# Adabas SOA Gateway

## Administration

Version 2.6.1

November 2016

ADABAS & NATURAL

This document applies to Adabas SOA Gateway Version 2.6.1.

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

Copyright © 2006-2016 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at http://softwareag.com/licenses.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at http://softwareag.com/licenses/ and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at http://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

**Document ID: ASG-AASGADMIN-261-20161123**

## Table of Contents


Security ..... vii
1 SOA Gateway Security ..... 1
2 SOA Gateway Control Centre ..... 3
3 Using SOA Gateway with ... ..... 5
4 SOA Gateway Server Admin ..... 7
    Stopping / Starting the SOA Gateway Server - Windows ..... 8
    Stopping / Starting the SOA Gateway Server - z/OS ..... 8
    Stopping the SOA Gateway Server - z/VSE ..... 9
    Stopping / Starting SOA Gateway Server - Linux, AIX and Solaris ..... 9
    Updating an expired license ..... 9
5 SOA Gateway 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 SOA Gateway Governance ..... 27
7 Web Services in SOA Gateway ..... 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 ..... 34
    Example 2 ..... 54
    Example 3 ..... 35
    Example 4 ..... 36
    Example 5 ..... 36
9 Using Username and/or Password ..... 39
    Authenticate with backend database ..... 40
    Apache's htpasswd ..... 40
    Authenticate with RACF ..... 41
    Enabling Web Service Authorization ..... 42
10 SSL Certificates ..... 47
    Setup ..... 48
    Introduction ..... 48
    Step 1: Set up your own CA ..... 48

Step 2: SOA Gateway Server key and certificate ..... 49
Server Configuration ..... 49
11 The SOA Gateway (legacy) Servers View ..... 57
Working with the Servers View ..... 58
Refresh the SOA Gateway configuration ..... 60
Save current Configuration to server file ..... 60
Import the SOA Gateway Configuration from server ..... 61
Export the SOA Gateway Server configuration ..... 62
Export SOA Gateway WebService definitions ..... 63
Remove a SOA Gateway Server entry ..... 63
Create new Services ..... 64
Modify the SOA Gateway Server definition entry ..... 65
Modify Global -> Security Options ..... 95
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 SOA Gateway ..... 74
IPv6 Considerations ..... 76
Reducing memory footprint using streaming ..... 77
13 Working with the SOA Gateway View ..... 79
Operations at the Server level ..... 80
Operations at the Driver level ..... 85
Operations at the Service level ..... 86
Operations at the DataView / BusinessDataView levels ..... 86
14 Drag-and-drop in the SOA Gateway View ..... 87
Driver definition as a drag source ..... 88
Service as a drag source ..... 88
DataView and BusinessDataView as a drag source ..... 90
15 SOA Gateway BusinessDataViews ..... 91
Overview / Concepts ..... 92
Using the BusinessDataView WebService ..... 97
Working with BusinessDataView Objects ..... 221
Summary ..... 100
Creating BusinessDataViews ..... 101
BusinessDataView Editor hints ..... 126
16 The SOA Gateway Control Center Action Log ..... 135
Working with the Action Log ..... 136
17 Setting Preferences for the SOA Gateway Control Centre ..... 139
Working with the Preferences ..... 140
18 The SOA Gateway (legacy) Configuration View ..... 143
Working with the Configuration View ..... 144
Filter displayed Services ..... 147

Edit a Service Definition ... 148
Add a Service ... 148
Duplicate a Service Definition ... 150
Duplicate a Service Definition to other server ... 150
Remove a Service from the server configuration ... 151
Import a DataView ... 152
Export a DataView ... 153
Generate Java Wrapper for a Service ... 154
Register a Service with an UDDI Business Registry (UBR) ... 154
Create a Resource ... 155
Delete a Resource ... 156
Replicate a Service ... 157
Refresh a Service ... 159
Import Service Definition files ... 160
19 Service Configuration ... 161
Service Properties ... 163
Service context menu ... 164
Lifecycle Governance ... 167
20 Creating a SYSOBJH extract file ... 171
Generating a SYSOBJH extract (Windows) ... 172
Generating a SYSOBJH extract (Mainframe) ... 183
21 SOA Gateway DataViews ... 193
Create a DataView from scratch ... 194
Opening a SOA Gateway DataView for editing ... 195
Editing a SOA Gateway DataView ... 196
Editing "special" fields ... 201
Enhanced Type Conversion ... 204
Exporting a DataView to a SOA Gateway server ... 205
Importing an existing SOA Gateway DataView ... 207
Creating a XML Schema for a DataView ... 207
22 Using SOA Gateway with Adabas ... 211
Creating a SOA Gateway DataView from an Adabas FDT ... 213
Create DataView(s) from a SYSOBJH unload file ... 213
23 SOA Gateway Service Creation ... 215
Initiating Service creation ... 216
Creating Service(s) from Adabas ... 218
Assigning a different DataView to existing service ... 219
Creating Services from a SYSOBJH extract file ... 221
Creating Service(s) from 3rd party web services ... 224
24 Using SOA Gateway to access Web Services ... 229
Introduction ... 230
Accessing Web Services ... 230
25 Message Oriented Middleware Support in SOA Gateway ... 237
Introduction ... 238
26 Usage Governance ... 239

Introduction ..... 240
Governance Data ..... 240
Service collection ..... 241
Controlling Collection ..... 242
27 Lifecycle Governance ..... 245
Concept ..... 246
Web Service Versioning ..... 247
Web Service Status ..... 248
Practical Example ..... 248
28 HTTP Proxy Driver ..... 251
How to use ..... 252
29 Monitor View ..... 253

# Security

*SOA Gateway Security*

*SOA Gateway Control Centre*

*Using SOA Gateway with ...*

*Usage Governance*

*SOA Gateway Statistics*

# 1 SOA Gateway Security

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

SOA Gateway makes use of the tried and trusted security support provided by Apache. This section will provide examples on how to secure SOA Gateway using Apache security. The security support should be added to the SOA Gateway 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.<br><br>E.g. C:\Program Files (x86)\Ostia\Portus 4.2.1\httpd_softwareag_adabas_asg.conf |
| z/OS | <ASG_DS>.CONF(HTCONF)<br><br>Note: Just editing and saving this file on z/OS is not sufficient. This file must be copied into the SOA Gateway file system (the <ASG_DS>.PFS dataset) and the SOA Gateway server restarted for the changes to take effect. To copy these changes into the SOA Gateway file system submit the job <ASG_DS>.JCL(SOAGINS3)<br><br><ASG_DS> is the HLQ where SOA Gateway has been installed. |

**Web Services in SOA Gateway**

**Securing your server to IP address or hostname**

**Using Username and/or Password**

**SSL Certificates**

# 2 SOA Gateway Control Centre

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

Please read the Introduction to the SOA Gateway Control Centre before continuing.

The following sections will explain the individual SOA Gateway Eclipse perspectives and views available to the administator, and how to work with each.

**The SOA Gateway Administration Perspective**

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

**The SOA Gateway "Legacy" Perspective**

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

Other aspects:

- **The SOA Gateway "DataView"**
- **Setting Preferences**

# 3 Using SOA Gateway with ...

| | |
|---|---|
| **Adabas** | Ultra-high performance, high-availibility transactional database |
| **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 |

# 4 SOA Gateway Server Admin


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

Most tasks will be handled by the **SOA Gateway Control Centre**, which presents a cordinated grapical view of the server. Tasks that are not possible from the Control Centre are documented here.

## Stopping / Starting the SOA Gateway Server - Windows

The SOA Gateway 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 -> SOA Gateway vN -> Apache -> Monitor Apache Servers**

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

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

The green arrow above indicates that SOA Gateway is up and running.

If SOA Gateway is not running the icon looks like this

## Stopping / Starting the SOA Gateway Server - z/OS

SOA Gateway can be stopped by issusing /p jobname at the terminal.

To start SOA Gateway, re-submit the ASGSTRT job.

## Stopping the SOA Gateway Server - z/VSE

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

## Stopping / Starting SOA Gateway Server - Linux, AIX and Solaris

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

## Updating an expired license

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

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

# 5 SOA Gateway 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 SOA Gateway.

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.

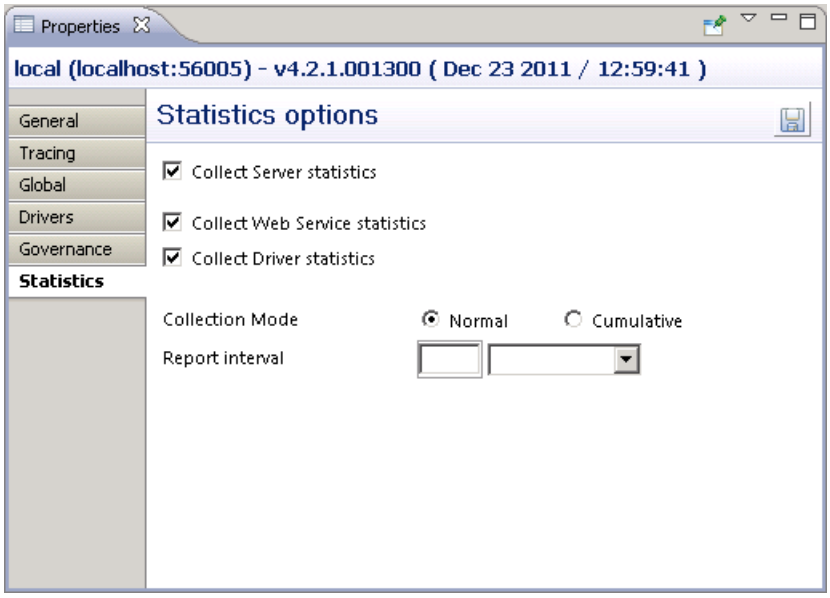
- errors_occurred

  This is the number of times the operation has failed.

## Server Setup

Setup is carried out using the SOA Gateway Control Centre.

Click on the server to be monitored in the list of SOA Gateway 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_millisec.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.

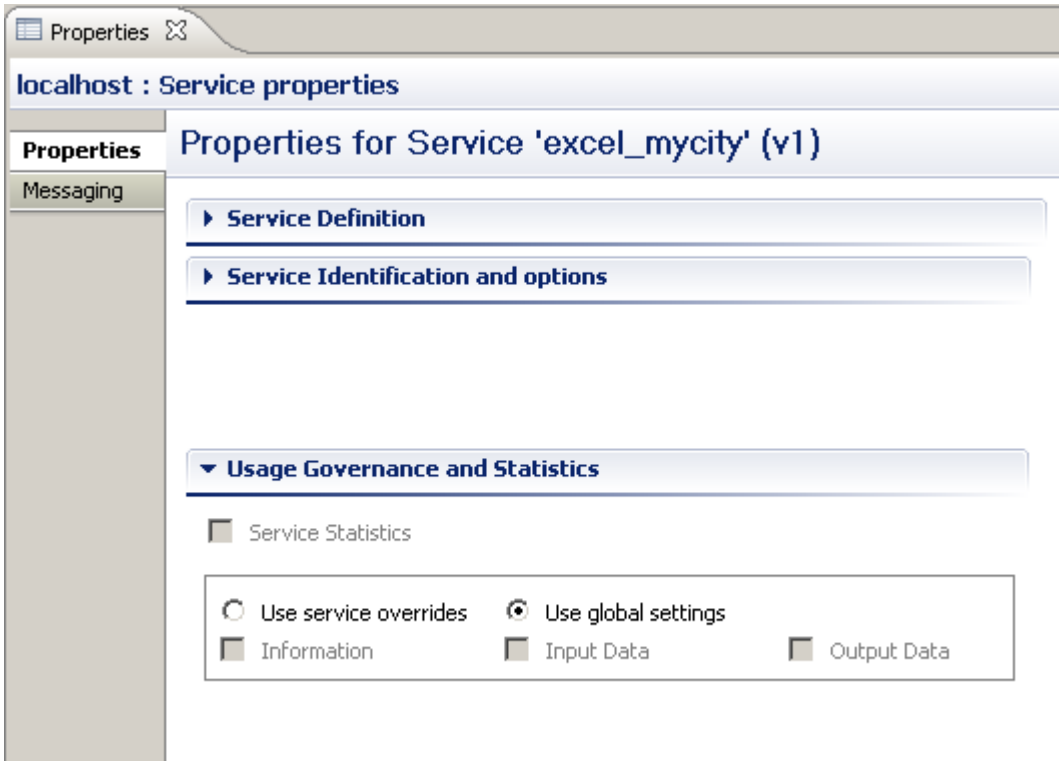
### Report Interval

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

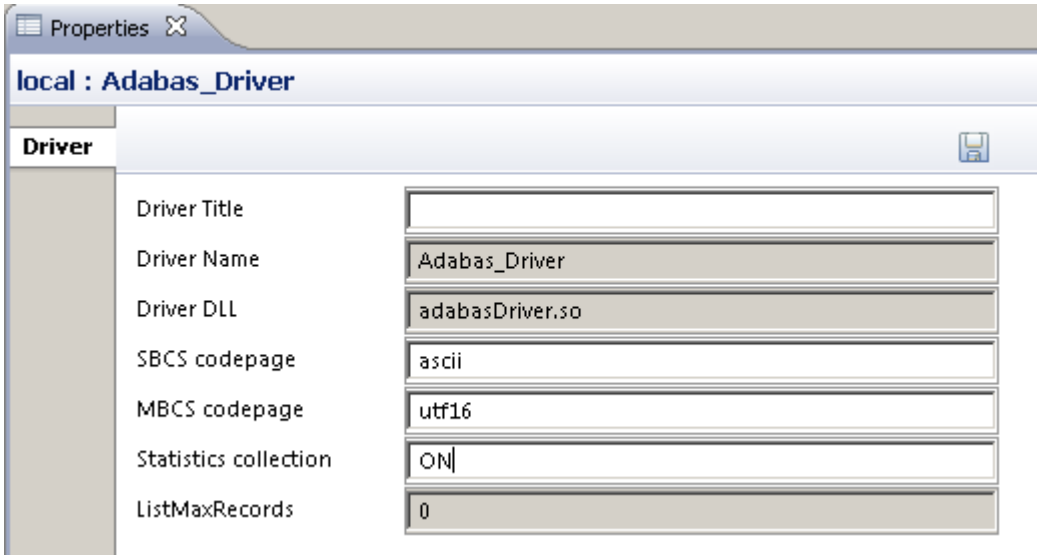
Click on the service to be monitored. In the Properties View set *Service Statistics* on/off. Note that the global setting overrides this setting and it will be greyed out if set on at the global level as shown below.

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



## Driver Setup

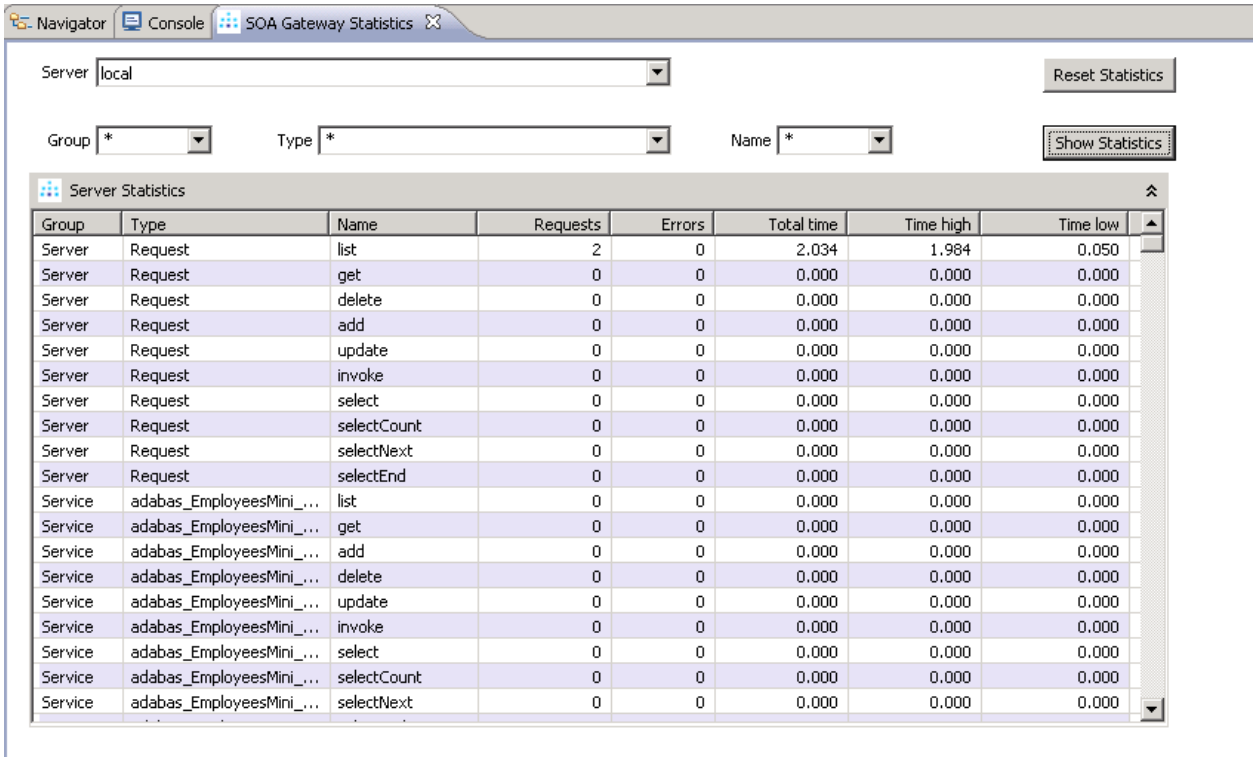
Click on the driver to be monitored. In the Properties View set Statistics collection to ON/OFF.




## Monitoring

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

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




| 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

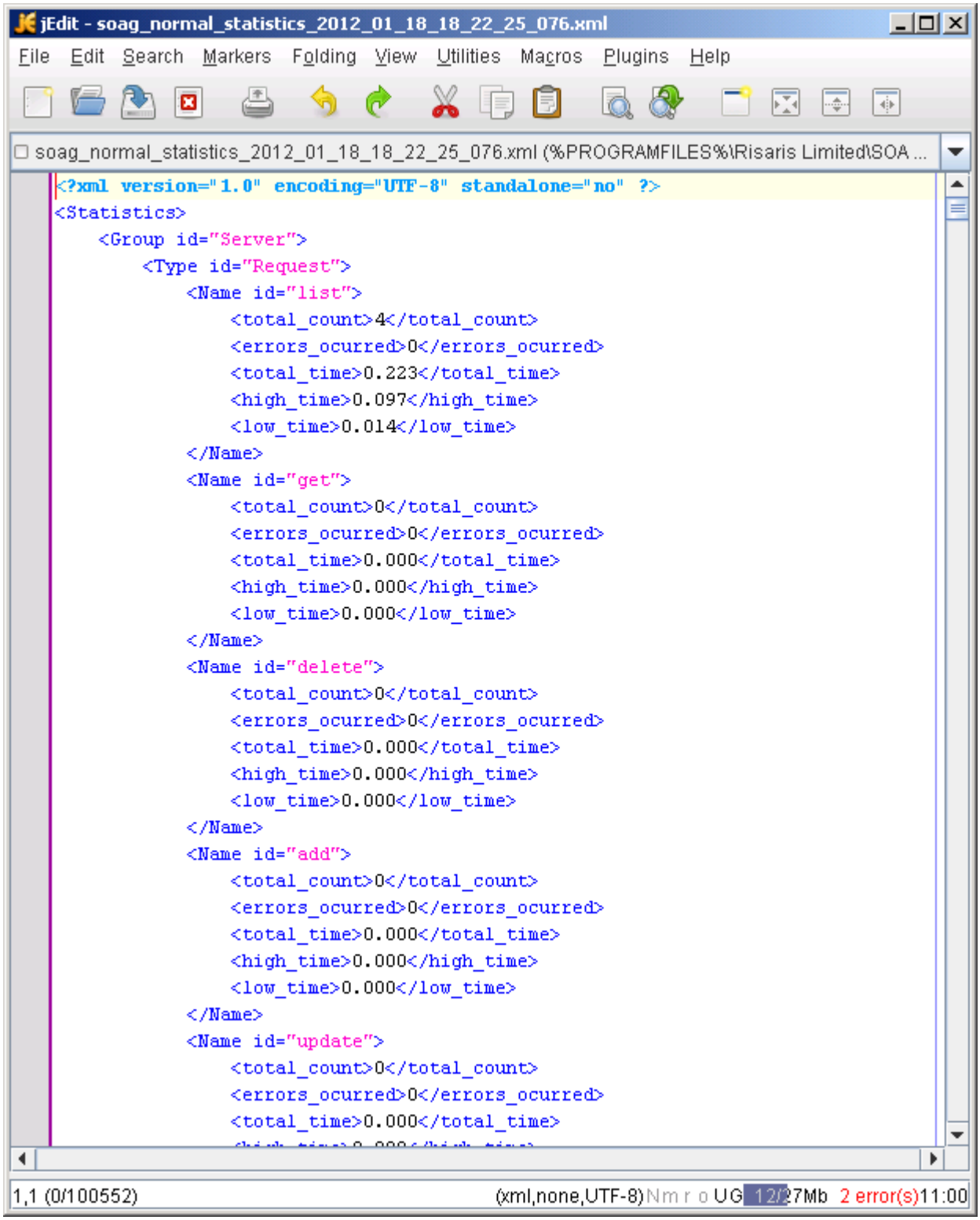
Note that the wildcarding is asterisk only i.e. Adabas* is not acceptable.

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:

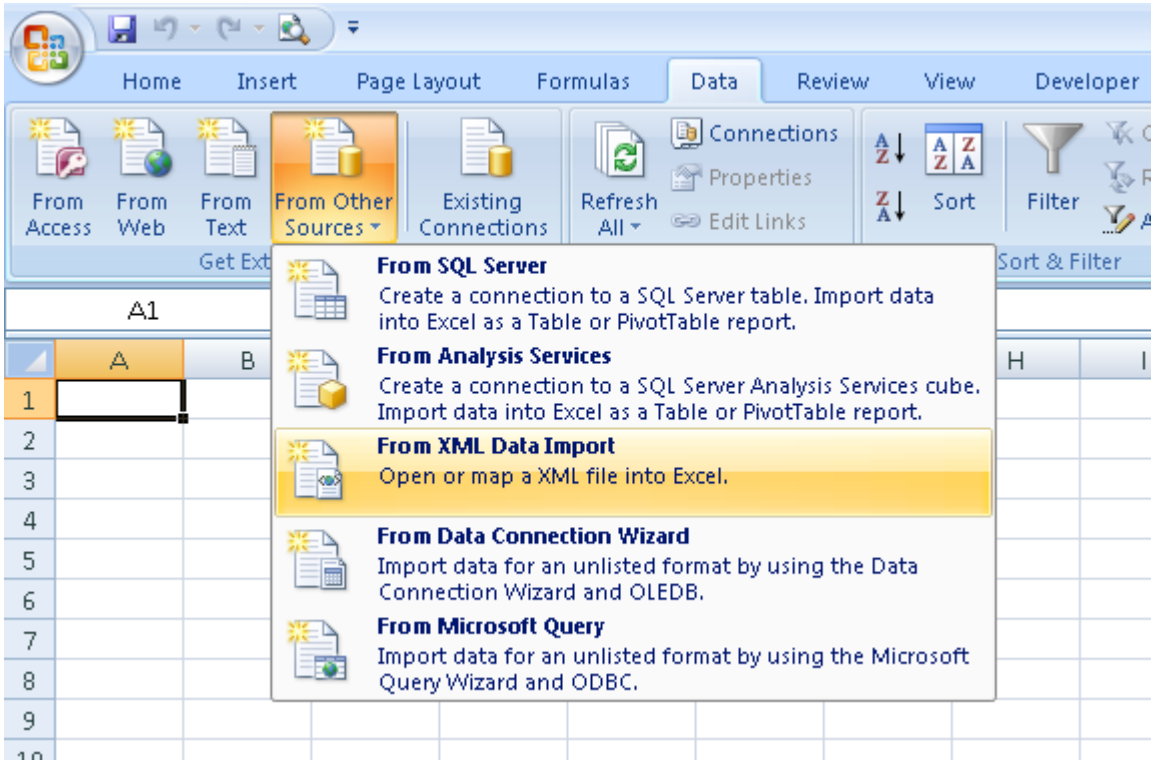



The statistic reports are written to the configuration folder where SOA Gateway is installed.

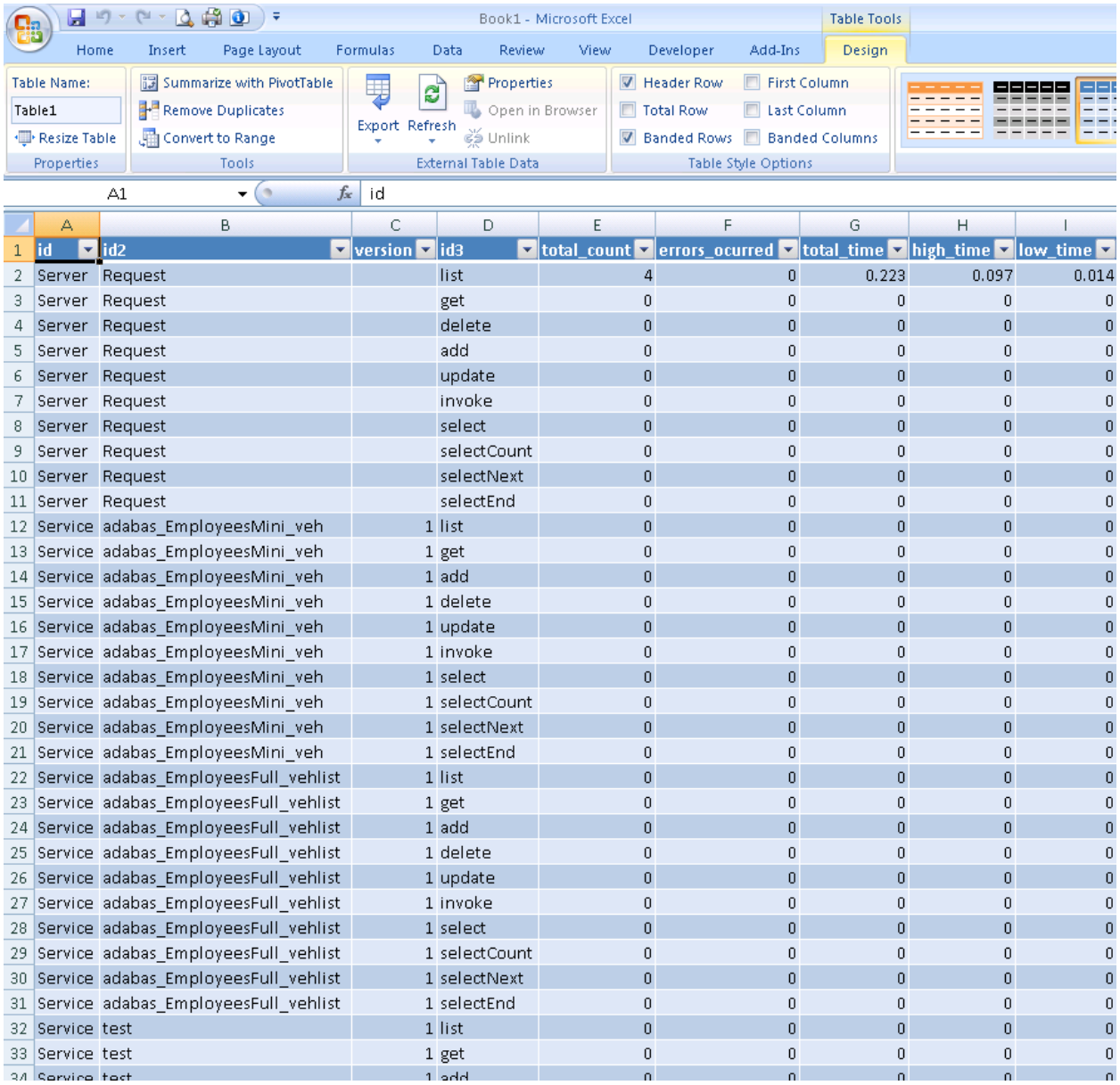
# XML

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 SOA Gateway configuration directory :




resulting in a very readable format::



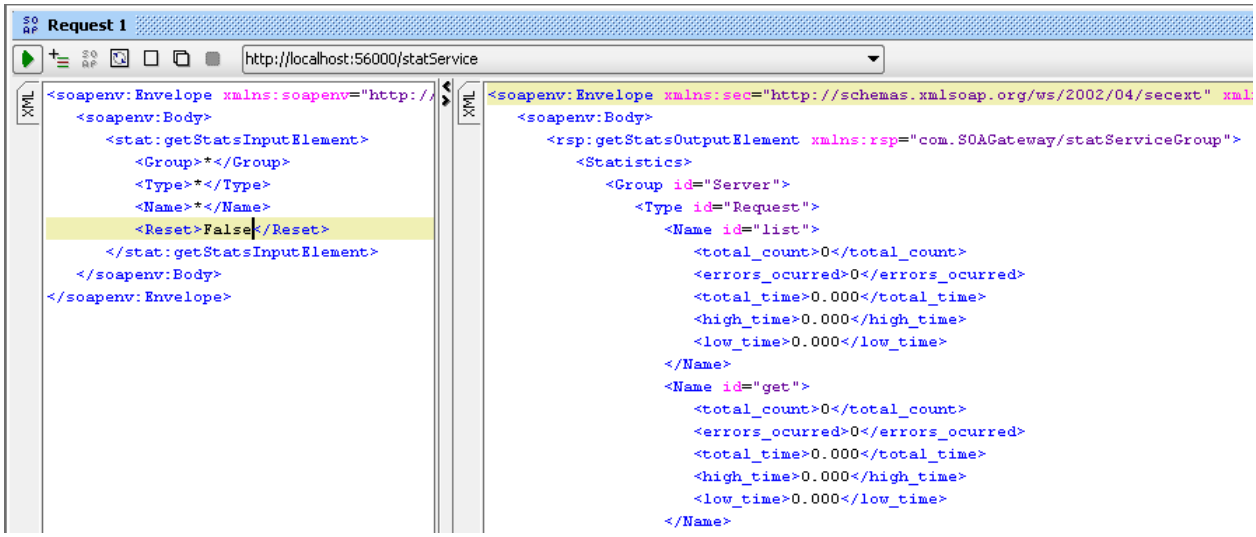

| id | id2 | version | id3 | total_count | errors_ocurred | total_time | high_time | low_time |
|---|---|---|---|---|---|---|---|---|
| Server | Request | | list | 4 | 0 | 0.223 | 0.097 | 0.014 |
| Server | Request | | get | 0 | 0 | 0 | 0 | 0 |
| Server | Request | | delete | 0 | 0 | 0 | 0 | 0 |
| Server | Request | | add | 0 | 0 | 0 | 0 | 0 |
| Server | Request | | update | 0 | 0 | 0 | 0 | 0 |
| Server | Request | | invoke | 0 | 0 | 0 | 0 | 0 |
| Server | Request | | select | 0 | 0 | 0 | 0 | 0 |
| Server | Request | | selectCount | 0 | 0 | 0 | 0 | 0 |
| Server | Request | | selectNext | 0 | 0 | 0 | 0 | 0 |
| Server | Request | | selectEnd | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | list | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | get | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | add | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | delete | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | update | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | invoke | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | select | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | selectCount | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | selectNext | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesMini_veh | 1 | selectEnd | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | list | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | get | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | add | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | delete | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | update | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | invoke | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | select | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | selectCount | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | selectNext | 0 | 0 | 0 | 0 | 0 |
| Service | adabas_EmployeesFull_vehlist | 1 | selectEnd | 0 | 0 | 0 | 0 | 0 |
| Service | test | 1 | list | 0 | 0 | 0 | 0 | 0 |
| Service | test | 1 | get | 0 | 0 | 0 | 0 | 0 |
| 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

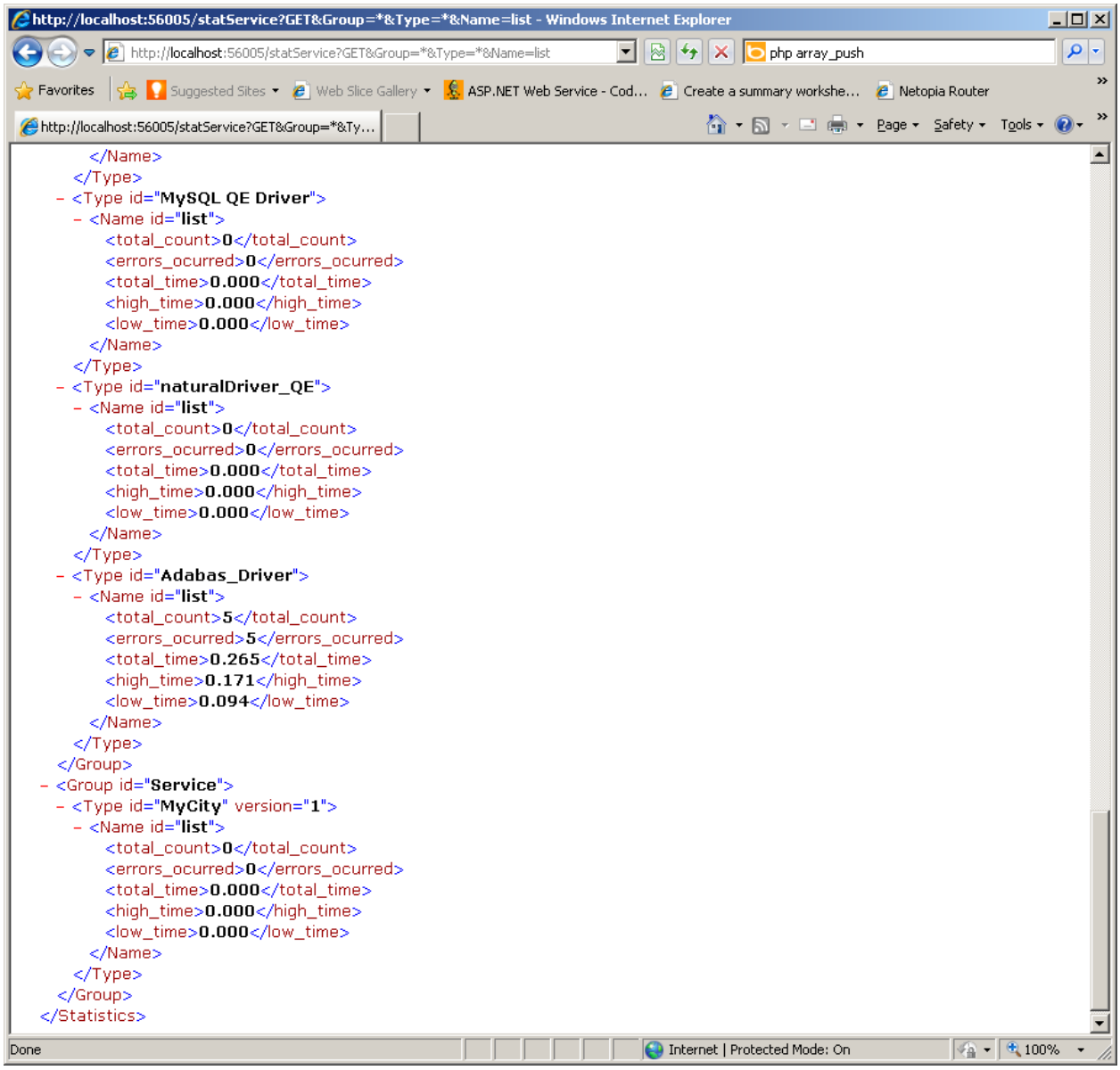
Statistics can be retrieved via a SOAP request using the statService web service. Using a WSDL consumer such as soapUI, as shown below, a request can be made using Group, Type and Name. As with the Control Centre, the '*' construct enables the entire set of statistics to be returned or a subset. The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




## 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



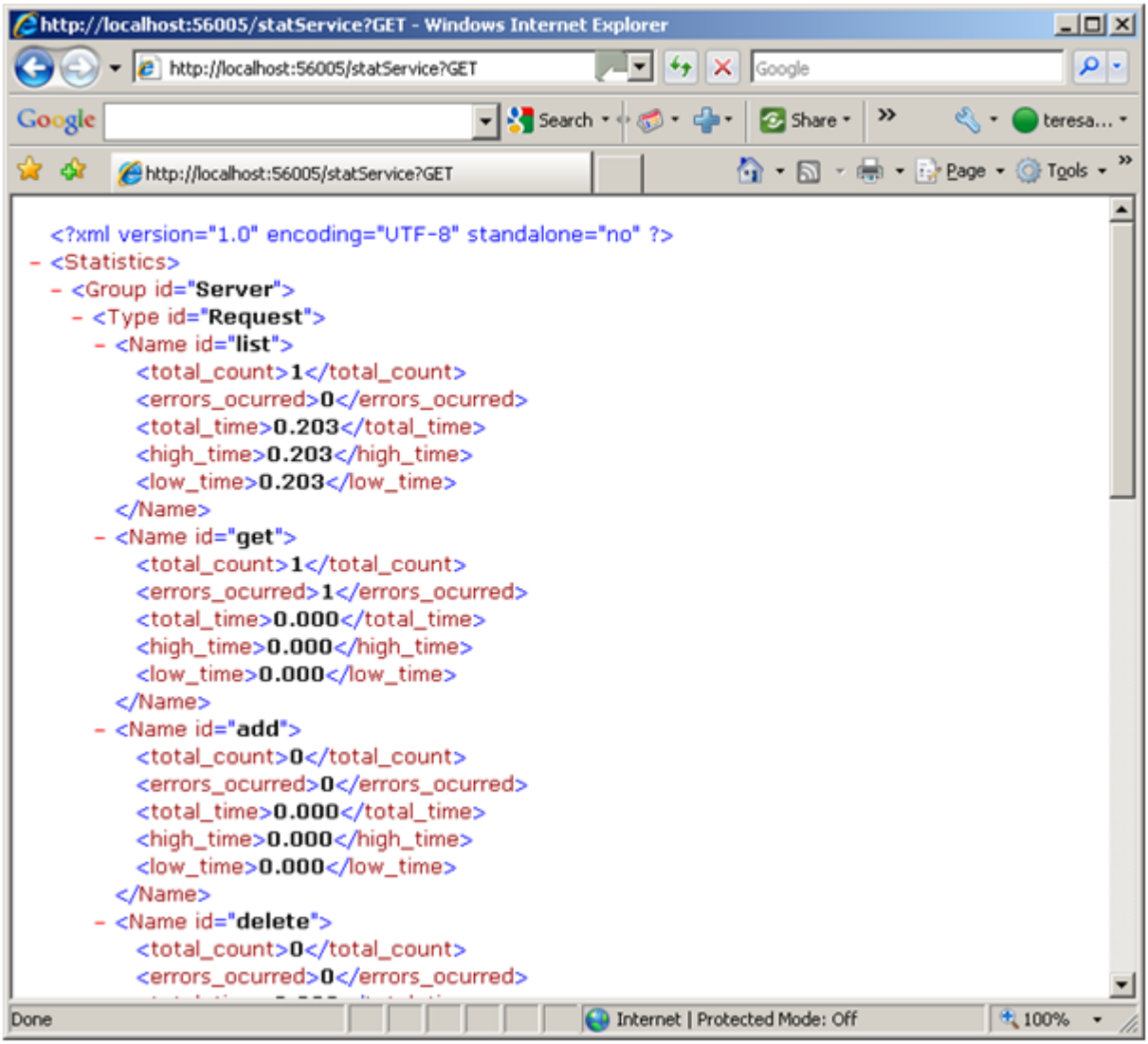

## Example

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 SOA Gateway Server.

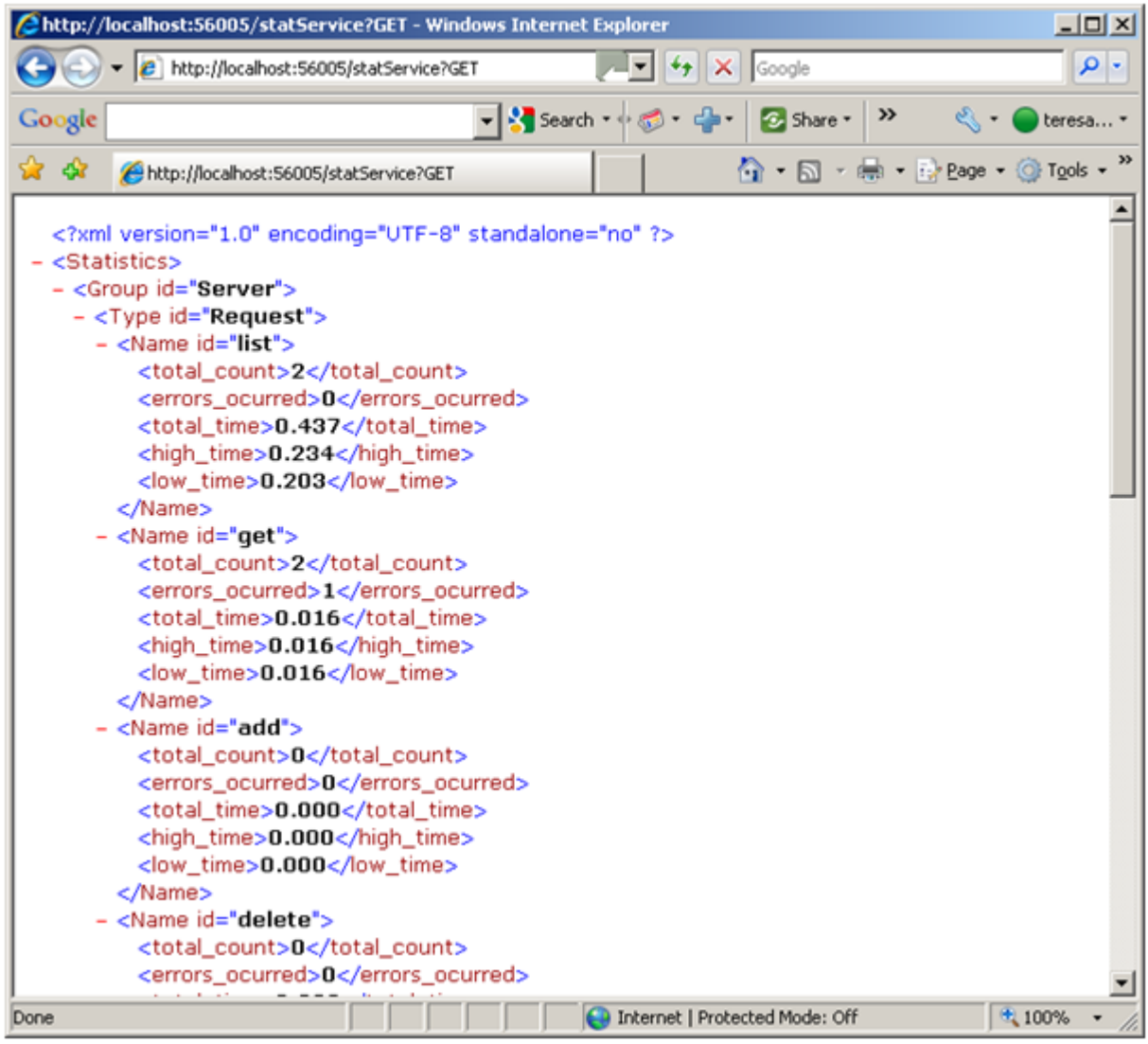



Let us issue the LIST and GET again, this time correcting the GET syntax and changing the LIST to retrieve ids starting with 20:

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

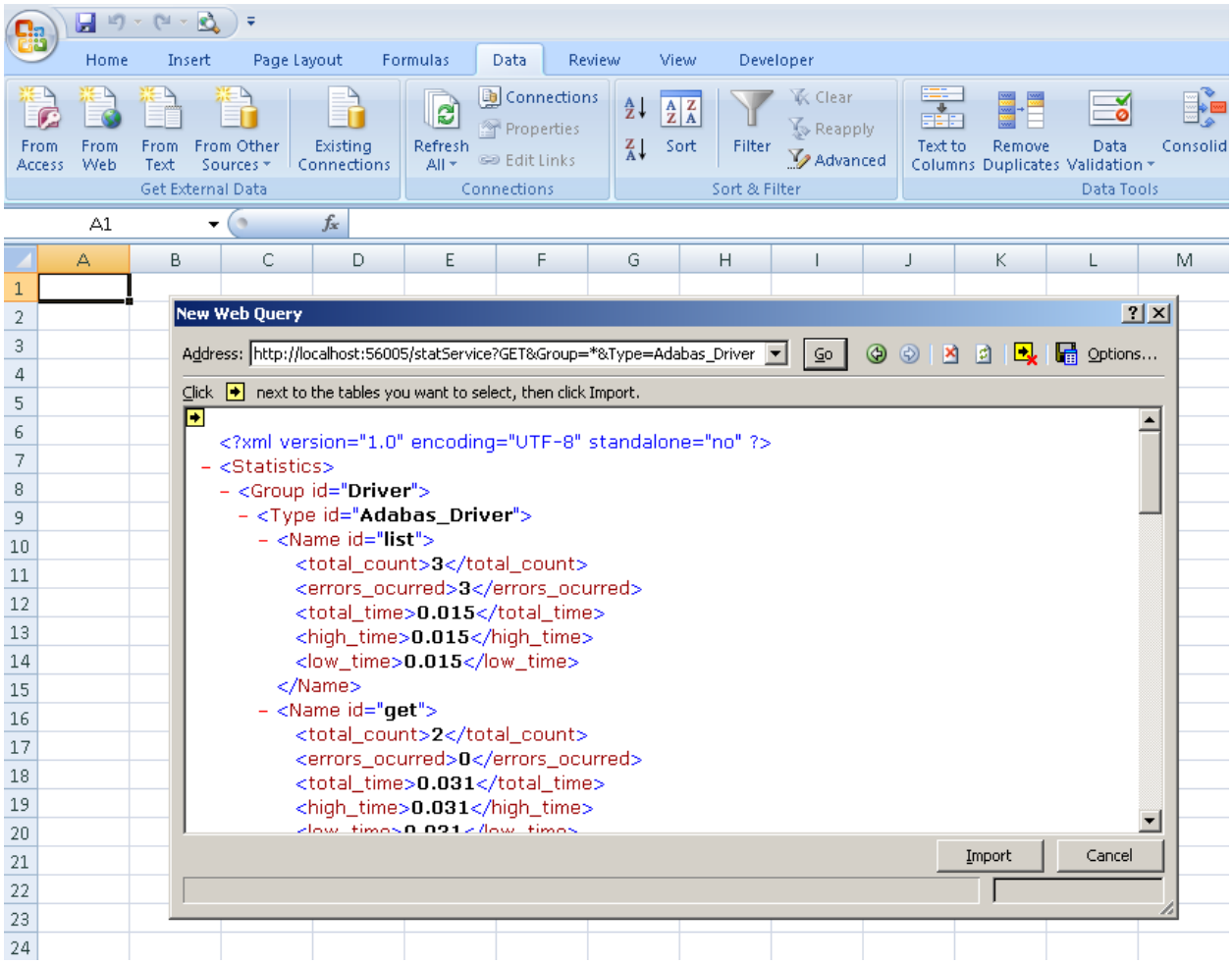
http://localhost:56005/Adabas_Driver_1_9?LIST&&AA=20*

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.

