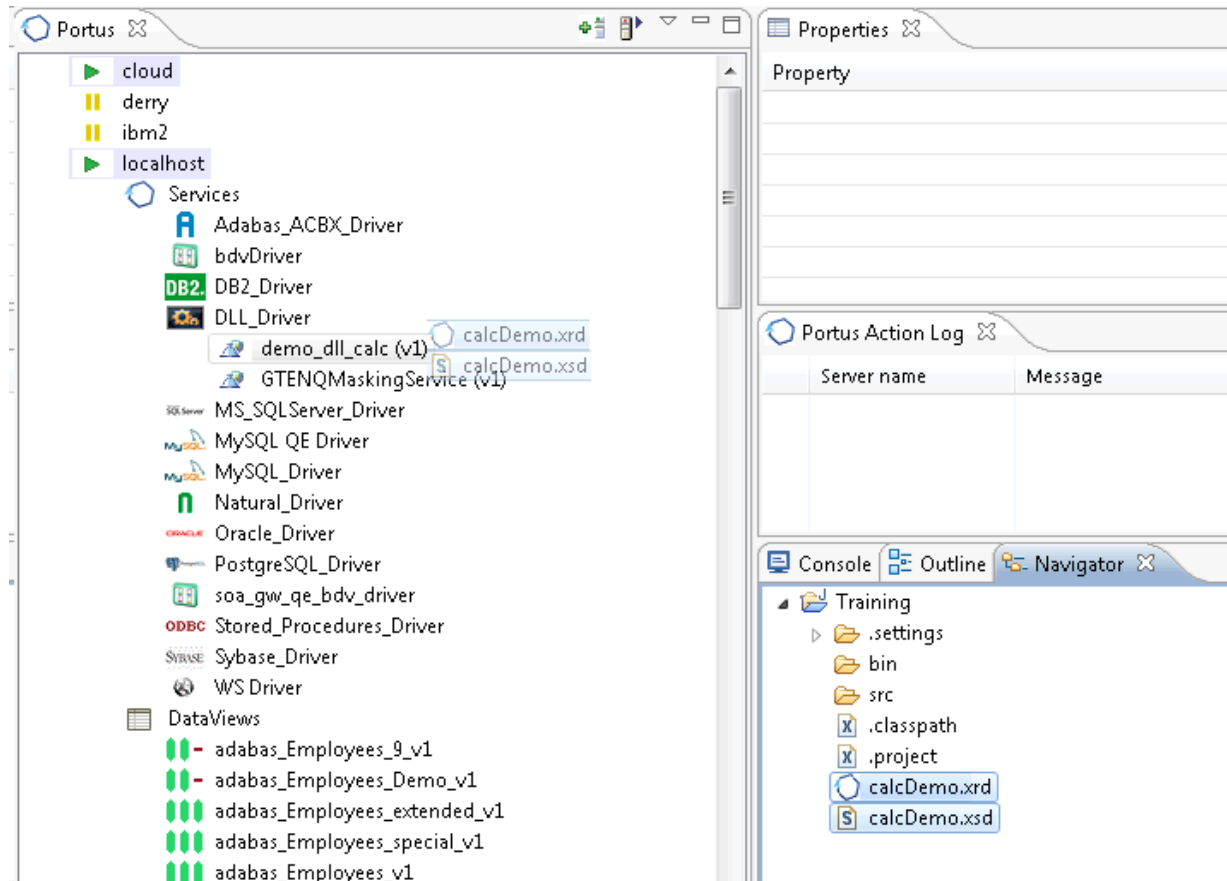


- Save the following XRD file to disk. This is file which maps the calc functions parameters to a format that Portus can understand.
- Open the Navigator View (Window -> Show View -> Navigator).
- Right-click your project, and select **Import**
- Expand **General** and select **File System**. Click **Next**

- Click **Browse** and select the directory where you saved the above XRD. Check the XRD, and click **Finish**



- Select both calcDemo.xsd and calcDemo.xrd. Drag and drop them both onto the demo_dll_calc Service:

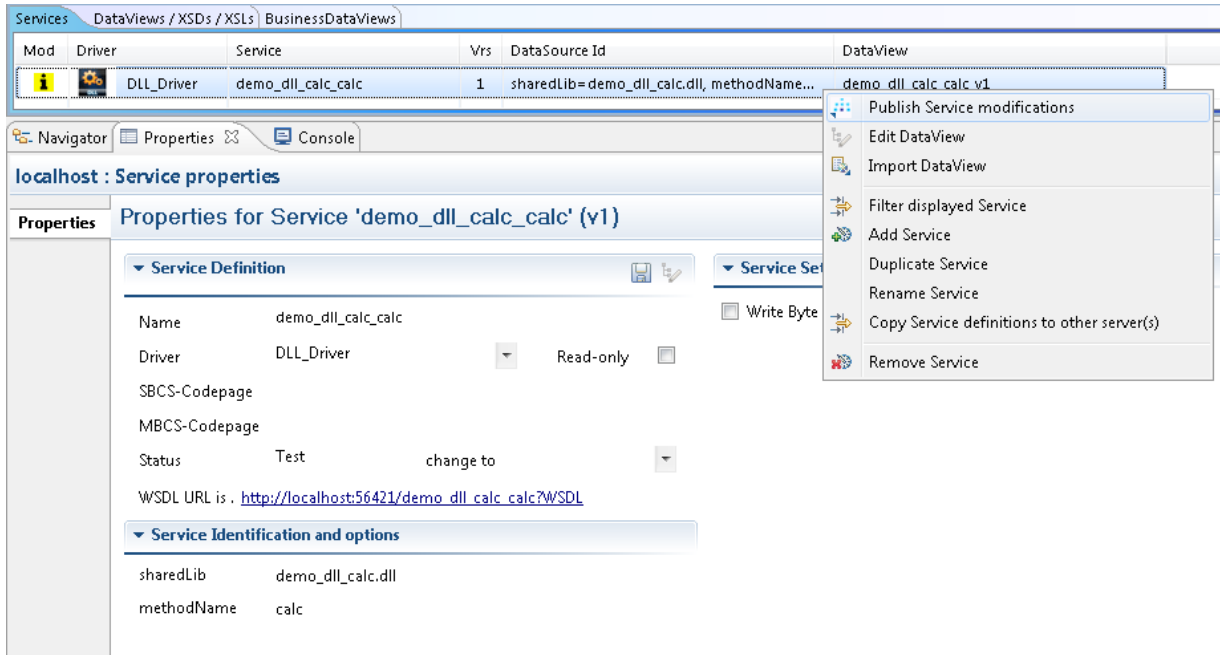


- Right-click the demo_dll_calc Service and select 'Refresh Service'.

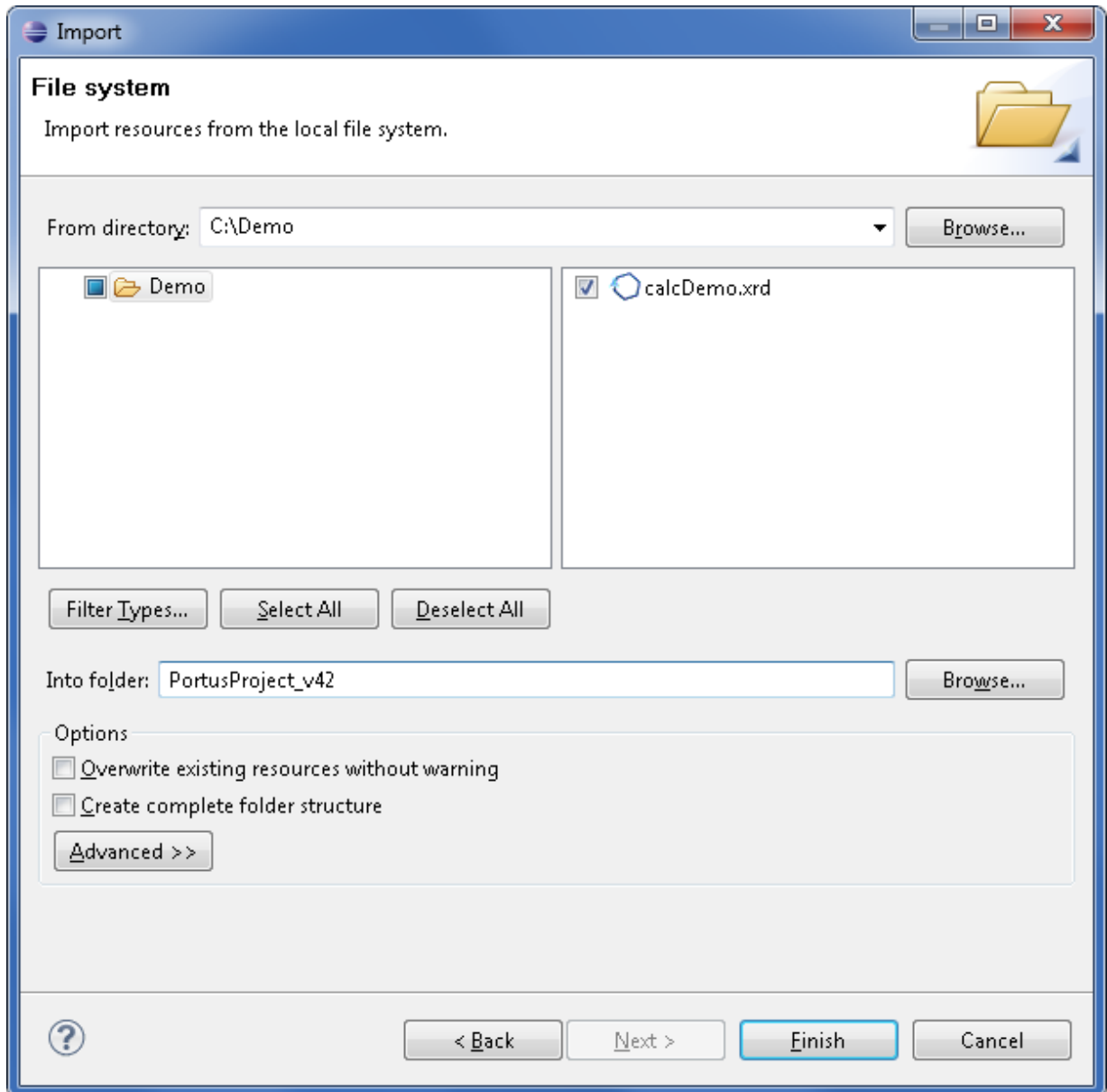
Define the Service manually (Portus Legacy Perspective)