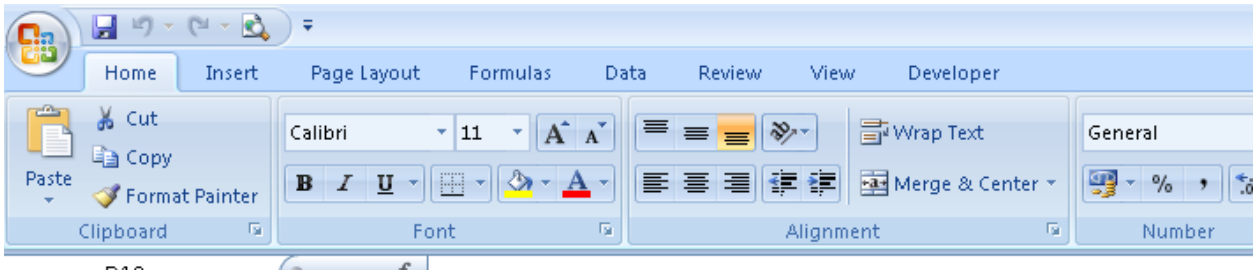



## Service Import

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:

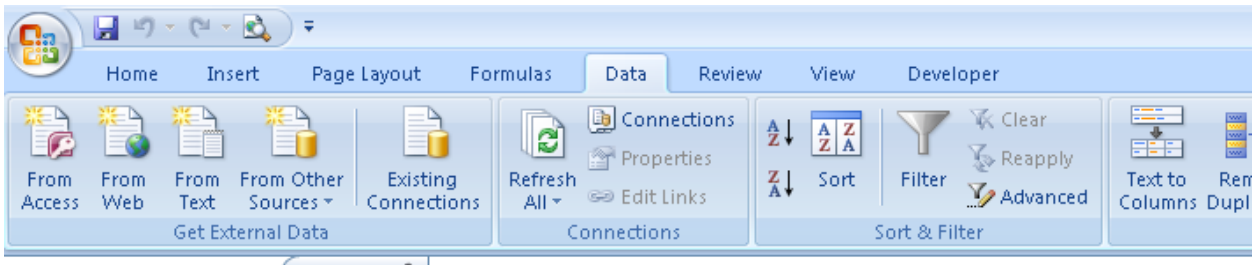



Select Import and accept the default location of A1.




| | 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 SOA Gateway Governance

There are 2 ways to apply goverance to SOA Gateway

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

# 7 Web Services in SOA Gateway


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

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

## Overview

SOA Gateway 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 SOA Gateway 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 SOA Gateway. 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 SOA Gateway 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 "SOA Gateway Create Resource" or "SOA Gateway 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 ..... 34
- Example 2 ..... 54
- Example 3 ..... 35
- Example 4 ..... 36
- Example 5 ..... 36

This section will outline how to secure the SOA Gateway 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 SOA Gateway 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 SOA Gateway server is running, not where the Eclipse IDE is running (if different).

1. Edit the SOA Gateway 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 SOA Gateway server.

1. Edit the SOA Gateway 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 SOA Gateway server. All others machines would be rejected access.

## Example 3

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

1. Edit the SOA Gateway 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 SOA Gateway 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 SOA Gateway. 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 SOA Gateway 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 ..... 40
- Apache's htpasswd ..... 40
- Authenticate with RACF ..... 41
- Enabling Web Service Authorization ..... 42

## Authenticate with backend database

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

This uses the SOA Gateway 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 SOA Gateway 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 SOA Gateway to check directly with using the SAF interface, the SOA Gateway address space must be APF authorized.
2. 2. If you have ADABAS installed and are using the ADABAS SAF Server, SOA Gateway 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 SOA Gateway 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 that 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 librart>,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.

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

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

Ìf 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway's security level should be set to "Password". This can be set either by the global setting in the control centre, or via the SOA Gateway Apache SoaGatewaySecurityLevel directive.

**Important:** On z/OS, the SOA Gateway 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, SOA Gateway will attempt to authenticate this user with RACF, providing the username "ASG" and the password "BOSTON1".

# 10 SSL Certificates


- Setup ..... 48
- Introduction ..... 48
- Step 1: Set up your own CA ..... 48
- Step 2: SOA Gateway Server key and certificate ..... 49
- Server Configuration ..... 49

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

## Setup

To enable the use of SSL Certificates for your SOA Gateway, **openssl** must be installed.

SSL support is not automatically built into SOA Gateway due to export restrictions in certain countries. Please contact your SOA Gateway 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 SOA Gateway 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: SOA Gateway Server key and certificate

This step will create a key and certificate for the SOA Gateway 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 out 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 SOA Gateway server, we need to enable SSL support. Choose next section depending on your system.

### SSL Configuration

The SSL configuration file is should be located in your SOA Gateway installation directory. Your SOA Gateway 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-crl    .crl

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 hostname and port are location where your SOA Gateway 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 SOA Gateway, 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 SOA Gateway. 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 "SOA Gateway
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 SOA Gateway 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 SOA Gateway (legacy) Servers View


- Working with the Servers View ..... 58
- Refresh the SOA Gateway configuration ..... 60
- Save current Configuration to server file ..... 60
- Import the SOA Gateway Configuration from server ..... 61
- Export the SOA Gateway Server configuration ..... 62
- Export SOA Gateway WebService definitions ..... 63
- Remove a SOA Gateway Server entry ..... 63
- Create new Services ..... 64
- Modify the SOA Gateway Server definition entry ..... 65
- Modify Global -> Security Options ..... 95
- 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.

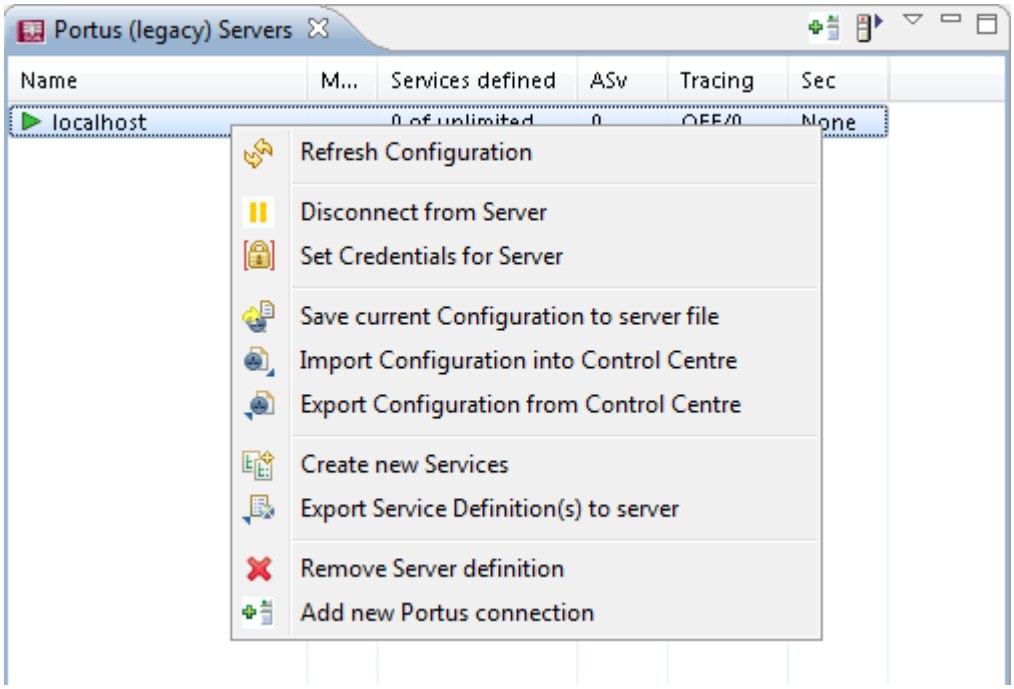
This view is your top level of configuration of one or many SOA Gateway servers.

**Note:** The Servers view is part of the SOA Gateway "legacy" perspective. This perspective is superseded by the new "SOA Gateway 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 SOA Gateway 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 SOA Gateway servers.<br><br>**Important:** To use HTTP credentials you must have a <Location /configurationService> directive enabled. |
| **Save current Configuration to server file** | Changes applied to a SOA Gateway 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 SOA Gateway server. The server must be restarted for the exported configuration to be activated. |

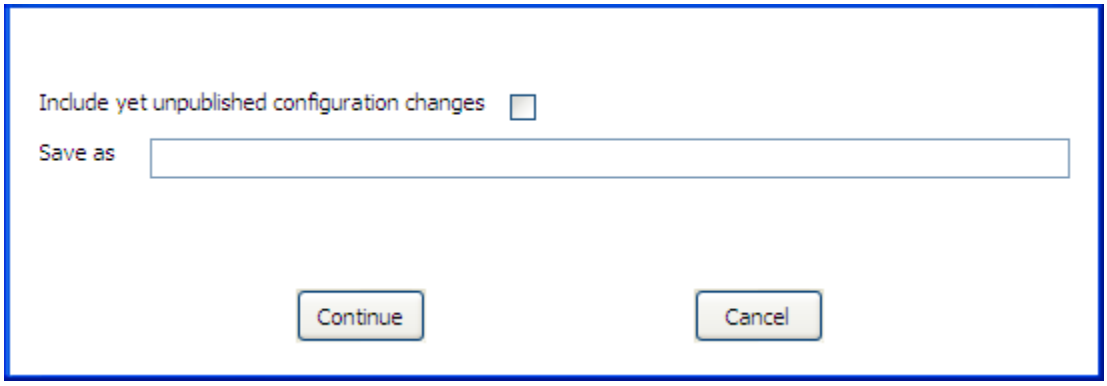
| Function | Description |
|---|---|
| **Create new Services** | Create SOA Gateway Resource related definition elements - DataViews and XML Schemata (XSDs) from native resource definition data, e.g. an Adabas FDT, SQL Tables, etc |
| **Export Service Definition(s) to server** | Export SOA Gateway Resource related definition elements - DataViews, XML Schemata (XSDs) and Stylesheets (XSLs). |
| **Remove Server definition** | Deletes the selected entry from the Servers View. |
| Add new SOA Gateway connection | Define a new SOA Gateway server to the Control Centre |

## Refresh the SOA Gateway configuration

When a SOA Gateway 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 SOA Gateway 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 SOA Gateway 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:




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 SOA Gateway

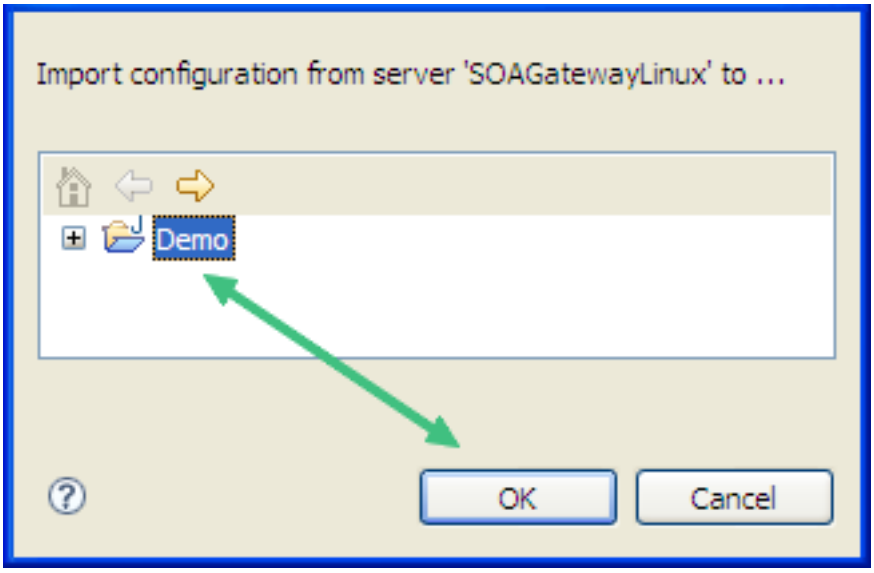
Server environment variables have to be adjusted to point to the new file. Click "Continue" to save the configuration, the successful operation will be confirmed by a message in the status line.

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

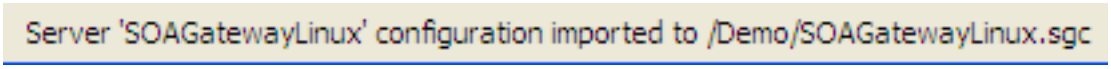
## Import the SOA Gateway Configuration from server

The current "live" SOA Gateway 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 SOA Gateway server etc.

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



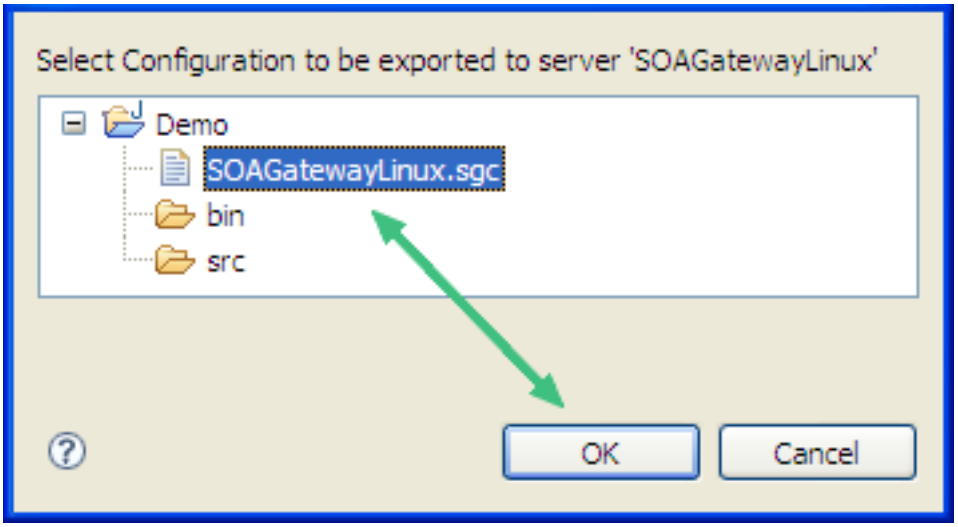

3. Successful import will be confirmed with a message in the status line as well as the Log View



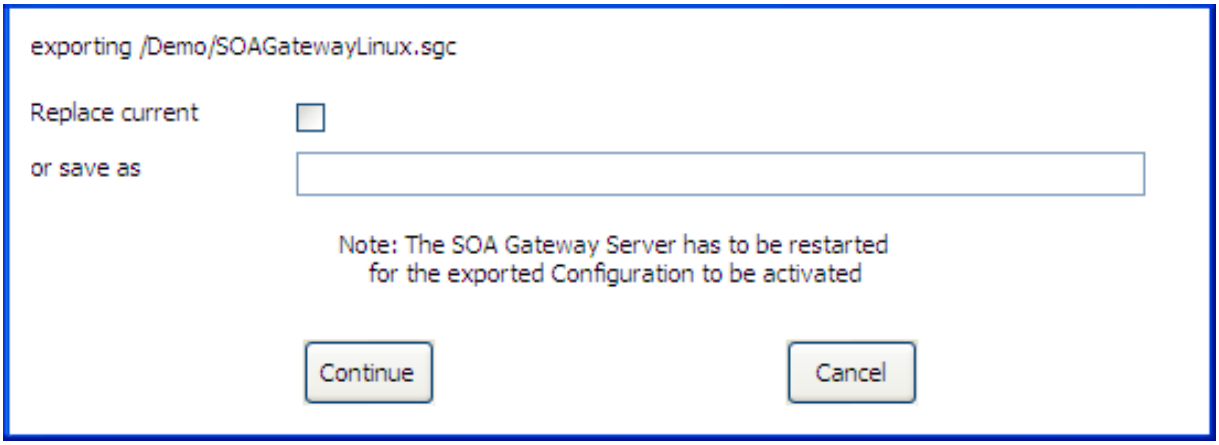
## Export the SOA Gateway Server configuration

An imported configuration can be exported back to any SOA Gateway 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 SOA Gateway configuration file, click "Continue" to perform the export operation.



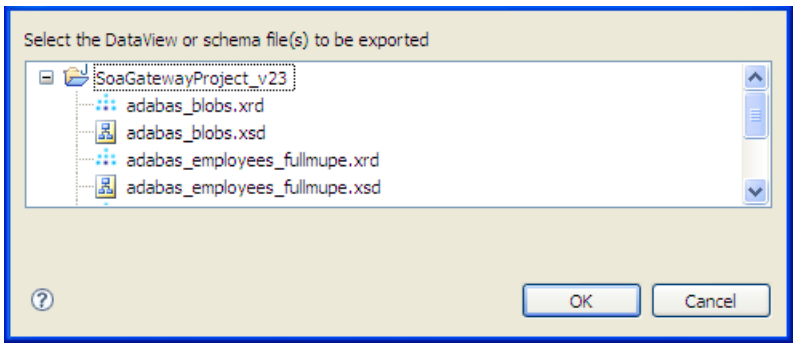

4. Successful export will be confirmed with a message in the status line

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

## Export SOA Gateway WebService definitions

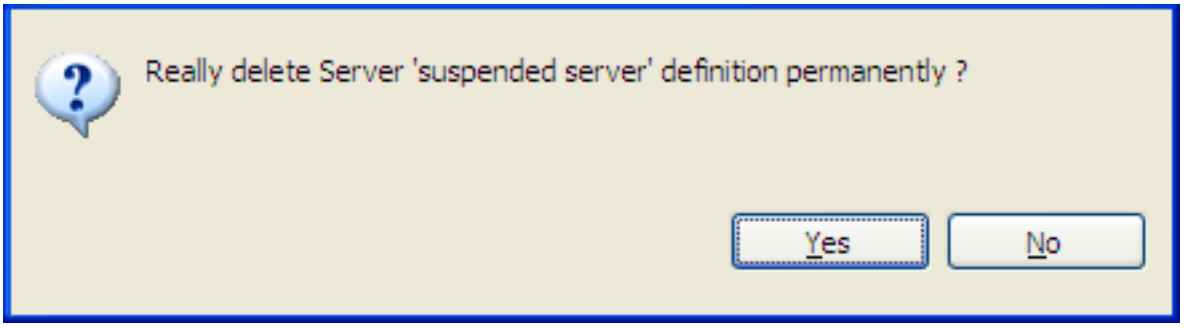
Export SOA Gateway WebService related definition elements - DataViews (XRDs) and/or XML Schemata (XSDs) and/or Stylesheets (XSLs).

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



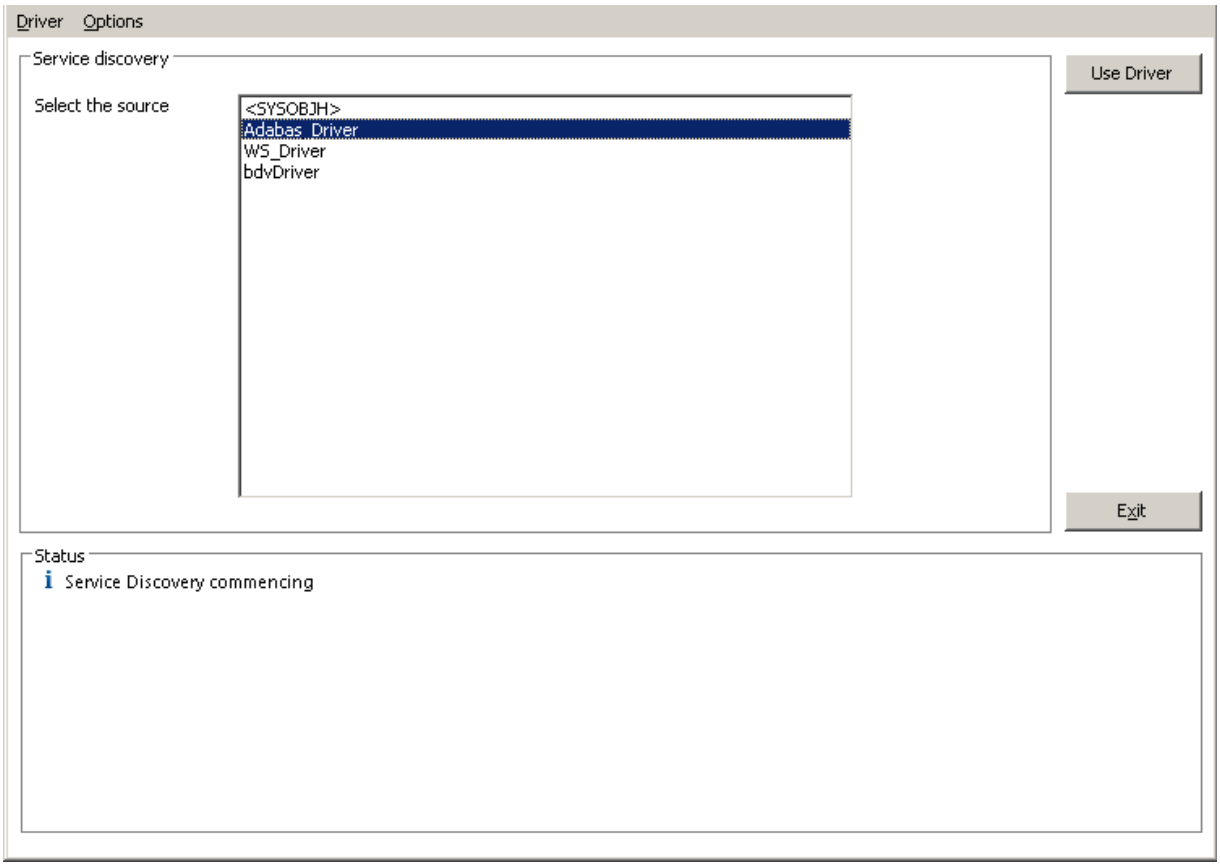

## Remove a SOA Gateway Server entry

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

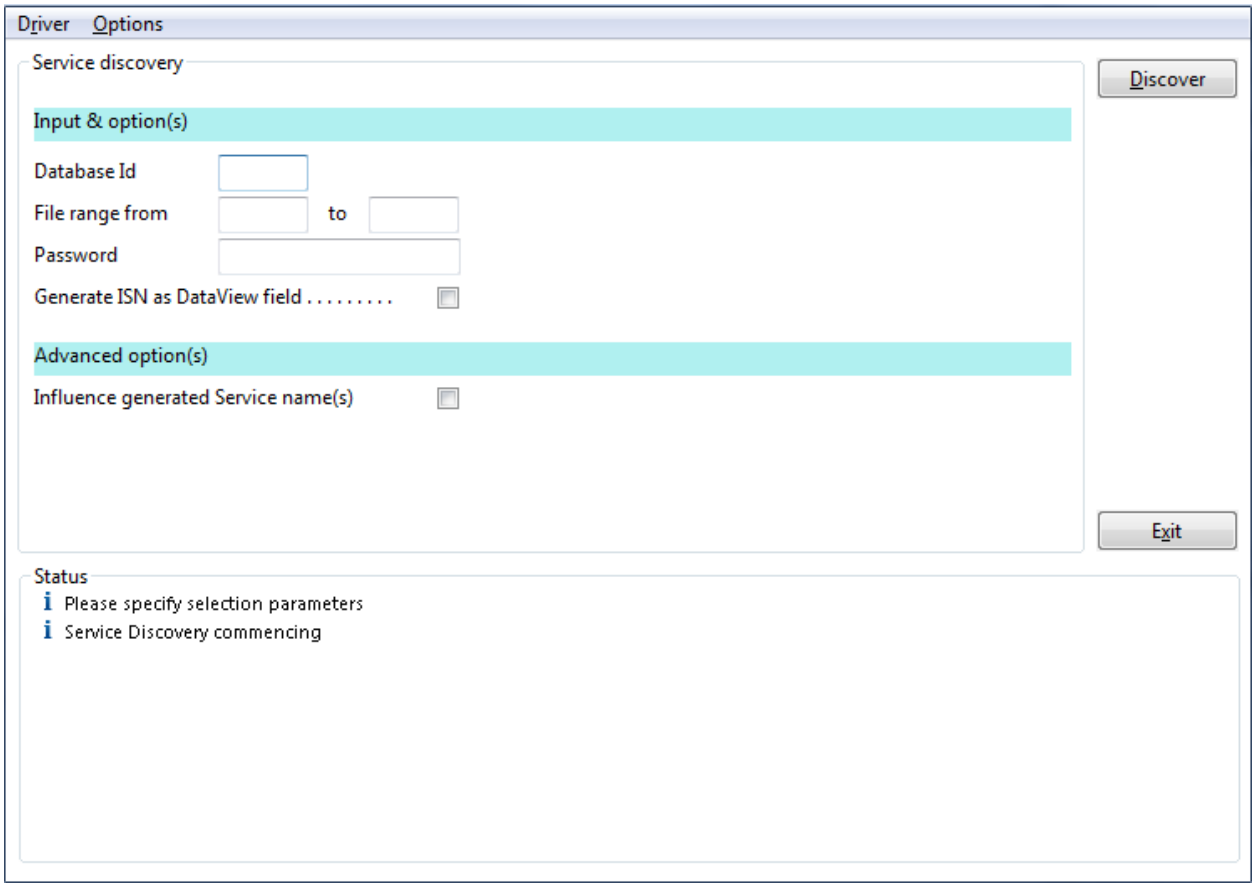
## Create new Services

1. Select the "Creat new Services" option from the context menu
2. 


   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




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 SOA Gateway WebService Creation section.

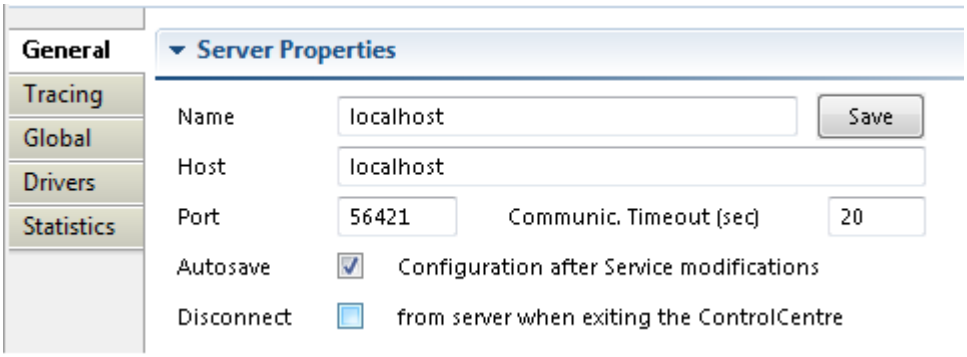
## Modify the SOA Gateway Server definition entry

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

To modify the SOA Gateway server entry:

1. In the SOA Gateway 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 SOA Gateway Status, and license information here too. This information is not modifable.

## Modify Global -> Security Options

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

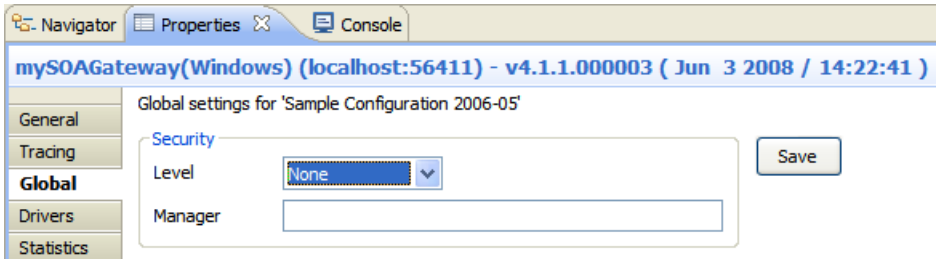
**Important:** We recommend you contact your SOA Gateway support team before modifying these parameters.

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 SOA Gateway can run at

1. None: No SOA Gateway 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 Certficate.

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.

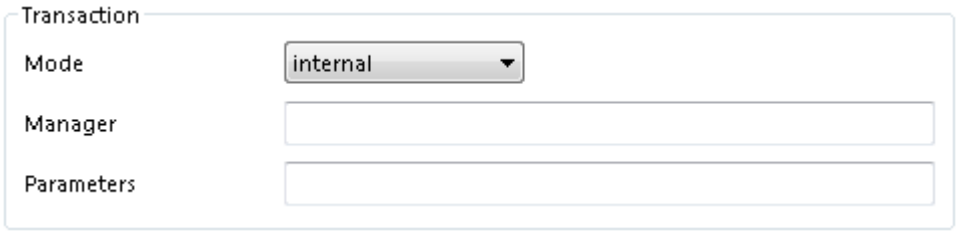



## Modify Global -> Transactional Options

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

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

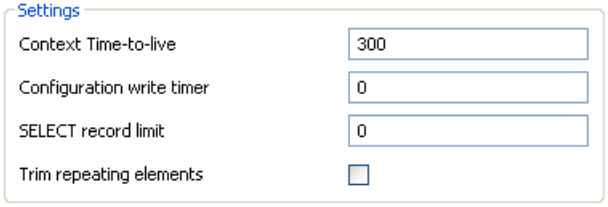
To open the Transaction view, left-click the server you wish to modify. In the Properties view, select the `Global` tab. Transaction options are in the `Transaction` group.




## Modify Global -> Settings

In this section you can set threshold values for SOA Gateway 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.




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 SOA Gateway 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.

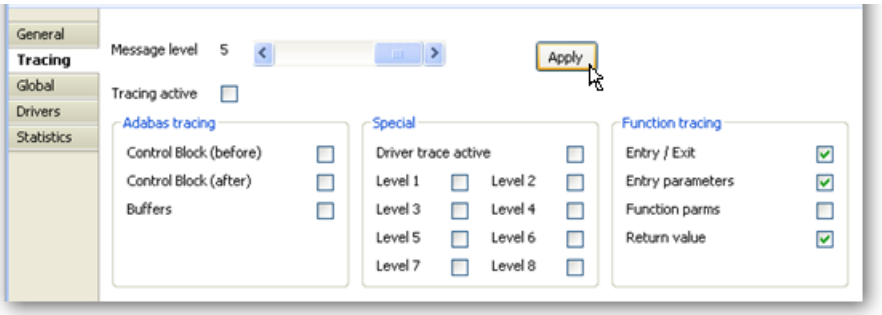
## Modify tracing / messaging options

The message level set for a SOA Gateway 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 SOA Gateway 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 SOA Gateway Server.

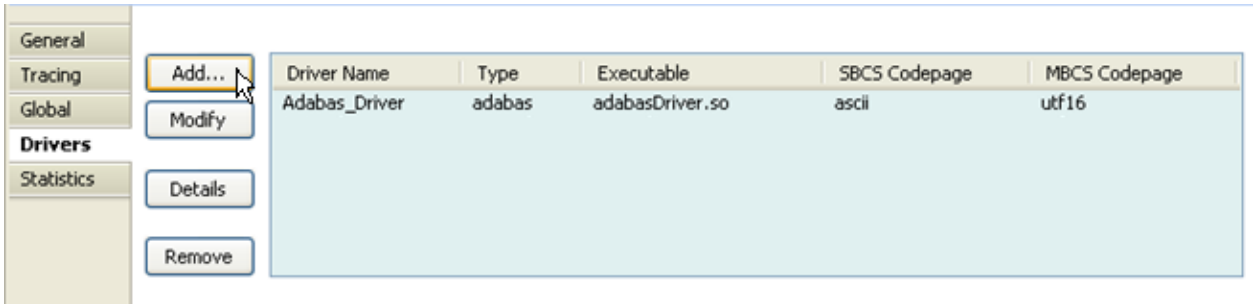


## Add / Modify Drivers

This section describes how you add/remove/modify the drivers that SOA Gateway uses to talk to the underlying resource

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.

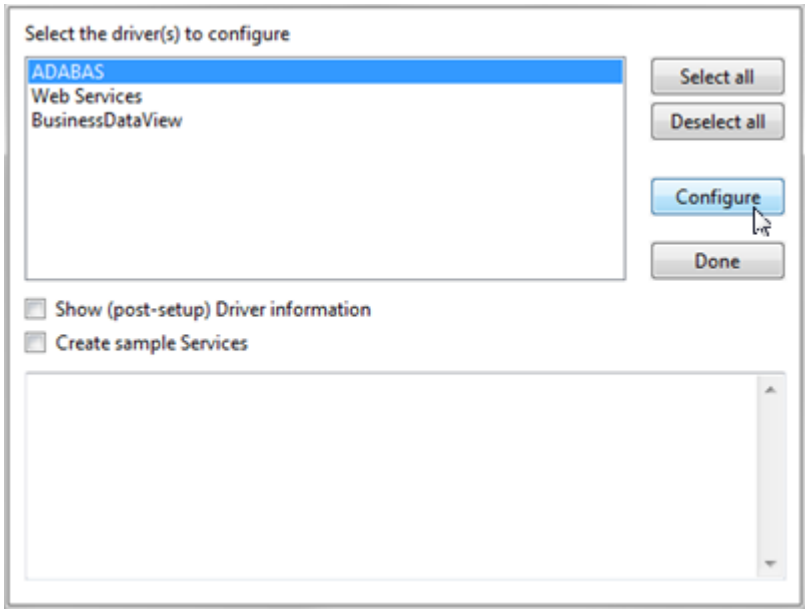



You may left-click a driver entry and select **Details** to display more information about this driver.

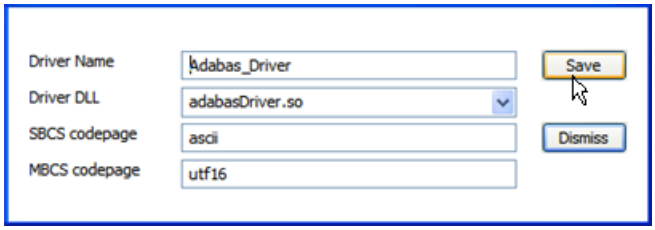
To modify an existing driver, left-click the driver,and select **Modify**

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 apper, 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 SOA Gateway 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 | |

And the following outlines the driver information for SOA Gateway 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 | |

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 SOA Gateway ..... 74
- IPv6 Considerations ..... 76
- Reducing memory footprint using streaming ..... 77

These tasks would typically be carried out by an experiened SOA Gateway user.

## How to install a server on a remote Windows

Typically when you install SOA Gateway on Windows, the Control Centre and the SOA Gateway server are on the same machine. If you wish to install SOA Gateway 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_2.6.1.NN

   where ../eclipse is the folder from which your Control Centre is launched.
2. Transfer the SOA Gateway 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 LICENSERC=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 SOA Gateway

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

- **Multiple instances of SOA Gateway on Linux**
- **Multiple instances of SOA Gateway on z/OS**

### Multiple instances of SOA Gateway on z/OS

The simplest way for setting up multiple SOA Gateway 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway installation, and then bring up both the existing and new installations using the same Apache webserver.

- If running, stop SOA Gateway. 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 apachectls scripts to start SOA Gateway
- Verify that the SOA Gateway start-up messages (found in the Apache logs) refer to the new SOA Gateway 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 SOA Gateway is supported on all non-Windows platforms. The underlying hosting infrastructure of SOA Gateway 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 SOA Gateway 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.

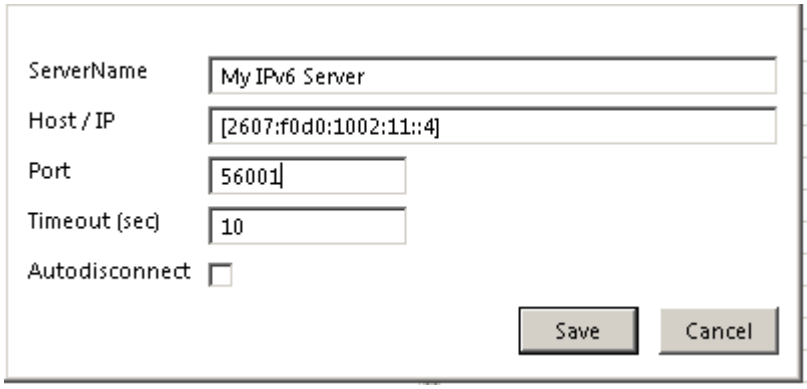
E.g.

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 SOA Gateway Control Centre, add a new server connection with the IPv6 address surrounded by [ and ] and the port number of your server




## Reducing memory footprint using streaming

When SOA Gateway 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, SOA Gateway has the ability to stream back data as it is retrieved from the database. The SOA Gateway 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, SOA Gateway does not have to build large internal strutures 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:** SOA Gateway 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

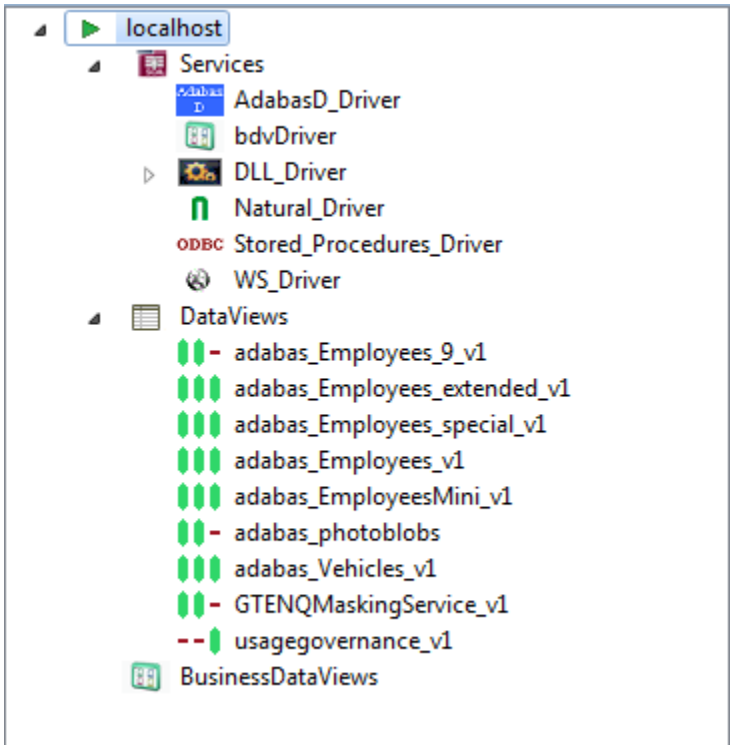
**Important:** Not all clients understand HTTP 1.1 or indeed support the chunked transfer encoding.

# 13 Working with the SOA Gateway View


- Operations at the Server level ..... 80
- Operations at the Driver level ..... 85
- Operations at the Service level ..... 86
- Operations at the DataView / BusinessDataView levels ..... 86

The SOA Gateway 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 SOA Gateway 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 SOA Gateway connection:* Define a new SOA Gateway 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 SOA Gateway 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 SOA Gateway 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 SOA Gateway servers.

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

4. ***Add driver to Server:***
5. *Add new SOA Gateway connection:* Define a new SOA Gateway 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 SOA Gateway server is running on. You may also view some detailed information about the server.

To modify the SOA Gateway server entry:

1. In the SOA Gateway 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 SOA Gateway Status, and license information here too. This information is not modifable.

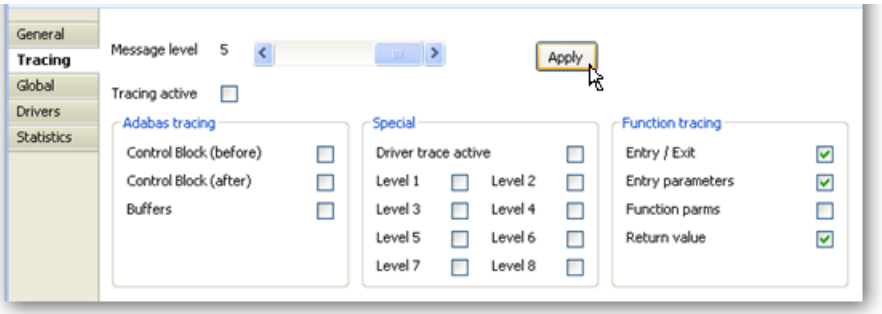
### Modify tracing and message level options

The message level set for a SOA Gateway 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 SOA Gateway Server.

To change either the message level or trace options:

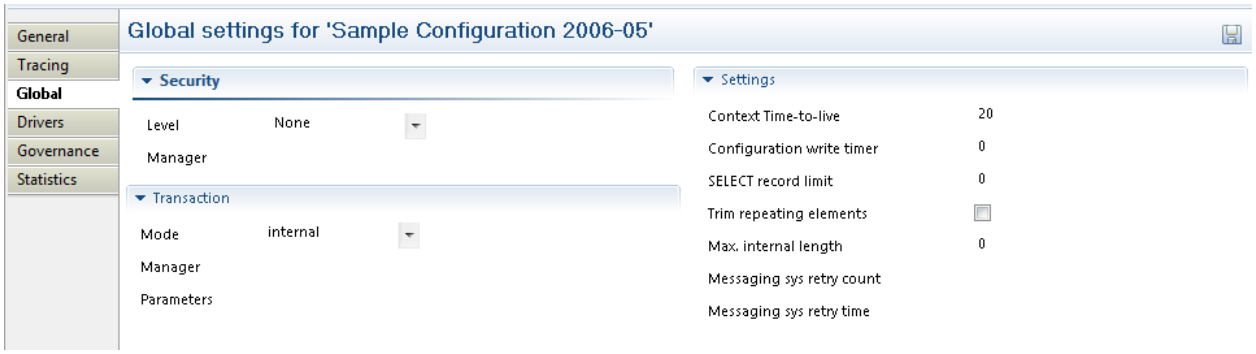
1. Select the "Tracing" tab from the Properties view
2. Modify the message level or tracing option(s), as appropriate, click "Apply" to send the changes to the SOA Gateway Server.



### Modify global settings

Global settings for the entire SOA Gateway Server can be changed here.

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




#### Security Settings

There are 4 security levels that SOA Gateway can run at

1. None: No SOA Gateway 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 Certficate.

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 SOA Gateway 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 SOA Gateway 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.

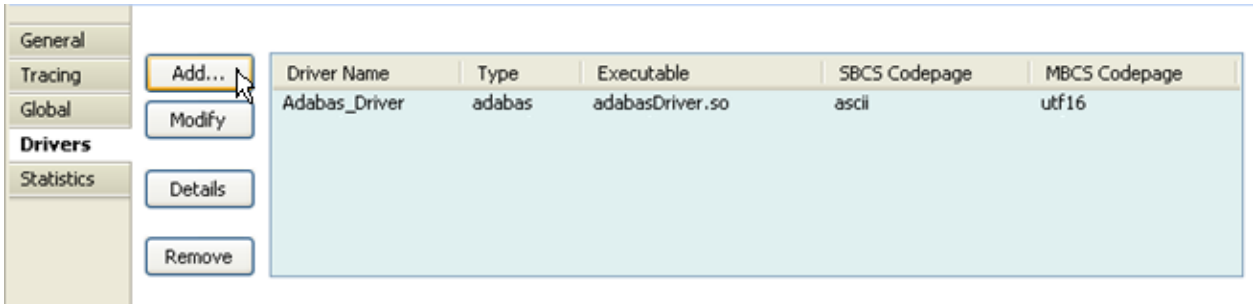
5. Max internal length: An upper limit on the size of a field in the XRD.
6. Messaging sys retry count: If using SOA Gateway to connect to a messaging system such as EntireX or MQ, it may be that a failure occurs attempting to connect due to a database not being started. A retry count can be set to allow 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 SOA Gateway uses to talk to the underlying resource

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.




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 SOA Gateway 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 | |

And the following outlines the driver information for SOA Gateway 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 | |

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.

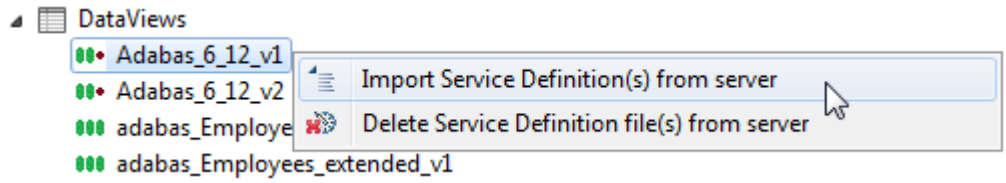
2. The Control Centre will automatically create an "empty" DataView or BusinessDataView for you, based on the type of the driver, and will open the editor on the newly created view.
3. A tutorial on creating a DataView from scratch is available here
4. Once the file is saved, the (Business)DataView will 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.

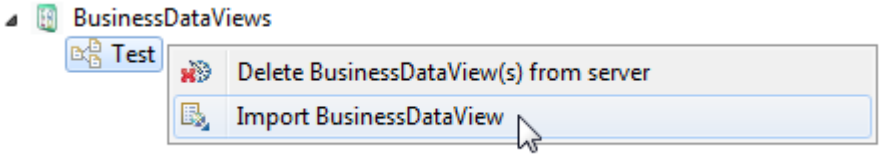
## Operations at the Service level

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 SOA Gateway View

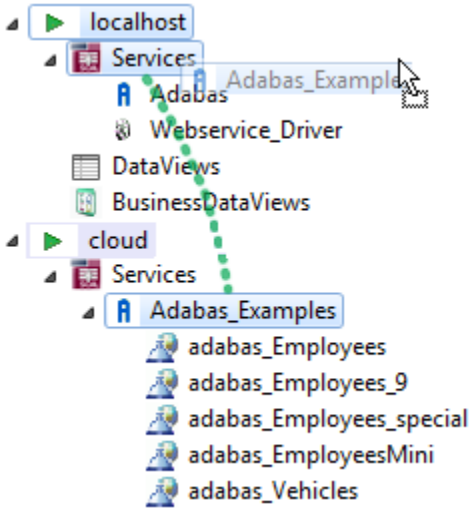

- Driver definition as a drag source ..... 88
- Service as a drag source ..... 88
- DataView and BusinessDataView as a drag source ..... 90

The SOA Gateway 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)

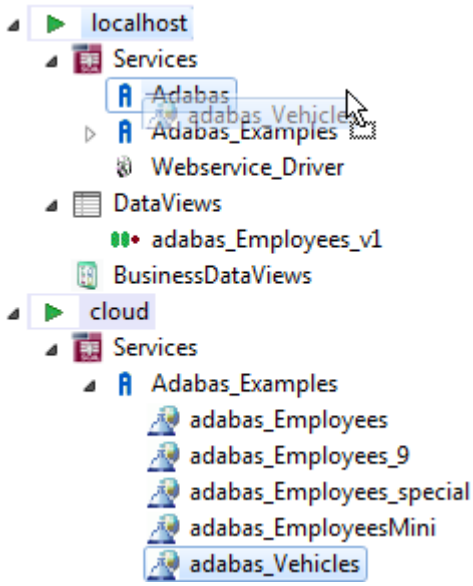
## Service as a drag source

- Dragging services from one server to another

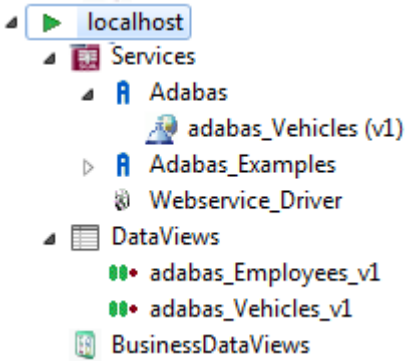
- Dragging services from one driver to another on the same server

### 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

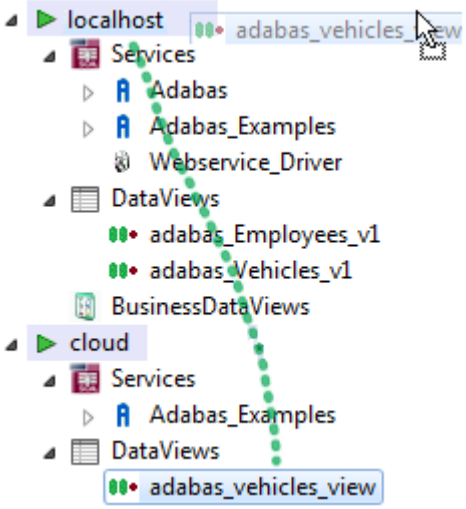
A DataView or BusinessDataView dropped on a service will replace the respective view currently used by that service.

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 SOA Gateway BusinessDataViews


- Overview / Concepts ..... 92
- Using the BusinessDataView WebService ..... 97
- Working with BusinessDataView Objects ..... 221
- Summary ..... 100
- Creating BusinessDataViews ..... 101
- BusinessDataView Editor hints ..... 126

SOA Gateway 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, SOA Gateway 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. SOA Gateway 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

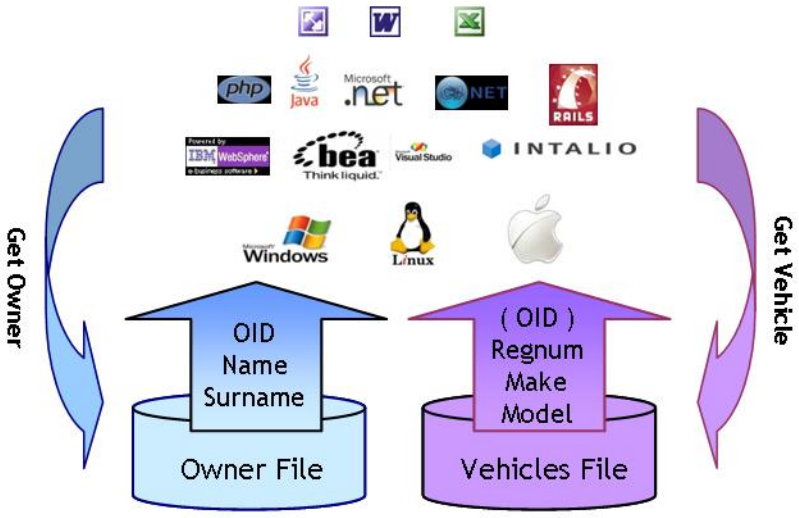
**Note:** This will contain details about the vehicles that are owned by owners. For the purposes of this illustration, we will assume the following fields:

- 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 SOA Gateway 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.

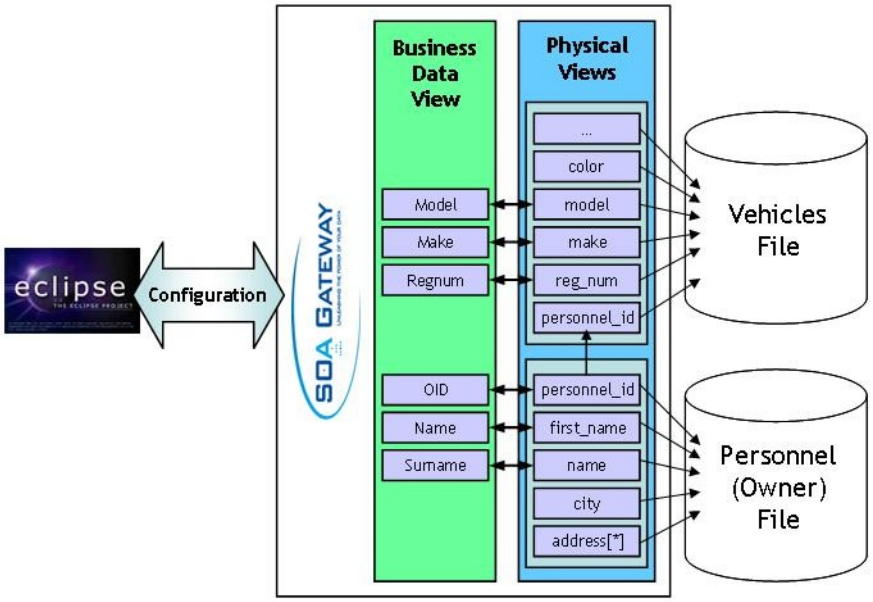
### The SOA Gateway approach

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. SOA Gateway 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. SOA Gateway 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, SOA Gateway will interface with it to guarantee transaction integrity.

### Creating a BusinessDataView Service

A BusinessDataView based WebService is created in SOA Gateway 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 SOA Gateway may be reused in one or more Business Data Views. The Eclipse Control Center supplied with SOA Gateway 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 SOA Gateway.

## Using the BusinessDataView WebService

Once the BusinessDataView Service has been created, it can be used in precisely the same way as any other SOA Gateway 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 SOA Gateway 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 Business-DataView 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:

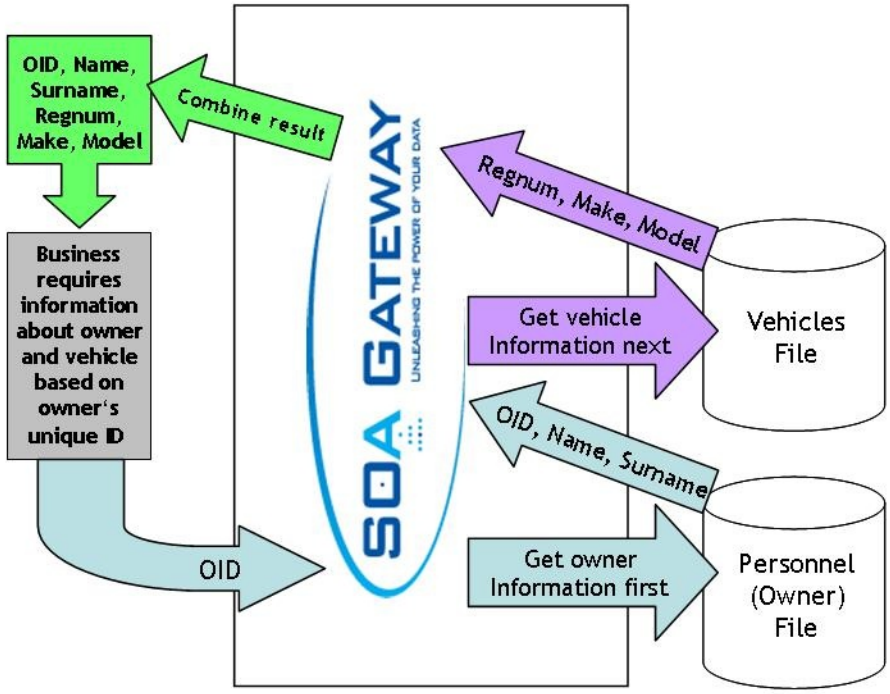
*http://<host>:<port>/OwnerVehicle?F=LIST&OID=1234**

## Working with BusinessDataView Objects

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

### Accessing a BusinessDataView Object

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

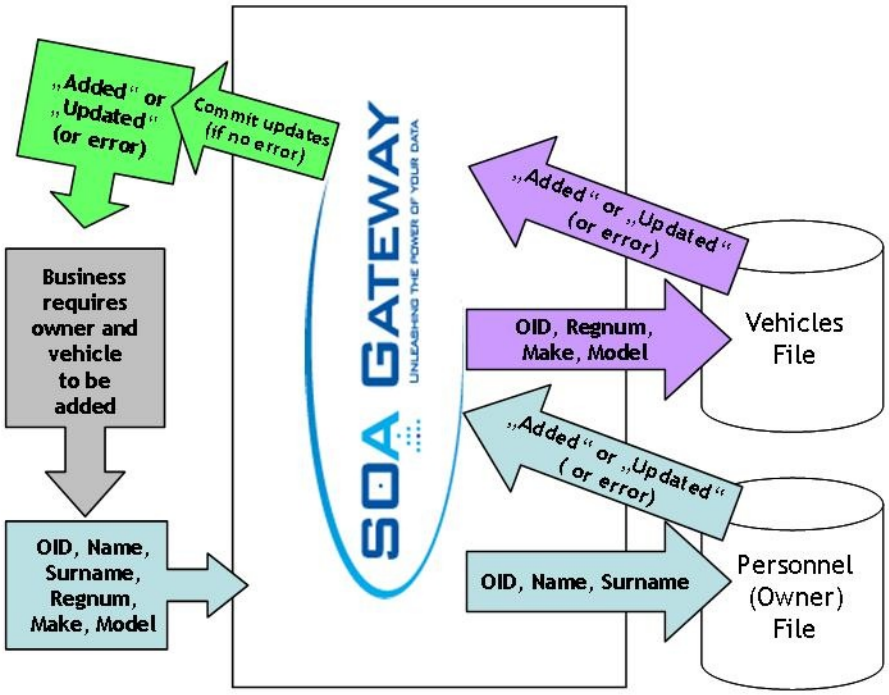



Some notes about the above:

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 SOA Gateway 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, SOA Gateway can achieve this as one standard REST or SOAP request:

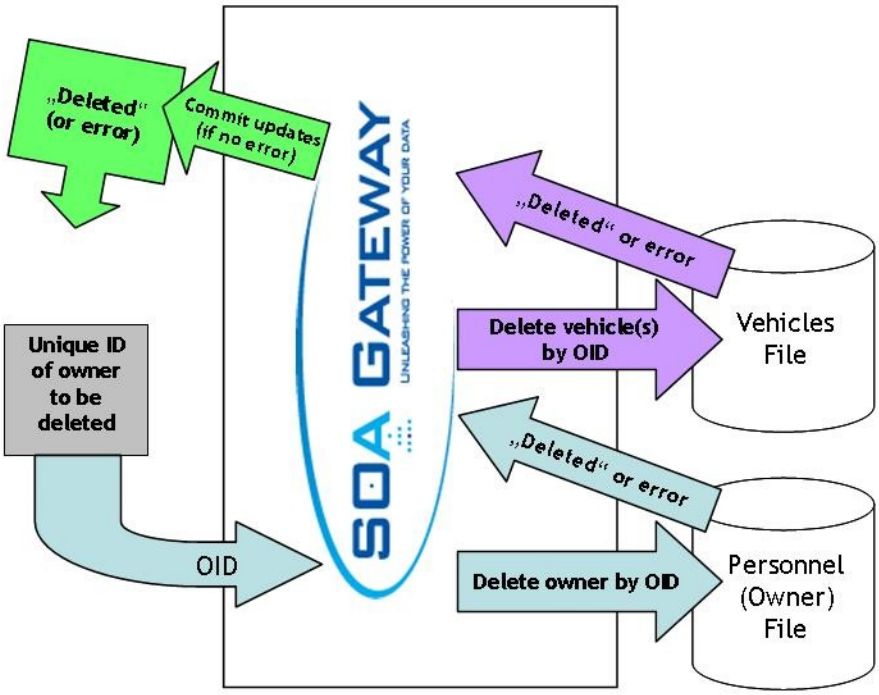



Some notes about the above:

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, SOA Gateway 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.

## Summary

It is clear that the Business Data View concept in SOA Gateway 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. Once the data model is understood and turned into a BusinessDataView new projects will benefit in terms of faster and more reliable implementations.

- Create Business Data View models from multiple sources including different databases as required.

- Ensure consistency in your data by exposing services that prevent applications creating inconsistencies in the data. Two-phase-commit (2PC) support is required when more than one database is involved in a transaction.

- 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.

## Creating BusinessDataViews

Follow these steps to create a BDV

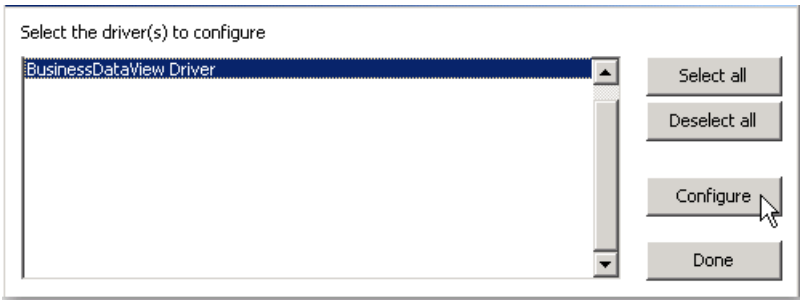
### Prerequisite - the BusinessDataView driver

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*.



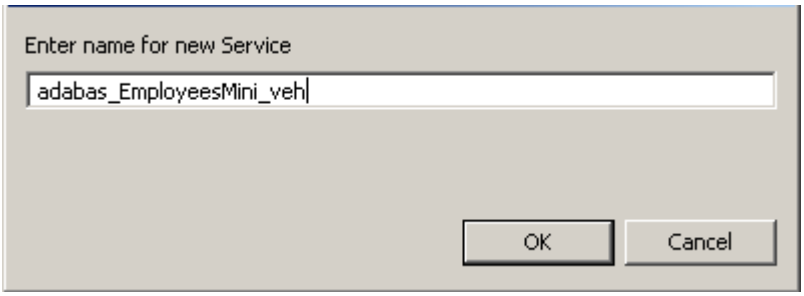
### Example 1

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.

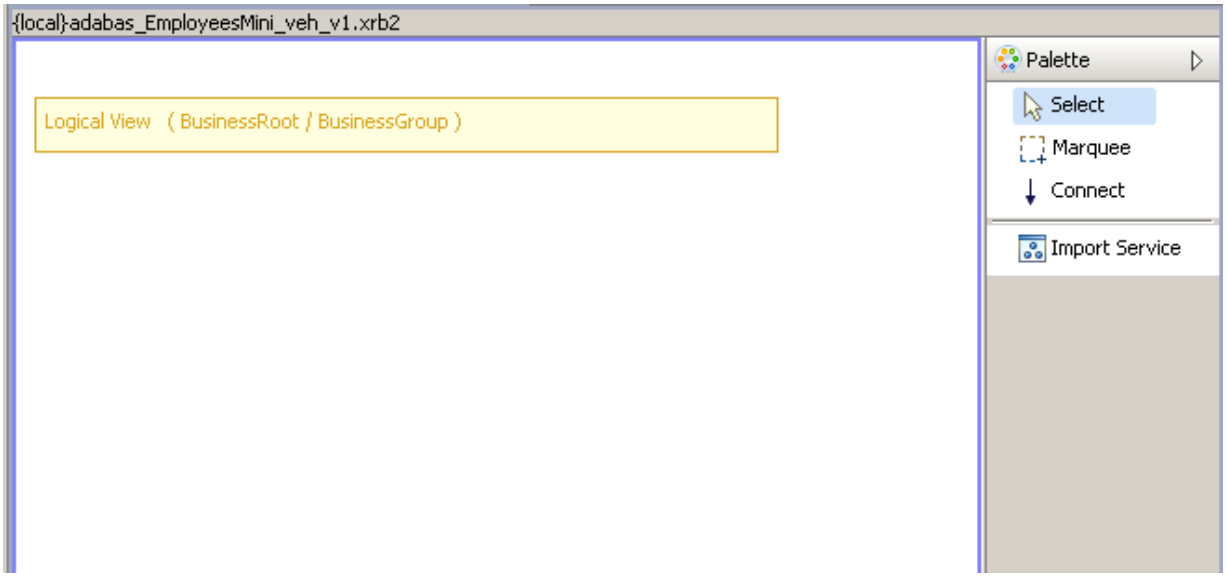
#### Creating the BusinessDataView

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

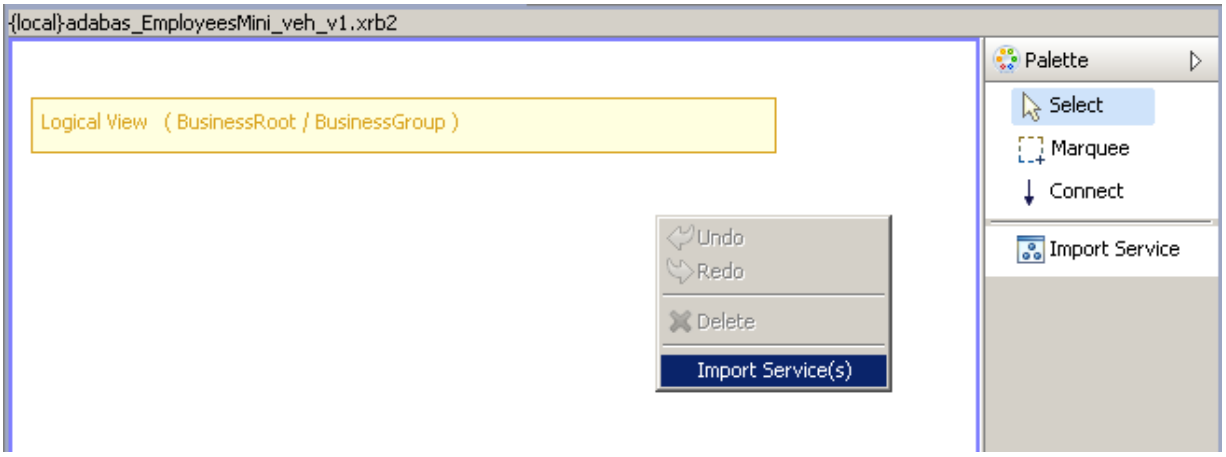
- Name it "adabas_EmployeesMini_veh", click OK




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

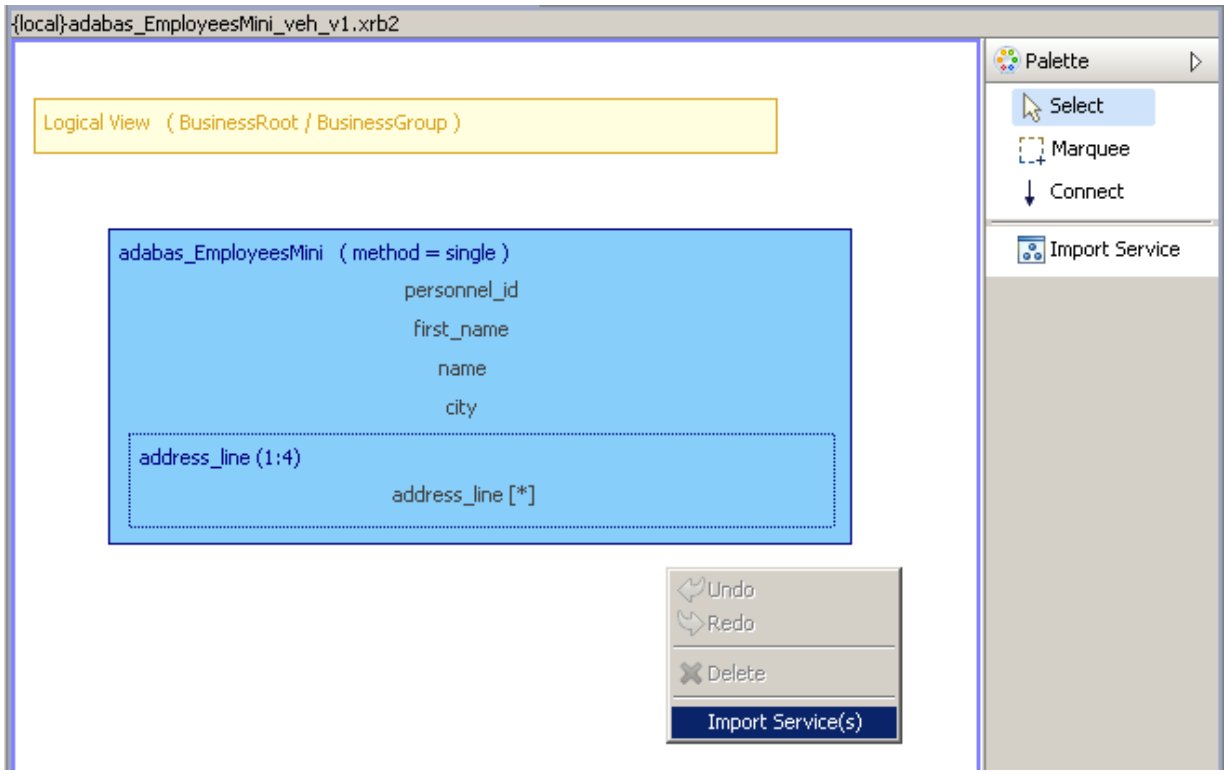



- Right-click anywhere within the blank area of the BusinessDataView Editor, select the *Import Service(s)* function from the context menu




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

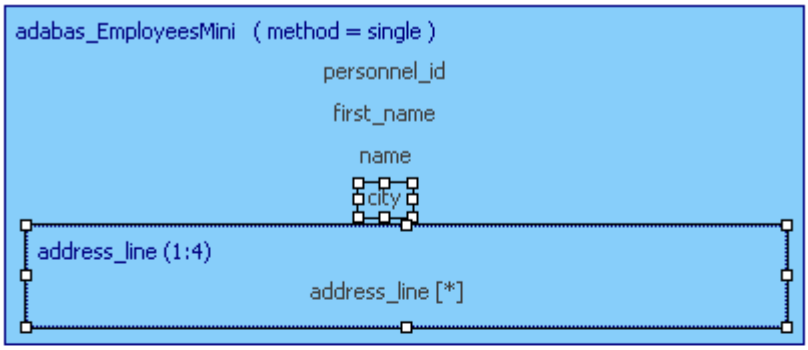
- A visual representation of the elements of this web service appears as shown below. Right-click beneath this view as shown and select the *Import Service(s)* function from the context menu

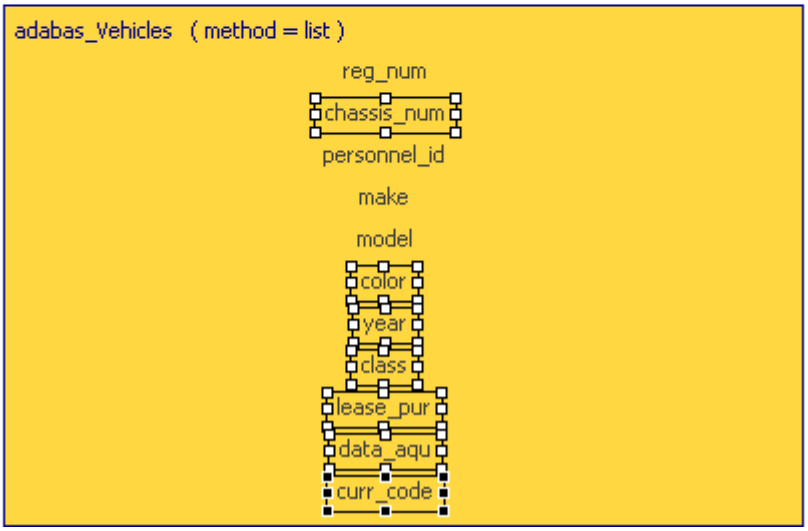



- Select the *adabas_Vehicles* WebService service and click *Import*.

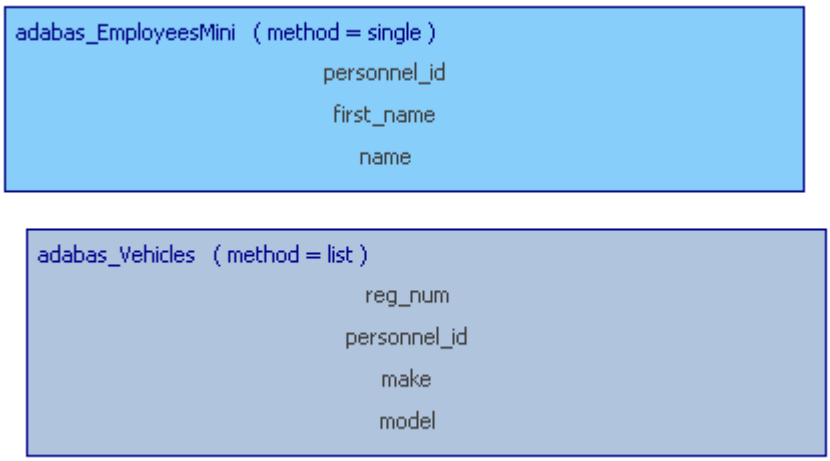
- Select the "Physical View" items not required for the time being (shown selected below) and hit the <*DEL*> key to remove them. There is *no* requirement whatsoever to use all of the "base WebService" fields within the BusinessDataView's "physical resource(s)". Note that there is also no requirement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

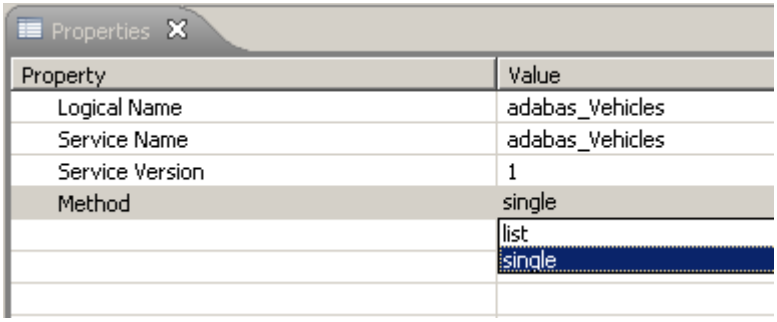


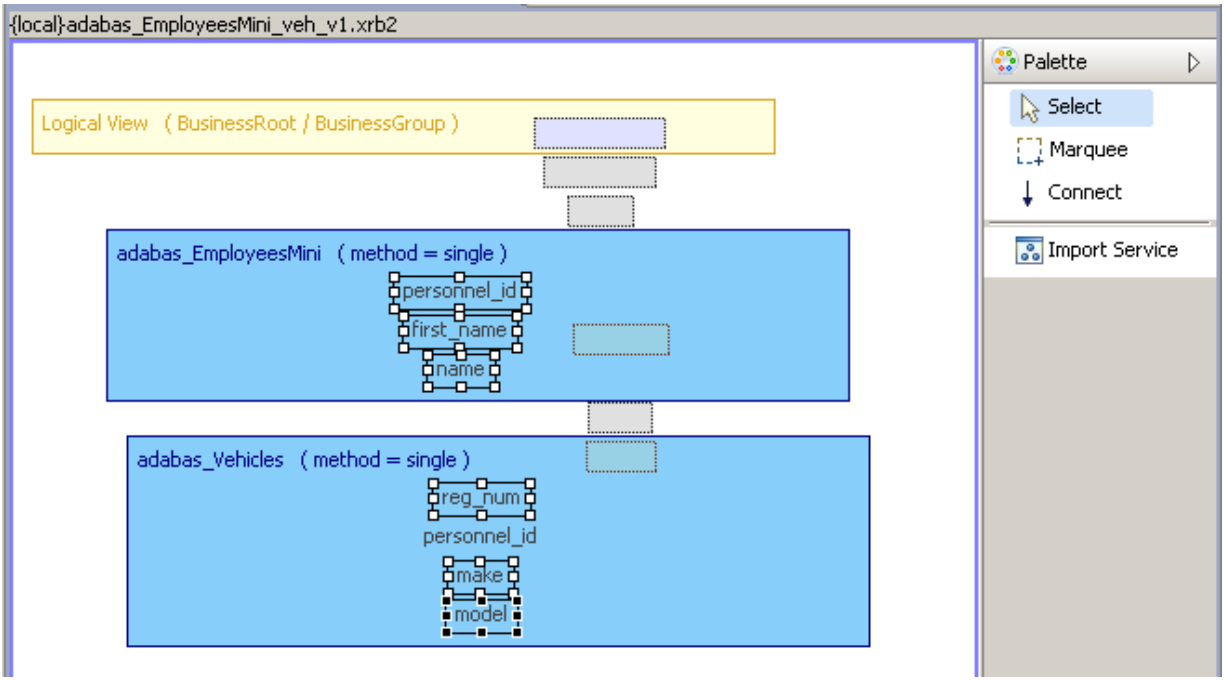

- This will leave you with the following items. For this example we are only interested in one vehicle per employee so we need to change the method from *list* to *single*. Our next example will demonstrate the use of the list method.



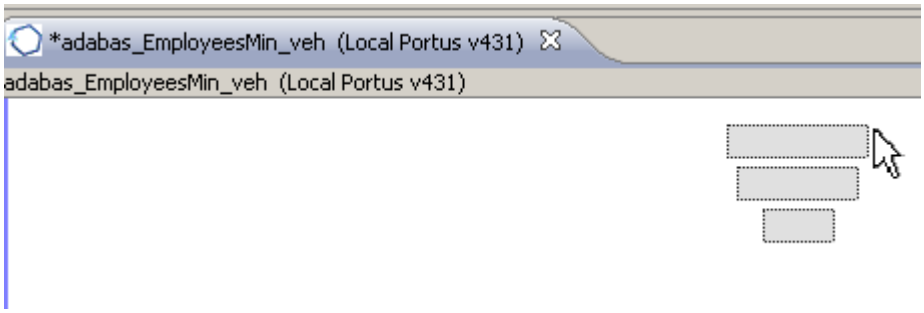

- Click on an empty area within the adabas_Vehicles physical view. The Properties view should display the properties for this Physical Resource. If the Properties view is not visible, select menu item Window -> Show View -> Properties. Change the Method to single as shown. Ctrl+S to save.



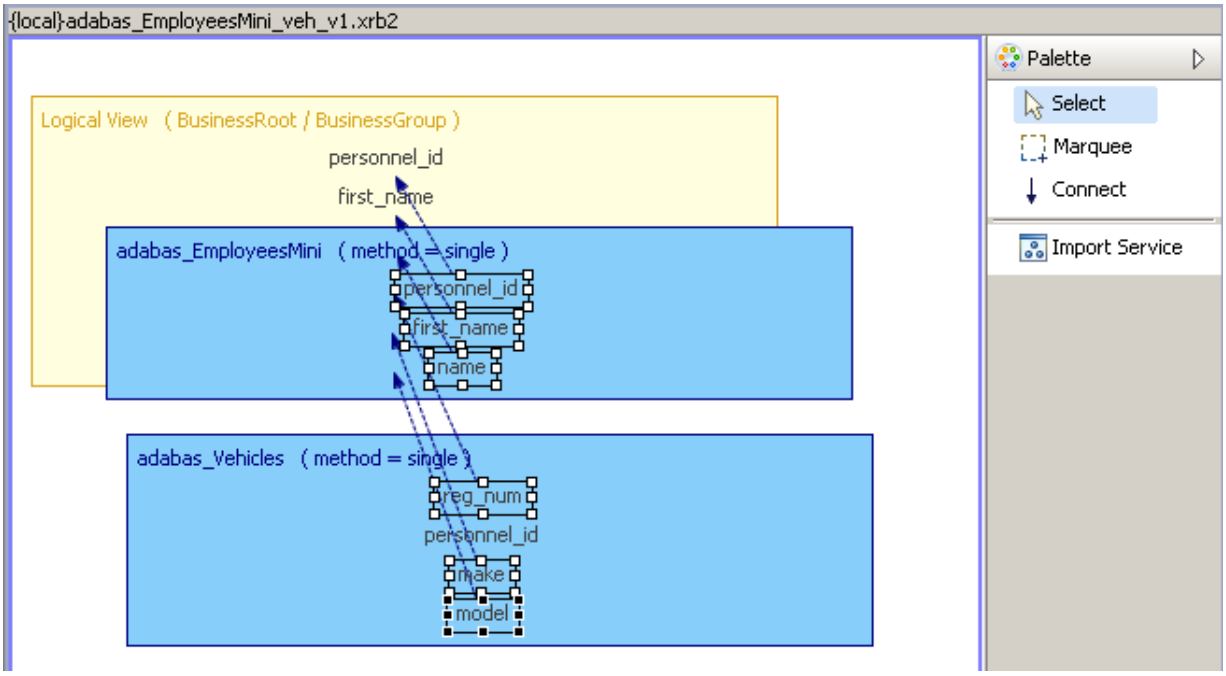

- Select all fields, *but NOT the personnel_id field within adabas_Vehicles*. Drag them across to the "Logical View" (LV) and drop them there. Note that there is a shadow view of the items selected. There is also a red circle with a line through it (not shown here) which disappears when the drop into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



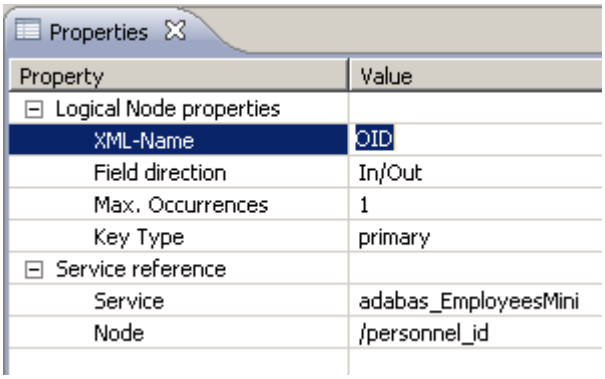

The BDV editor screen will scroll if the elements being dragged are held close to the editor boundary



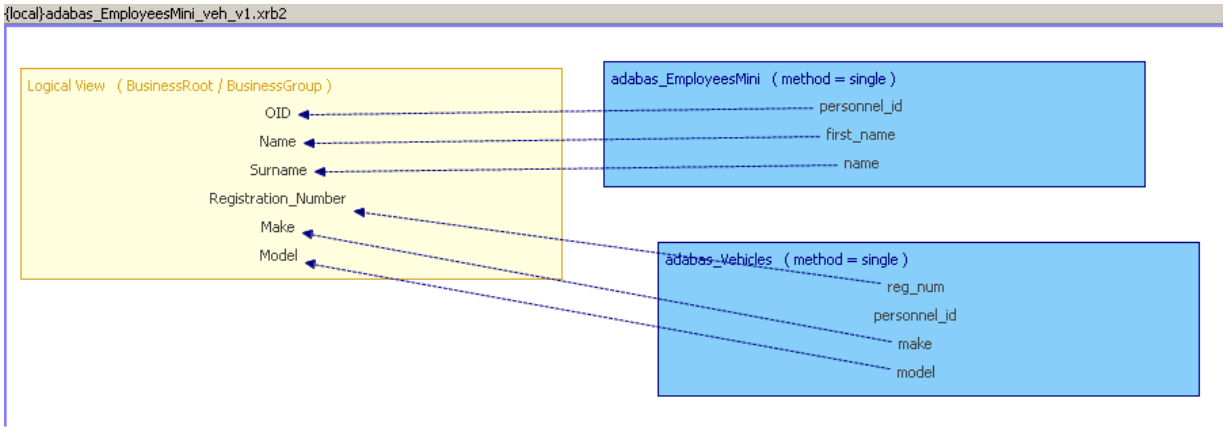

- This will add the selected fields to the "Logical View", and also create their connections to the source "Physical View" fields at the same time. Note that the various views can be moved around to get a clearer picture. It is a good idea to save the BDV at this stage (Ctrl+S).



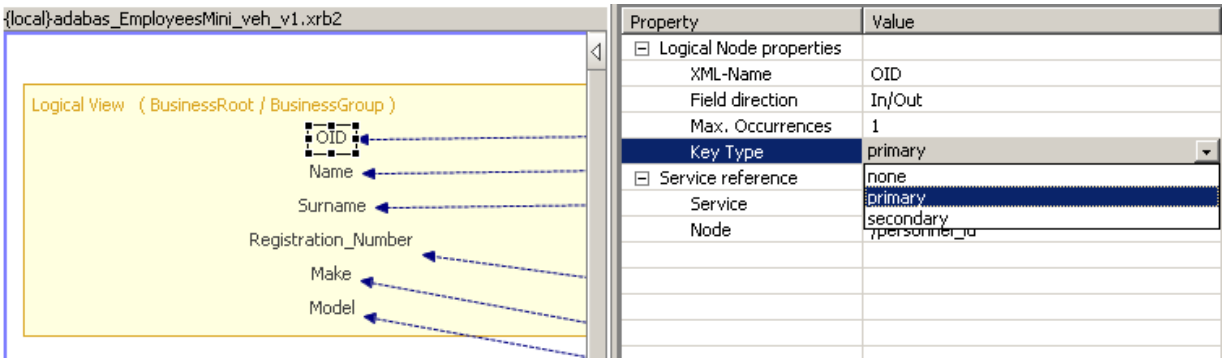

- Now go to the "Logical View" and select the *personnel_id* field. In the Properties view change the node name to *OID* by changing its XML-Name value.



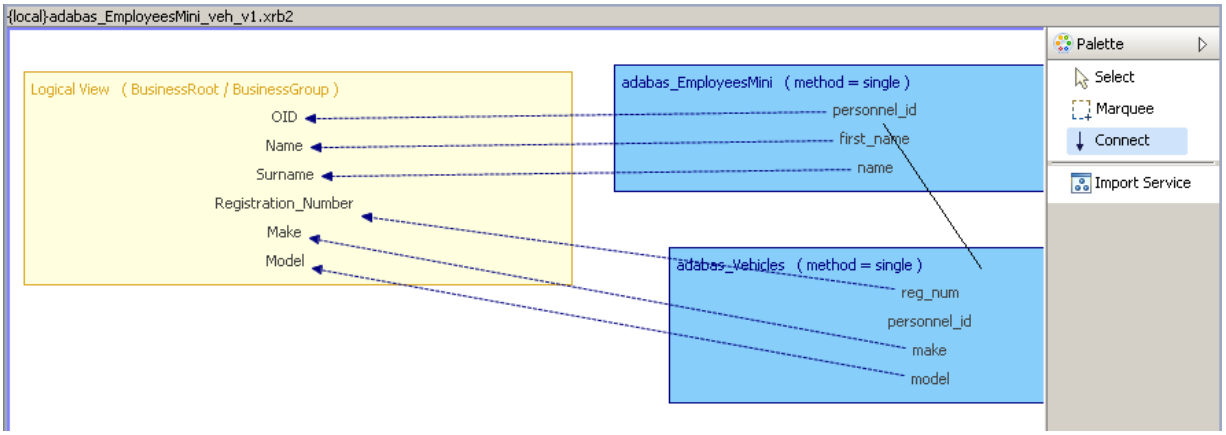

- Change all field names as required, the picture should now look like this. Note again that this is not compulsory. The names can be left as is if so desired.