- Add a new Service. See here for more information
- Add the following information
 - Name : demo_dll_calc_calc
 - Driver : DLL Driver (or equivalent on your system)
 - sharedLib : demo_dll_calc.dll (the name of the DLL you've built)
 - methodName : calc

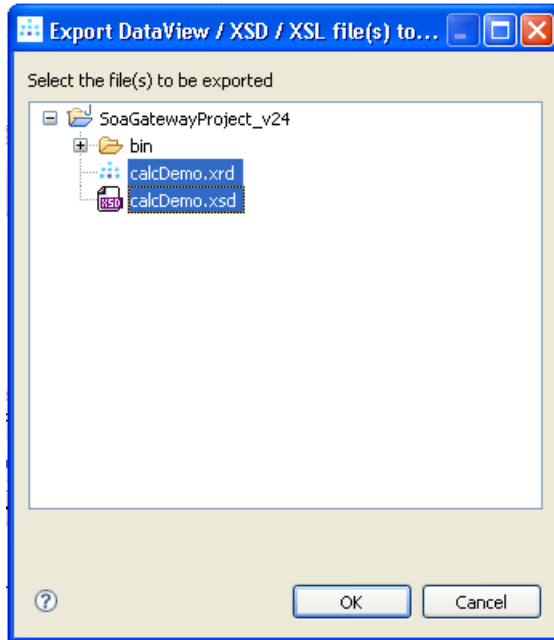
Right-click on the Service and select 'Publish Service modifications'



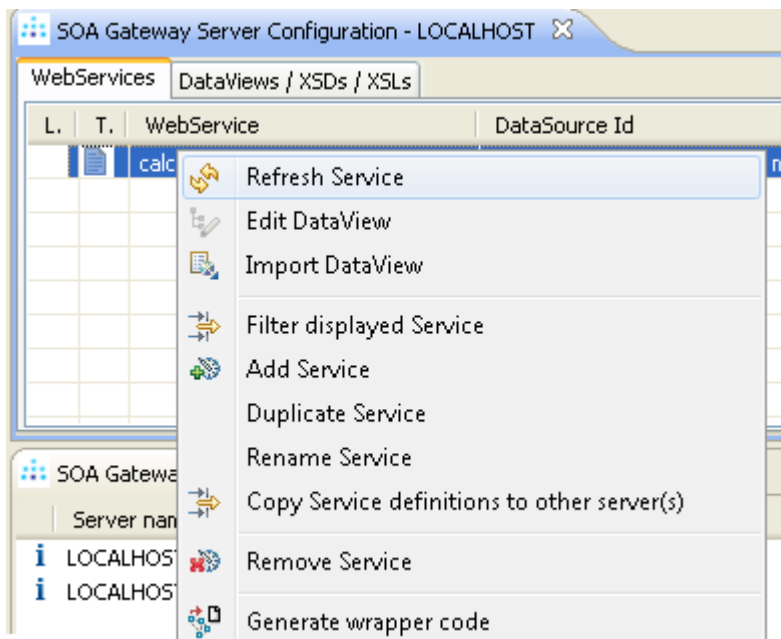
- Save the following XRD file to disk. This is file which maps the calc functions parameters to a format that Portus can understand.
- Open the Navigator View (Window -> Show View -> Navigator).
- Right-click your project, and select **Import**
- Expand **General** and select **File System**. Click **Next**.
- Click **Browse** and select the directory where you saved the above XRD. Check the XRD, and click **Finish**



- In the Server View, right-click your server, and select 'Export Service Definition(s) to server'
- Select both the calcDemo files, and click **OK**



- In the Configuration view, Right-click your Service and select **Refresh Service**



Accessing DLL using Portus

This section will show you how to access our new DLL Web service.

- In the Web Service properties, click the WSDL URL to open your the WSDL for your web service in a browser.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="uri://localhost:56000/calculator_Demo"
  xmlns:tns="uri://localhost:56000/calculator_Demo" xmlns:asg="com.SOAGateway/CalcGroup"
  xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext" name="CalcRootCollection">
- <types>
- <schema targetNamespace="http://schemas.xmlsoap.org/ws/2002/04/secext"
  xmlns="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://schemas.xmlsoap.org/ws/2002/04/secext">
- <xs:element name="Security">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="UsernameToken">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="Username" type="xs:string" />
- <xs:element minOccurs="0" name="Password" type="xs:string" />
- </xs:sequence>
- </xs:complexType>
- </xs:element>
- </xs:sequence>
- </xs:complexType>
- </xs:element>
- </schema>
- <schema targetNamespace="com.SOAGateway/CalcGroup" xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="com.SOAGateway/CalcGroup">
- <xs:element name="invokeInputElement">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="CalcRoot">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="CalcGroup">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="operation" type="xs:string" />
- <xs:element name="operand1" type="xs:int" />
- <xs:element name="operand2" type="xs:int" />
- </xs:sequence>
- </xs:complexType>
- </xs:element>