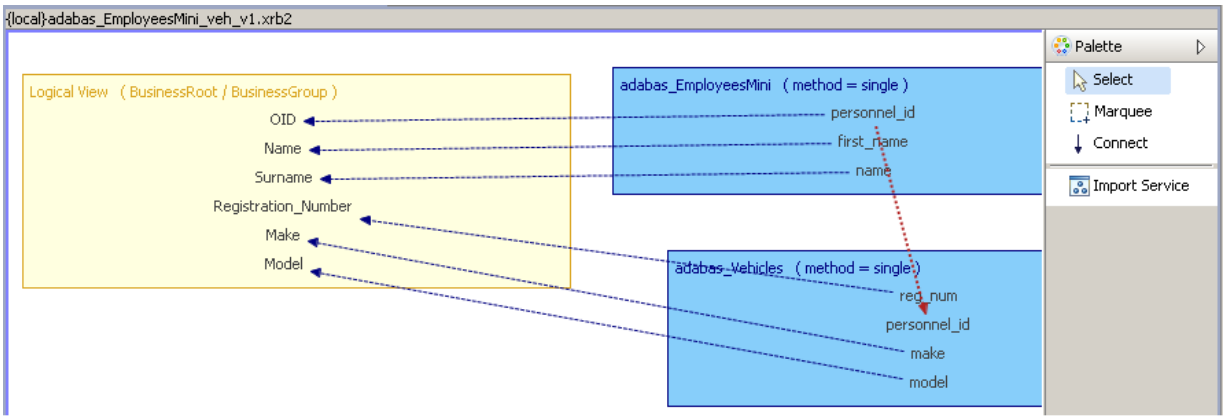

- Recall that this BDV will enable us to query employee details and their vehicle details. There are 2 steps to creating this link in the BDV. The first step involves setting the appropriate field to be a primary or secondary key. In this example we need to set the OID field in the LV to primary. This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



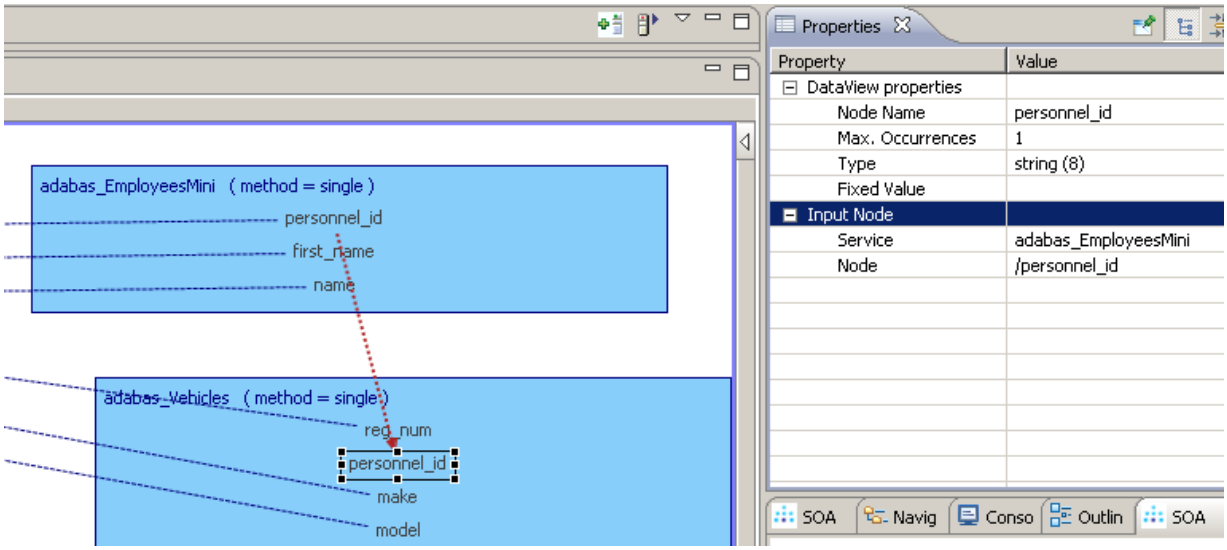

- The next step involves creating a connection or link between the appropriate field of one physical resource to a field in another physical resource. Select *Connect* item from the Palette. Select the *personnel_id* field of *adabas_EmployeesMini* view and move the cursor away. As you can see this marks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




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



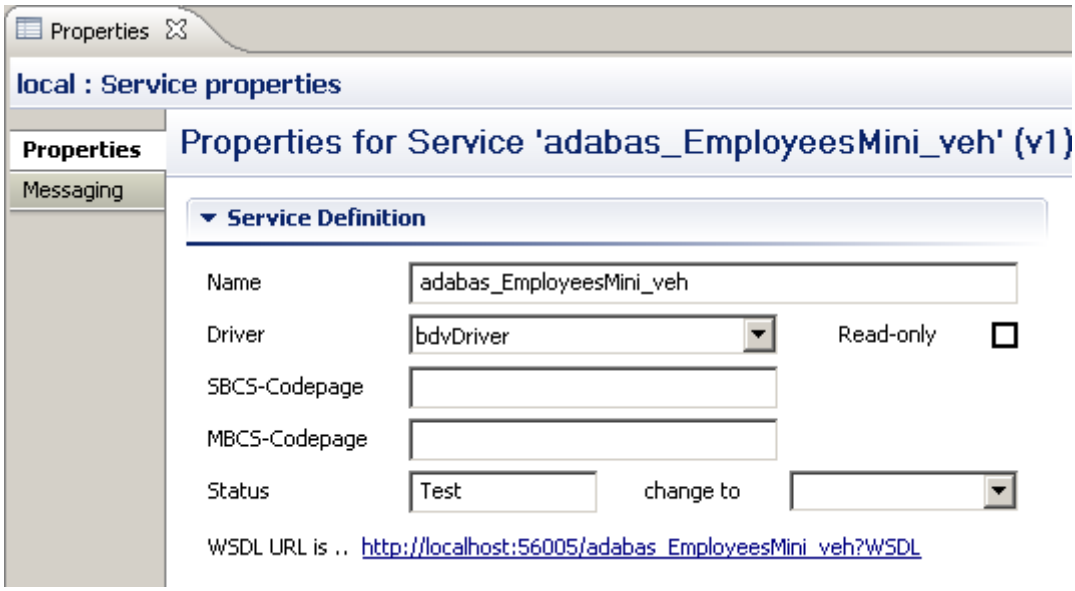

- This "links" the "base (physical) WebServices" together and indicates that the *personnel_id* from *adabas_EmployeesMini* is to be passed on and input into the *personnel_id* field of *adabas_Vehicles*. When the WebService based on the BusinessDataView is invoked, the "Employees" file is read first,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



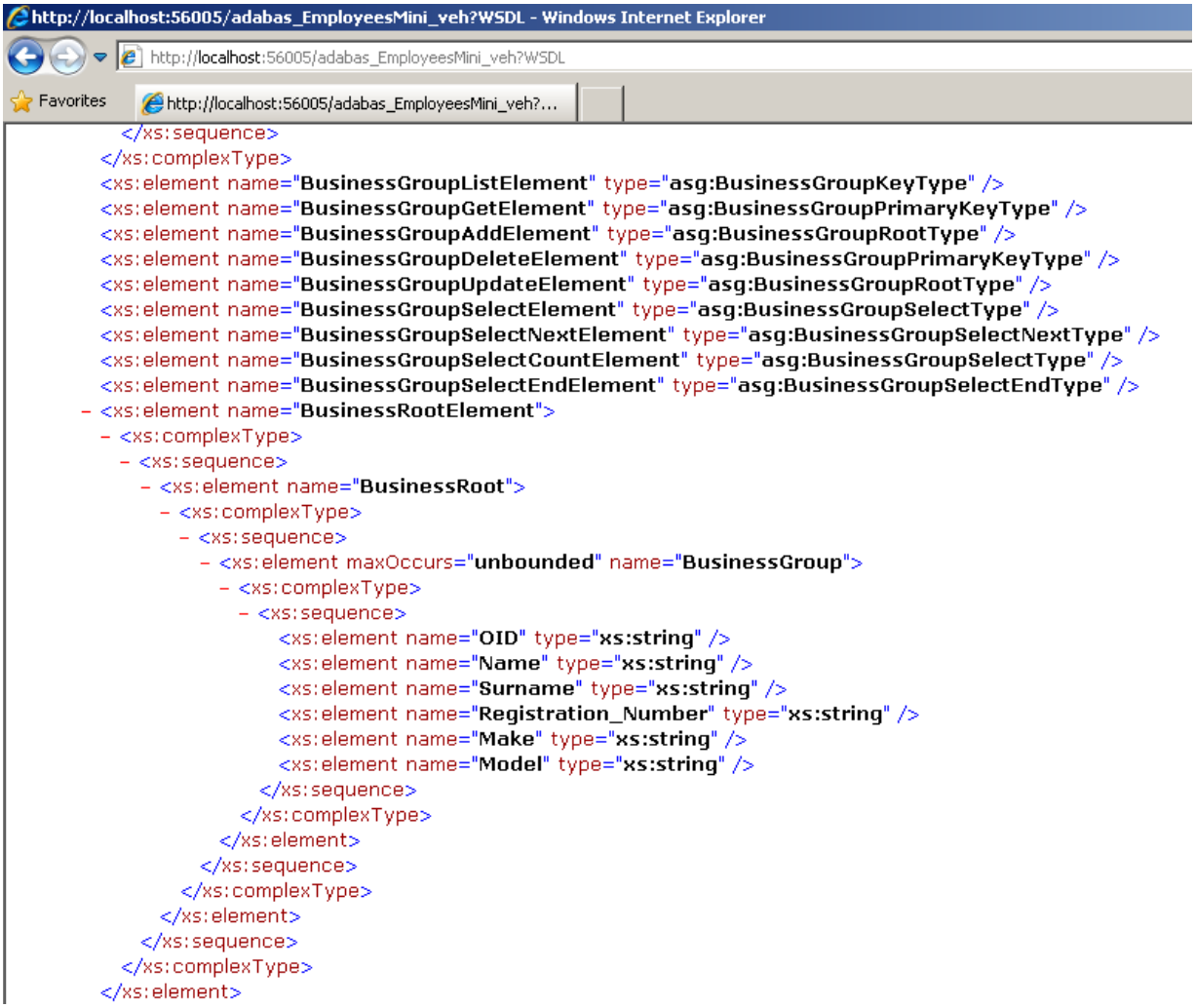

- Click the *Save* icon, or File->Save, or Ctrl+S to save the BusinessDataView on your SOA Gateway server

### 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).




- 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.

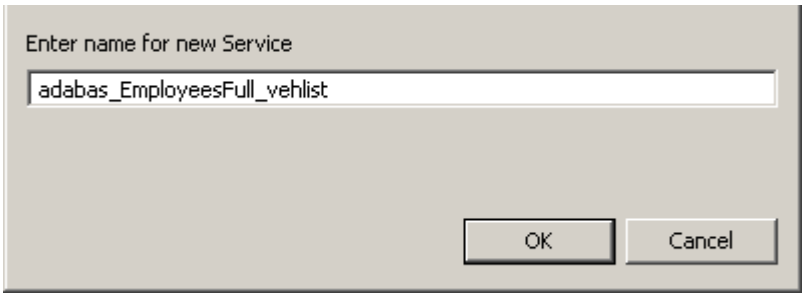
### Example 2

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

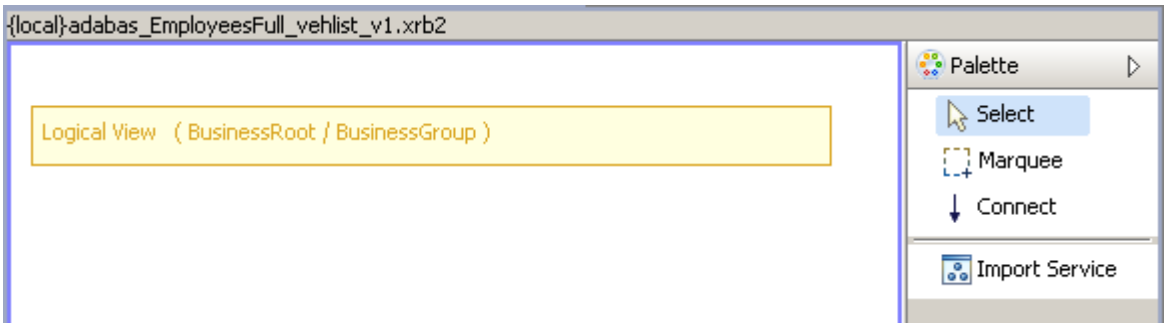
### Create the BusinessDataView

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

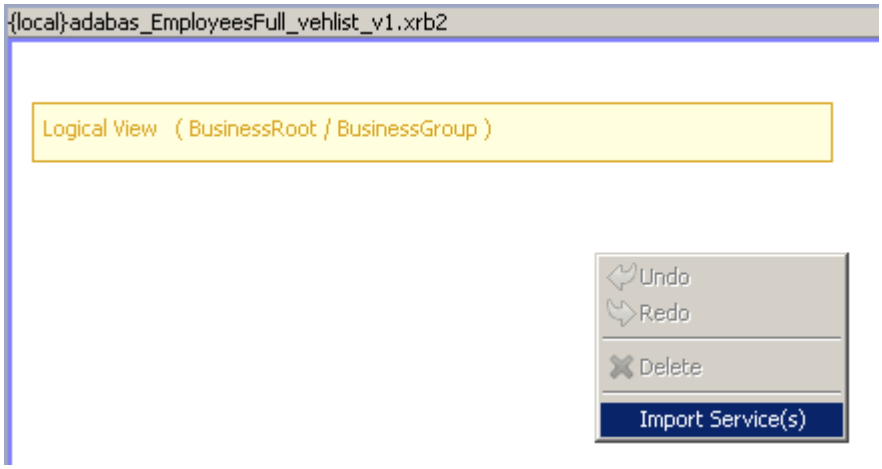
- Name it *adabas_EmployeesFull_vehlist* and click OK



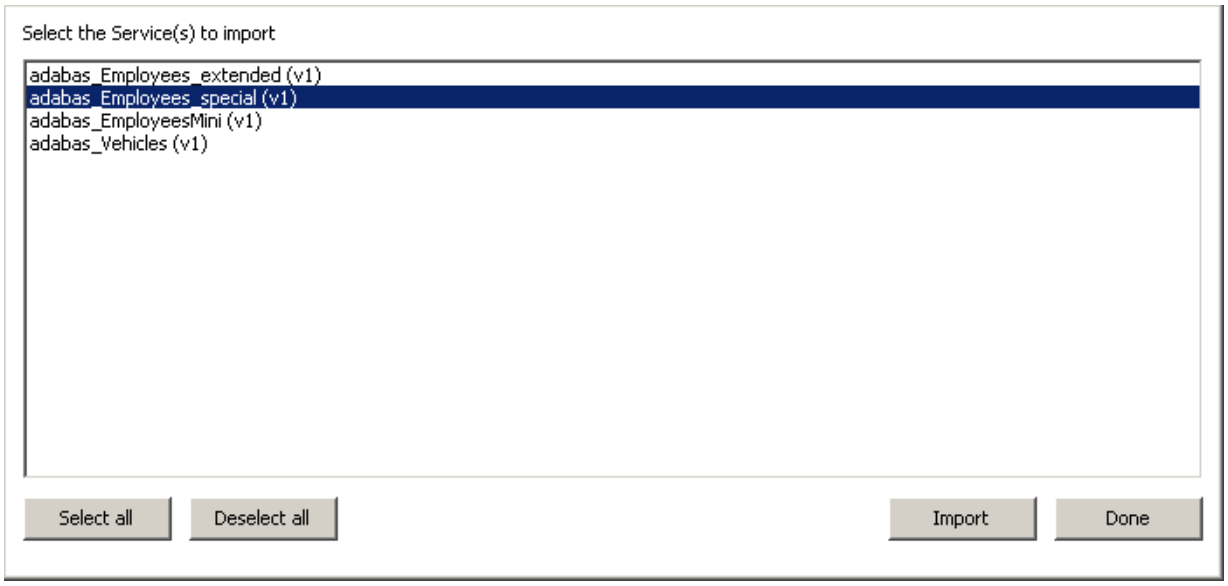

- This will open the BusinessDataView Editor with an initially empty "Logical View" (LV) created



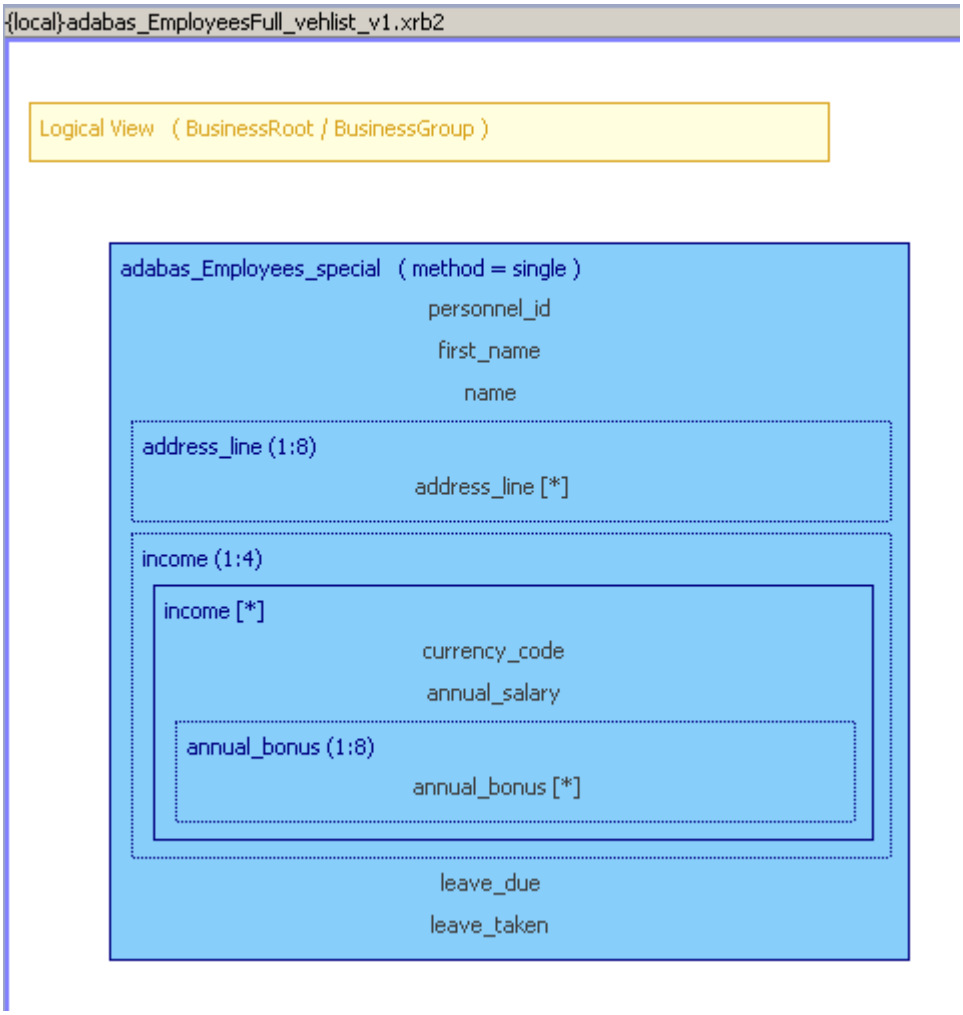

- Right-click anywhere within the blank area of the BusinessDataView Editor, select the *Import Service(s)* function from the context menu



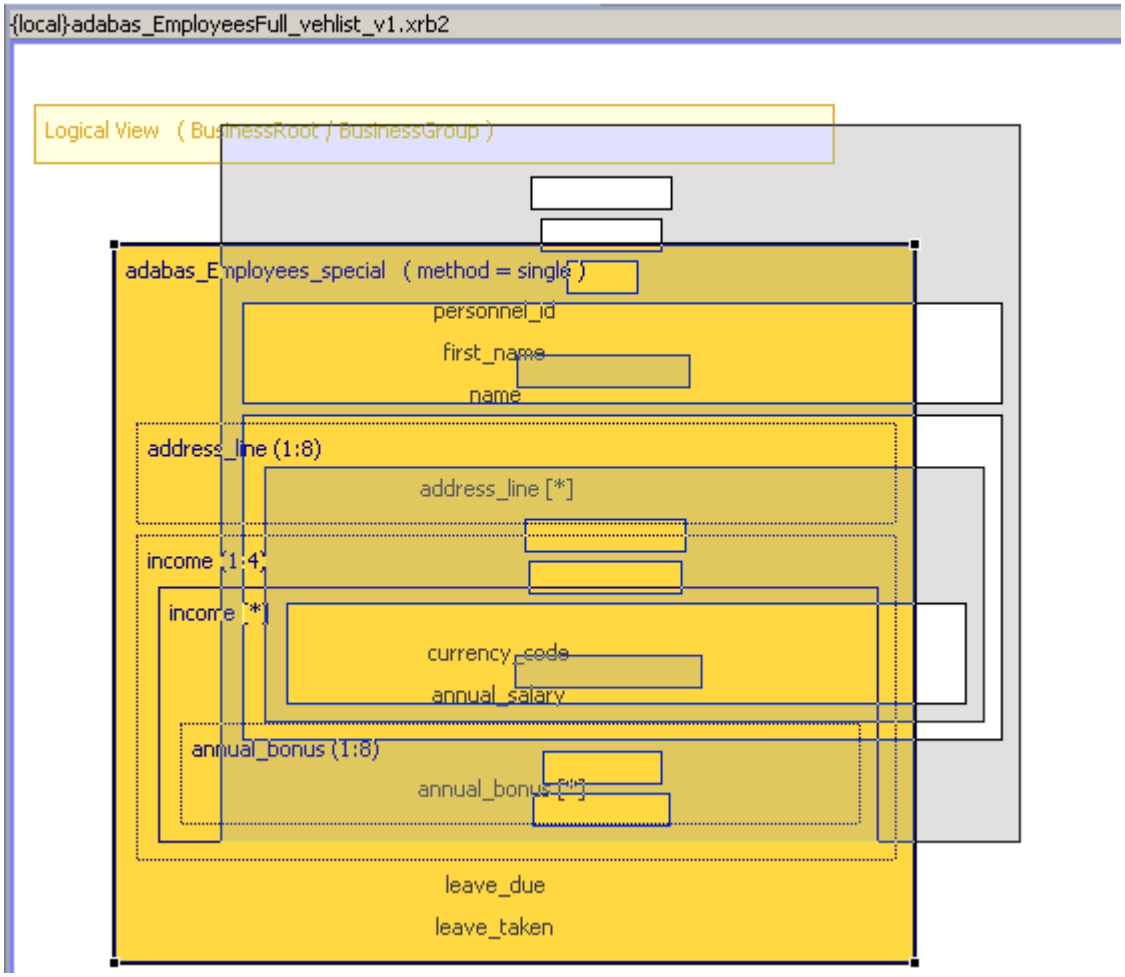

- Select the *adabas_Employees_special* service, click *Import*



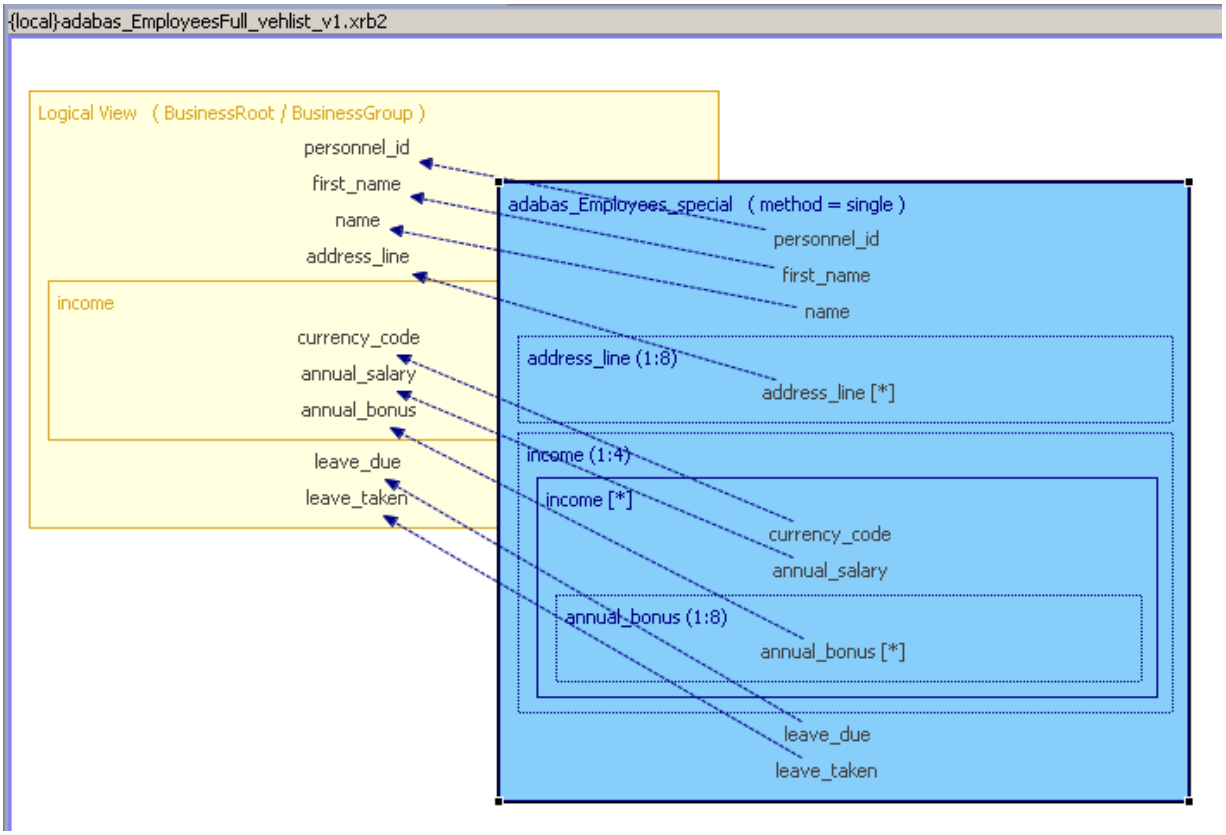

- Again in this example, we wish to use only certain fields in our BDV. Select the items not required and hit the <*Del*> key to remove them. Select the first item and, holding down the Ctrl key select the others. This can be done in batches if so desired. Note again that this is not compulsory to do.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



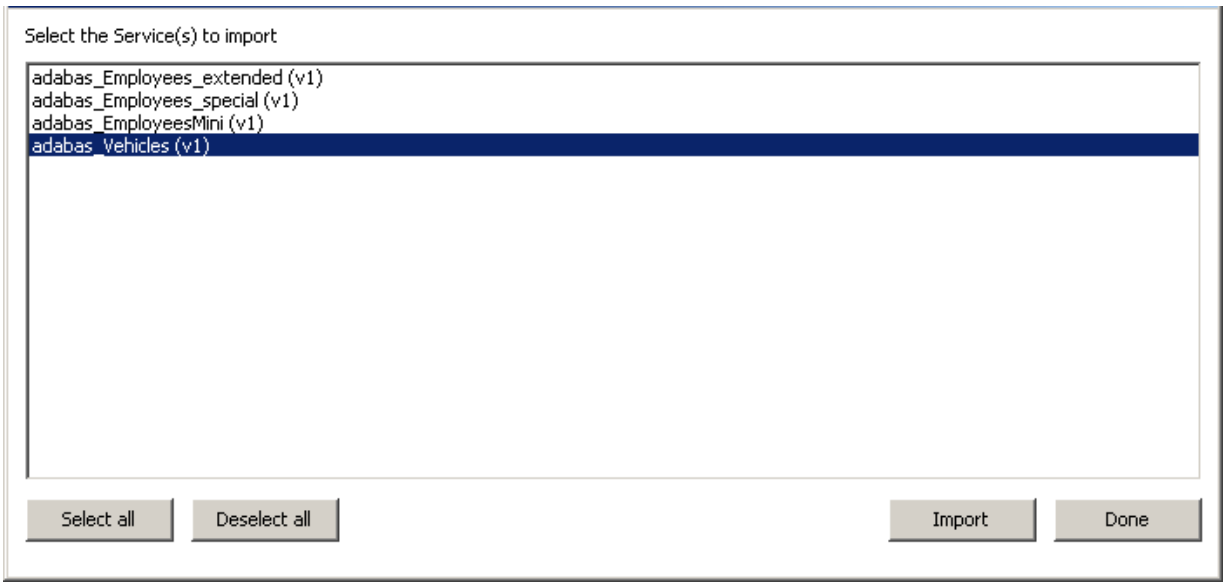

- It is good practice to save frequently (Ctrl+S, select the Save icon, File->Save). Select the entire *adabas_Employees_special* view, drag it to the LV and drop it there.



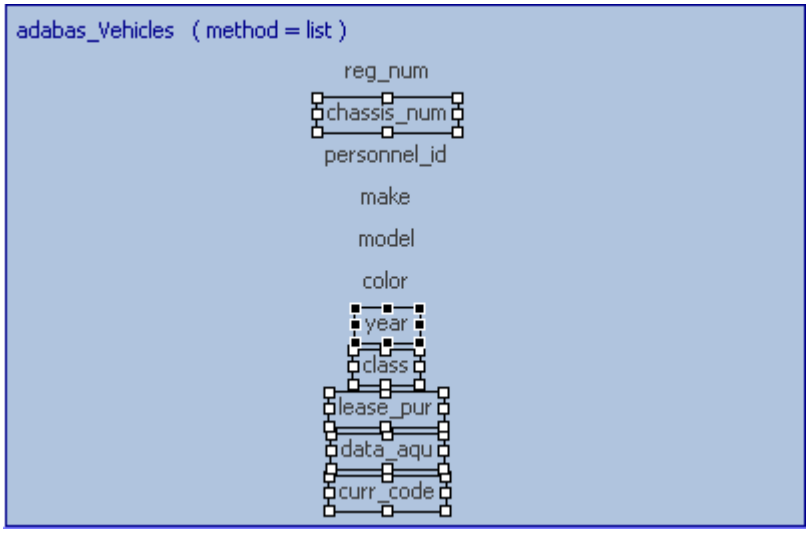

- This will add the fields to the LV, and also create their connections to the source "Physical View" (PV) fields at the same time.



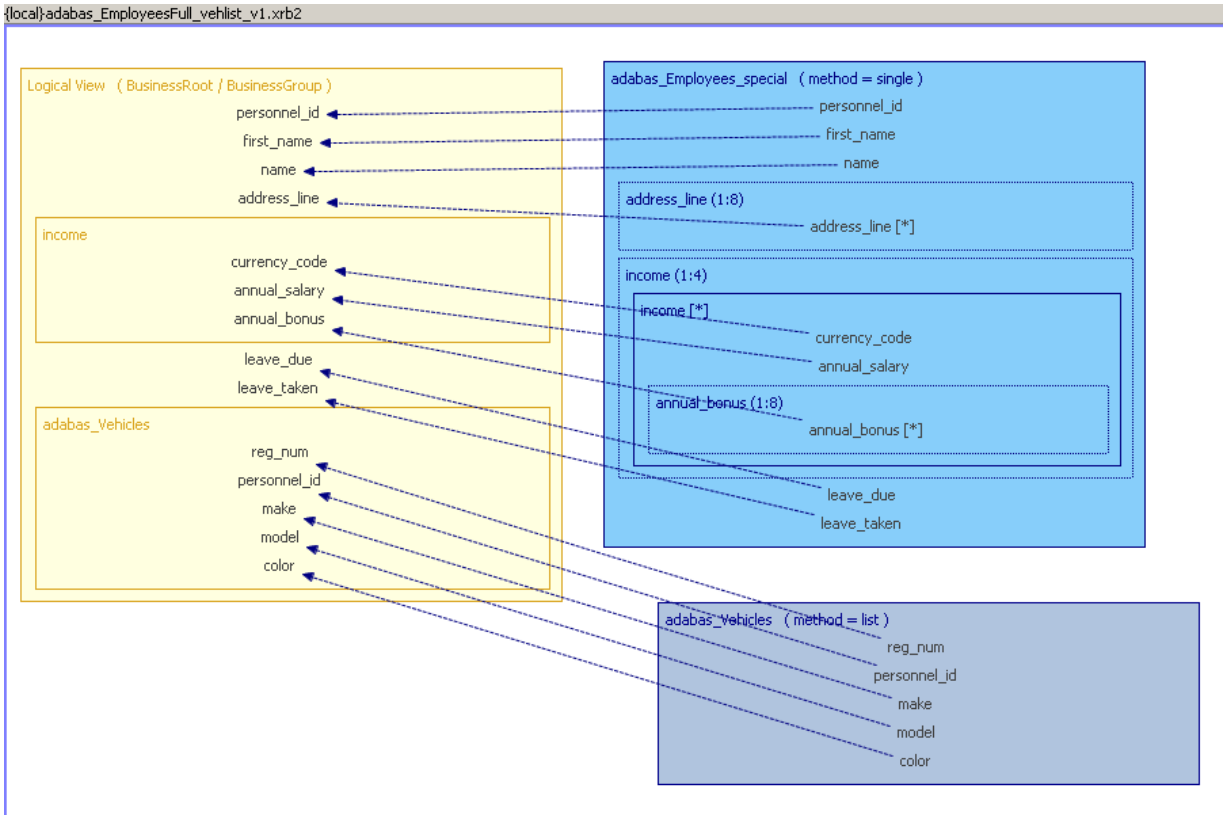

- Import the *adabas_Vehicles* WebService.




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




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



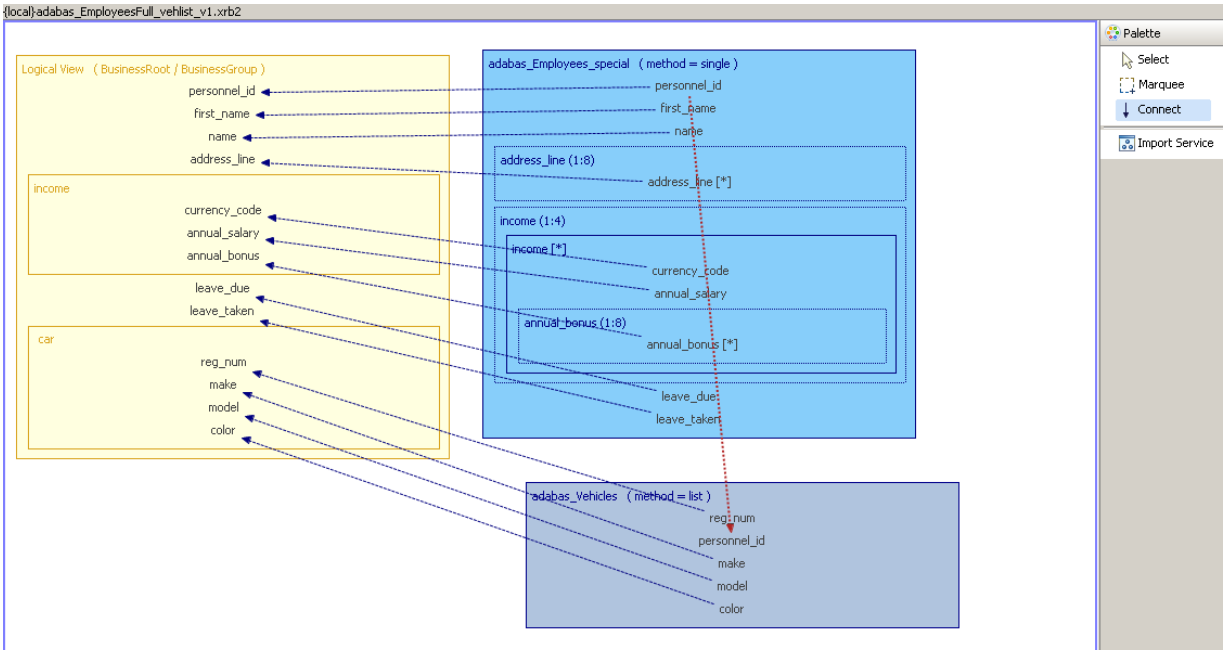

- Note that the elements from the PR have been wrapped in a container or logical group which by default is given the same name as the source PR. This happens automatically when the PR method is set to list. The group name can be changed as follows. Click the newly created group to select it. In the Properties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

Properties

| 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. Therfore 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:

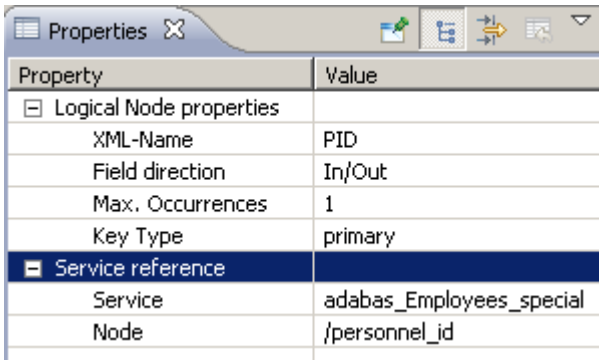



- We have already seen that the LV can be a subset of the underlying data. It can also represent that data in a different format e.g. the fields names can be changed and occurances of fields such as addreess_line can be reduced.

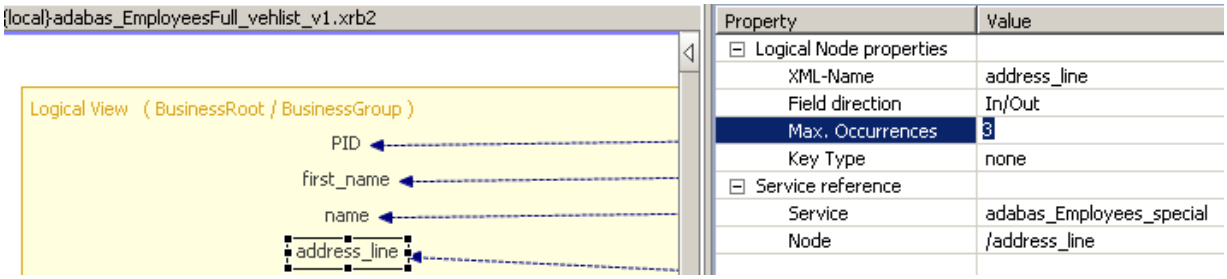
  For these changes select the field and open its Property View.

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

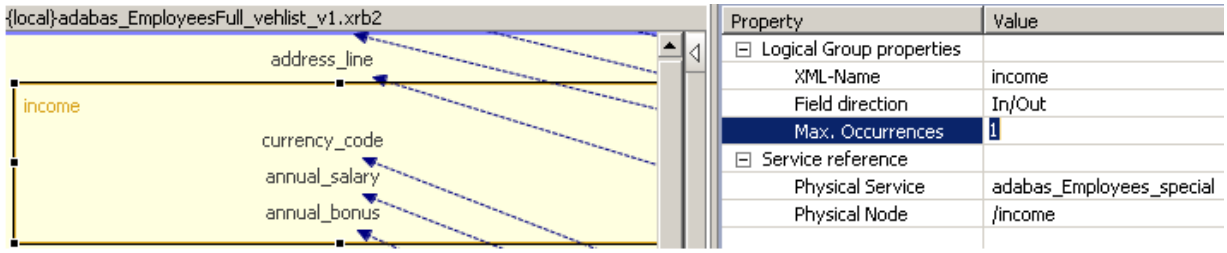
Change the key type for the PID field to *primary* (mandatory).




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



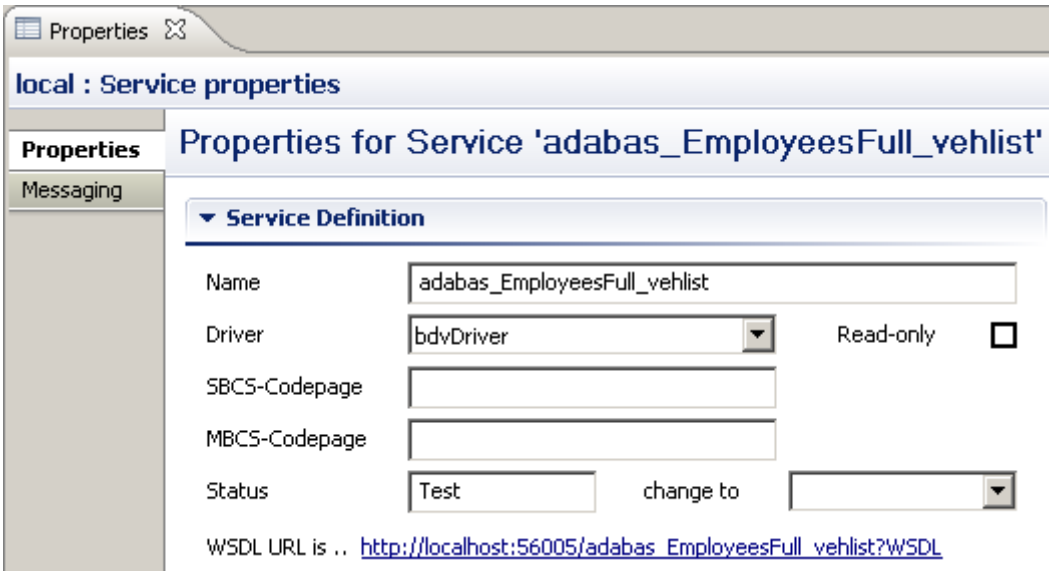

- Change the *Max. Occurrences* property for the *income* group field to 1.



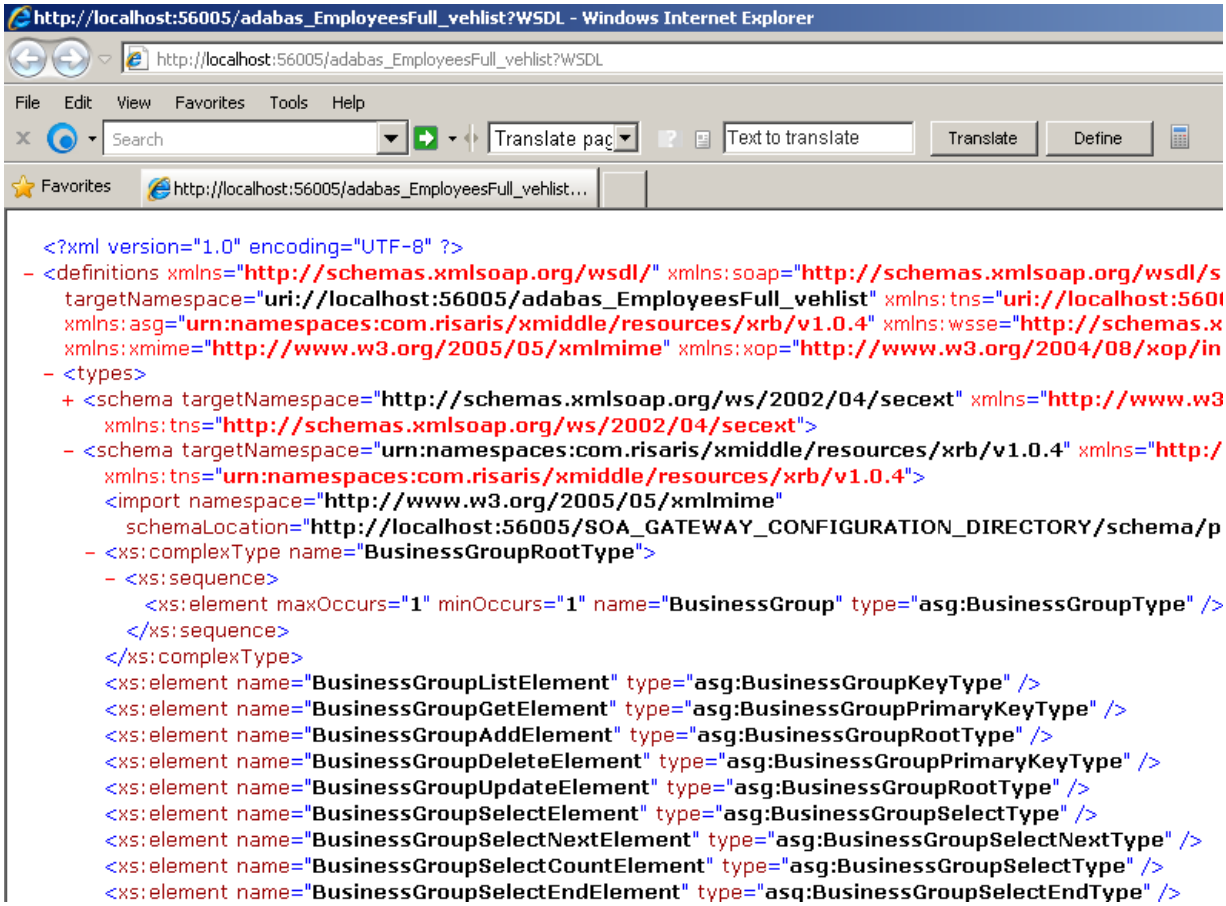

- Click the *Save* icon, or File->Save, or Ctrl + S to save the BusinessDataView on your SOA Gateway server

### Test the WebService

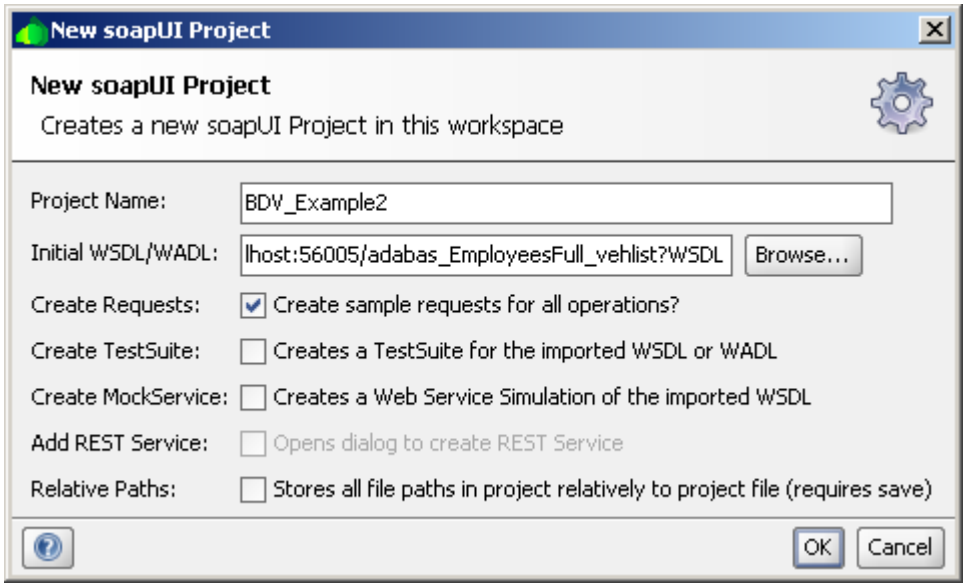
- Click the URI next to *WSDL URL is ...* (This can be found in the Service properties for the BDV just created. Select the web service under bdvDriver as shown and open the Property View.



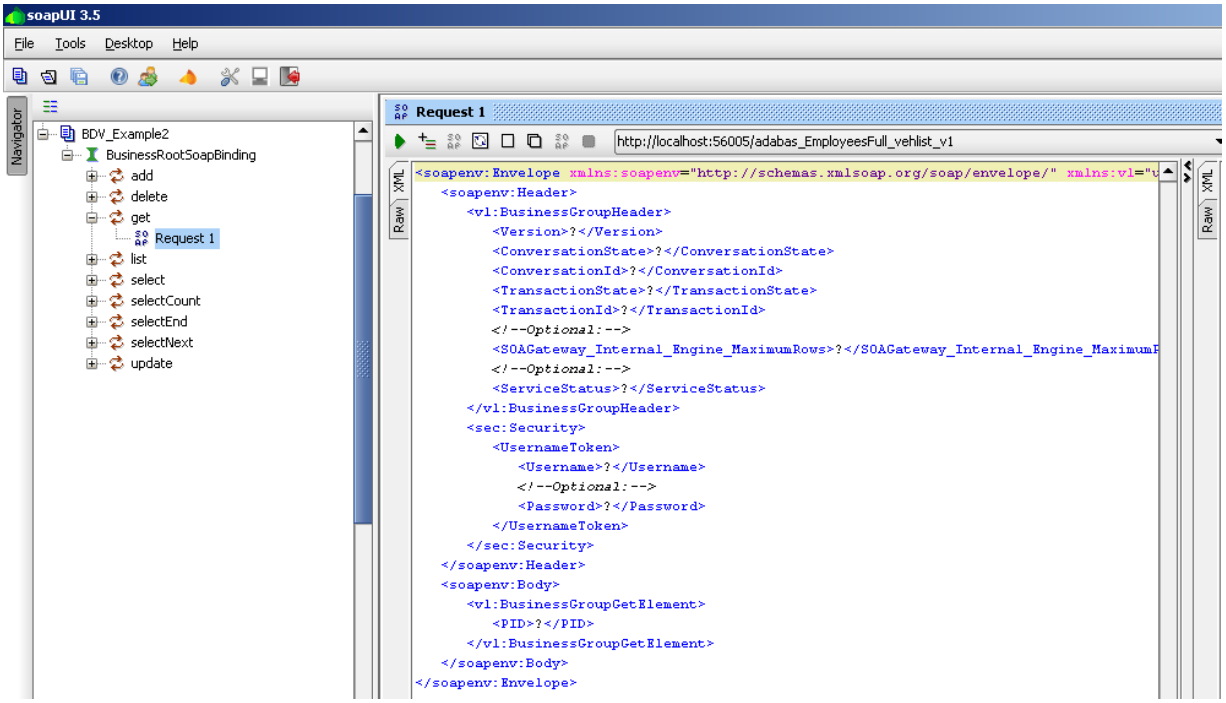

- The WSDL for a WebService based on a BusinessDataView looks just like one for a "basic" (= "physical") WebService, exposing all available access functions (add, delete, get, list, select, etc).



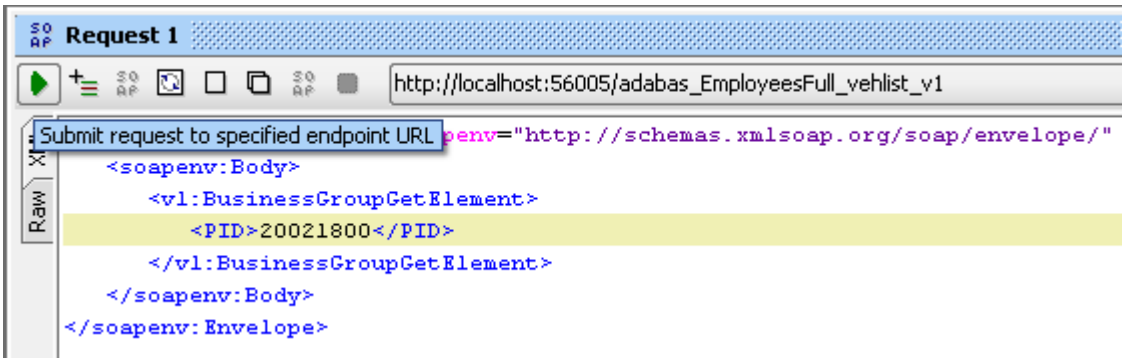

- We will now use *soapUI* (see the Introduction in the Tutorials section) to access the BDV-based WebService. First of all, create a new soapUI project based on the WSDL URL shown above.



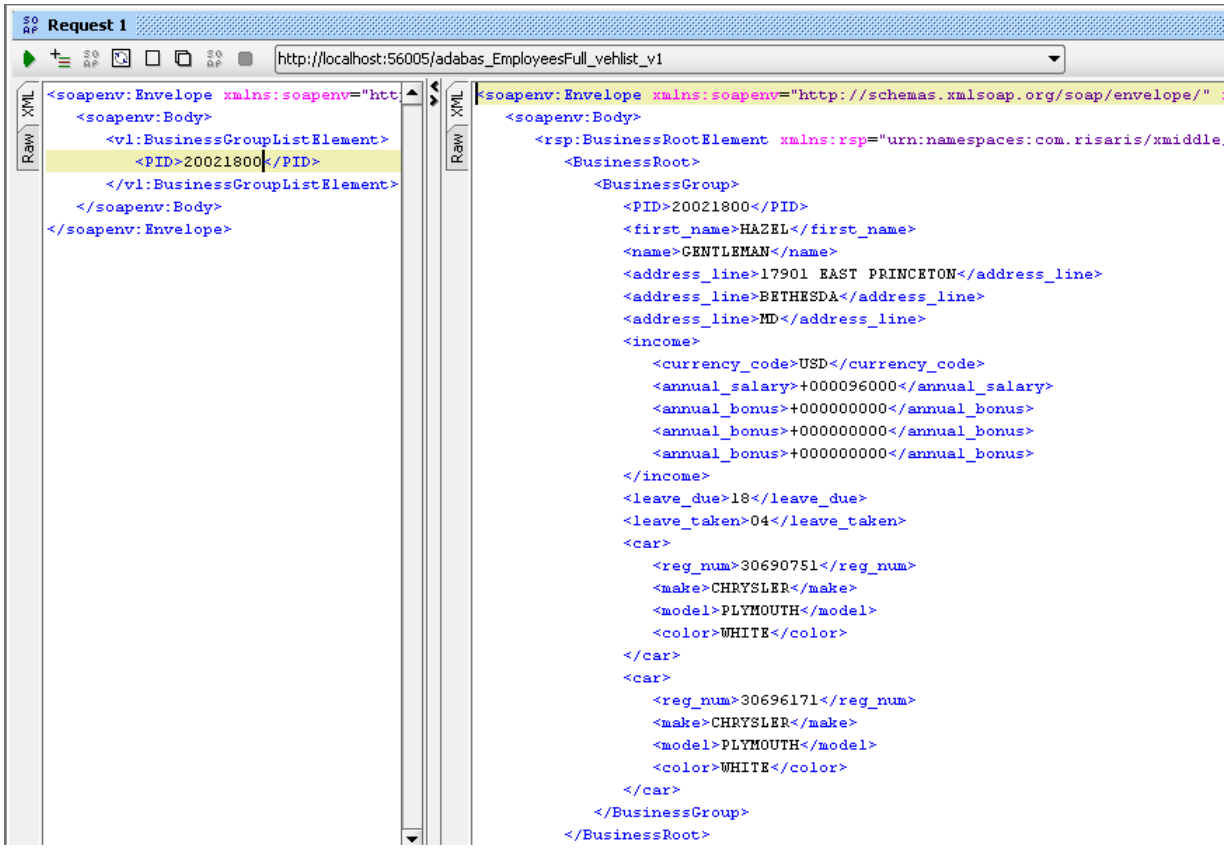
- Double-click the default request created for the *get* method, this opens up the request window.




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




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

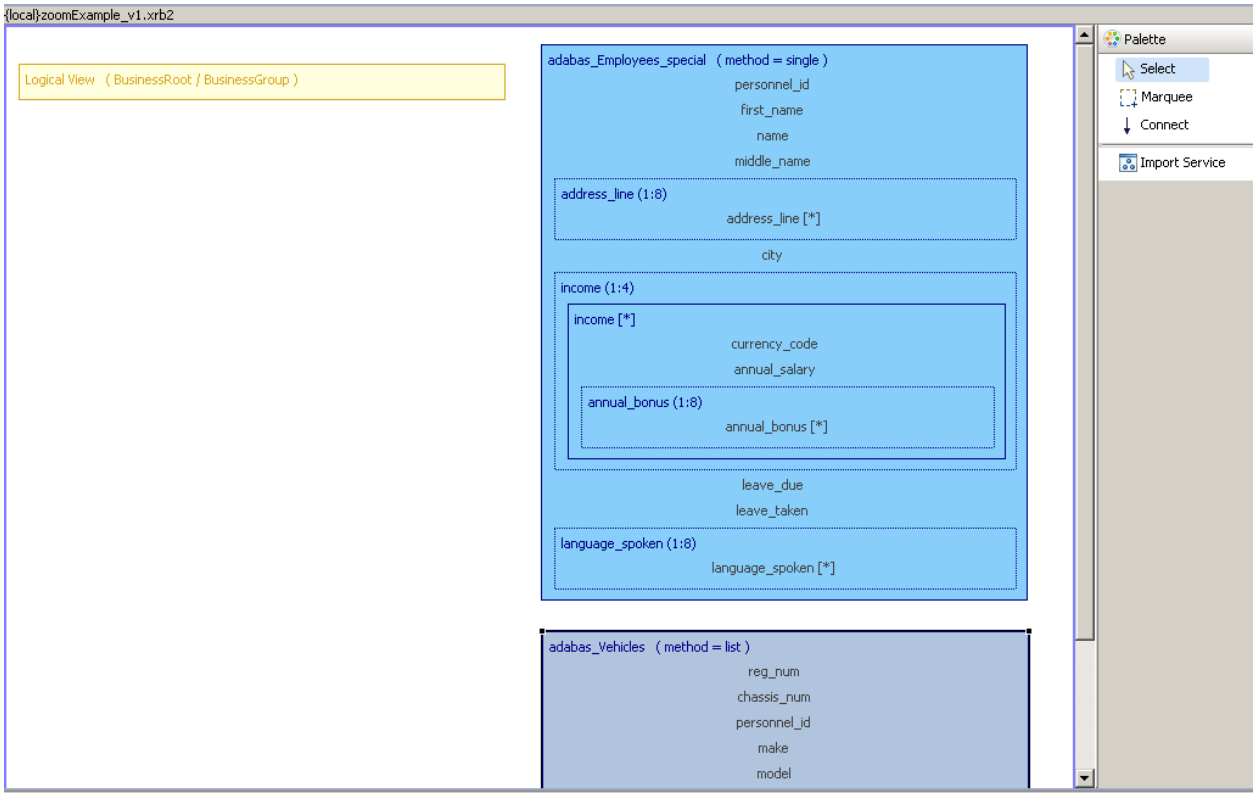



## BusinessDataView Editor hints

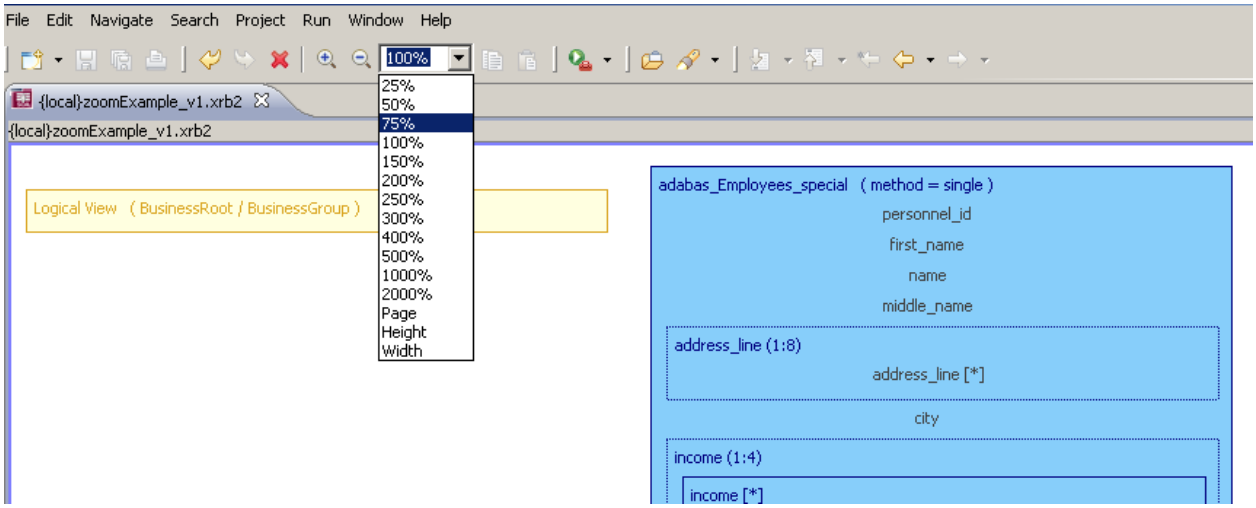
This section provides you with a number of general purpose hints for making your life easier in respect to editing BusinessDataViews.

### Zooming

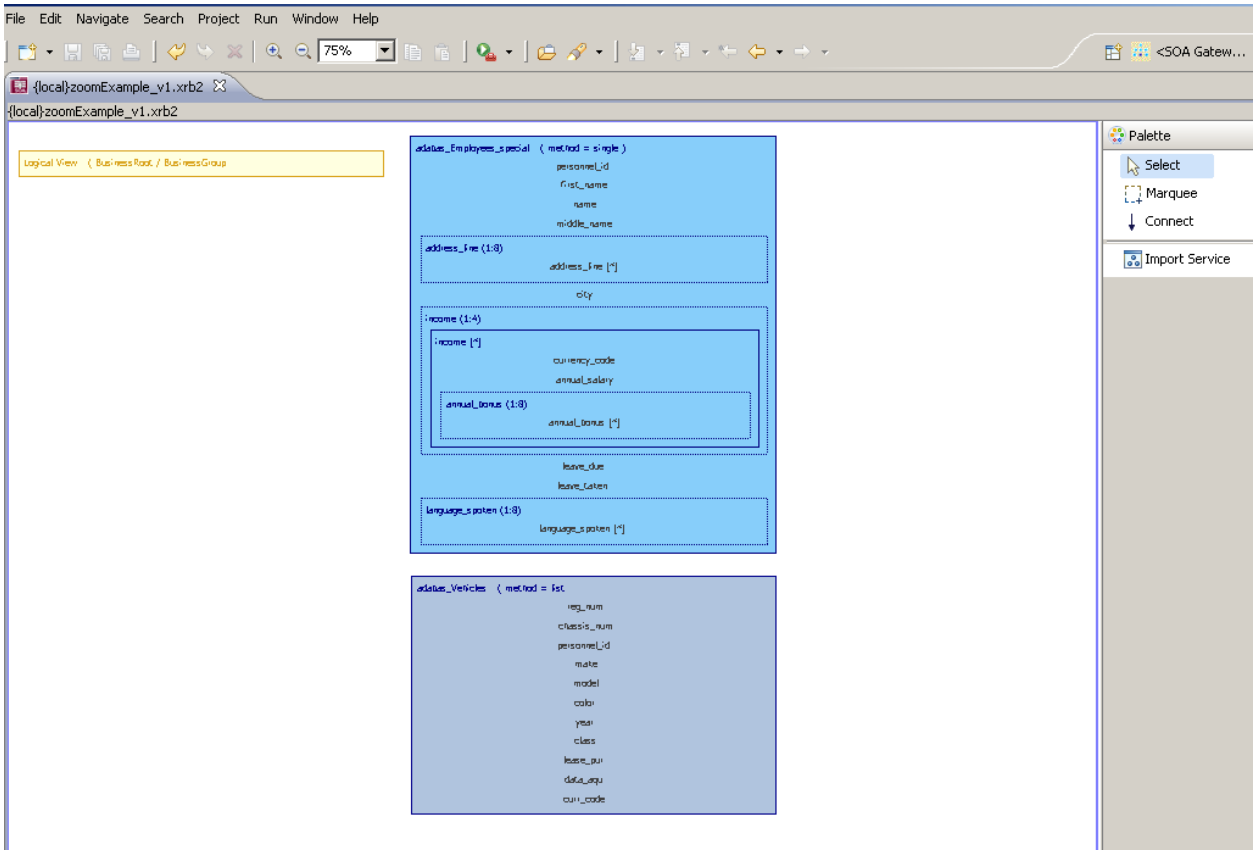
When the BusinessDataView is too big, or the screen is to small to display it in its entirety at a resolution of 100%, or if you want to see some element(s) in more detail, the zoom level can be changed accordingly.



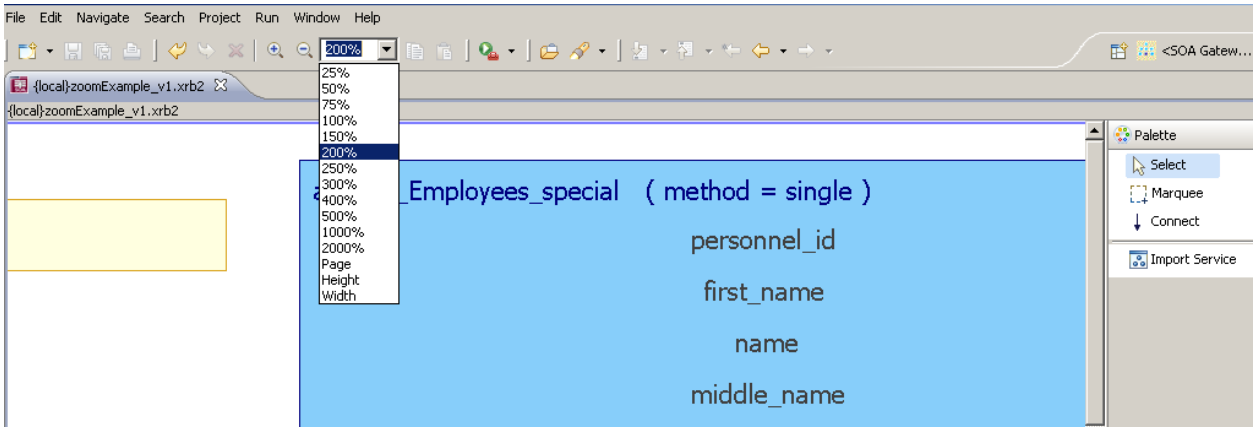

The initial display at 100%, not all details are visible.




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



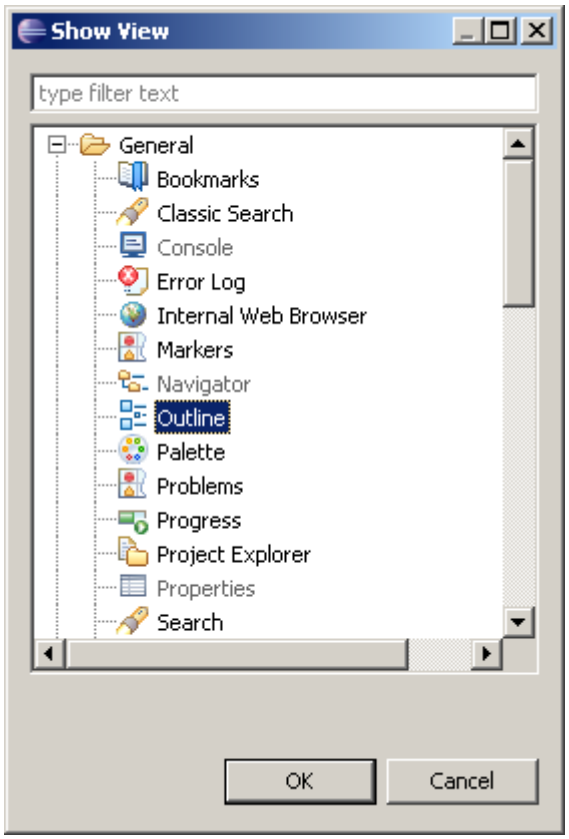

Increase the zoom level to a value above 100% to see more details.



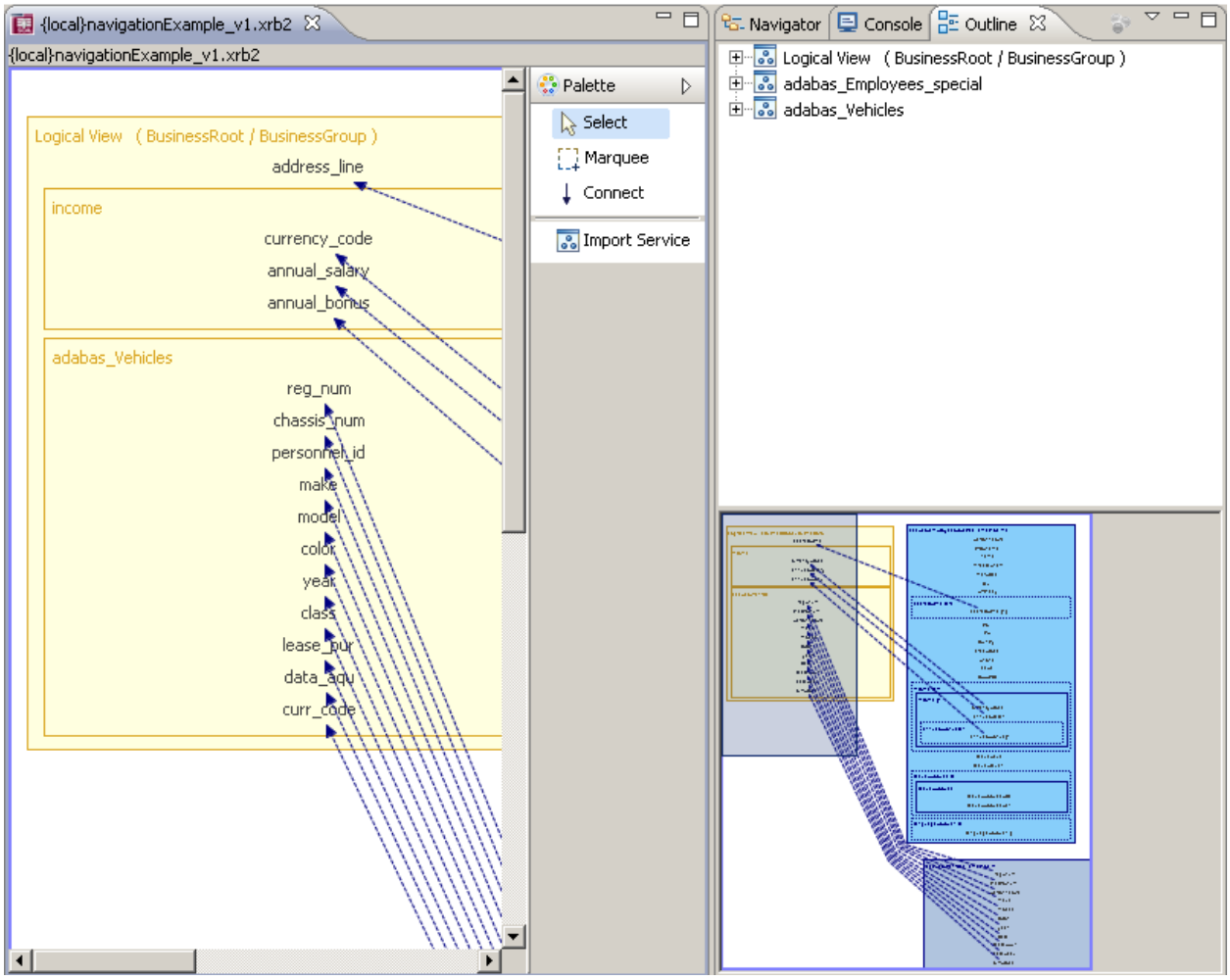
### Making the BDV elements visible

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.

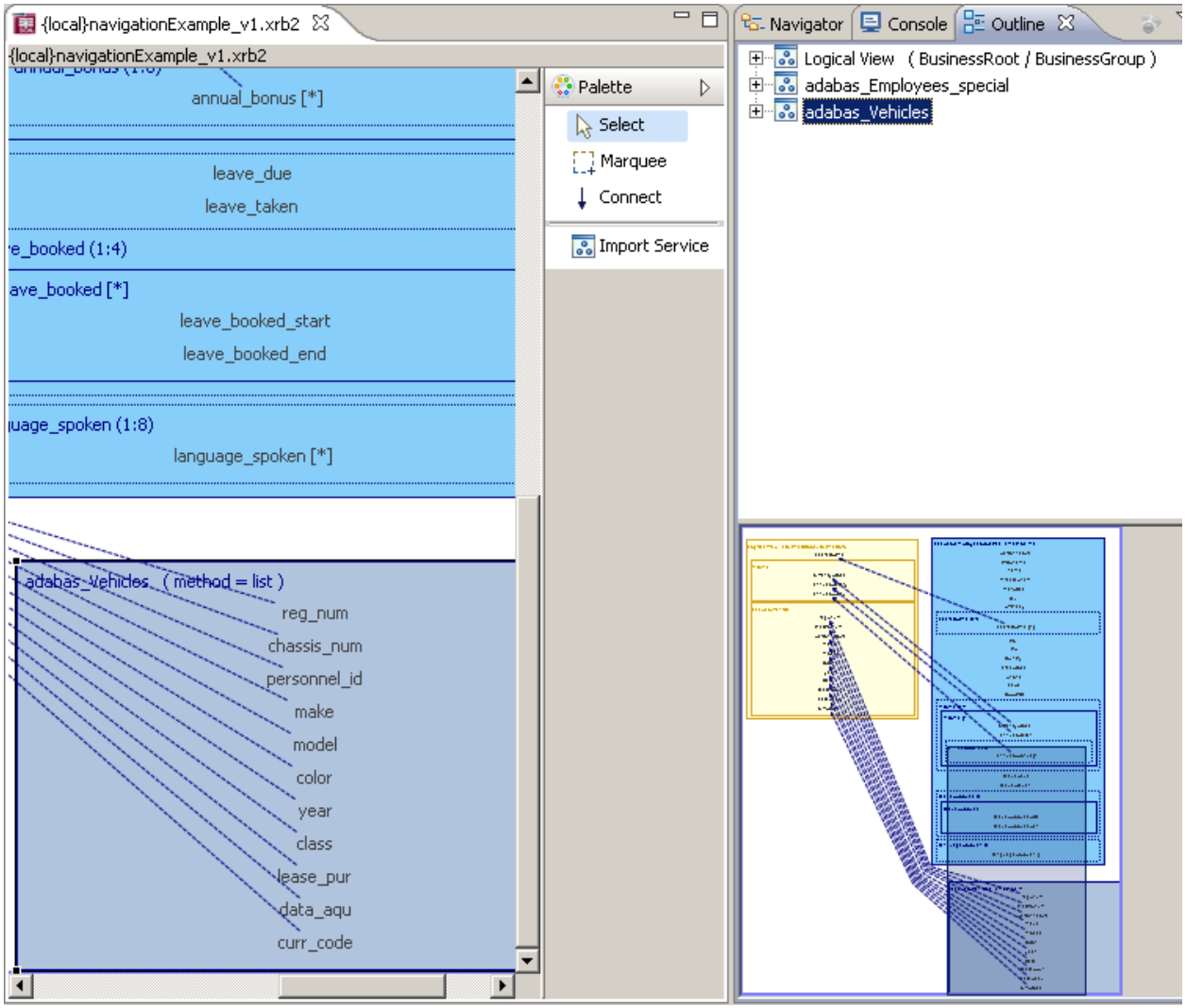
If the Outline view is not visible, from the menu select Window -> Show View -> Other..., select Outline in the General folder and hit OK.



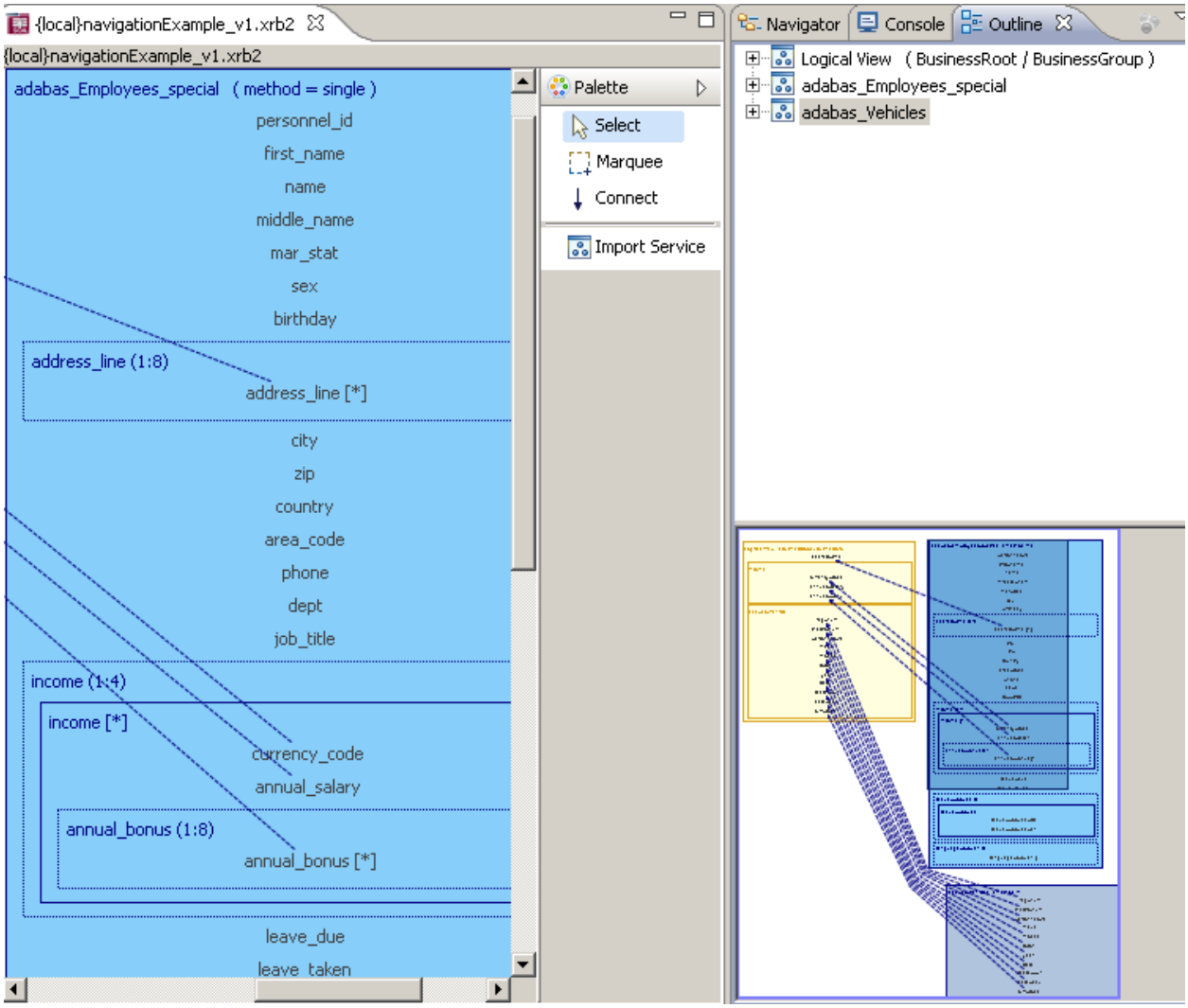

In this example we have a populated LV plus 2 PRs, with the employees PR partially visible. To make the vehicles, select the entry in the Outline view as



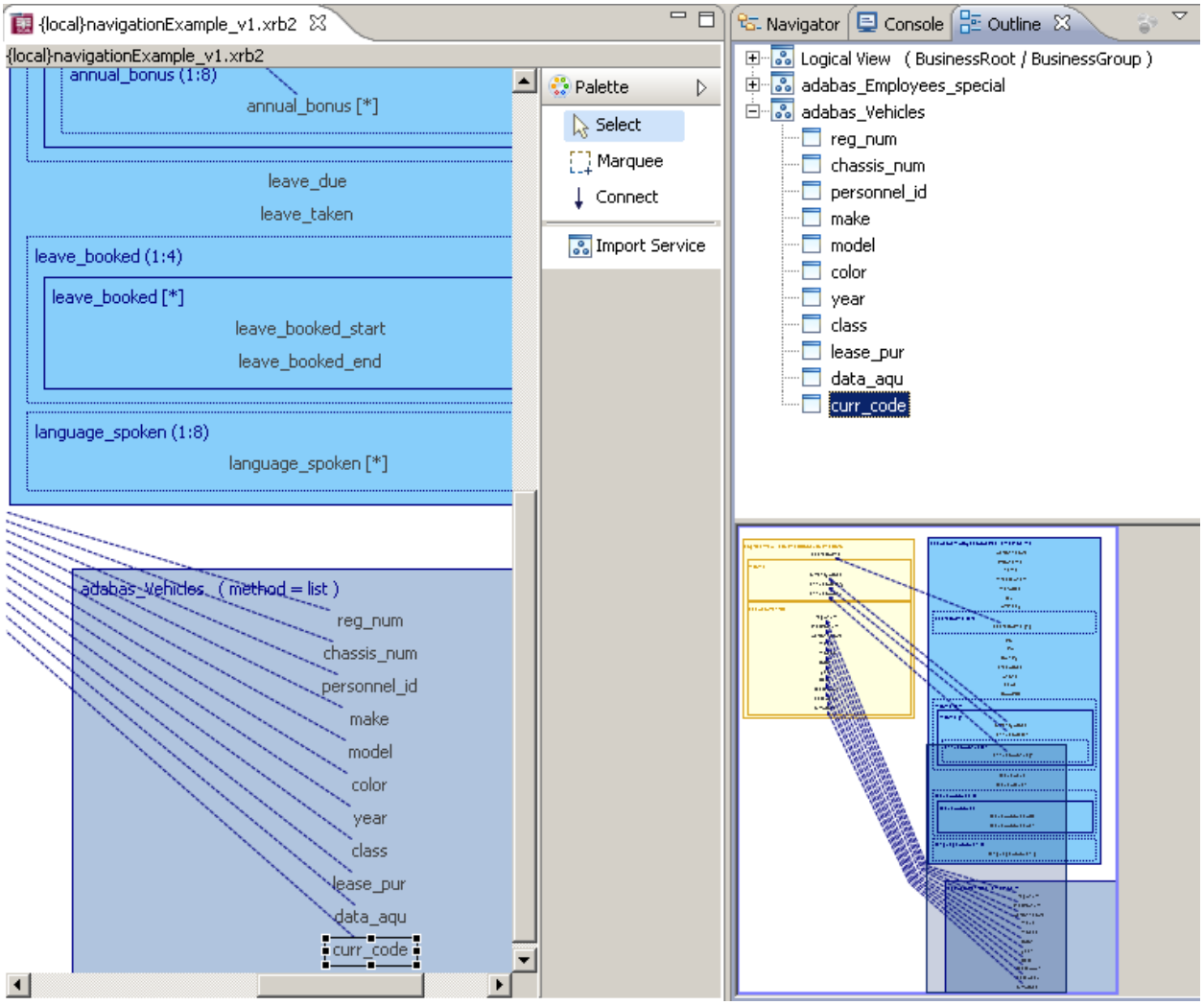

To make the vehicles PR visible, select the entry (adabas_Vehicles in this instance) in the Outline view as shown below. The view will align itself with the PR selected. Notice also how the aerial view in the bottom right-hand corner has also moved the focus to the PR, with the darker blue rectangle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

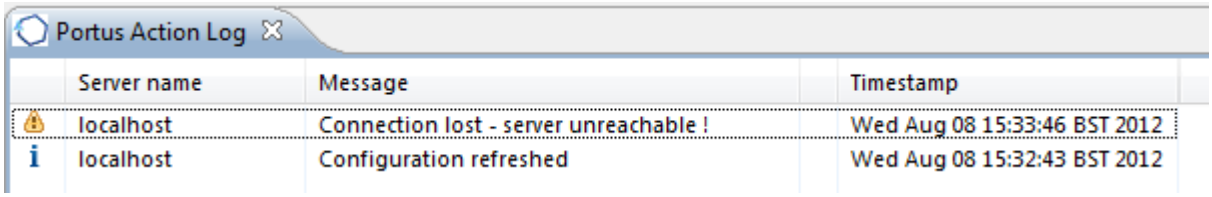

- Working with the Action Log ..... 136

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).

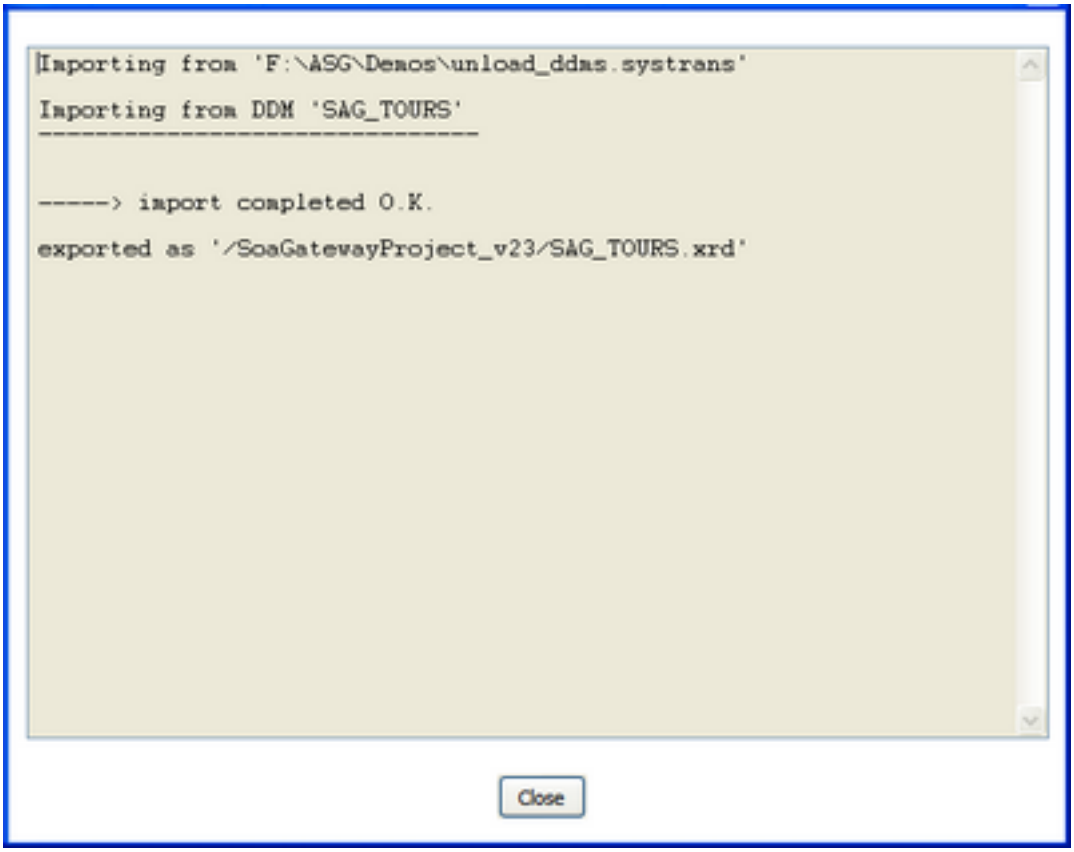
## Working with the Action Log

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.




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




Click the 'Close' button to dismiss the window.

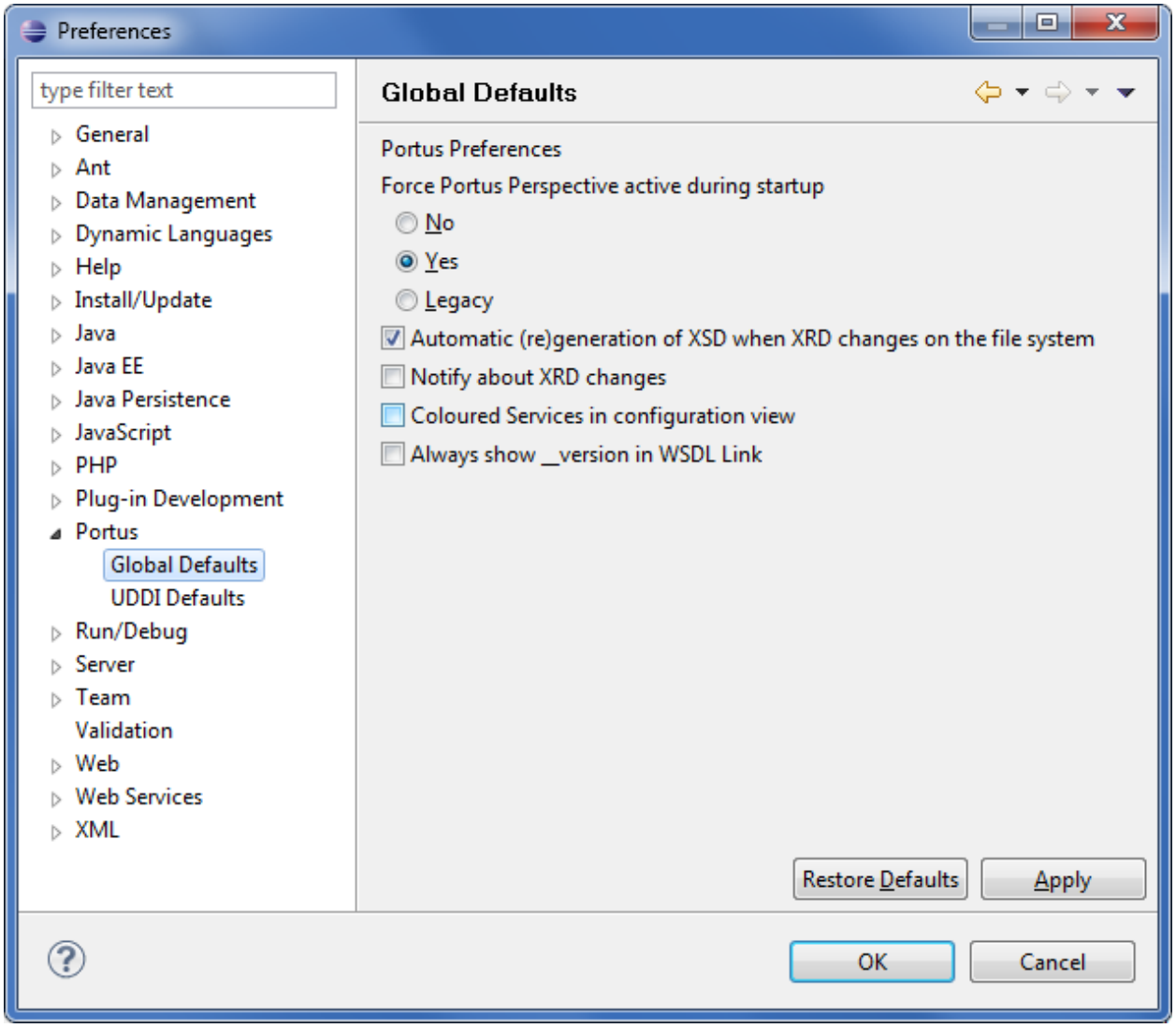
# 17 Setting Preferences for the SOA Gateway Control Centre


- Working with the Preferences ..... 140

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

## Working with the Preferences

Invoke the preferences via Window->Preferences...




- "Force SOA Gateway 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 SOA Gateway Administration perspective will be activated when Eclipse is started
  - *Legacy*: The SOA Gateway "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 SOA Gateway (legacy) Configuration View


- Working with the Configuration View ..... 144
- Filter displayed Services ..... 147
- Edit a Service Definition ..... 148
- Add a Service ..... 148
- Duplicate a Service Definition ..... 150
- Duplicate a Service Definition to other server ..... 150
- Remove a Service from the server configuration ..... 151
- Import a DataView ..... 152
- Export a DataView ..... 153
- Generate Java Wrapper for a Service ..... 154
- Register a Service with an UDDI Business Registry (UBR) ..... 154
- Create a Resource ..... 155
- Delete a Resource ..... 156
- Replicate a Service ..... 157
- Refresh a Service ..... 159
- Import Service Definition files ..... 160

The Configuration View provides context based interfaces to

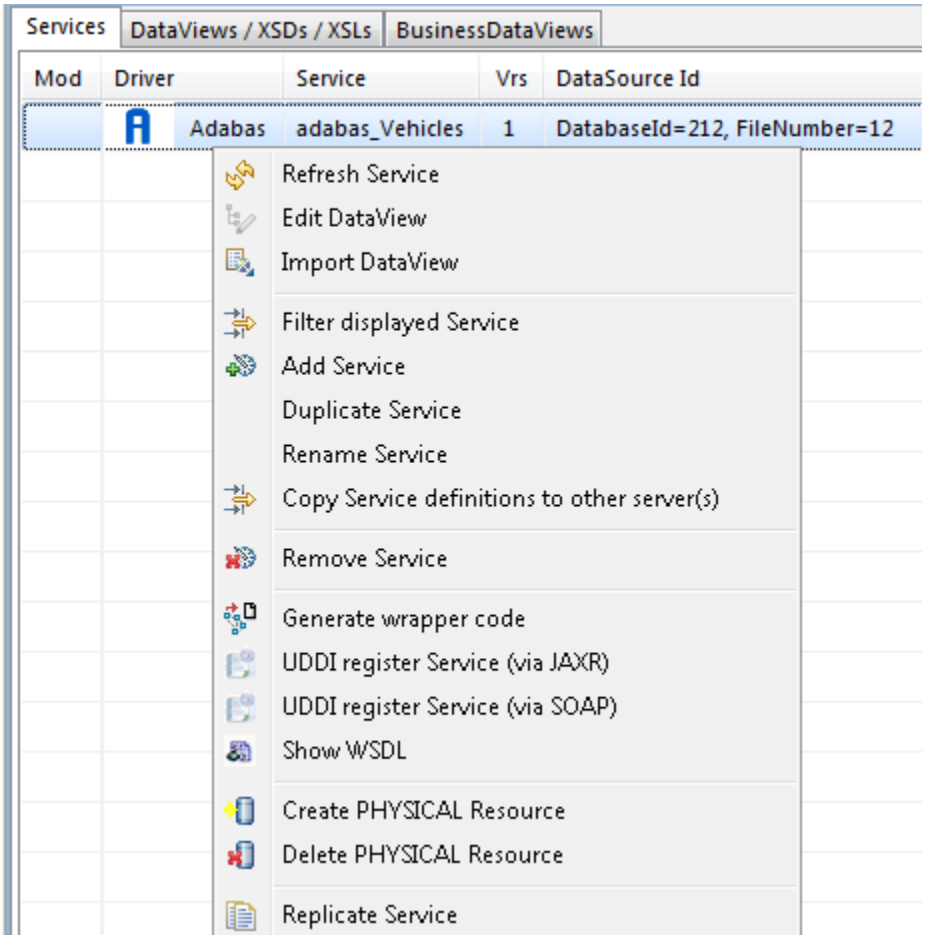
| |
|---|
| All Service related functions. A Service is the SOA Gateway definition exposing an Adabas file, ODBC table etc. to the world.. |
| The DataViews (XRDs) , XML schema files (XSDs) and Stylesheets (XSLs) available on the target server. |
| The BusinessDataViews (BDVs) available on the target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

**Note:** The Configuration view is part of the SOA Gateway "legacy" perspective. This perspective is superseded by the new "SOA Gateway 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 SOA Gateway 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.