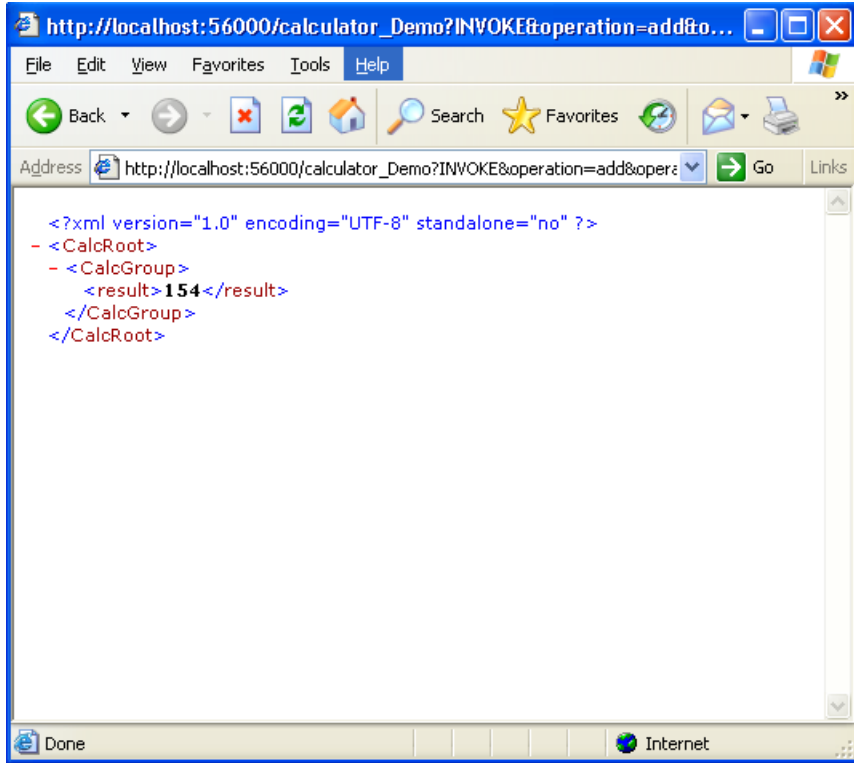
```

- You may import this WSDL to the WS Client of your choice, but for the purpose of this demo, we will use REST-style requests to access the DLL. More information about REST can be found [here](#).

In the browser URL box, remove the ?WSDL characters, and add the following

?INVOKE&operation=add&operand1=55&operand2=99

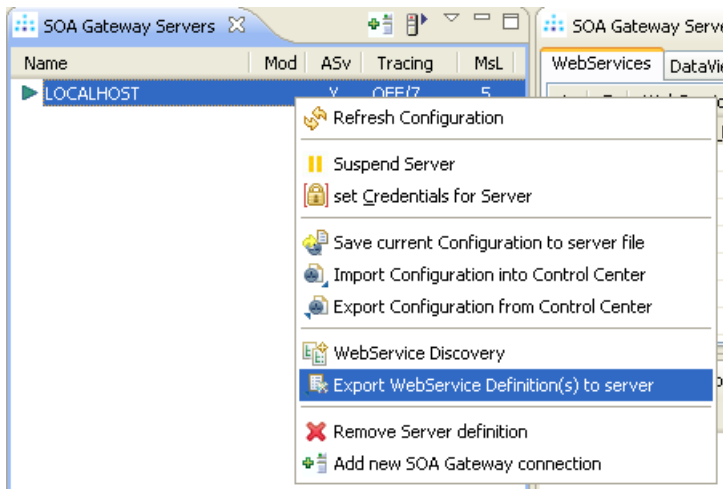
You should get a results something like this



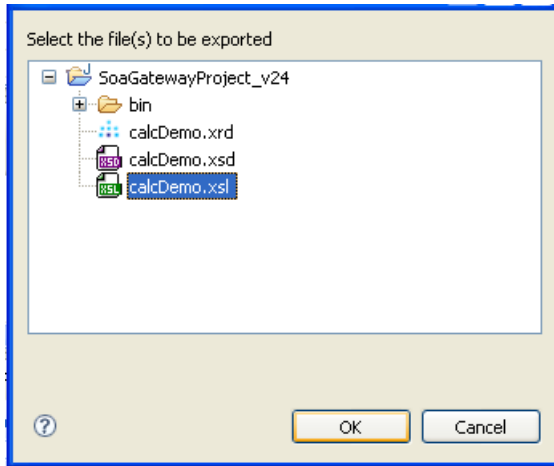
Congratulations! You have now accessed a DLL using Portus

- If you look back at the original C program, you can see how the name=value pairs match up to the parameters of the calc function call. Try changing the operation and operands for different mathematical operations.

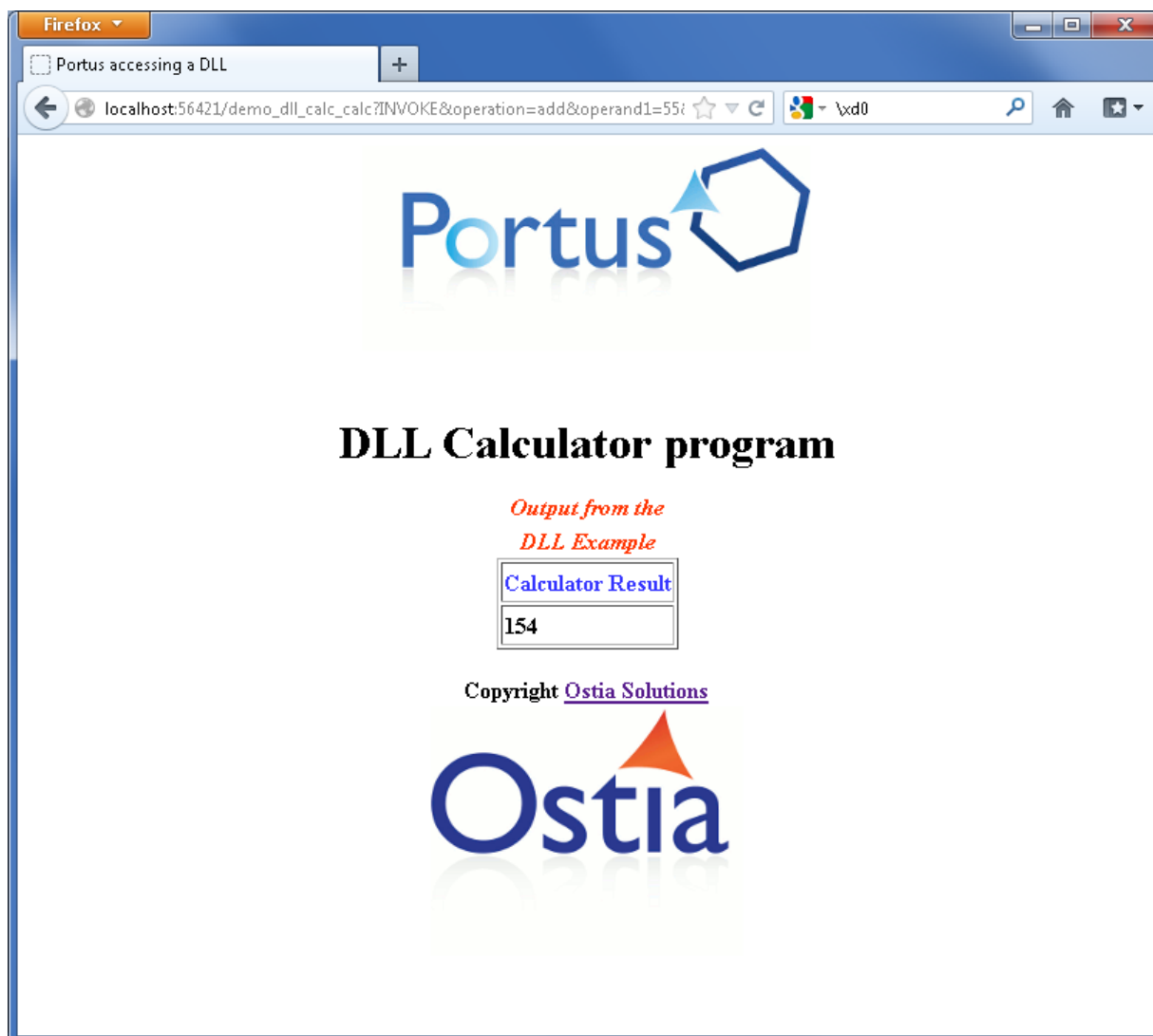
You can also use a XSL Stylesheet which can be used to change the appearance of the output XML. It is available here. As with the XRD previously, import this file into the project, and export it to the server.



And then ...



- Now when you refresh your browser, the stylesheet will be applied and the output should look like this



29

Using Portus to invoke stored procedures

- Introduction 308
- Executing Stored Procedures using Portus 308

Using Portus to invoke stored procedures

Introduction

A stored procedure is a group of SQL statements compiled into a single execution plan. Portus uses the ODBC interface to communicate to a relational database management system (RDBMS) using stored procedures.

Microsoft SQL Server is a relational database management system (RDBMS) produced by Microsoft.

This HOWTO assumes that the Portus and SQL Server database are both running on Windows.

It is assumed that the SQL Server has the ODBC Connector installed.

For the purpose of this HOWTO, we've installed the "AdventureWorks" sample that is provided with the SQL Server installation. See [here](#) for more information about AdventureWorks and other SQL Server sample databases.

Executing Stored Procedures using Portus

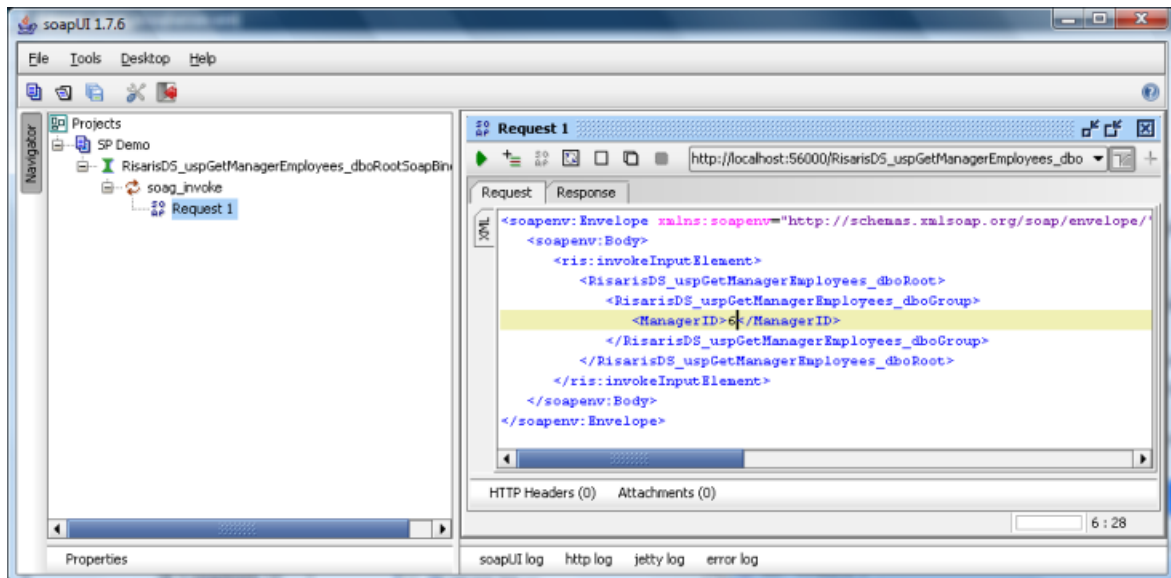
Previously (here) we showed how to create a Service based on an MS SQL stored procedure, uspGetManagerEmployees. The following steps show what needs to be done to execute it.

This WSDL is the starting point to invoking your stored procedures using Portus. There are many clients available to consume and use web services, for example soapUi, XMLSpy, and Infopath. For this example we are using soapUi.

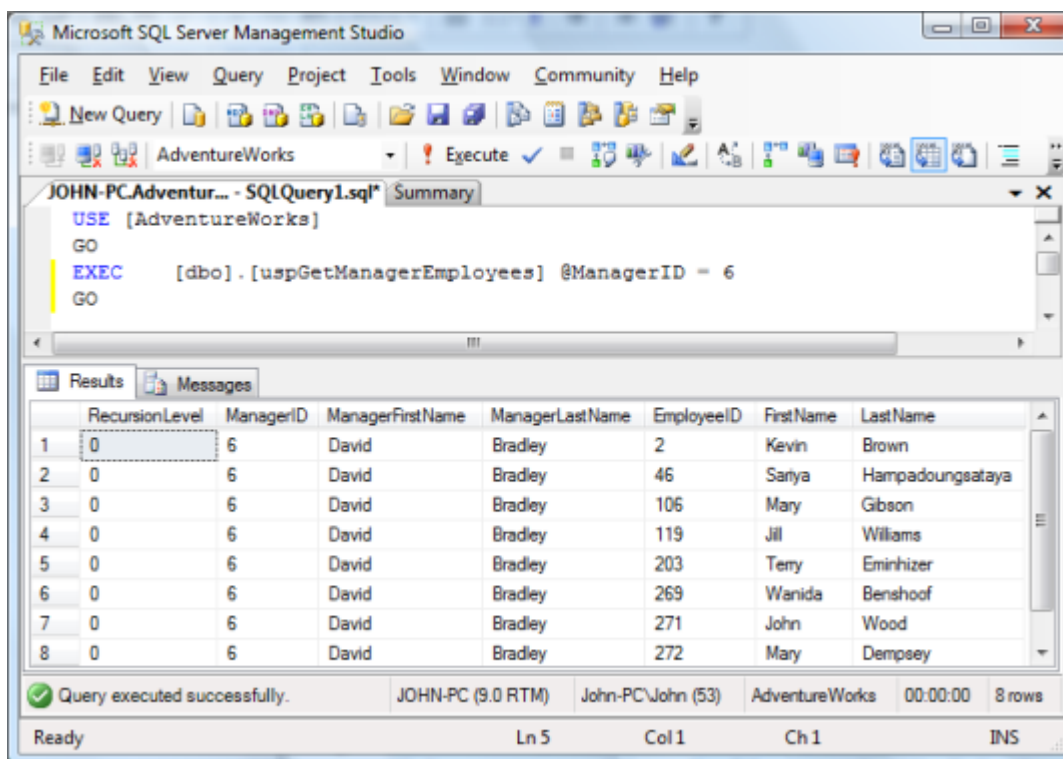
If you are unfamiliar with soapUI a tutorial on using it is available [here](#).

- Start soapUi and create a new WSDL project.
- Import the uspGetManagerEmployees WSDL, E.g. `http://localhost:56000/RisarisDS_ospGetManagerEmployees_dbo?WSDL`
- Edit the soag_invoke request
- Completely remove the `<RisarisDS_ospGetManagerEmployees_dboGroupHeader>` element from the `<soapenv:Header>` element
- In the `<Security>` element, add Username and Password for invoking uspGetManagerEmployees if required. Otherwise, the `<Security>` element, and indeed the rest of the `<Header>` element can be removed as shown below.
- Add the Manager ID value as the content of the `<ManagerID>` element (same as `exec uspGetManagerEmployees 6;)`

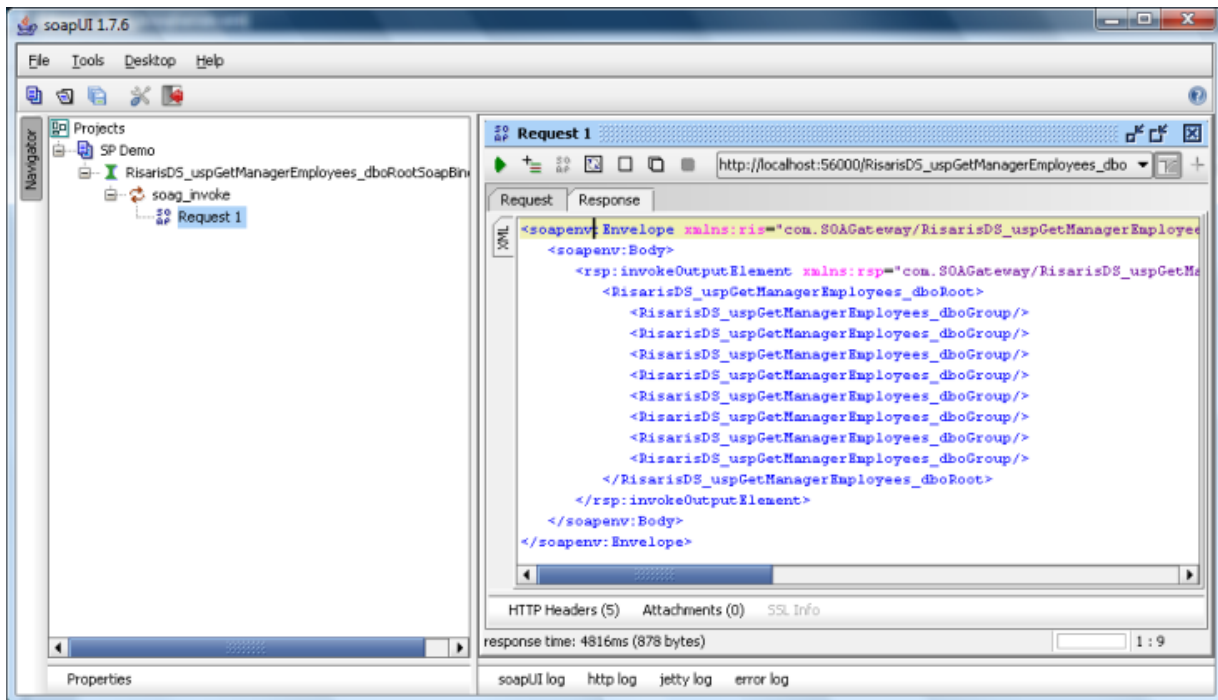
E.g.



- At this point you need to decide which of the resultset columns you want returned. As the resultsets of stored procedures can vary depending on the parameters being passed it is important to know what is expected. In this example you can see the results from executing the `uspGetManagerEmployees` stored procedure in Microsoft SQL Server Studio.

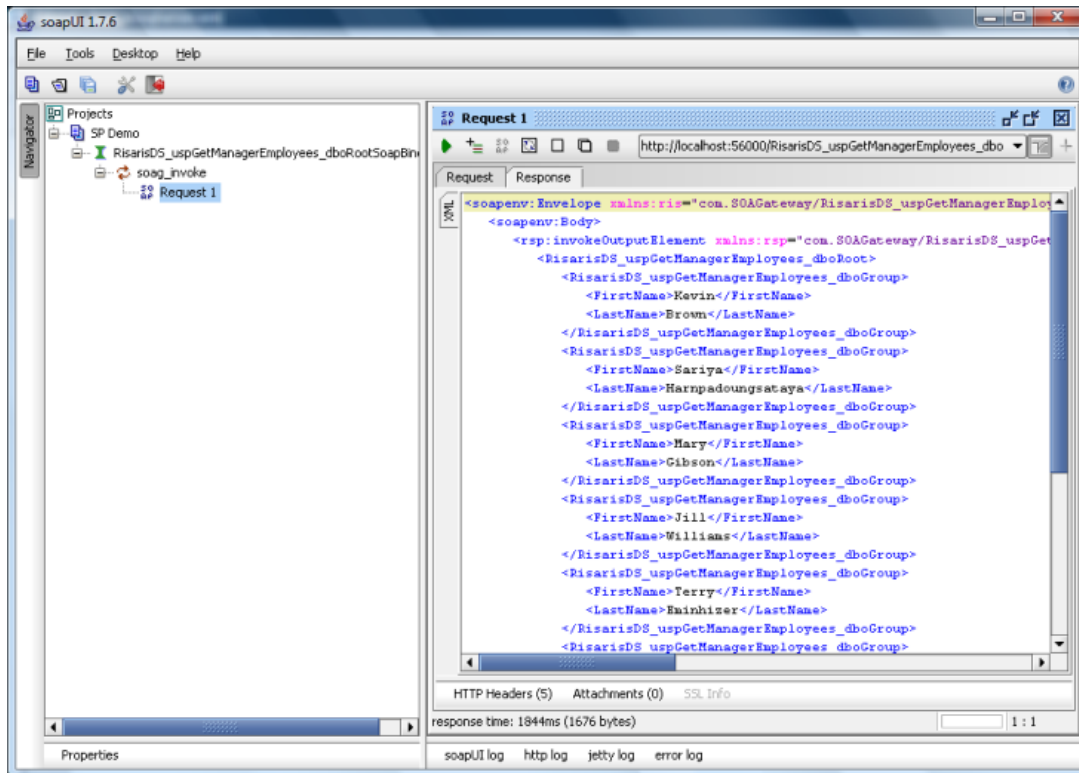


- For illustration in soapUi hit the green arrow, and the results of the request should be displayed as:



Note that the numbers of 'rows' returned is as expected but empty. For each column you wish to return you need to add a definition for it to the XRD previously generated for the Service.