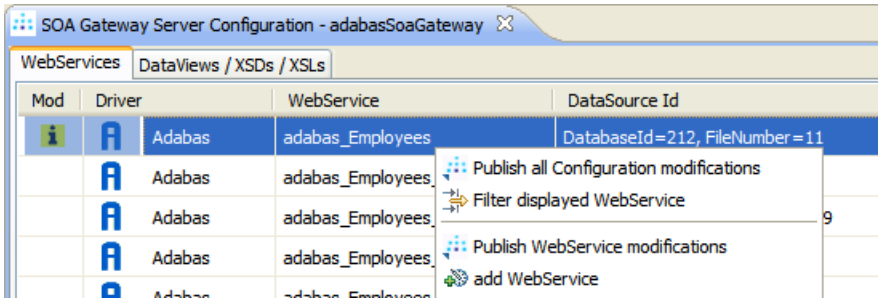



Note: All functions will make 'published' changes to Service(s) available within a running SOA Gateway server immediately, but these changes are only preserved over a server restart when the configuration is either saved manually, or the server is defined with the "autosave" option.

| 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 SOA Gateway Server Configuration |

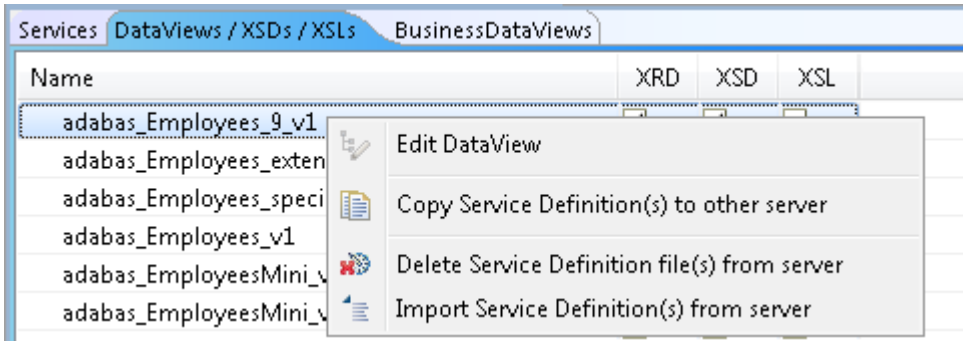
| Function | Description |
|---|---|
| **Generate wrapper code** | Generate a Java wrapper, based on the Axis2 framework, for a Service. |
| **UDDI Register Service (via JAXR or SOAP)** | Register a datasource - a Service - with a UDDI Business Registry |
| Show WSDL | Opens the WSDL for the selected Service in a browser window. |
| **Create PHYSICAL Resource** | Create a Resource (Adabas file), physically, based on the SOA Gateway Definitions. |
| **Delete PHYSICAL Resource** | Physically delete a Resource (Adabas file etc.) pointed to by a SOA Gateway 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**.

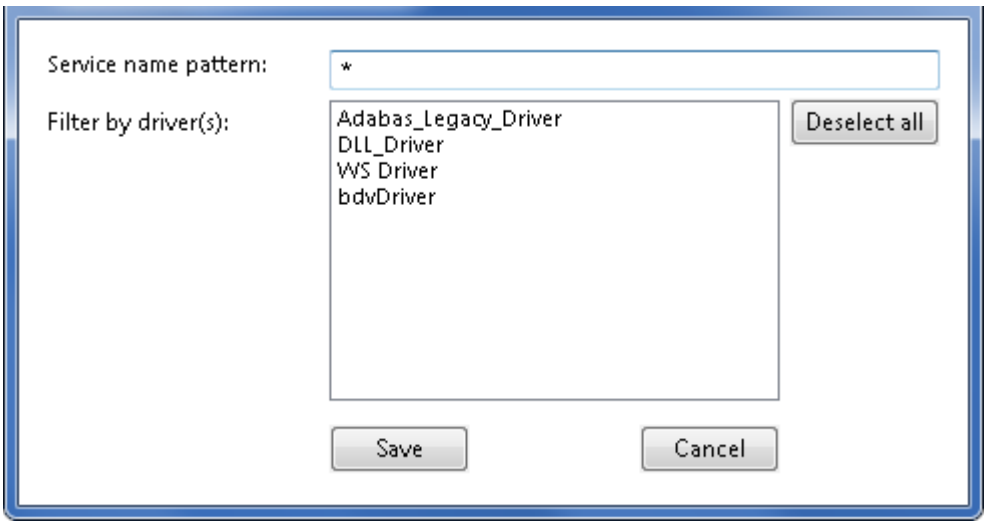


## Filter displayed Services

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 mathcing 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

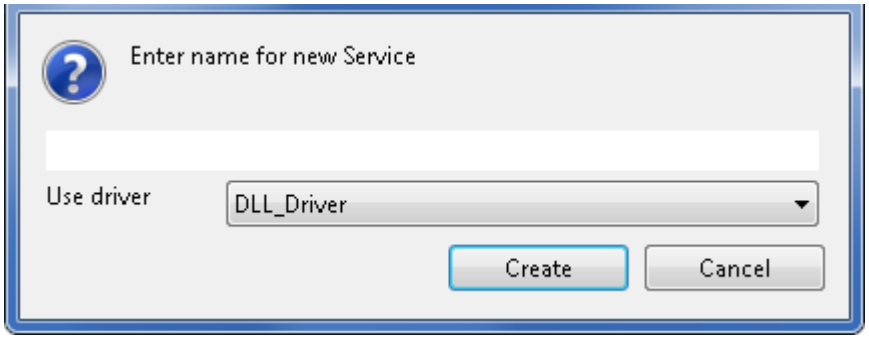
Click on the Service name in the Configuration View, modify the Service definition in the Properties View. Modified Services will be indicated by an information icon in the 'Mod'(ified) column. Modifications stay local until one of the **publishing functions** is executed.

**Important:** Changing the Service Name will rename the Service.

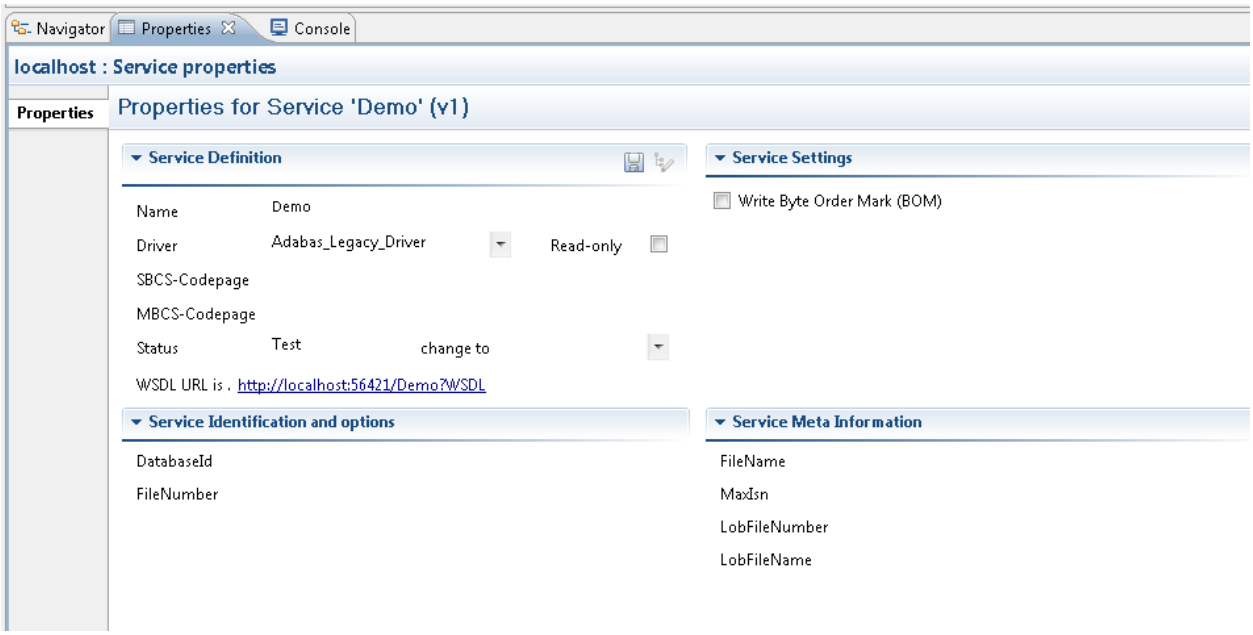
## Add a Service

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.




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 SOA Gateway:

1. Service Definition:

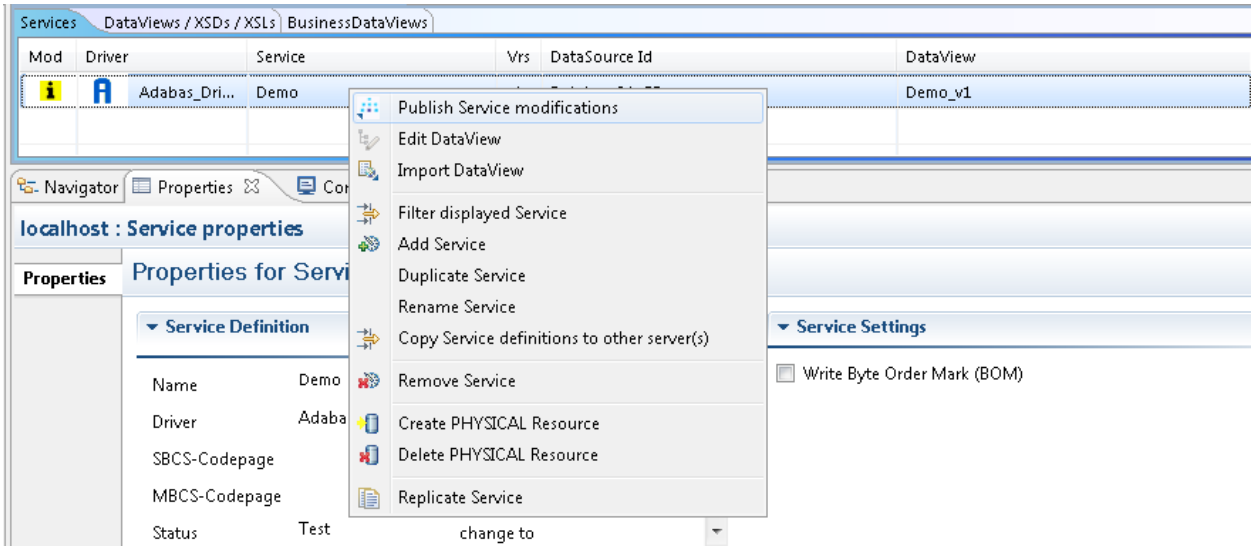
   Name: The Service name which is used to identify the Adabas file to the outside world.

   A Service is accessible by its URI http://<your_server>/<Service_name> When retrieving the WSDL for a SOA Gateway 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 SOA Gateway 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.

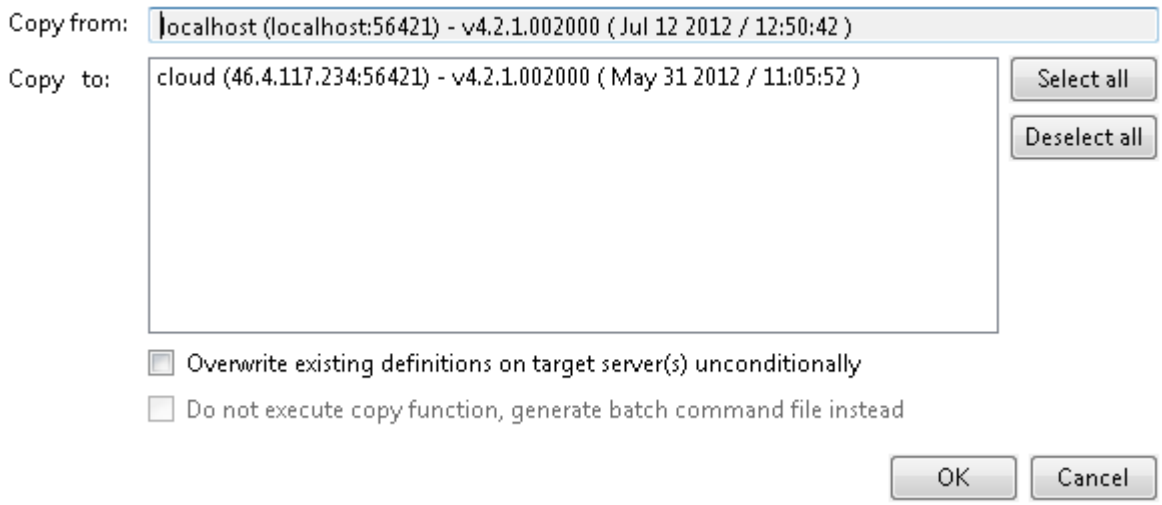
## Duplicate a Service Definition

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**.



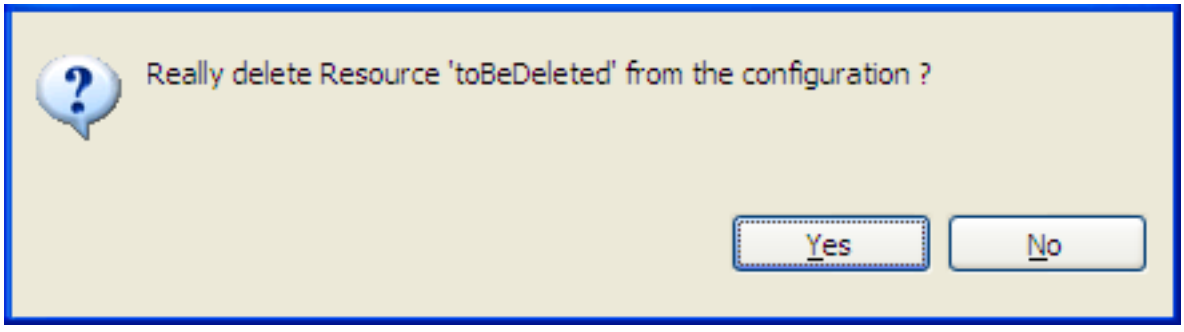

Successful execution of the copy operation will be indicated by a message in the SOA Gateway Action Log

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 SOA Gateway Action Log entry.

## Remove a Service from the server configuration

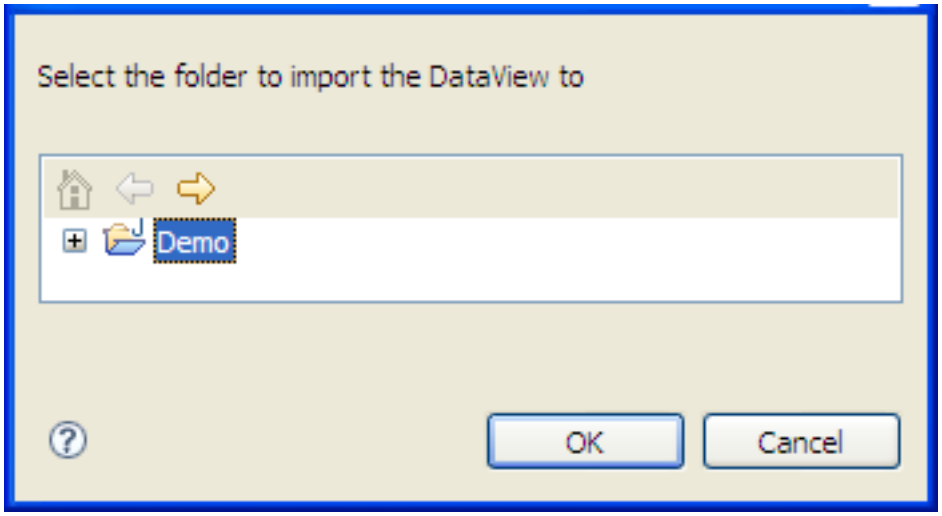
Select the **Remove Service** function from the context menu to delete a Service from the server configuration. You will be requested to confirm the deletion:




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

## Import a DataView

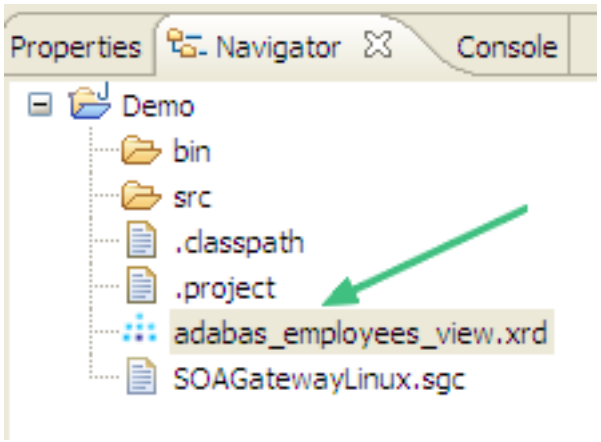
Select the **Import DataView** function from the context menu to import the physical layout description (DataView) into the Eclipse workspace for editing, archiving etc.




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

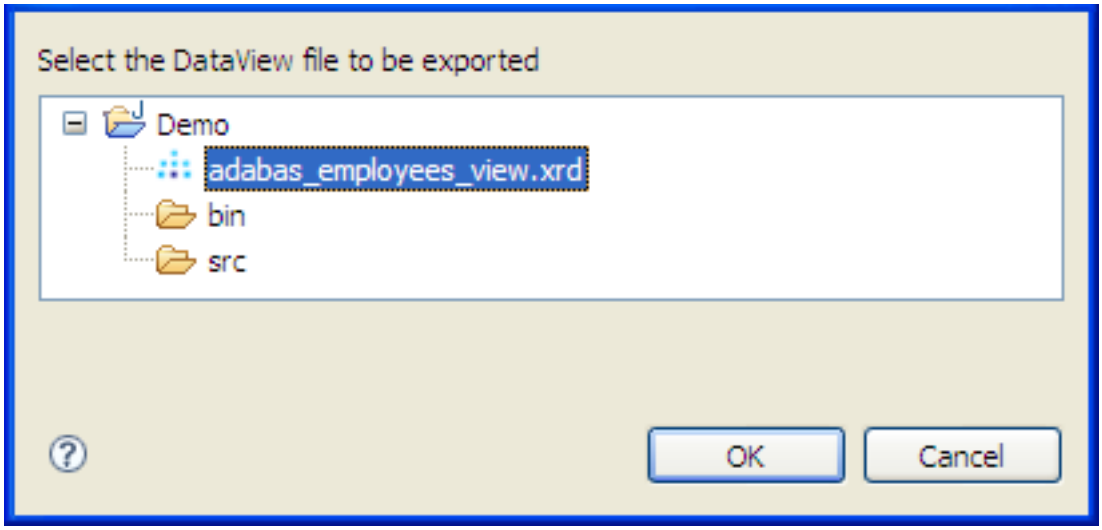
The import will be confirmed by a message in the Status pane.

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



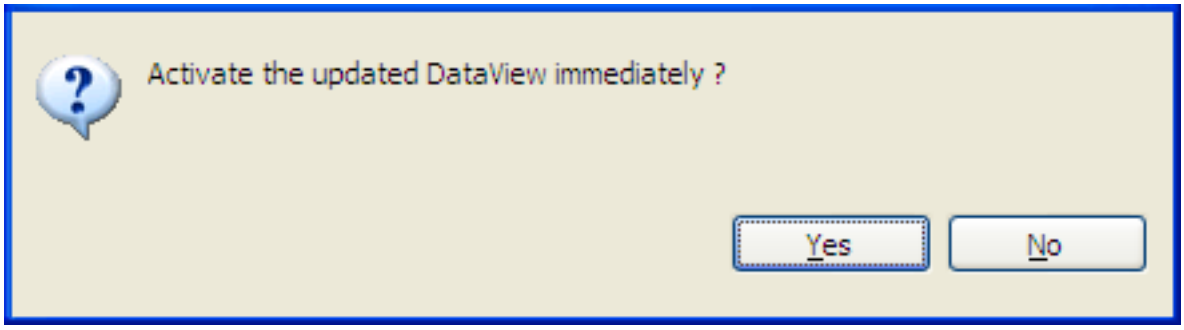
## Export a DataView

Select the **export DataView** function from the context menu to export the physical layout description (DataView) to the SOA Gateway 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 SOA Gateway 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

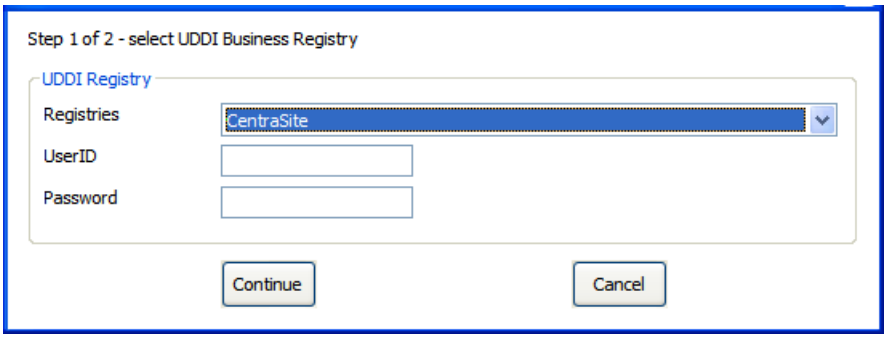
Java wrapper/stub classes are generated using the **Apache Axis2** feature **WSDL2Java**.

## Register a Service with an UDDI Business Registry (UBR)

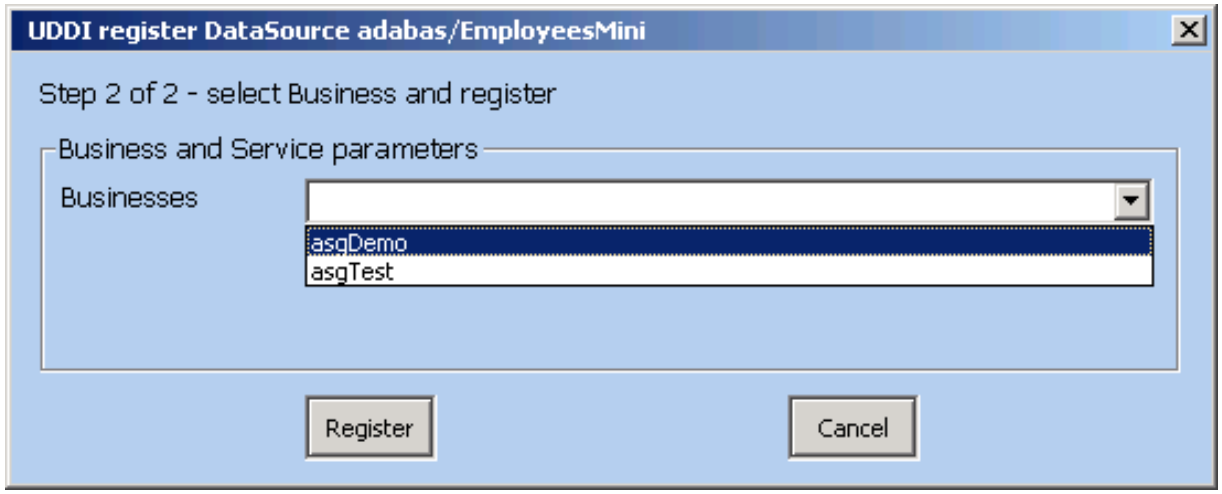
SOA Gateway Services, which are essentially "Services", can be registered with an UDDI Business Registry, for example Software AG's Centrasite, so that any UDDI enabled client application can find an exposed service, retrieve its signature, issue requests agains it etc.

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

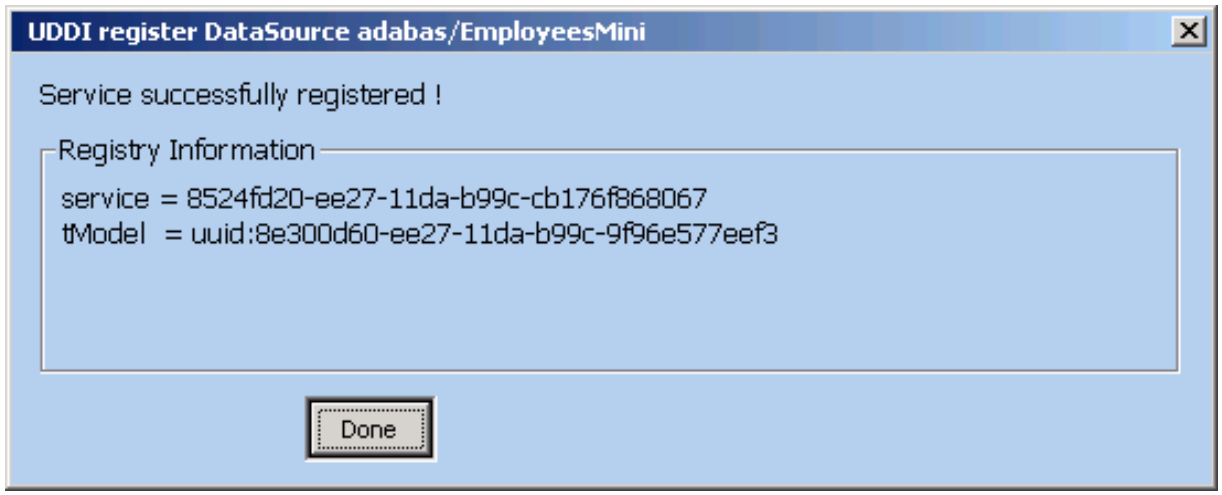
1. Define a UDDI Business Registry to the SOA Gateway Control Centre (from the SOA Gateway Eclipse Preferences dialog)
2. Select the **UDDI register Service** function from the context menu
3. Select one of the predefined UDDI Business Registries, enter the user credentials required to be able to publish




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

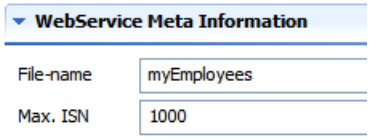



6. Click **Done**

## Create a Resource

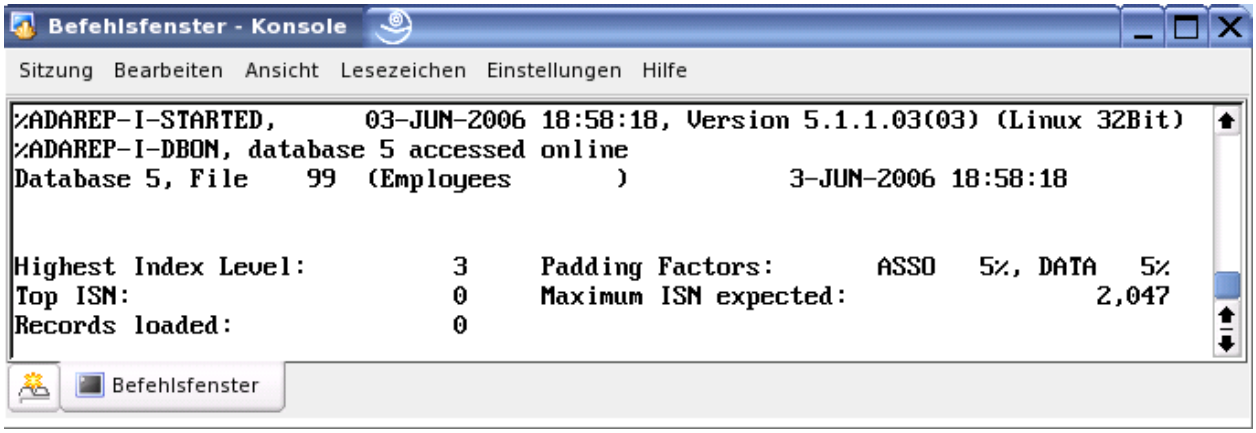
A physical Adabas file can be created from a SOA Gateway Service definition.

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.




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

The ADABAS file has been created on the target database.




## Delete a Resource

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

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

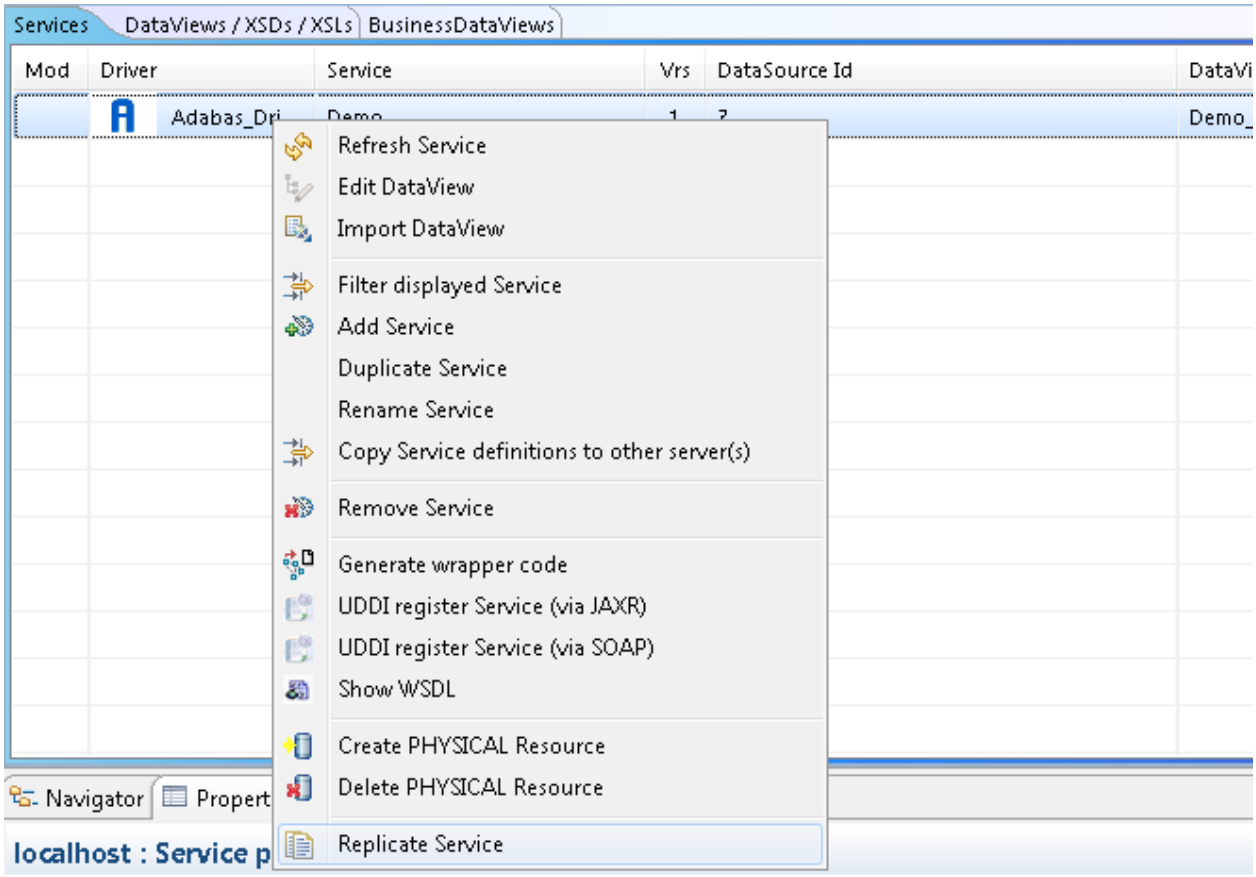
As the Resource (the Adabas file) will be deleted physically from the database, you will be asked to confirm a second time.

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

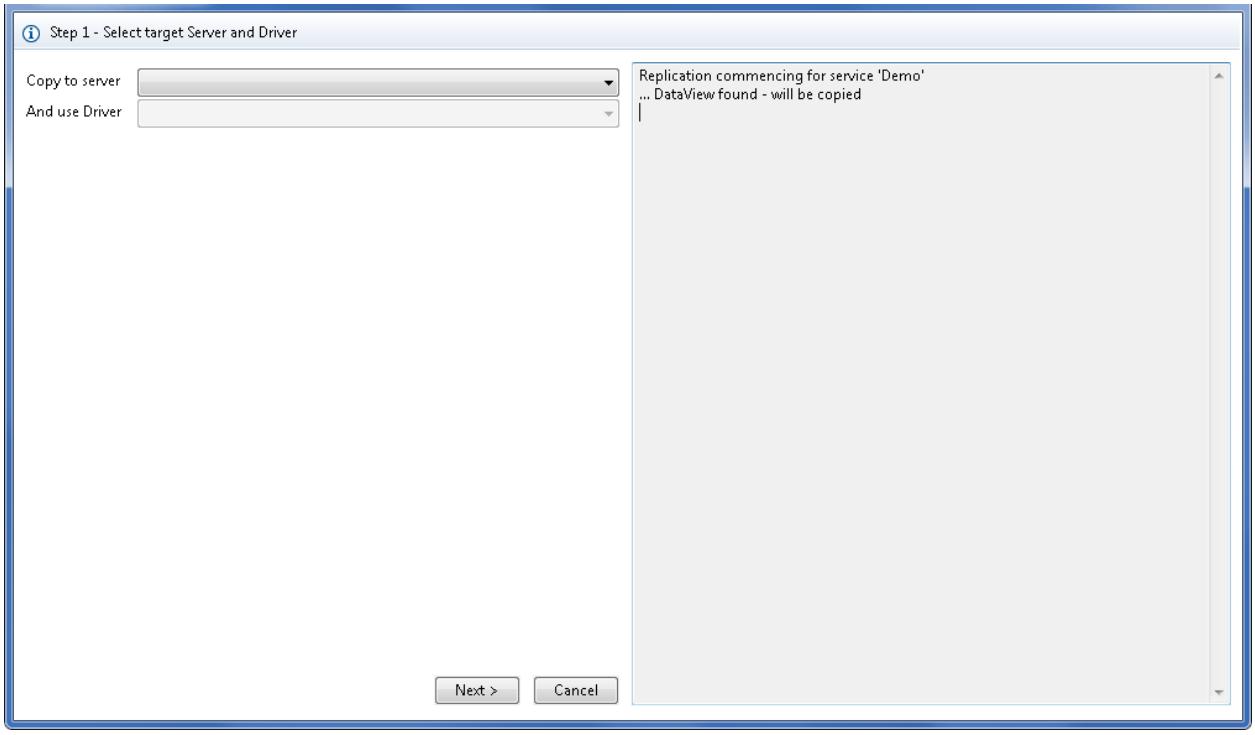
## Replicate a Service

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

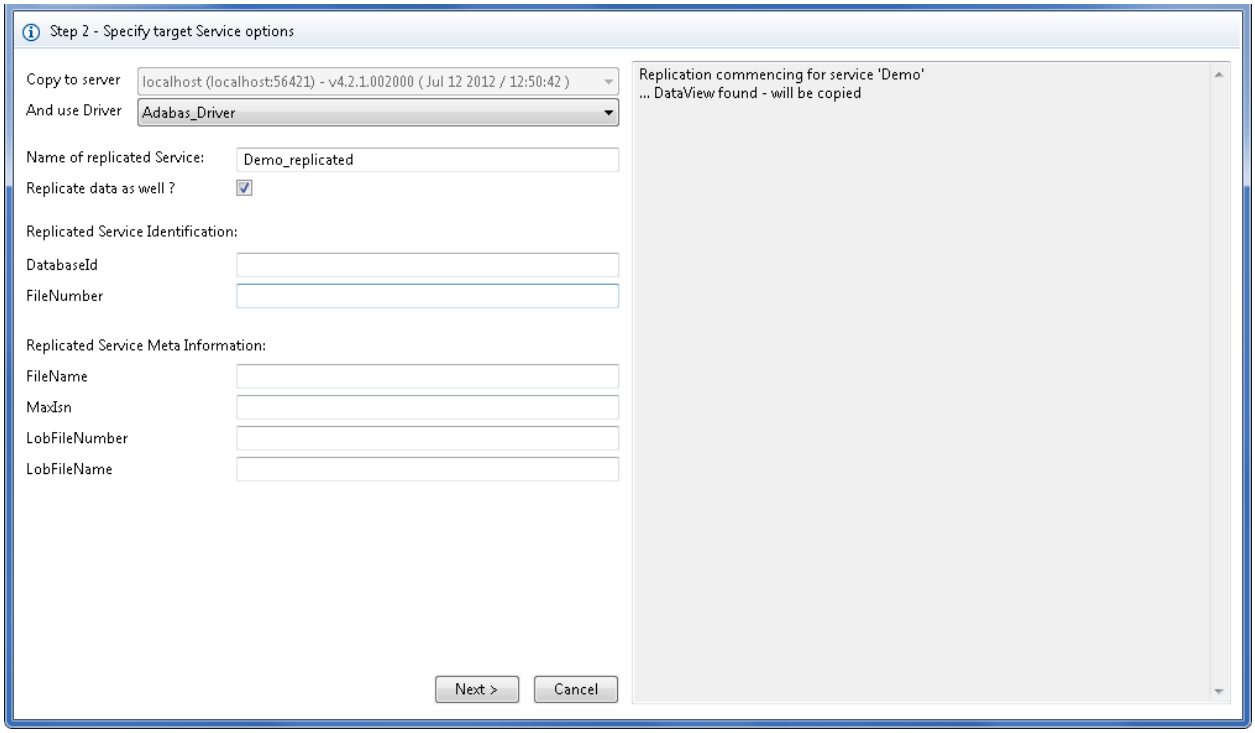
Select the **Replicate Serice** function from the context menu



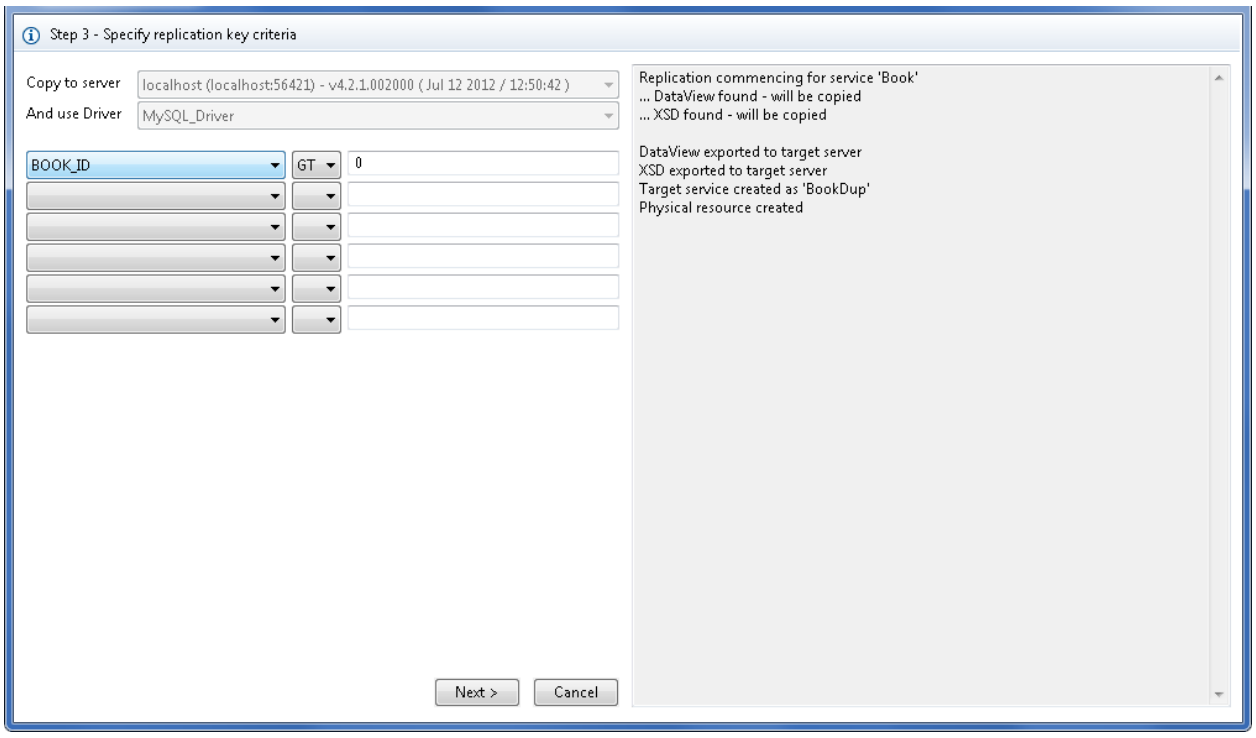

You will be asked on which server the copy will be created, and which driver is to be used to create the copy. Click **Next** after having selected the required information.




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**



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 fine has already been allocated on the target database, and will not be deleted automatically, even if the actual data replication is not carried out.

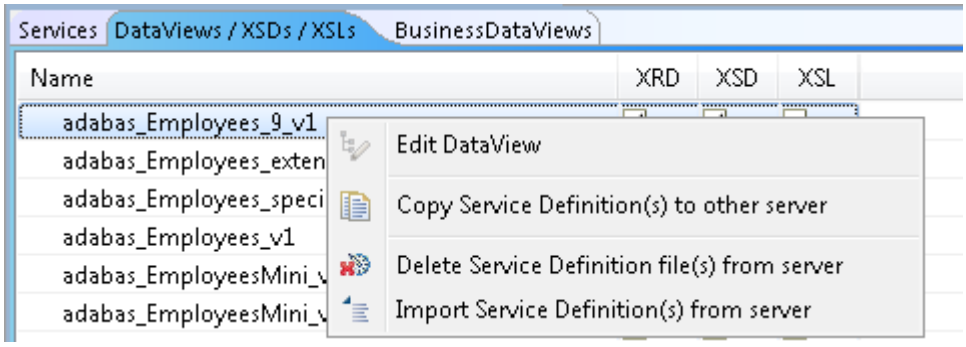
At each stage messages are displayed in the right-hand pane. Click **Done** to dismiss the Replication dialog.

## Refresh a Service

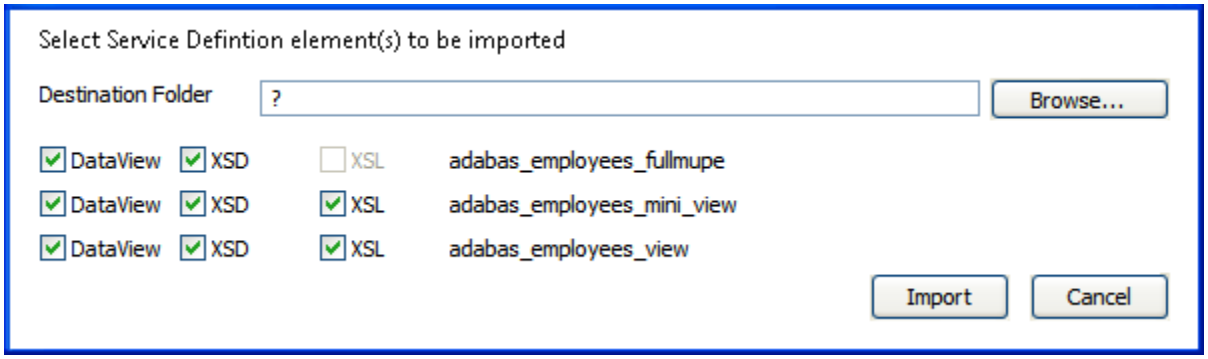
When modified Service definitions files (DataView or XSD) are exported to the SOA Gateway server, Service(s) using these file(s) may need to be 'refreshed' on the server to pick up the modifications.

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 ..... 163
- Service context menu ..... 164
- Lifecycle Governance ..... 167

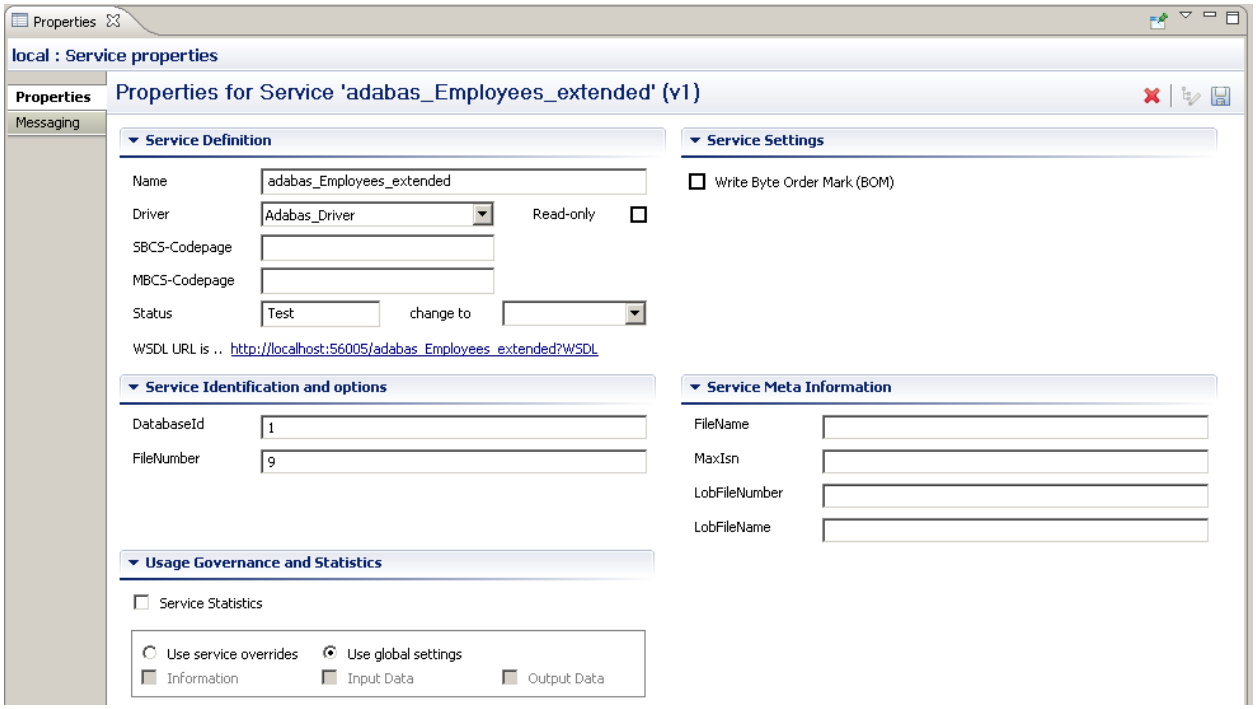
When a driver entry is expanded, the list of services associated with this driver is also displayed.