- If not already opened, open the Portus Control Centre. Select the appropriate Portus Server and the Service definition e.g. RisarisDS_uspGetManagerEmployees_dbo.
- We are now going to add 2 new fields to the XRD, FirstName and LastName. Ensure that the internal name matches exactly that in the expected resultset. Take care also to set the correct length for the data type. Select the correct direction option which is output in this case.
- Save the changes. Export the new XRD definition. Refresh the Service.
- Return to soapUI and hit the green arrow, and the results of the request should be displayed as :



Congratulations! You have now executed an MS SQL Server stored procedure using Portus!

30

Using Portus with CICS

- Creating a Portus DataView for a CICS program 314

Portus interfaces with CICS ...

The following is a sample Portus 'WebService' definition for a CICS program exposed as a 'Web Service'

Before a Portus WebService can be used, the mapping between the physical layout (the CICS program's parameters) and the Portus (XML) representation must be in place. This mapping is called the Portus 'DataView'. In addition to the manual approach described in detail in the Data Views section, there are a number of semi-automated methods aiding in the generation of Portus DataViews for CICS resources

1. **dynamically create a DataView for a CICS program**

Creating a Portus DataView for a CICS program

A new DataView can be generated for a CICS program using the 'WebService Discovery wizard', which will not only generate the DataView (XRD), but also a XSD, plus the Portus Resource definition itself.

1. Use the 'WebService creation Wizard' to generate the WebService(s)

31 Using Portus with LE

- Creating a Portus DataView for a LE program 316

Portus interfaces with LE ...

The following is a sample Portus 'Service' definition for a LE program exposed as a 'Service'

Before a Portus Service can be used, the mapping between the physical layout (the LE program's parameters) and the Portus (XML) representation must be in place. This mapping is called the Portus 'DataView'. In addition to the manual approach described in detail in the Data Views section, there are a number of semi-automated methods aiding in the generation of Portus DataViews for LE resources

1. **dynamically create a DataView for a LE program**

Creating a Portus DataView for a LE program

A new DataView can be generated for a LE program using the 'Service Discovery Wizard', which will not only generate the DataView (XRD), but also a XSD, plus the Portus Resource definition itself.

1. Download the COBOL-source for the LE sample program to your local disk drive from here and compile it.
2. Start the Discovery Wizard and select the LE_Driver as described here. Select Next >.
3. Enter the name of the Load Library the sample has been compiled into. Click **Generate**
4. Select the *COBENVIR* sample, click **Import**
5. You will now associate the load module with the source, which will be used to generate the Portus DataView. From the *Connect to* column's dropdown box select *Source from File System*.

Advance to the *Connect with* column, click it, then click the push-button appearing at the right hand corner

6. In the File Dialog popping up, navigate to the location where the COBOL-source for the demo program COBENVIR has been downloaded to, select it

For files with an extension of *.cob* the *Language* will automatically be set to *Cobol*. Click the **Next** button.



Note: For COBOL sources not conforming to the standard COBOL source layout check the "Free format source" option to still have them parsed and the DataView generated correctly.

Progress and completion of the generation process will be shown in the Status pane.

7. The Service has been created and is now usable. Select the Service and view it's properties. If not visible then open Window -> Select View -> Properties
8. Click the URL next to "WSDL URL is .." to display the WSDL exposed by the generated Service:

```

- <schema targetNamespace="http://www.risaris.com/namespaces/COBENVIR">
- <xs:element name="invokeInputElement">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="COBENVIRRoot">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="COBENVIRGroup">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="COBOL_ENV" type="xs:string"/>
- <xs:element name="OS_ENV" type="xs:string"/>
- <xs:element name="COBOL_DATE" type="xs:string"/>
- </xs:sequence>
- </xs:complexType>
- </xs:element>
- </xs:sequence>
- </xs:complexType>
- </xs:element>
- </xs:sequence>
- </xs:complexType>
- </xs:element>

```

9. Overtyping the keyword *WSDL* with *INVOKE&COBOL_ENV=&OS_ENV=&COBOL_DATE=* and execute the request

```

- <COBENVIRRoot>
- <COBENVIRGroup>
- <COBOL_ENV/>
- <OS_ENV/>
- <COBOL_DATE>2009/06/21 10:52:02.271</COBOL_DATE>
- </COBENVIRGroup>
- </COBENVIRRoot>

```

Note: Although the three fields defined in the DataView are in fact "output only" they have to be specified in the request because the Discovery process generates them as "input / output" because it simply cannot know if input is required. This can be changed with the DataView editor manually, but this is beyond the scope of this description.

32 Using Portus with VSAM

- Creating a VSAM Portus Service 320

Portus can interface with VSAM .

Portus can access data in VSAM datasets. RRDS, ESDS and KSDS are supported.

The following will guide you through the process of creating a VSAM based Service.

Creating a VSAM Portus Service

A new DataView can be generated to access a VSAM resource using the 'Service Discovery Wizard', which will not only generate the DataView (XRD), but also a XSD, plus the Portus Resource definition itself.

1. Start the Discovery Wizard using the VSAM_Driver as described here.
2. Enter the High Level Qualifier of your VSAM file. You may also optionally enter the Catalog name. Click **Discover**
3. Select the VSAM dataset from the list returned, click **Generate**
4. Under *Connect to* , select the DataView entry.

Advance to the *Connect with* column, click it, then click the push-button appearing at the right hand corner, and associate it with a DataView entry.

5. Alternatively, instead of a DataView, you can associate the VSAM file with Cobol code. Under *Connect to* , select Source from File System.
6. Advance to the *Connect with* column, click it, then click the push-button appearing at the right hand corner, and associate it. Now browse for the source to associate with this VSAM Service.
7. The selected cobol source will be parsed and data structures in the Linkage and Work Storage Sections will be presented.
8. Select the data structures from the list and click Continue. The Service will be created and published.

33

Using Portus to access Web Services

- Introduction 322
- Accessing Web Services 322

Web services are Web based applications that use open, XML-based standards and transport protocols to exchange data with clients.

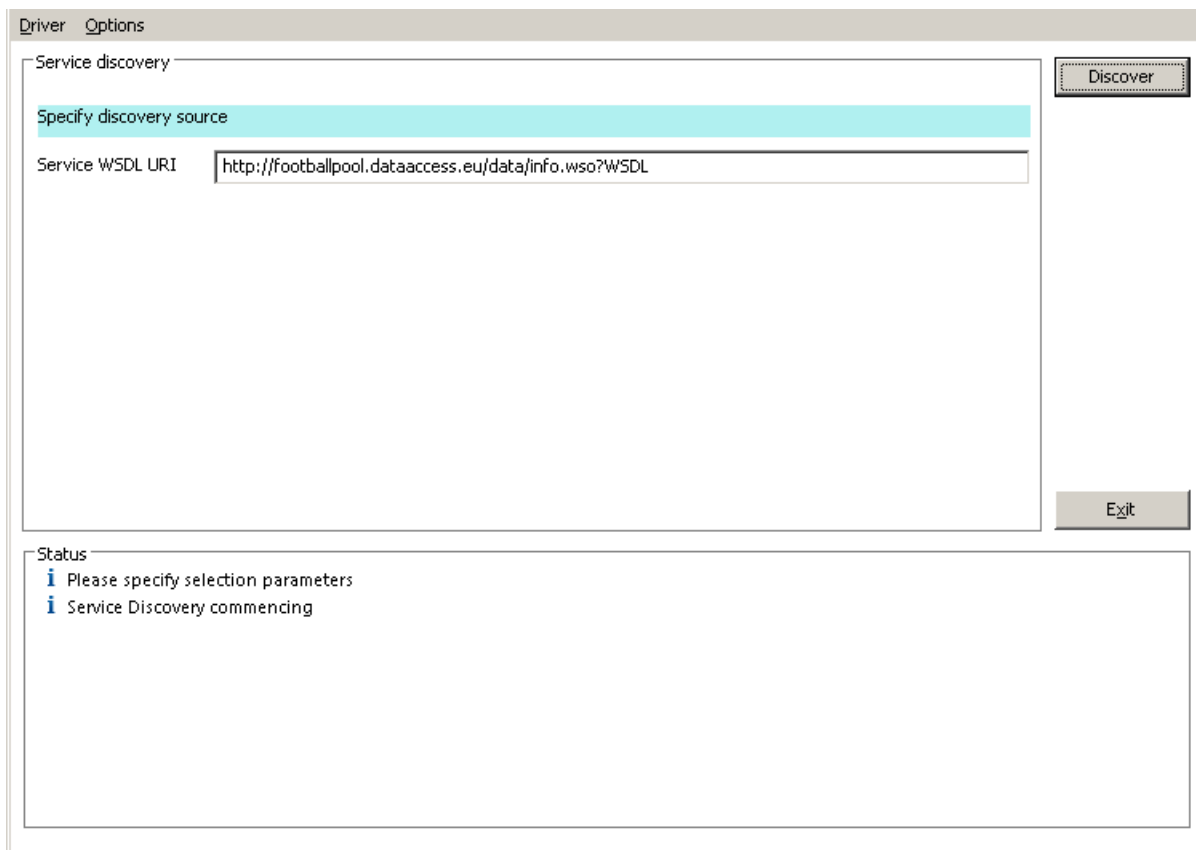
Introduction

Portus can be used to access other web services. The web service driver queries the WSDL exposed by these services and creates a service to enable these to be called.

Accessing Web Services

In this example we will use the web service <http://footballpool.dataaccess.eu/data/info.wso>.

- In the Control Centre Portus tab, select the chosen server. Under Services select the web service driver. Right-click and select Discover Service(s) :
- Specify the Service WSDL URI and click the 'Discover' button:



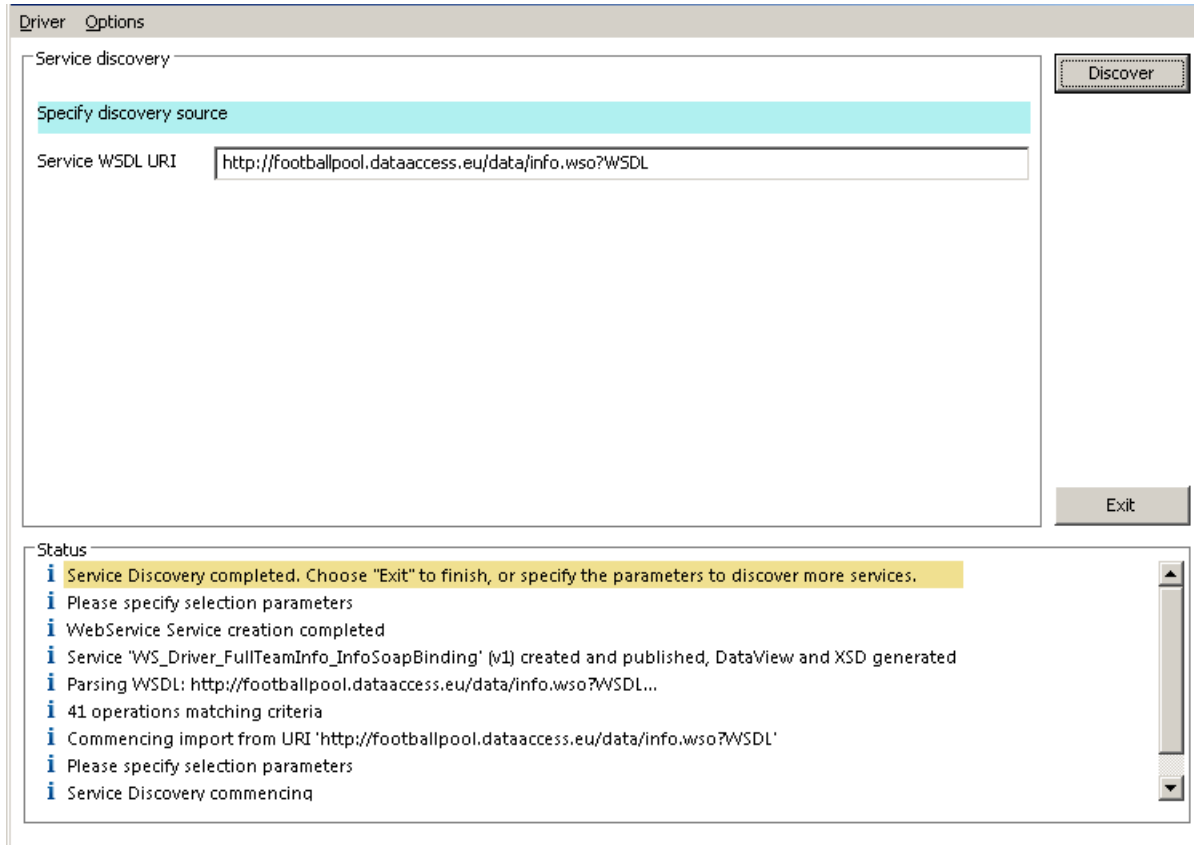
- The Status section will display the progress of each stage of the web service discovery and the Service discovery section will display the following information retrieved from the service description:

The screenshot shows the Portus interface with the following components:

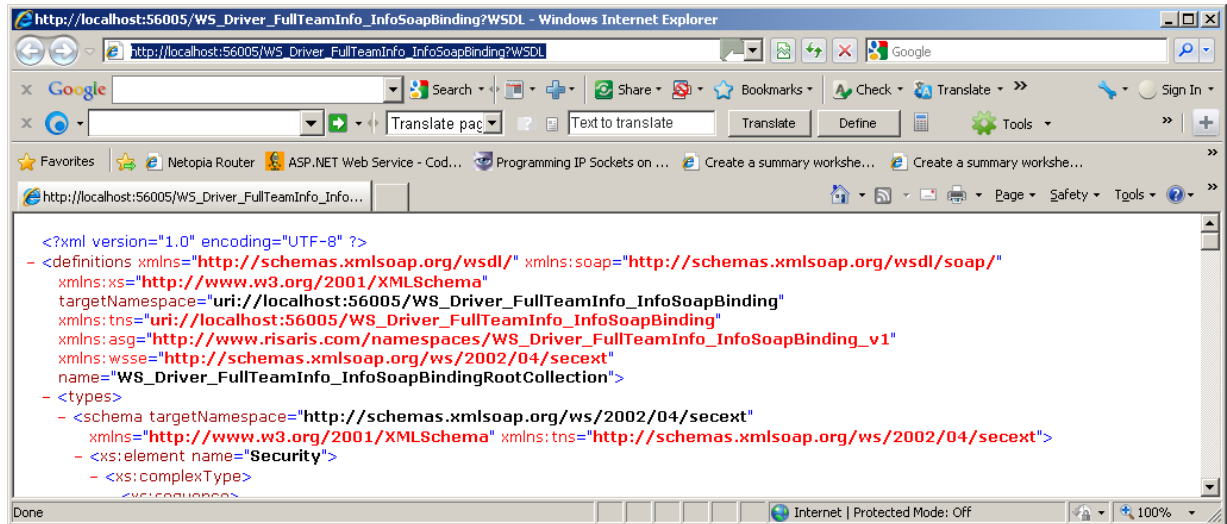
- Service discovery table:**

Service name	Operation	Binding
<input type="checkbox"/> WS_Driver_AllPlayersWithYellowCards_In...	AllPlayersWithYellowCards	InfoSoapBinding
<input type="checkbox"/> WS_Driver_AllPlayersWithRedCards_Info...	AllPlayersWithRedCards	InfoSoapBinding
<input type="checkbox"/> WS_Driver_AllCards_InfoSoapBinding	AllCards	InfoSoapBinding
<input type="checkbox"/> WS_Driver_Cities_InfoSoapBinding	Cities	InfoSoapBinding
<input type="checkbox"/> WS_Driver_StadiumNames_InfoSoapBinding	StadiumNames	InfoSoapBinding
<input type="checkbox"/> WS_Driver_StadiumURL_InfoSoapBinding	StadiumURL	InfoSoapBinding
<input type="checkbox"/> WS_Driver_StadiumInfo_InfoSoapBinding	StadiumInfo	InfoSoapBinding
<input type="checkbox"/> WS_Driver_AllStadiumInfo_InfoSoapBinding	AllStadiumInfo	InfoSoapBinding
<input checked="" type="checkbox"/> WS_Driver_FullTeamInfo_InfoSoapBinding	FullTeamInfo	InfoSoapBinding
<input type="checkbox"/> WS_Driver_Teams_InfoSoapBinding	Teams	InfoSoapBinding
<input type="checkbox"/> WS_Driver_GroupCount_InfoSoapBinding	GroupCount	InfoSoapBinding
<input type="checkbox"/> WS_Driver_Groups_InfoSoapBinding	Groups	InfoSoapBinding
<input type="checkbox"/> WS_Driver_GroupCompetitors_InfoSoapB...	GroupCompetitors	InfoSoapBinding
- Buttons:** Generate >, < Back, Select all, Deselect all, Exit.
- Status section:**
 - i 41 operations matching criteria
 - i Commencing import from URI 'http://footballpool.dataaccess.eu/data/info.wso?WSDL'
 - i Please specify selection parameters
 - i Service Discovery commencing

- Use Select all to choose all operations or select on an individual basis. For clarity we have chosen the FullTeamInfo operation only. When chosen click the Generate button.
- The Status section will be updated with the progress of the generation process, listing the operations selected and the results for each one:



- The Wizard is still active and ready for further discovery, click Discover to start the process again, or Exit to dismiss the wizard.
- The generated Service(s) will be listed under the Web Service driver.
- To view the WSDL generated select 'WSDL URL is..' : which can be found in the Properties View for the service. This will open in your default browser e.g.

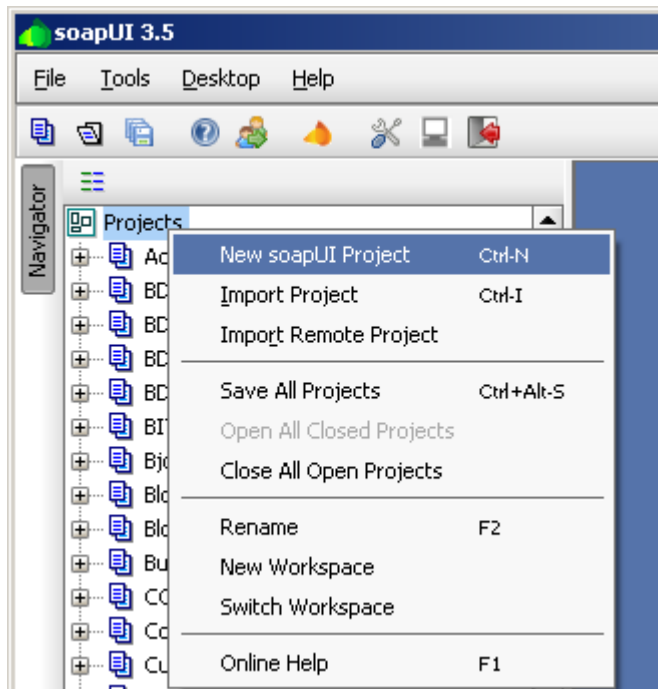


```

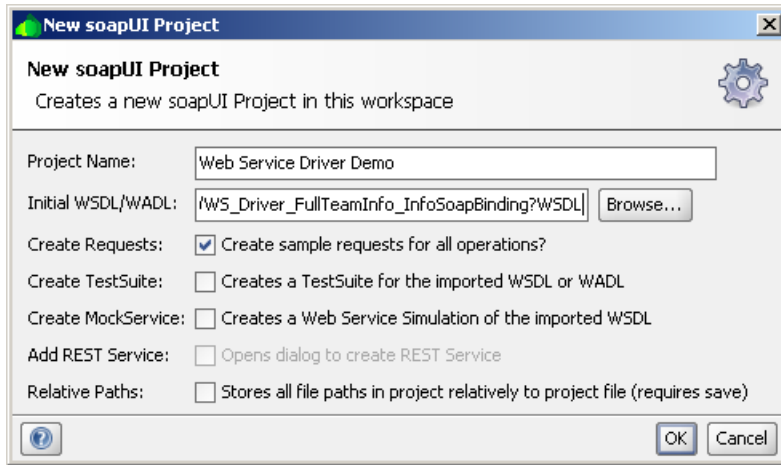
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="uri://localhost:56005/WS_Driver_FullTeamInfo_InfoSoapBinding"
  xmlns:tns="uri://localhost:56005/WS_Driver_FullTeamInfo_InfoSoapBinding"
  xmlns:asg="http://www.risaris.com/namespaces/WS_Driver_FullTeamInfo_InfoSoapBinding_v1"
  xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
  name="WS_Driver_FullTeamInfo_InfoSoapBindingRootCollection">
- <types>
- <schema targetNamespace="http://schemas.xmlsoap.org/ws/2002/04/secext"
  xmlns="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://schemas.xmlsoap.org/ws/2002/04/secext">
- <xs:element name="Security">
- <xs:complexType>
- <xs:sequence>

```

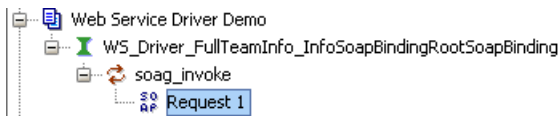
- Here we are using soapUI to demonstrate how this web service can be invoked. Open soapUI, right click on Projects and choose New soapUI Project :



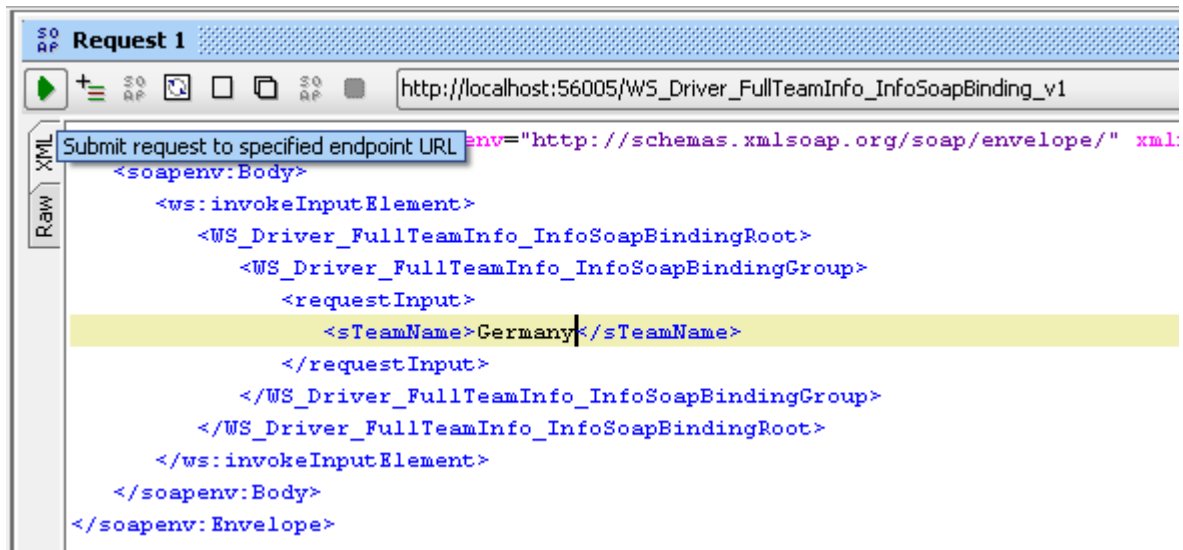
- Enter the name of the project and set the WSDL to the value copied previously :



- When the import is complete double click on the Request 1 entry to open the dialog :



- For this request there is no need for the <soapenv:Header> so select all of this and delete it. Set "Germany" as the content of the <sTeamName> element and issue the request by selecting the play button :



- The results will be displayed as shown:

The screenshot shows a web service client interface with two panels displaying XML data. The top panel shows the request, and the bottom panel shows the response. The response is a SOAP envelope containing a request output with a full team info result for Germany.