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

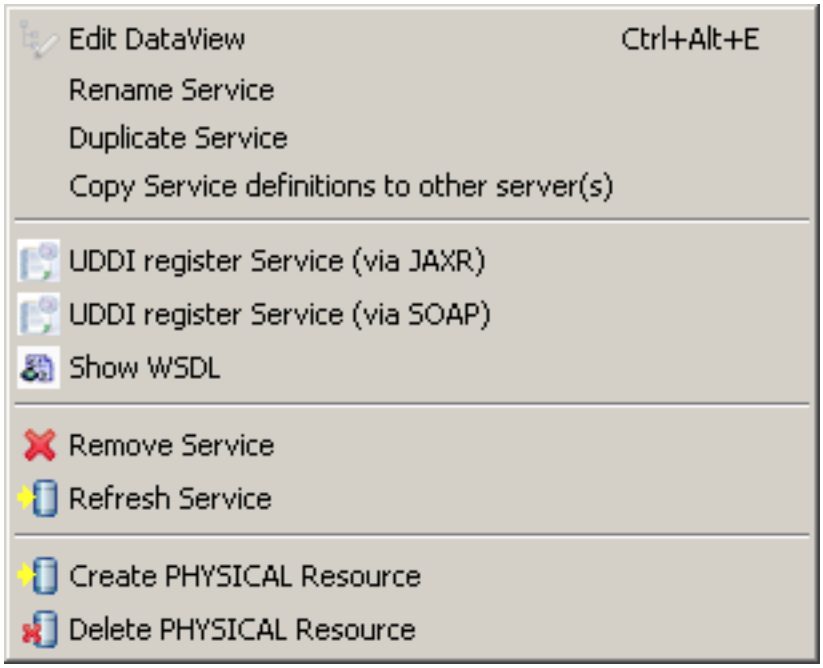



When a web service is selected, the following information is displayed.

- 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 SOA Gateway 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 SOA Gateway 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), physically, based on the SOA Gateway Definitions.
- *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 SOA Gateway Servers are accessible from the Control Centre.
  - If migrating services from a previous version of SOA Gateway, the utility is invoked from the most recent Control Centre.
  - Both SOA Gateway 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:

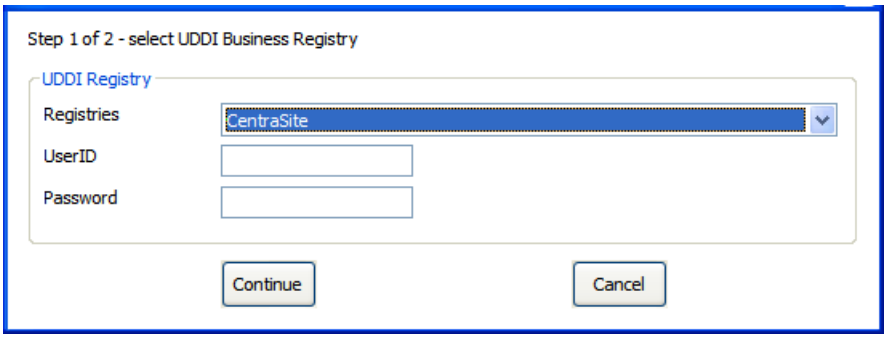
1. In the SOA Gateway you wish to *copy from*, choose the service you wish to copy by right-clicking and choosing **Copy Service Definitions**
2. Choose the server you wish to *copy the service to*.
3. If no driver with the same name exists on the target server you will be given the option to choose another driver of the same type as the source.
4. If you wish to overwrite the service on the target server, check the **Overwrite exsting....** 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)

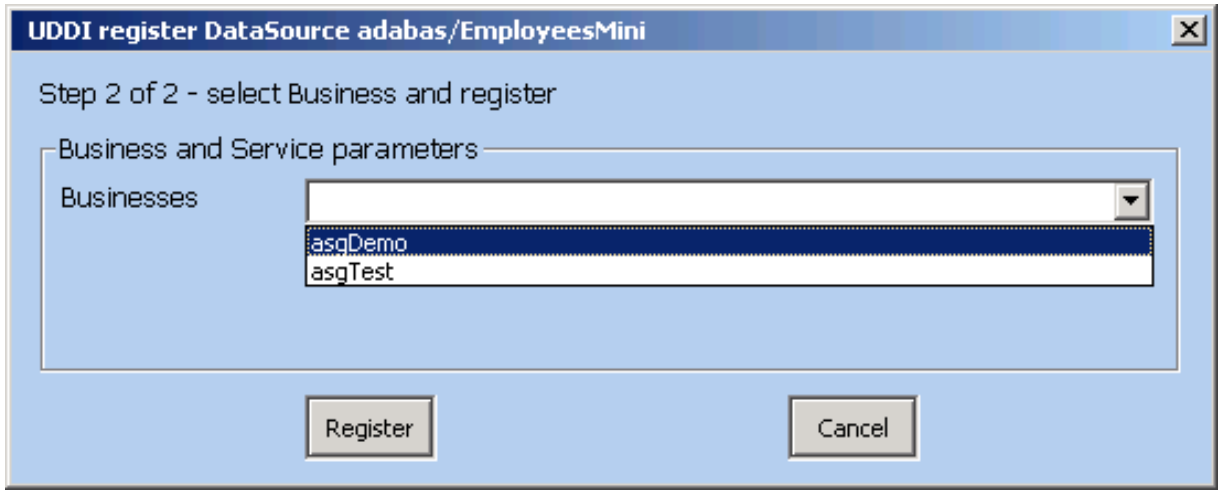
SOA Gateway Services, which are essentially "Services", can be registered with an UDDI Business Registry, for example Software AG's Centrasite, so that any UDDI enabled client application can find an exposed service, retrieve its signature, issue requests agains it etc.

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

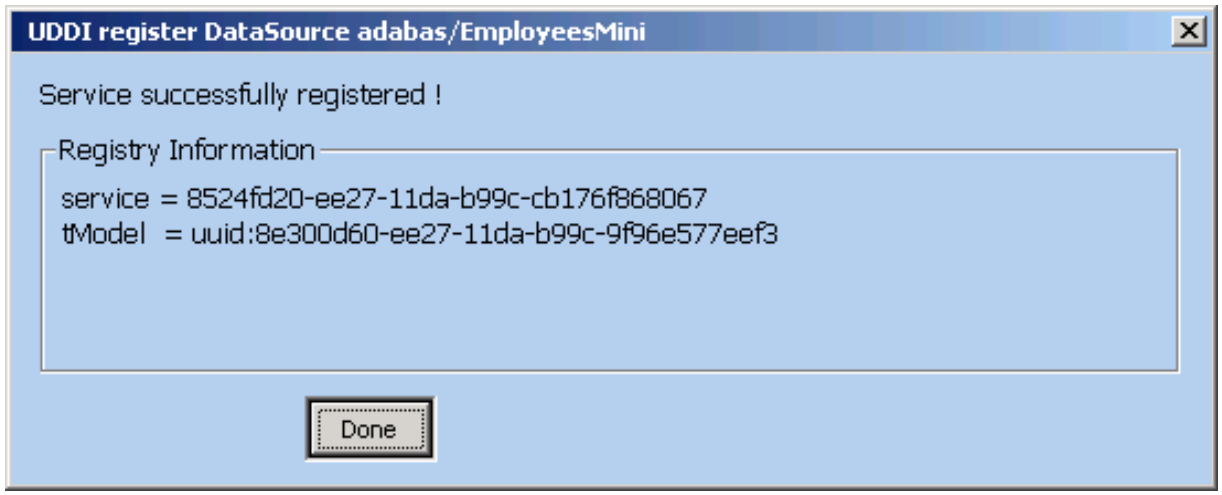
1. Define a UDDI Business Registry to the SOA Gateway Control Centre (from the SOA Gateway Eclipse Preferences dialog)
2. Select the **UDDI register Service** function from the context menu
3. Select one of the predefined UDDI Business Registries, enter the user credentials required to be able to publish




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 SOA Gateway 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 SOA Gateway web services. As part of the SOA Gateway 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, SOA Gateway 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 SOA Gateway 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 SOA Gateway 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

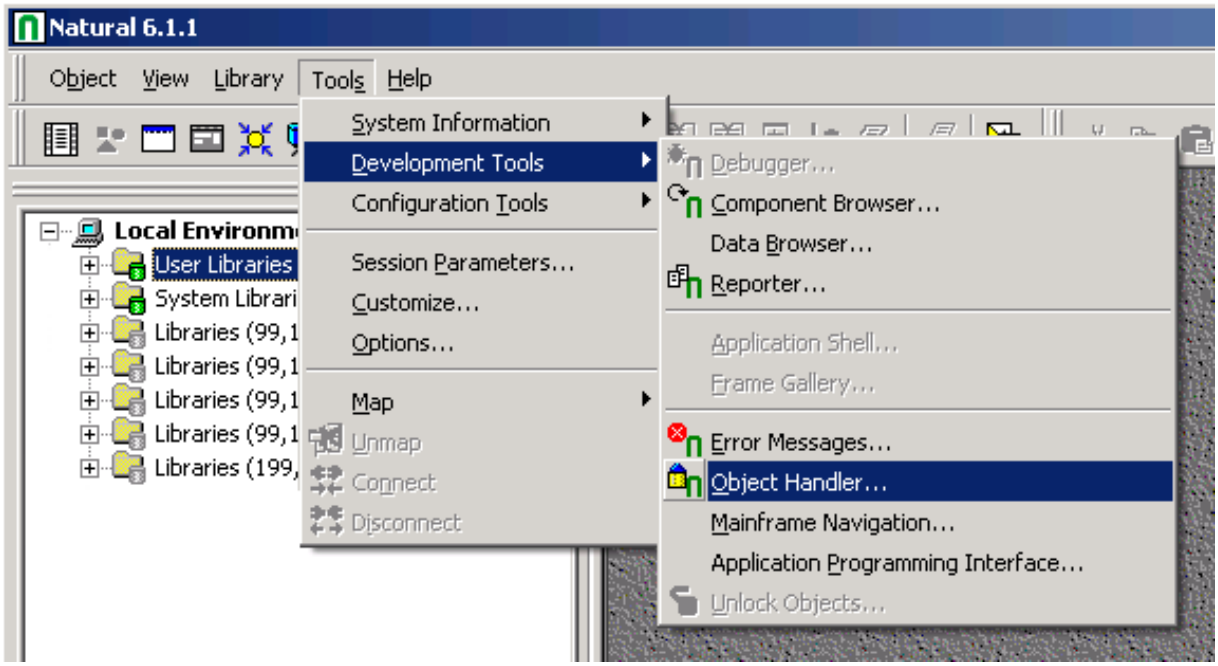
| *Current Status* | *Next Status* |
|---|---|
| Test | Frozen |
| Frozen | Test, Deprecated |
| Deprecated | Test, Historical |
| Historical | Deprecated |

# 20 Creating a SYSOBJH extract file

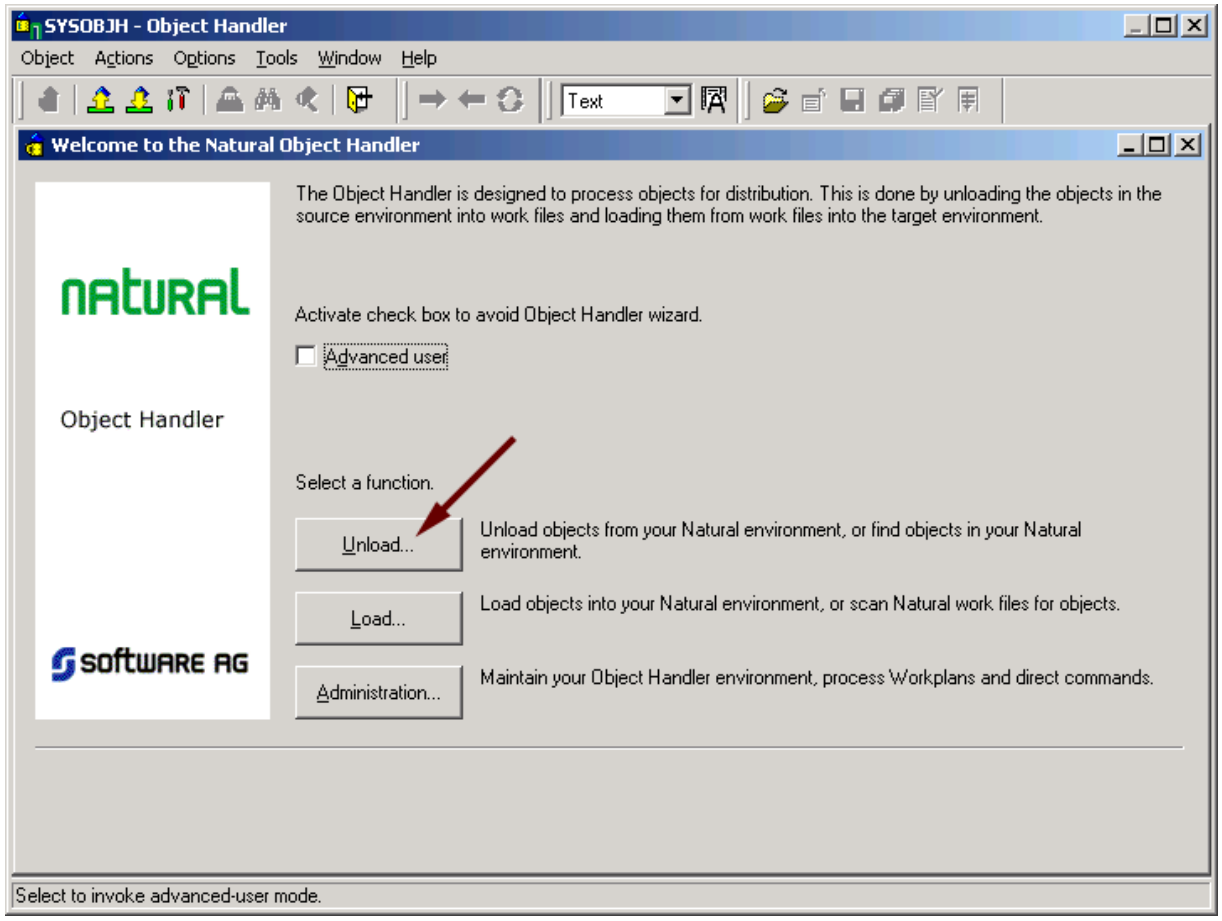

- Generating a SYSOBJH extract (Windows) ..... 172
- Generating a SYSOBJH extract (Mainframe) ..... 183

## Generating a SYSOBJH extract (Windows)

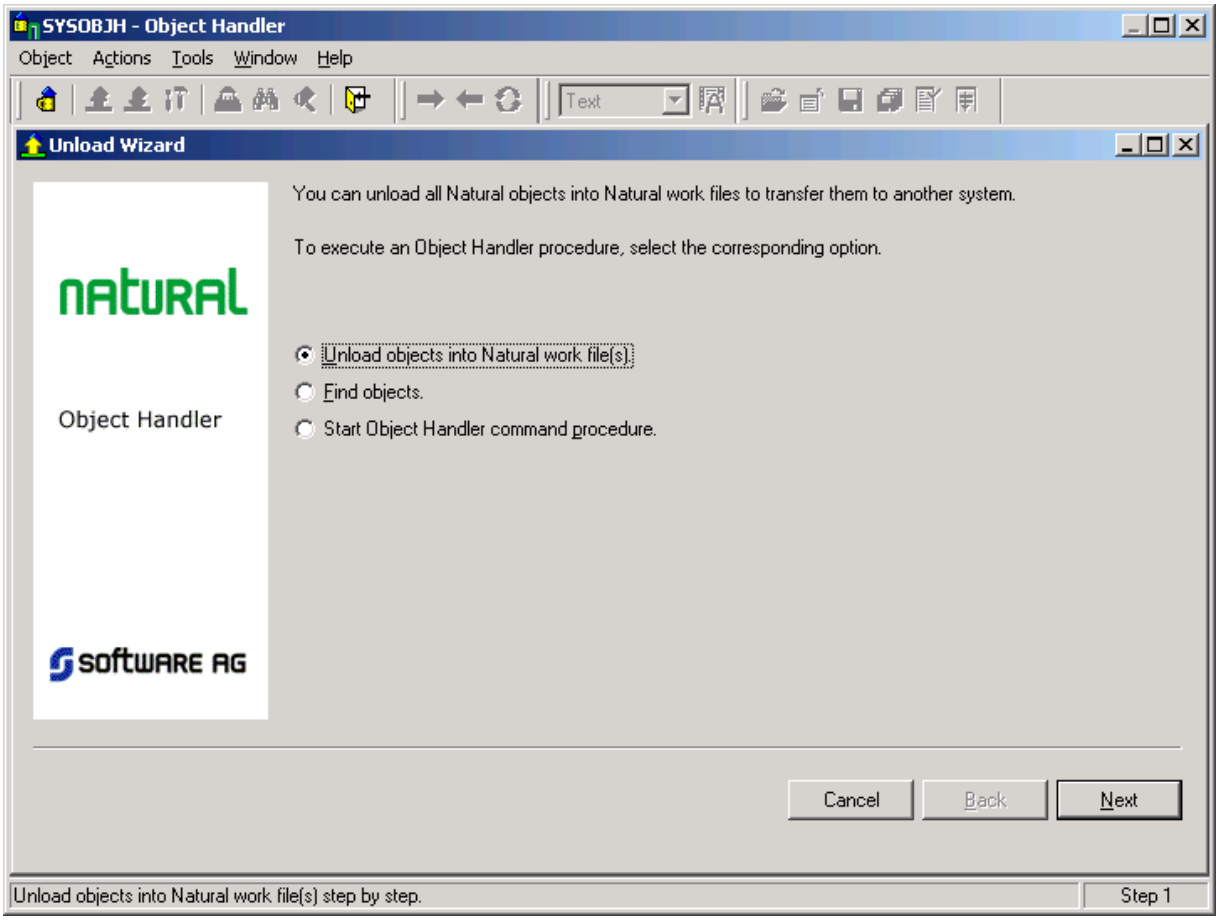
1. Start Natural, from the **Tools** menu bar select **Development Tools**", then select **Object Handler**.



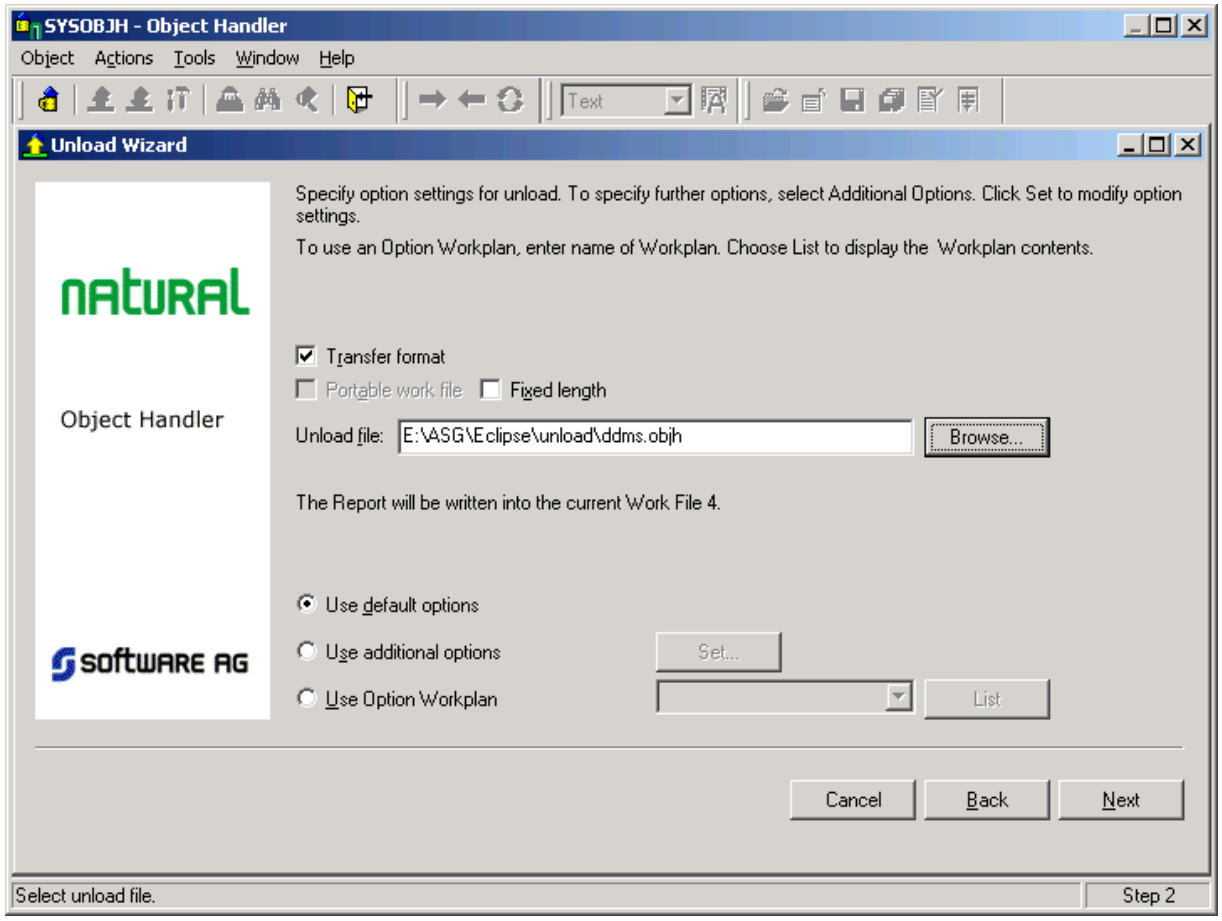

2. Click the **Unload** button to start the Unload Wizard.



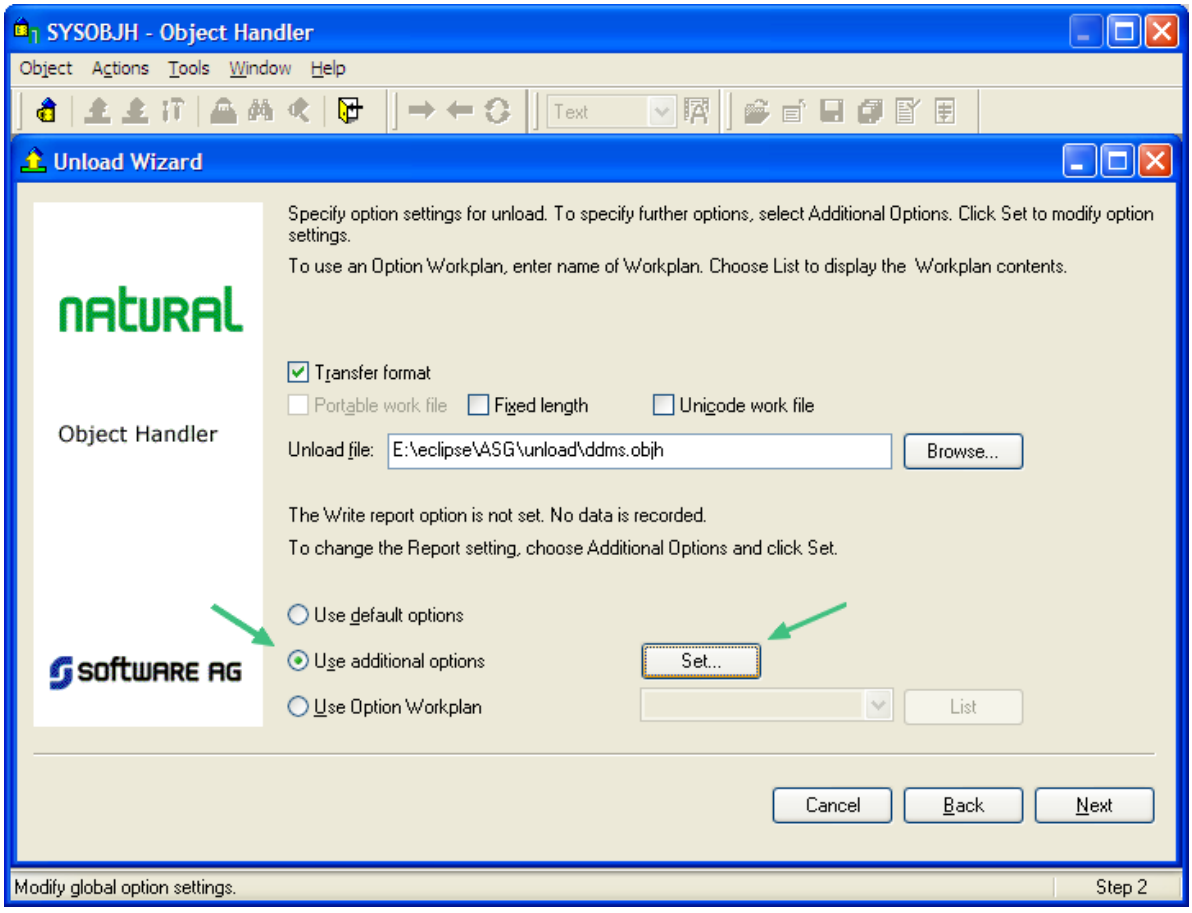

3. Check the **Unload objects into Natural work file(s)** radio button. Click **Next**.



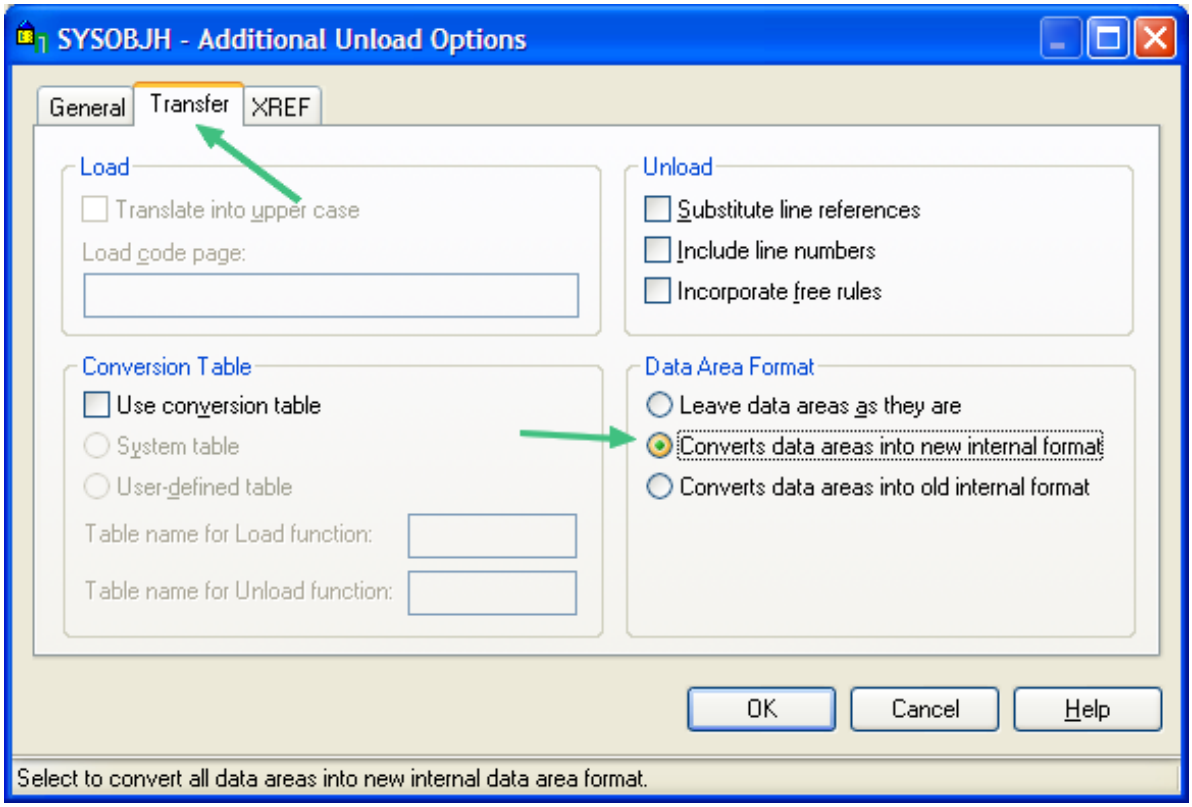

4. Select the **Transfer format** check box, enter the location to which you want the files to be unloaded in the Unload file text box (or click the **Browse** button to bring up a folder/file selection dialog). Click **Next**.



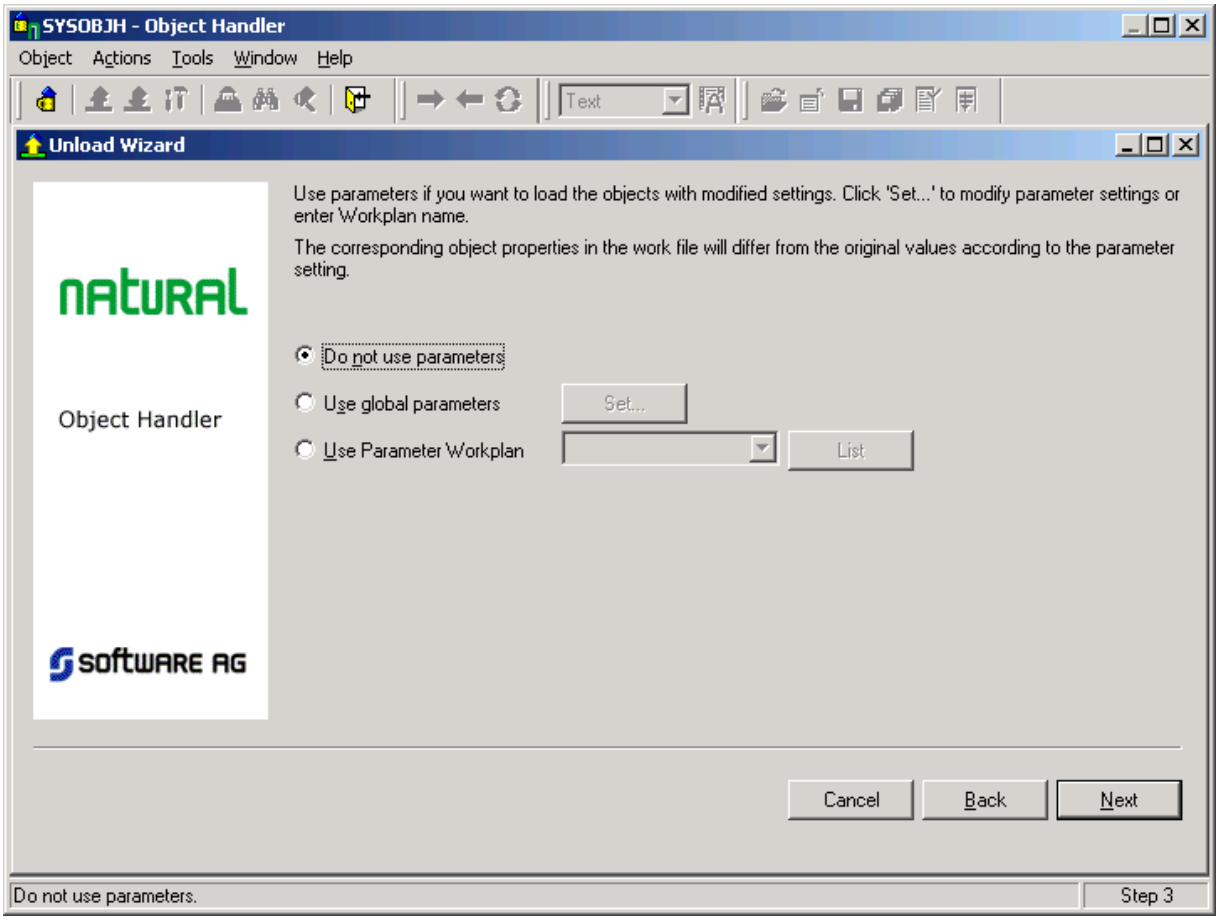

5. When generating an OBJH extract including PDAs, it is essential to set the following:
   - Check "Use additional options" and click the "Set..:" buttom



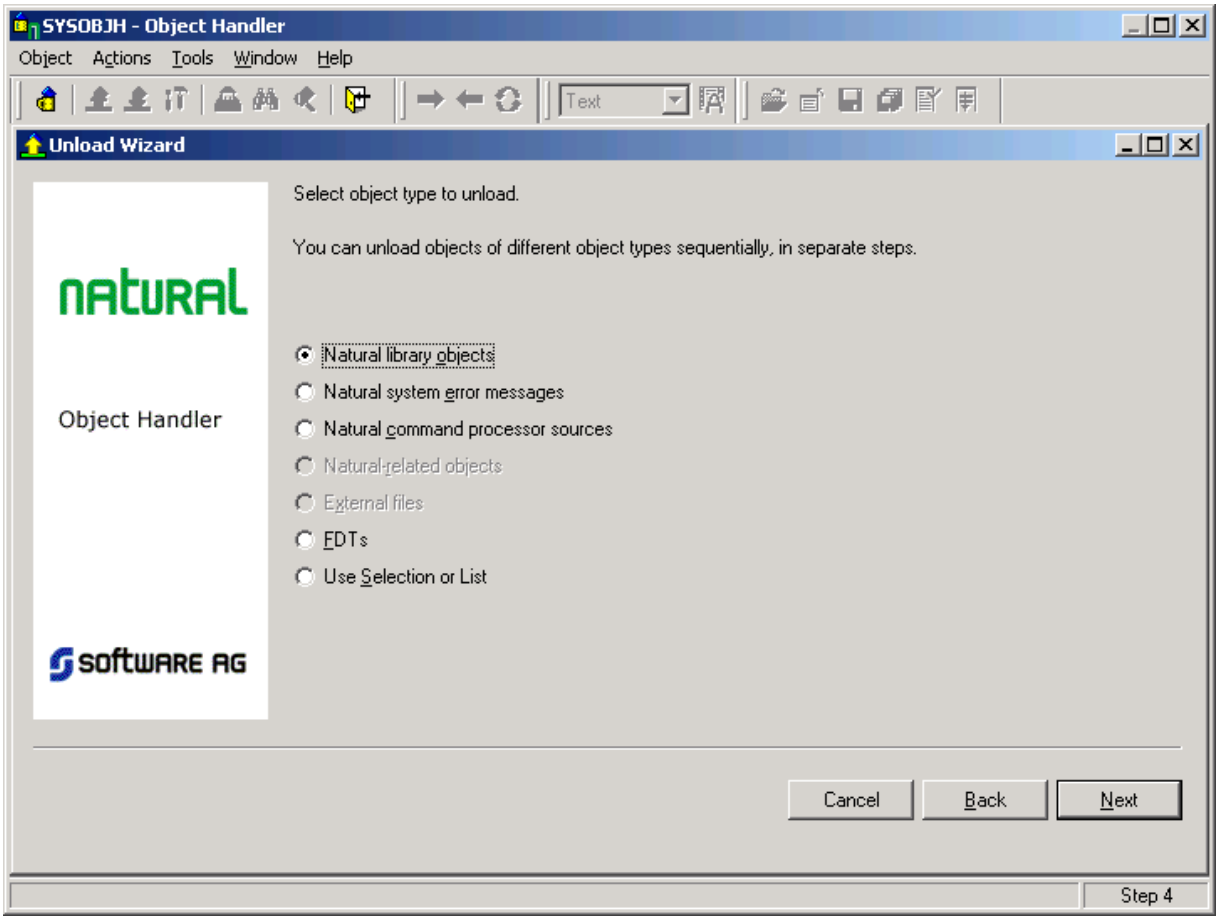

- Select the "transfer" tab and check the "Converts data areas into new internal format", then click "OK"



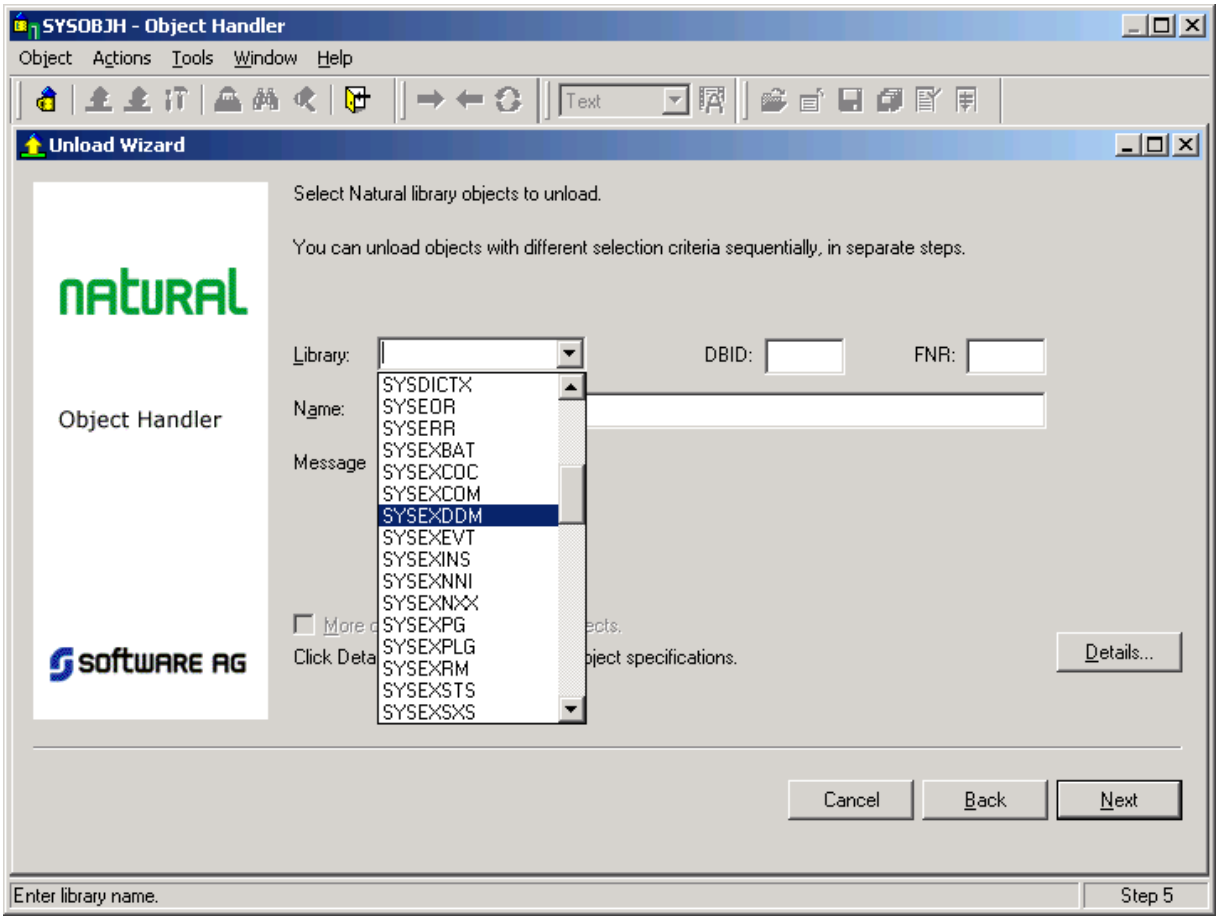

6. Select the **Do not use parameters** radio button, click **Next**.



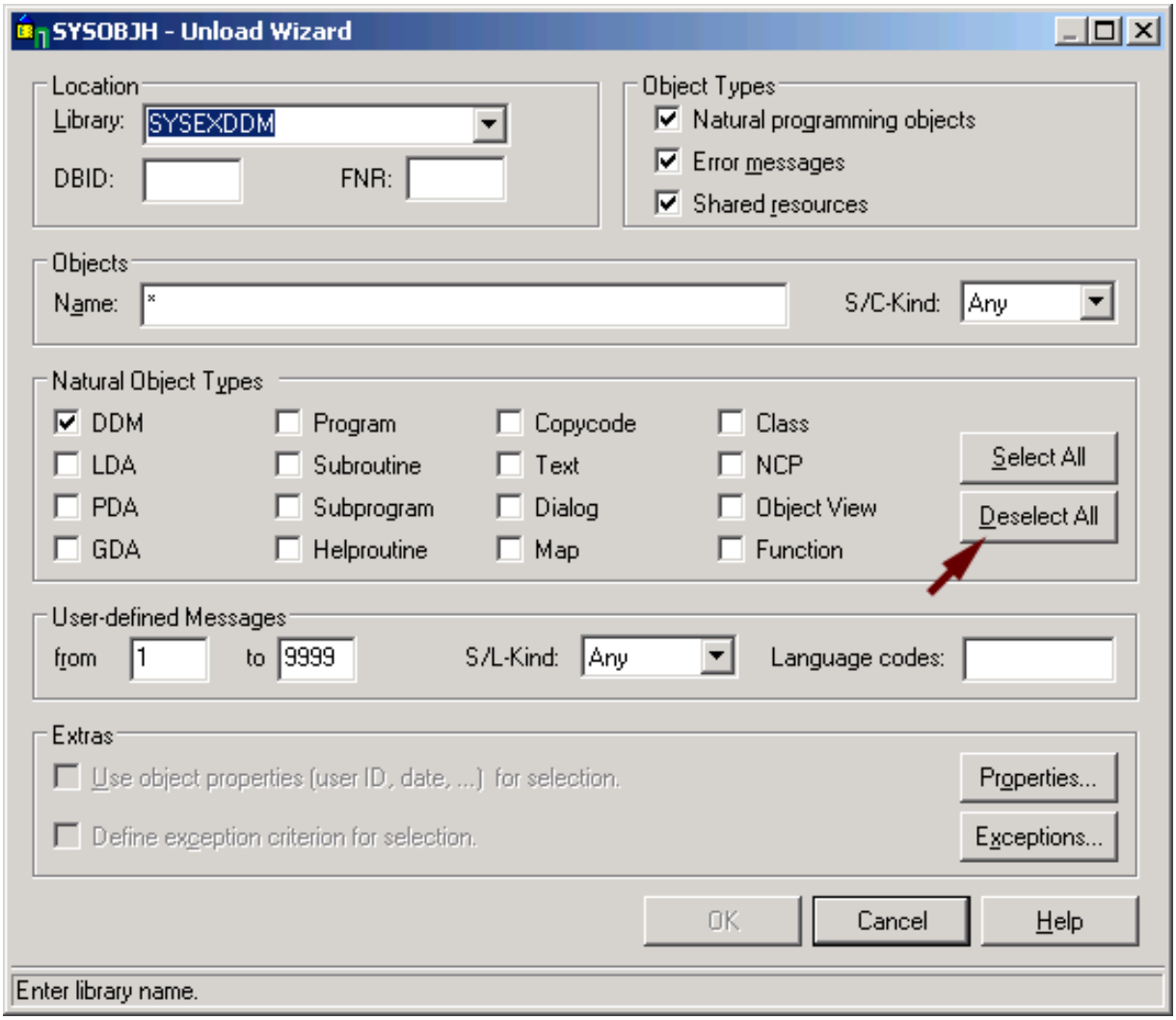

7. Select the **Natural library object** radio button, click **Next**.



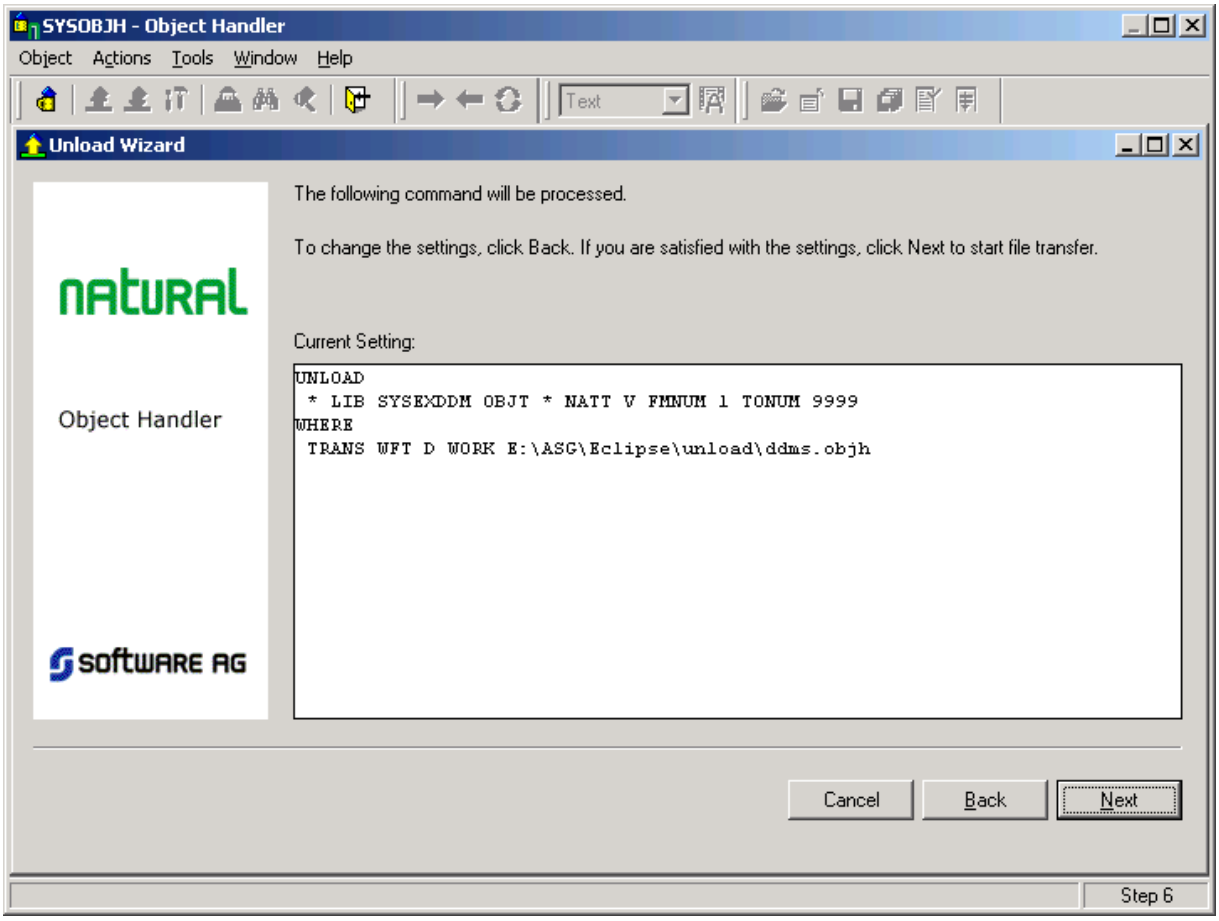

8. Select the Library containing your DDM definitions (e.g. SYSEXDDM) from the Library list box, then click the **Details** button. If you also enter a specific DBID and/or FNR, only the relevant DDMs are selected, otherwise all valid DDMs are returned.