```

Request 1
http://localhost:56005/WS_Driver_FullTeamInfo_InfoSoapBinding_v1

Raw XML
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sec="http://schemas.xmlsoap.org/security/2001/04">
  <soapenv:Body>
    <ws:invokeInputElement xmlns:ws="http://schemas.xmlsoap.org/ws/2003/05/soap-invoke">
      <WS_Driver_FullTeamInfo_InfoSoapBindingRoot>
        <WS_Driver_FullTeamInfo_InfoSoapBindingGroup>
          <requestOutput>
            <FullTeamInfoResult>
              <sName>Germany</sName>
              <sCountryFlag>http://footballpool.dataaccess.eu/images/flags/de.gif</sCountryFlag>
              <sCoach>Joachim Löw</sCoach>
              <iCompeted>16</iCompeted>
              <sGoalKeepers>
                <string>Hans-Jörg Butt</string>
                <string>Manuel Neuer</string>
                <string>Tim Wiese</string>
              </sGoalKeepers>
              <sDefenders>
                <string>Arne Friedrich</string>
                <string>Dennis Aogo</string>
                <string>Holger Badstuber</string>
                <string>Jérôme Boateng</string>
                <string>Marcell Jansen</string>
                <string>Per Mertesacker</string>
                <string>Philipp Lahm</string>
                <string>Serdar Tasci</string>
              </sDefenders>
              <sMidFields>
                <string>Bastian Schweinsteiger</string>
                <string>Marko Marin</string>
                <string>Mesut Özil</string>
                <string>Piotr Trochowski</string>
              </sMidFields>
            </FullTeamInfoResult>
          </requestOutput>
        </WS_Driver_FullTeamInfo_InfoSoapBindingGroup>
      </WS_Driver_FullTeamInfo_InfoSoapBindingRoot>
    </ws:invokeInputElement>
  </soapenv:Body>
</soapenv:Envelope>

Raw XML
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sec="http://schemas.xmlsoap.org/security/2001/04">
  <soapenv:Body>
    <rsp:invokeOutputElement xmlns:rsp="http://www.risaris.com/namespaces/WS_Driver_FullTeamInfo_InfoSoapBinding_v1">
      <WS_Driver_FullTeamInfo_InfoSoapBindingRoot>
        <WS_Driver_FullTeamInfo_InfoSoapBindingGroup>
          <requestOutput>
            <FullTeamInfoResult>
              <sName>Germany</sName>
              <sCountryFlag>http://footballpool.dataaccess.eu/images/flags/de.gif</sCountryFlag>
              <sCoach>Joachim Löw</sCoach>
              <iCompeted>16</iCompeted>
              <sGoalKeepers>
                <string>Hans-Jörg Butt</string>
                <string>Manuel Neuer</string>
                <string>Tim Wiese</string>
              </sGoalKeepers>
              <sDefenders>
                <string>Arne Friedrich</string>
                <string>Dennis Aogo</string>
                <string>Holger Badstuber</string>
                <string>Jérôme Boateng</string>
                <string>Marcell Jansen</string>
                <string>Per Mertesacker</string>
                <string>Philipp Lahm</string>
                <string>Serdar Tasci</string>
              </sDefenders>
              <sMidFields>
                <string>Bastian Schweinsteiger</string>
                <string>Marko Marin</string>
                <string>Mesut Özil</string>
                <string>Piotr Trochowski</string>
              </sMidFields>
            </FullTeamInfoResult>
          </requestOutput>
        </WS_Driver_FullTeamInfo_InfoSoapBindingGroup>
      </WS_Driver_FullTeamInfo_InfoSoapBindingRoot>
    </rsp:invokeOutputElement>
  </soapenv:Body>
</soapenv:Envelope>

```

Congratulations! You have now accessed a Web Service using Portus!

34 Message Oriented Middleware Support in Portus

- Introduction 330

Message Oriented Middleware (MOM) is a category of connectivity middleware that provide program-to-program communications by message passing. MOM, because it generally supports multiple protocols, comprises an infrastructure that will support reliable and scalable high-performance distributed application networks. Most Message Oriented Middleware is implemented with queued message store-and-forward capability and it is this feature the Portus supports.

In particular the systems supported by Portus are IBM WebSphere MQ and Software AG EntireX Broker.

Introduction

There are 2 ways to integrate a messaging system with Portus.

1. To monitor for Portus specific requests, to process these and issue a response.
2. To log Usage Governance Data.

Tutorials are available which demonstrate these 2 capabilities.

The first tutorial uses Portus to write usage governance data to a messaging system and a second Portus to process these and store them in a database. See [here](#) for details on this.

This tutorial uses a Java client to access the WSDL of a web service supporting MQ and call it using SOAP over MQ. See [here](#) for details on this.

35 Usage Governance

- Introduction 332
- Governance Data 332
- Service collection 333
- Controlling Collection 334

Once a service is deployed and operational, it may be necessary to govern its usage. This section describes this functionality and how it works in Portus.

Introduction

When a service has been created and is deployed, there are a number of things that may need to be understood about the usage of service.

- For audit purposes, it may be necessary to know who has used a specific service and when.
- It may be necessary to log the data provided to the service and the data returned by the service to record what the user did.
- It may be necessary to understand who is using a service on specific dates or even within a time range.
- When a service has been deprecated (i.e. replaced by a newer version of the service), it is necessary to understand who is still using the service so that they may be upgraded to the newer level of the service and the older version retired.
- For capacity planning purposes, it is necessary to understand how often a service is being called.
- In order to comply with Service Level Agreements (SLAs), the behaviour of the service under normal circumstances must be understood so that a commitment can be made to a SLA and it can be detected when a service is performing outside its normal band.

Governance Data

The following describes the information that is available to enable an organization to properly govern their various services based on their core software assets.

Version

The web service version.

Status

The web service status i.e. Test, Frozen, Deprecated or Historical.

Identity

The user calling the web service.

Time

Time is recorded at various stages of the service invocation e.g. the time the server received the request, the time the input was processed, the time the output was processed and the time the request was returned.

Location

This records the IP address from where the request originated and the IP address of the server.

Operation

This records the HTTP SOAP method and the request operation.

Data

This records the request input data and the request output data.

Service collection

Portus can be configured to collect data or not at the service name level. This represents the logical name of the service which may have different versions of the service active at any one time in a given Portus server instance. The following information may be collected about each service.

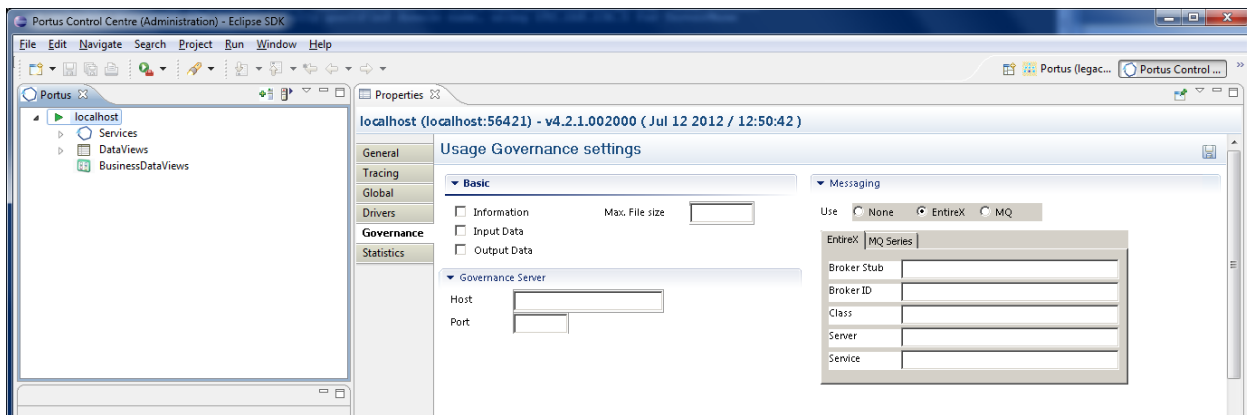
- No governance information about the service at all.
- Detailed access information about the service. This will record the access data (e.g. date and time of access, identity of caller, location of caller etc.) for each request made to the service.
- Request information provided to the service by the caller. This will record the message data provided to the service for each request made to the service.
- Response information returned by the service to the caller. This will record the message data returned by the service to the caller for each request made to the service.

Controlling Collection

Collection will be controlled by higher level defaults for the entire Portus server instance. These may then be overridden at the service level.

Global Collection

This is set on/off via the Portus Control Centre. Selecting the server displays its properties. Click on the Governance tab. Shown below are the items which can be set :



Important: To activate/deactivate usage governance, these settings must be saved, the server stopped and restarted

Basic

Here is where Governance collection is set on/off. Selecting *Information* turns collection on. Selecting *Input Data* and/or *Output Data* adds this information to the collection. If writing to a local file then a maximum file size can be set. When this value is reached the file will be written and collection continues with a new file.

Governance Server

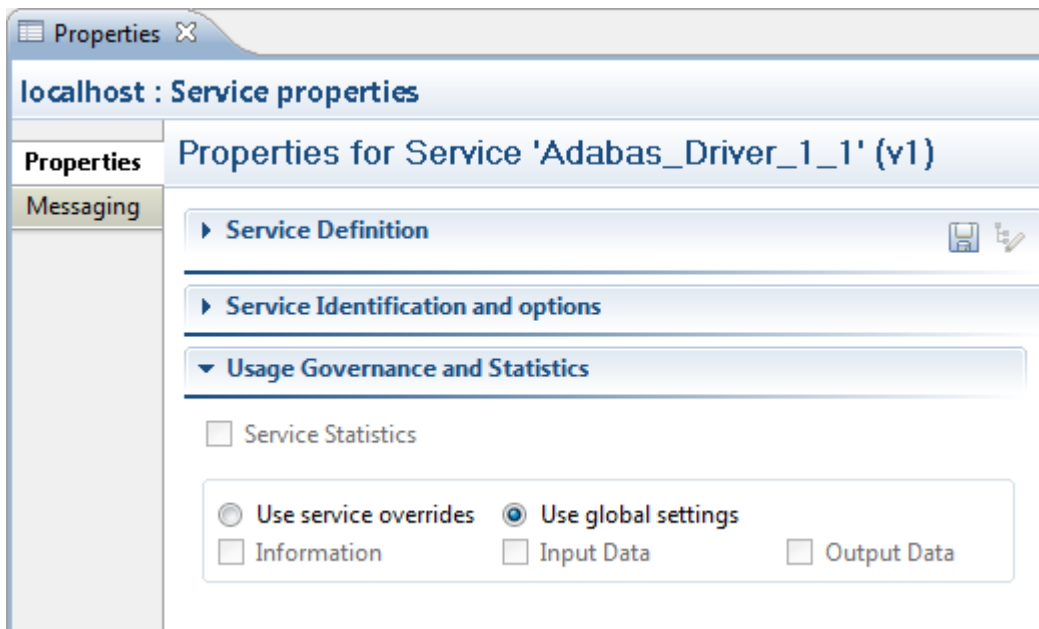
The option to send collection data to another Portus is achieved by setting the *Host* and *Port* details. Prior to this a governance web service must be created and the server up and running. See here for details on how to do this.

Messaging

A third option is to send data to an existing Messaging system. Currently this can be IBM WebSphere MQ or Software AG EntireX Broker. See bla bla bla for details on Portus Messaging.

Collection at Web Service Level

Select the web service (under Services) in the Control Centre. Here the collection can be set as required for a specific service (Use service overrides). Otherwise the value of the global setting is used (Use global settings).



The fourth option shown here, *Service Statistics*, is used to collect various metrics at the web service level. See here for details on this.

See here for tutorials on Usage Governance.

36 Lifecycle Governance

- Concept 338
- Web Service Versioning 339
- Web Service Status 339
- Practical Example 340


This section describes how the Portus web services can be governed in terms of their availability, their versions and their status. This concept is referred to as “Lifecycle Governance” – the governance of Portus web services throughout their life time.

The availability (or unavailability) of a web service is something that an organization may wish to set for a period of time. For example, if there is maintenance required on a specific database table, the web service which uses this table could be made unavailable without affecting other services. In this case, clients will receive a specific Portus message to indicating the service is unavailable.

The versioning of web services allows similar services to co-exist, transparent to existing clients, while allowing new clients to take advantage of the latest service modifications. For example, what happens when the type of a parameter needs to change from xs:string to xs:int? This change must be coordinated so that existing clients of the web service can continue unaffected, and ensuring that new clients receive the most up-to-date implementation. Portus can make this possible seamlessly.

The status of a service is a concept where the service is in the process of being migrated from different states as it evolves. For example, a service may be in a “test” or “frozen” state.

For the purposes of this example, we will use the “MySQL City” sample service. See here for more information. The process is the same for any other service defined in the SOA Gateway.

 **Important:** The Portus legacy perspective should not be used for configuring services with lifecycle governance. All further references to the Control Centre assume the Portus Administration perspective.

Concept

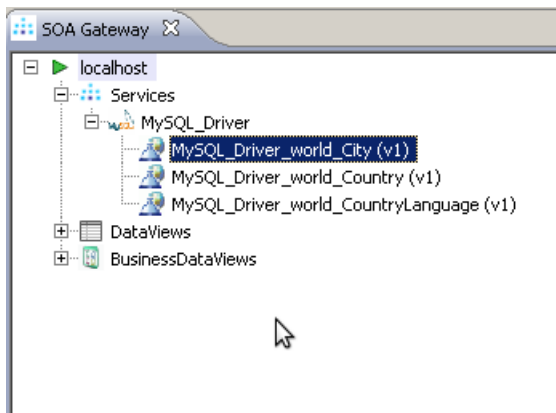
The basic concept of lifecycle governance is that it allows changes to be made to existing services without forcing existing clients to upgrade. Each client that wishes to consume a web service starts with retrieving the service WSDL. Thus, Portus will return different WSDLs based on the different version numbers of the service. The key point is that the endpoint will also have the version number as part of the URL. Thus a client may use “version 1” of a Portus web service. If the administrator decides that the web service needs to be changed, a new WSDL with a new endpoint is made available. New clients can use version 2, while existing clients can continue to use version 1 seamlessly.

From a clients perspective, this process is seamless, there is no requirement for them to have any understanding of web service versions. New clients just start using the WSDL, existing clients work with their services as before.

From an administrators perspective, Portus will control the allocation and incrementation of service version numbers.

Web Service Versioning

Every web service that is created in Portus has an associated version. By default, this version number starts a “1”. The control centre will display the version number of each service after its service name.



When retrieving the WSDL for a service, Portus will always return the most recent version of the WSDL. The most recent version is determined by the incrementally highest version number. For example, version 3 is “more recent” than version 1. It is still possible to retrieve a WSDL other than the most recent, but you must pass an extra “__version=N” argument on the URL. For example:

```
http://host:port/MySQL_Driver_world_City?WSDL&__version=1
```



Important: You will notice that the WSDL endpoint now has a unique version number appended on the end of the URL in the <service> section.

Web Service Status

There are 4 possible states a Portus Web Service can be in.

1. Test

In this state, the service may be modified as often as is required to bring the service to a point where it is to be made available to others. The service can be edited and/or modified, and the version number will stay the same.

2. Frozen

When the service is to be deployed or made available to others for use, it must be frozen. At this point, the version assigned to the service will represent this version of the service forever. There can be multiple frozen versions of any service, but they must have different version numbers.

3. Deprecated

When a service has been marked as deprecated, clients will still be able to call the service, but they will receive a warning message in the response, and the administrator will see a warning message in the system log. In this scenario, it is expected that administrators should recommend a service upgrade to their clients.

4. Historical

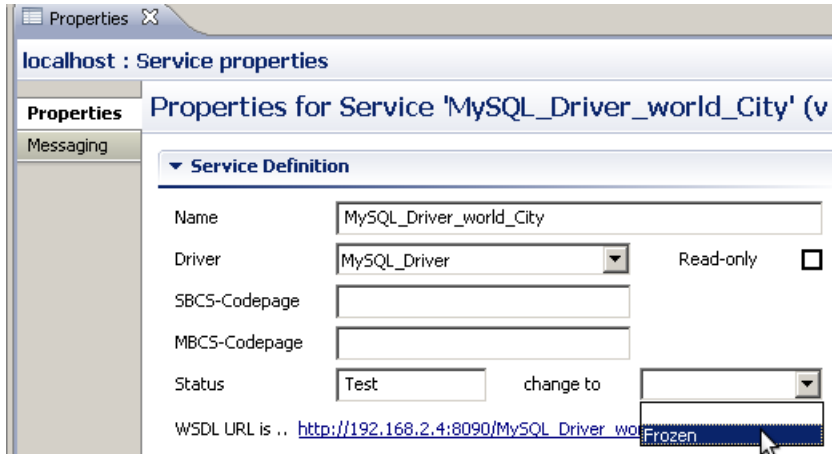
This status will be assigned to a service which is no longer active and has essentially been deleted. It is maintained in the configuration for historical purposes. If this service is invoked, the client will be returned a soap fault and a record logged that an attempt was made to use a historical version of the service.

Practical Example

In this example, we will use the “adabas_EmployeesMini” service that is created when adding a new Adabas driver. The principal is the same for all other Portus web services. Initially, the version number of the web service is “1”. This service can be modified as often as required, and neither the version number nor the service state will change.

Freezing a service

When the administrator is happy that the service is ready to be put into general use, they can mark the service as “Frozen” using the drop down box in the web service properties.



Once the service is frozen, it is not possible to change the DataView of this service. It is still possible however to modify the properties of this web service. For example, it is still possible to change the database id/name of the frozen web service.

Modifying a service

If the administrator now wishes to make changes to the DataView of the frozen service, it is still possible to right-click the service, and choose “Edit DataView”. This will bring up the data view editor as usual. When this view is saved, the control centre will create a new version of the modified service, by incrementing the version number.

Therefore, no modifications have been made to the frozen service (version 1), but a new service has been created (version 2) which contains these modifications. The services names will be the same (adabas_EmployeesMini). The version 2 service will be created with its status set to Test.

Accessing WSDL and calling the service

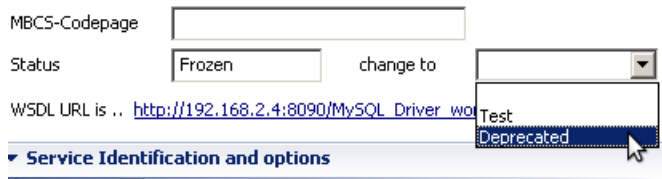
When a client requests the WSDL for this service, by default they will receive the version 2 WSDL. Thus new clients will get the most up to date version of the service. It is assumed that existing clients will only import the WSDL once (when version 1 was the most recent). To access the version 1 WSDL, provide the `__version=1` argument.

```
http://host:port/MySQL_Driver_world_City?WSDL&__version=1
```

Any client that uses the existing version 1 service will now receive a informational message in the SOAP response Header indicating that a newer version of this service exists. Clients could be configured to recognised this information, and take appropriate action, such as synchronizing with the SOA Gateway to get the latest version of this service.

Deprecating a service

When the administrator wishes to move clients off an old service, they should mark it as deprecated.



Clients will not be able to retrieve the WSDL for this service version, so new clients cannot use this service. Existing clients will receive a warning message in the SOAP headers when they try to use this service. A warning message will also be written to the server log.

```
Fri May 13 21:12:46.00558327, pid: 00003267, tid: -1237358848. WARNING: in file uriCache.cpp, at line 1358. indicating The service MySQL_Driver_world_City version 1 has been deprecated.
```

Historical Service

When the lifecycle of a service has come to an end, it can be marked as Historical. This means the WSDL is no longer available, and any client trying to call this service version will receive an error message.

```
Fri May 13 21:12:46.00558327, pid: 00003267, tid: -1237358848. ERROR: in file uriCache.cpp, at line 1358. uriCache::checkForCacheEntry() returned -33044, indicating The service MySQL_Driver_world_City version 1 is no longer available
```


37 HTTP Proxy Driver

- How to use 344

The HTTP Proxy driver consumes external non-Portus web services via their WSDL, and then allows them to be called via Portus.

How to use

Right-click the "Proxy Pass-thru Driver" entry and select Discover

Enter the URL of the WSDL. Alternatively, click the Browse button, and choose the file from disk

The wizard will now create a Portus "Proxy service" for the selected WSDL. The WSDL for this service will be equivalent to the initial WSDL, with the exception of the service endpoint.

When Portus receives a request from a client to this Proxy Service, it will relay the request to the endpoint in the original WSDL, and return the response from this endpoint.

38 Monitor View

The Control Centre Monitor View allows the server log files to be viewed from within the Control Center.

The Monitor log is not available on mainframe server installations

To start the Monitor View, select Window -> Show View -> Other -> Portus Monitoring

Select your server from the drop-down menu. The Control Centre will connect to this server and download the server logs (Apache Error and Access logs).

 **Important:** If the logs are very large, this may take some time

The **Refresh** will download the latest log changes