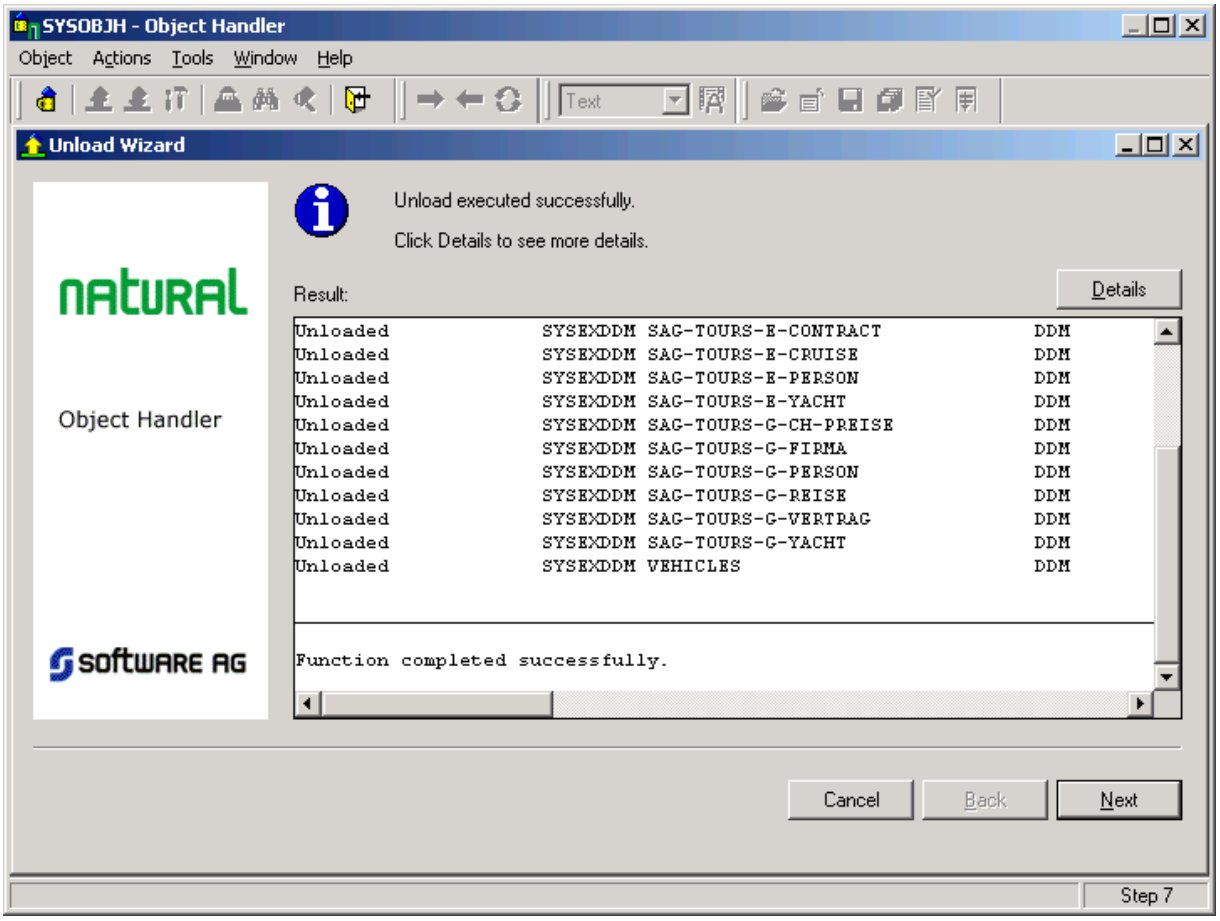

9. In the Unload Wizard details window, first click the **Deselect All** button to deselect all Natural Object Types, then select **DDM** and/or**PDA** . Click the **OK** button to return to the Unload Wizard.




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

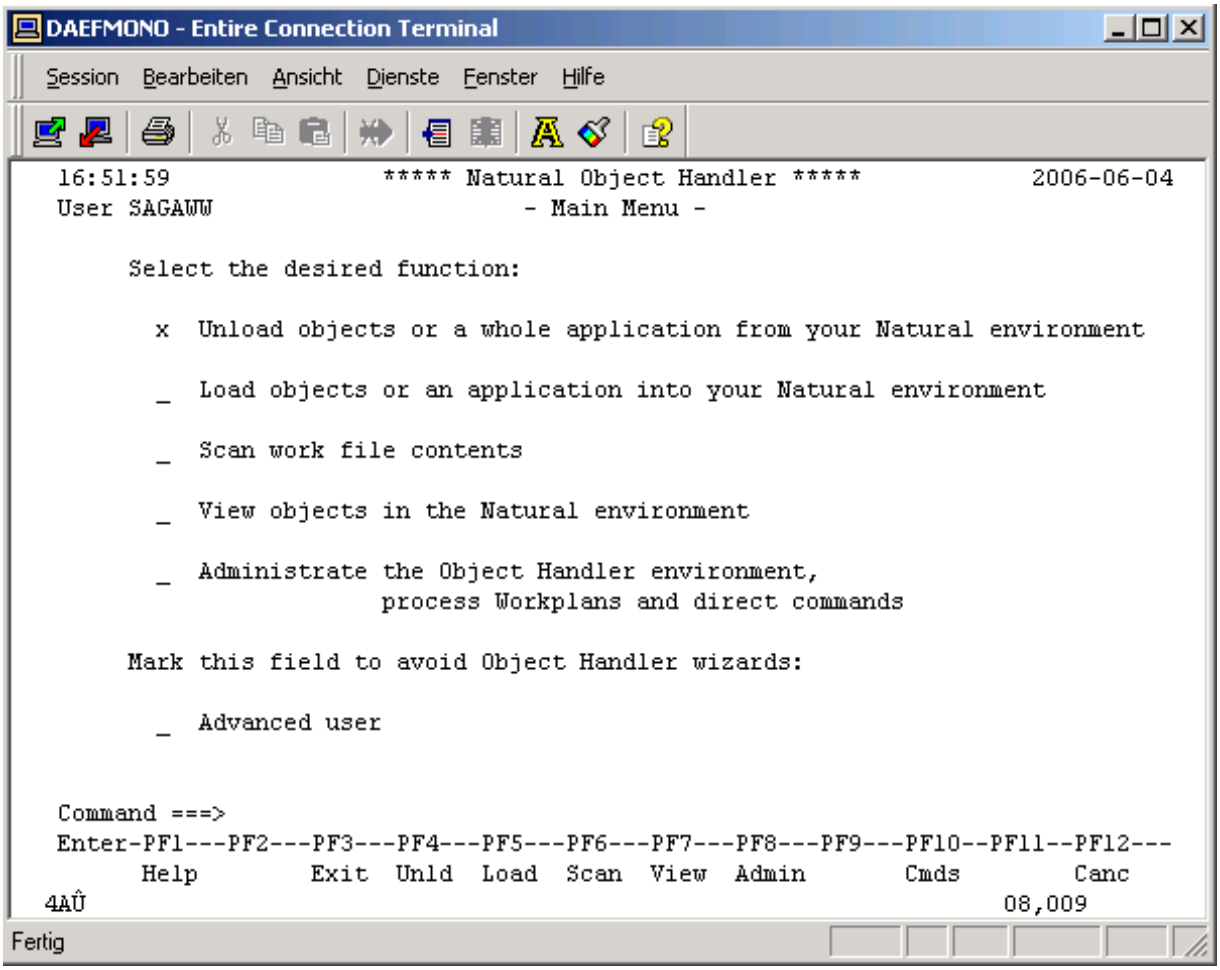



11. The status of the unload is shown. If it was successful, a file is created under the path and file name specified. Click the **Next** button to complete the unload, close the SYSOBJH wizard.

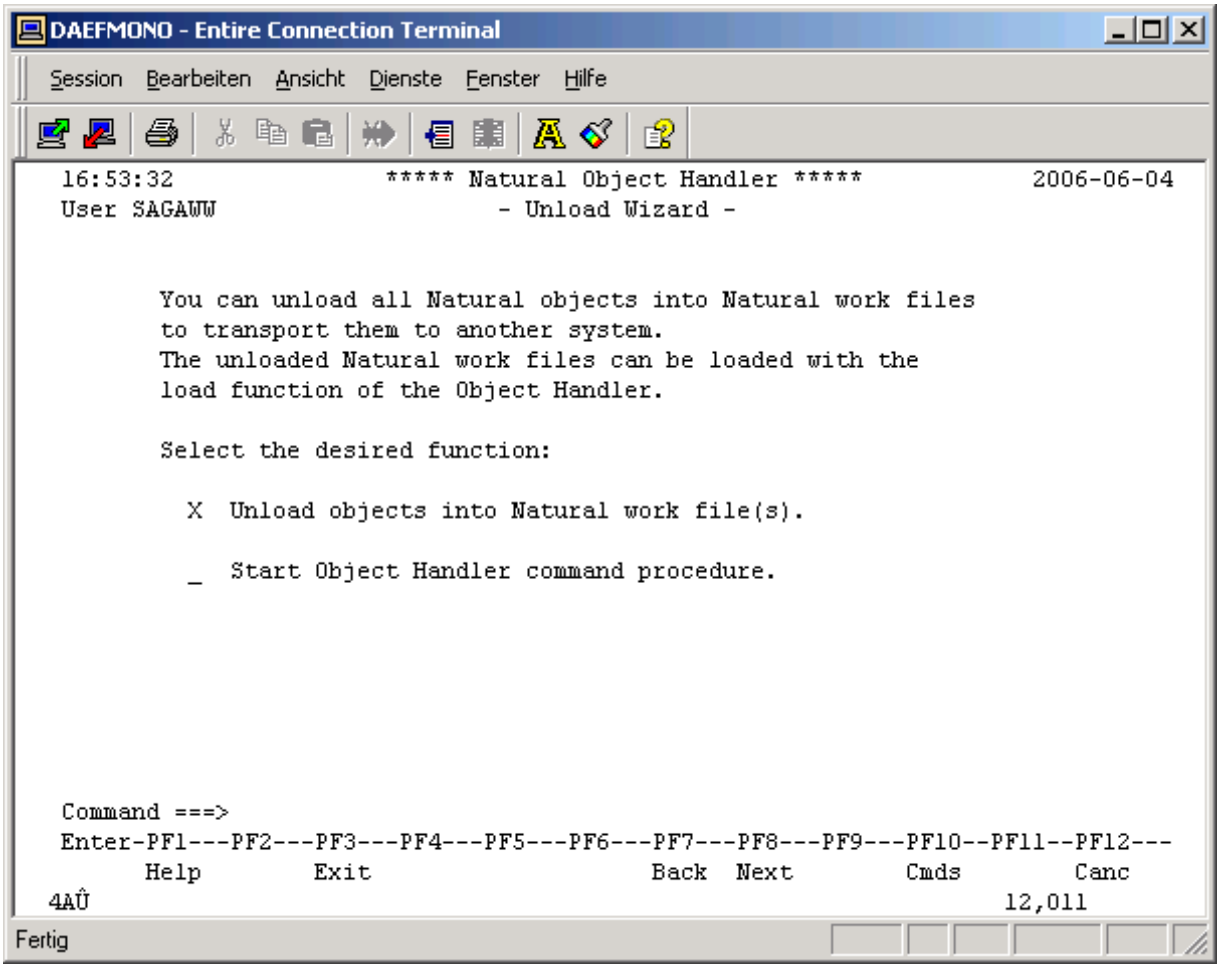



## Generating a SYSOBJH extract (Mainframe)

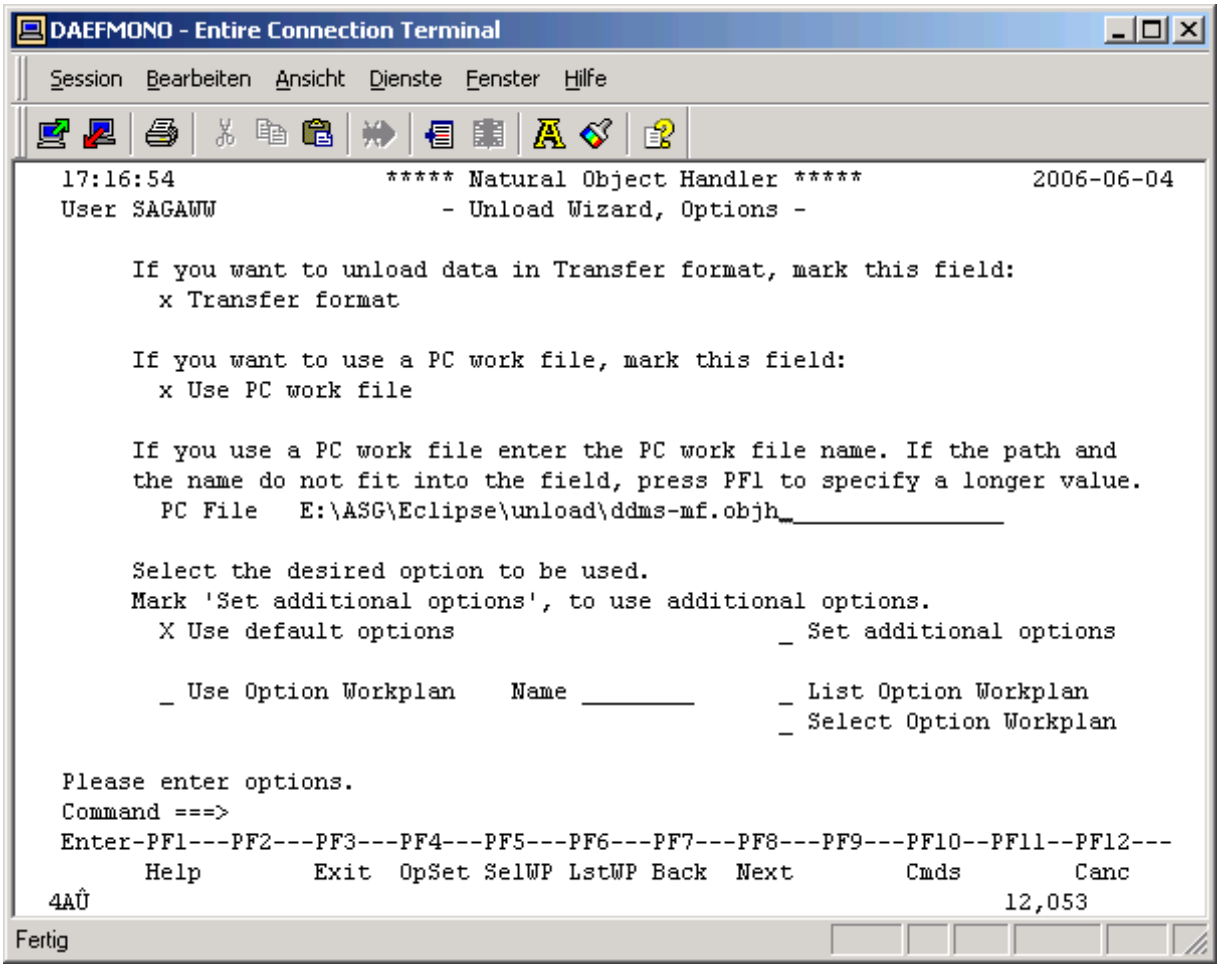
1. Start Natural, enter the command `SYSOBJH` to start the Natural Object Handler.
2. Select the **Unload** fnuction to start the Unload Wizard.



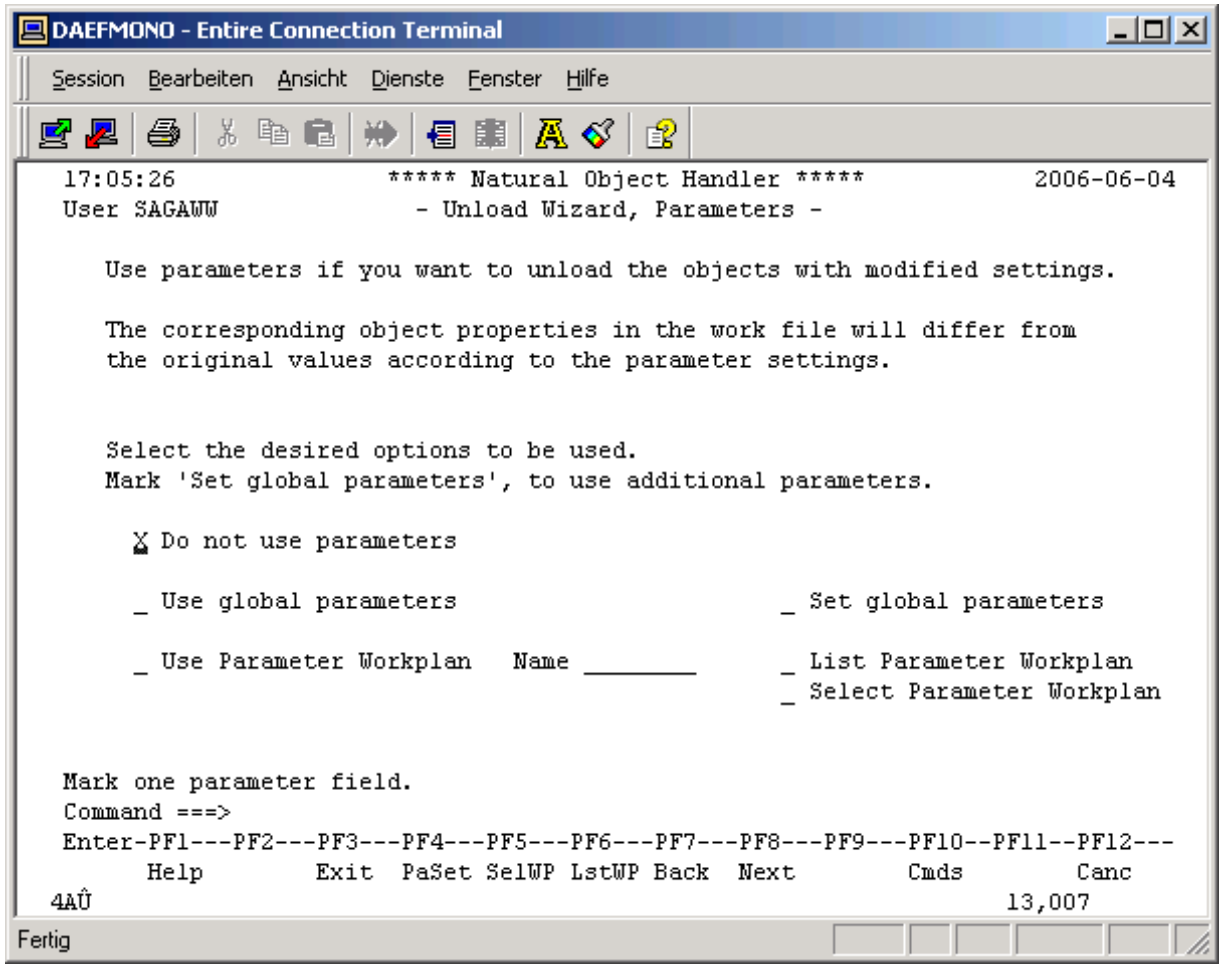

3. Check **Unload objects into Natural work file(s)** radio button.



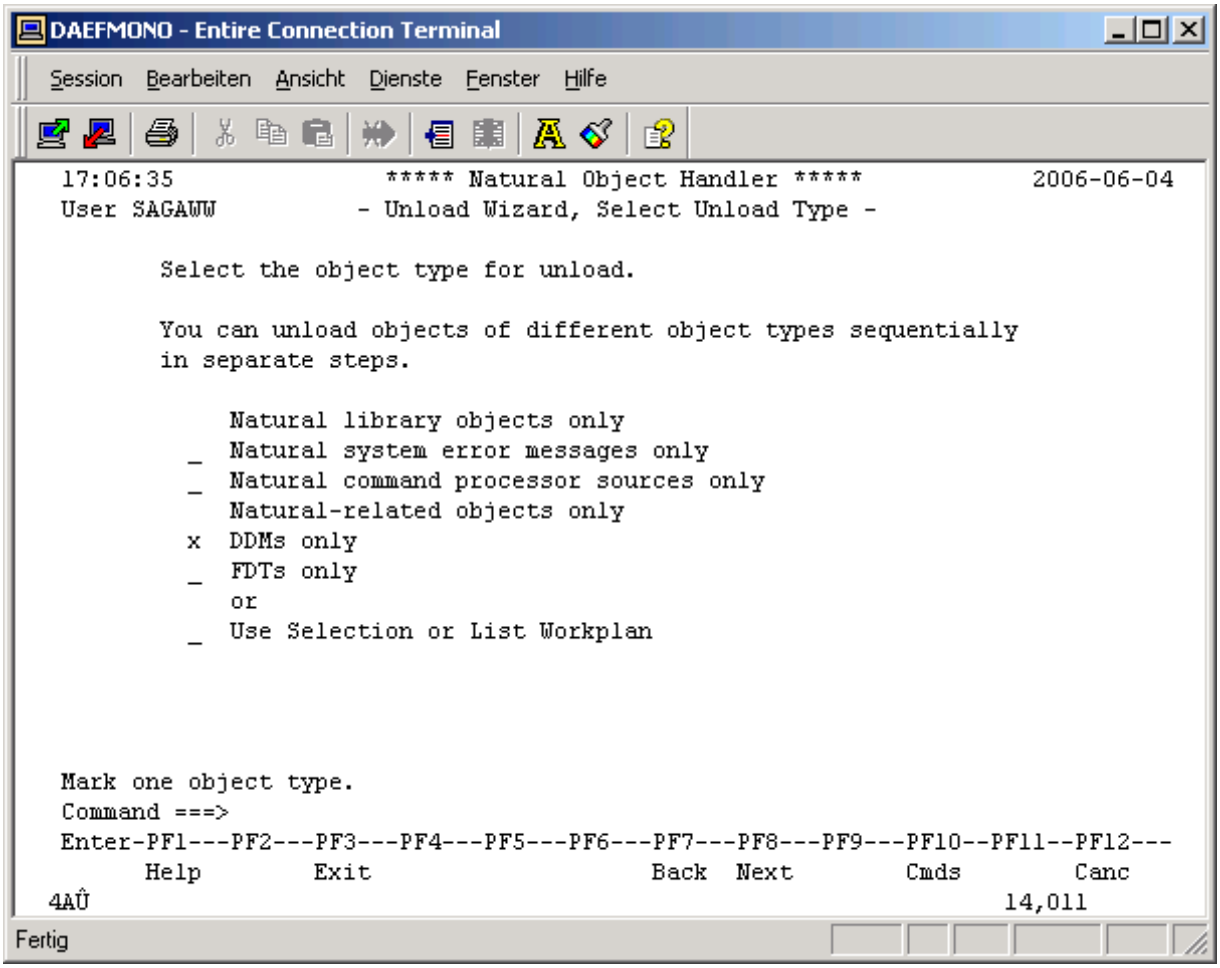

4. Select Transfer format, this example uses Entire Connection to transfer the unload file to your PC, enter the location to which you want the files to be unloaded in the PC File input field.



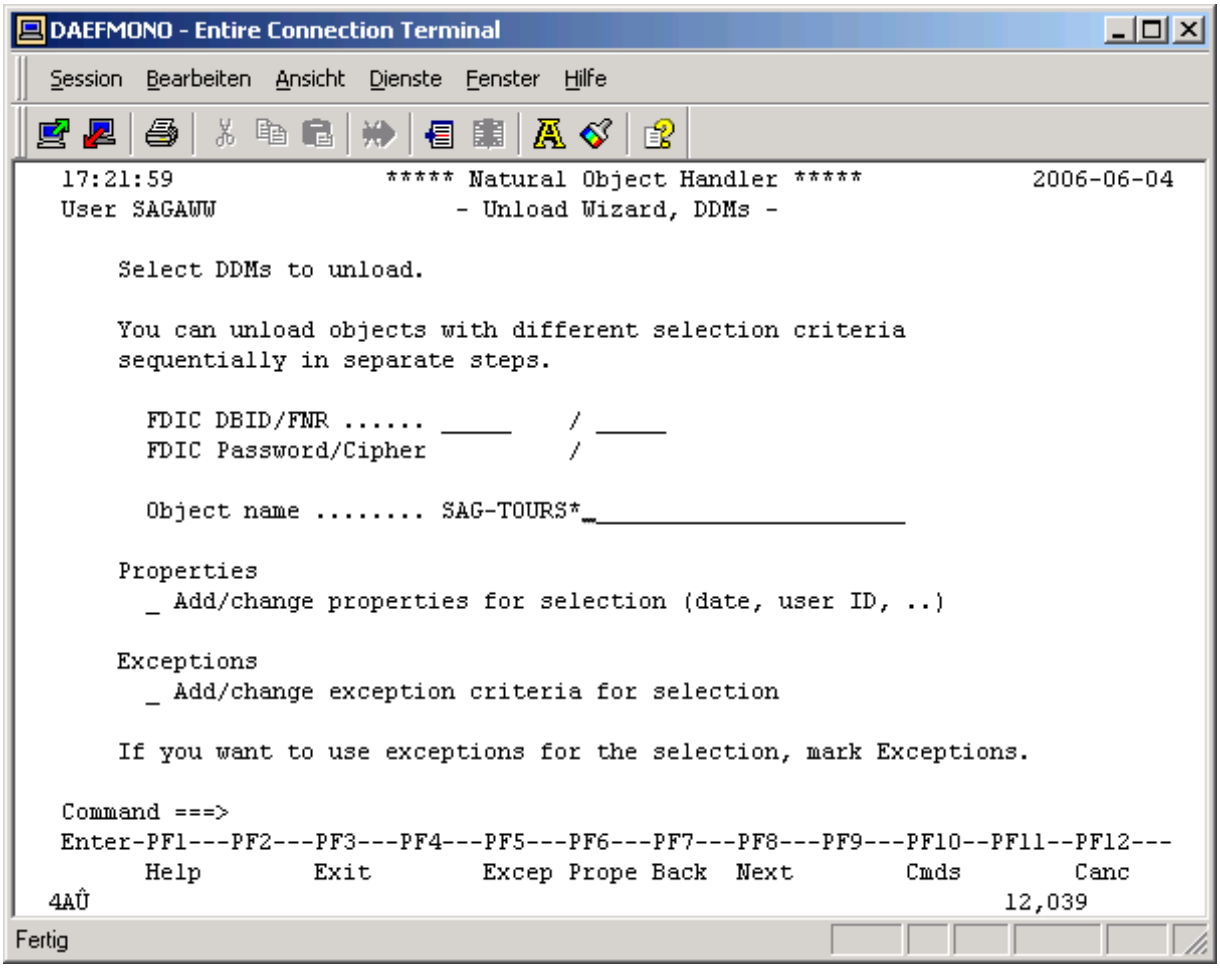

5. Select **Do not use parameters.**



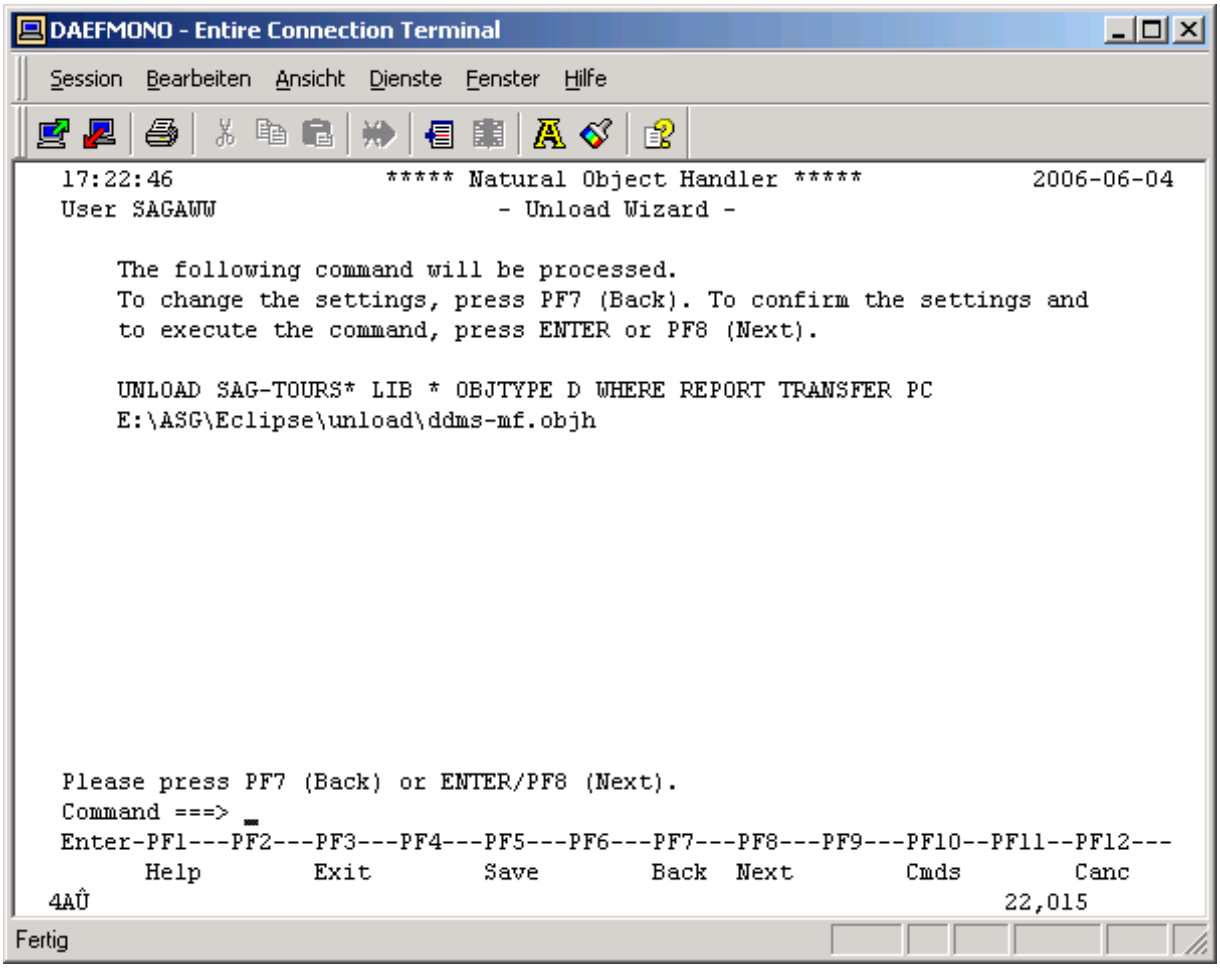

6. Select **DDMs only.**



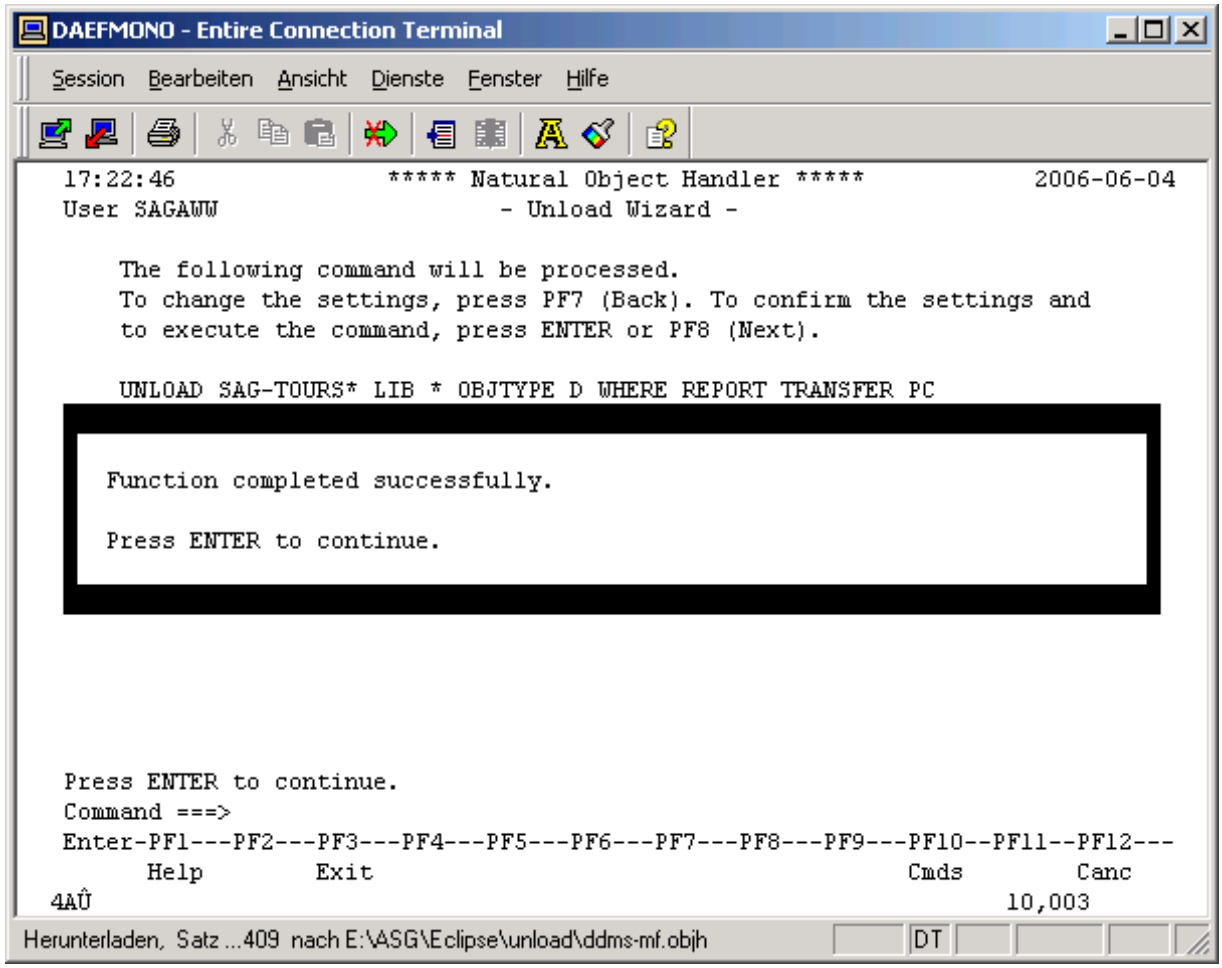

7. Select the DDM(s) to be unloaded, optionally specify additional selection criteria to narrow the search.




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




9. 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.


DAEFMONO - Entire Connection Terminal
Session Bearbeiten Ansicht Dienste Fenster Hilfe
17:22:46 ***** Natural Object Handler ***** 2006-06-04
User SAGAWW - 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
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 Cmds Canc
4AÛ 10,003
Herunterladen, Satz ...409 nach E:\ASG\Eclipse\unload\ddms-mf.objh
DT

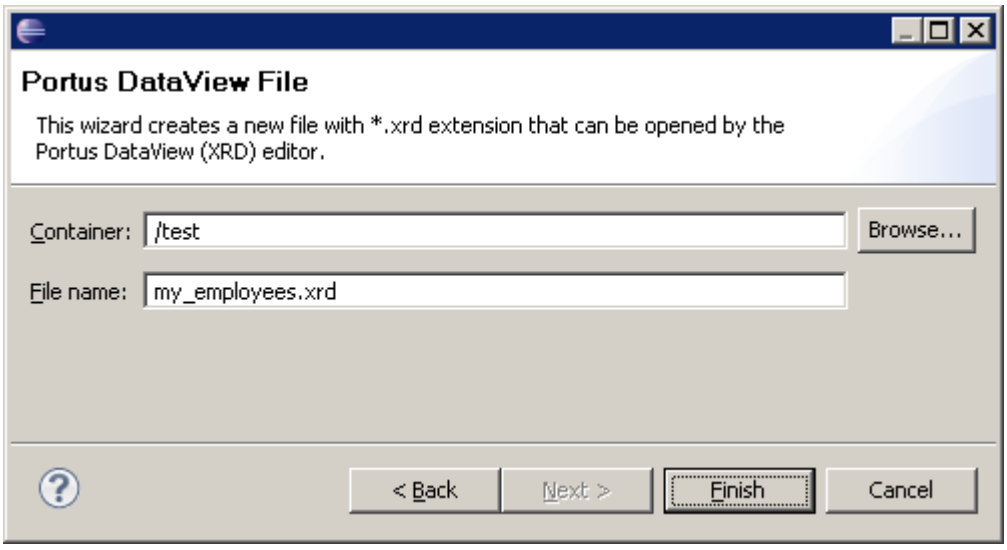
# 21 SOA Gateway DataViews


- Create a DataView from scratch ..... 194
- Opening a SOA Gateway DataView for editing ..... 195
- Editing a SOA Gateway DataView ..... 196
- Editing "special" fields ..... 201
- Enhanced Type Conversion ..... 204
- Exporting a DataView to a SOA Gateway server ..... 205
- Importing an existing SOA Gateway DataView ..... 207
- Creating a XML Schema for a DataView ..... 207

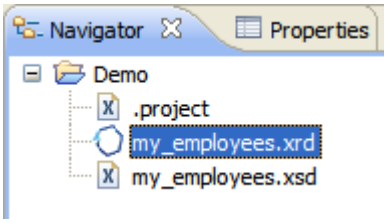
- Create a DataView from scratch
  - adding MU (muliple value) fields
  - adding PE (periodic group) fields
  - adding "special" fields (Adabas Super-Descriptors etc.)
- Opening a SOA Gateway DataView for editing
- Editing a SOA Gateway DataView
- Export a DataView to a SOA Gateway server
- Importing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


## Create a DataView from scratch

1. A vanilla SOA Gateway 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-> SOA Gateway 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.



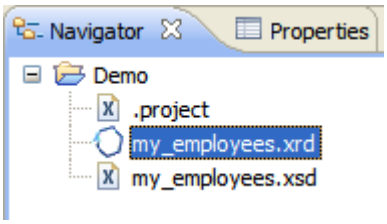

3. You have now created an "empty" (i.e. no fields defined yet) SOA Gateway DataView file in the selected folder.



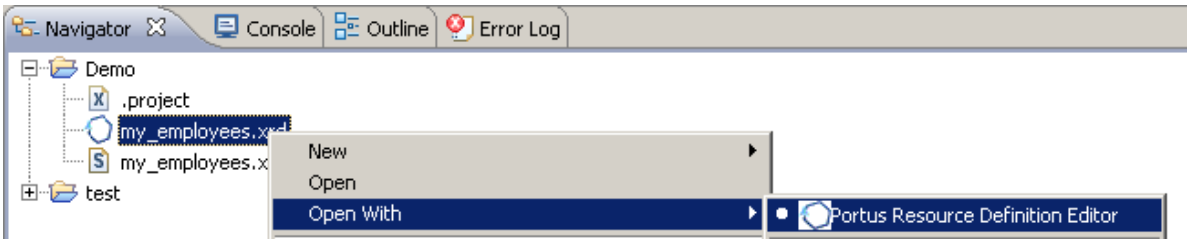

4. For information on actually editing the DataView, adding fields etc., please continue reading at **Edit a SOA Gateway DataView**

## Opening a SOA Gateway DataView for editing

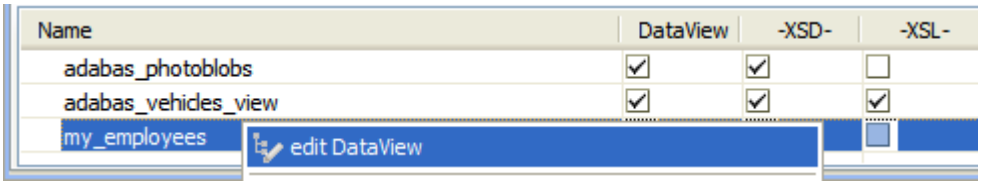
1. Open a "local" file, contained in an Eclipse project within the active workspace...
   - by double-clicking on it's name in the Package Explorer or Navigator




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



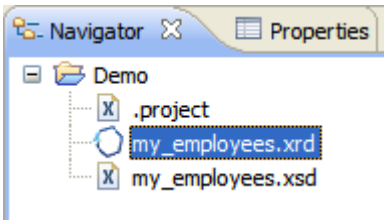

2. Open a "remote" DataView file, directly on the server, without importing it into a workspace / project first, by right-clicking the DataView name in the Configuration View to bring up the context menu, then select "edit DataView"




## Editing a SOA Gateway 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.

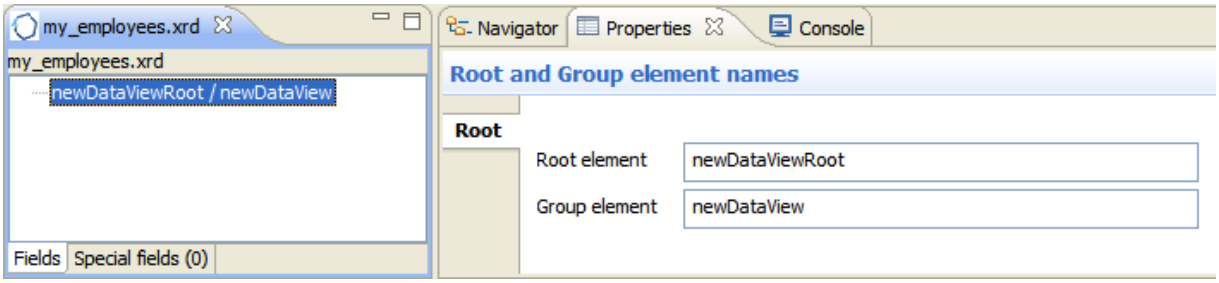
   DataViews are not tied to a specific "resource type", the same mapping can be used to access an Adabas file or a SQL table. There are, however, elements of a DataView which are only meaningful in the context of a specific resource type, for example "special descriptors" (super-, sub-, ...) for Adabas.




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:

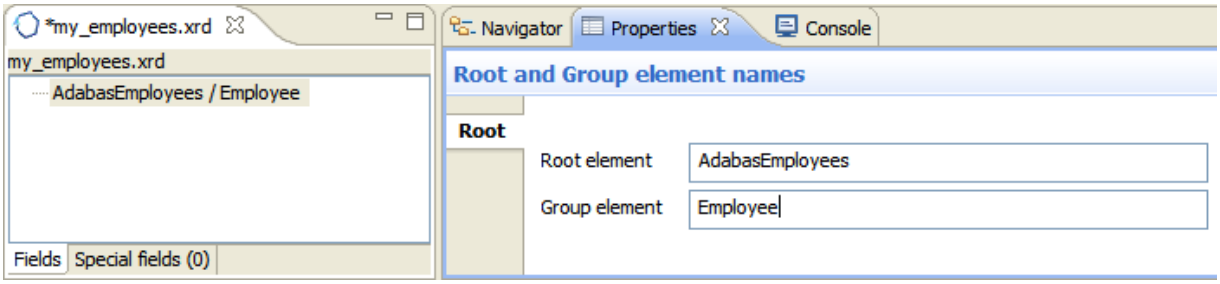


3. First of all, enter the Root element name: The (XML) "structure" or "set" name under which items (records) for a Resource linked to this DataView will be referred to. E.g.: AdabasEmployees

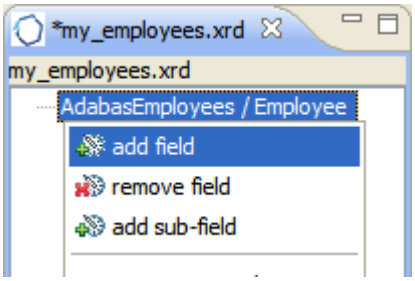
   Enter the Group element name, this is the SOA Gateway "record name". (E.g.: Employee).

   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.

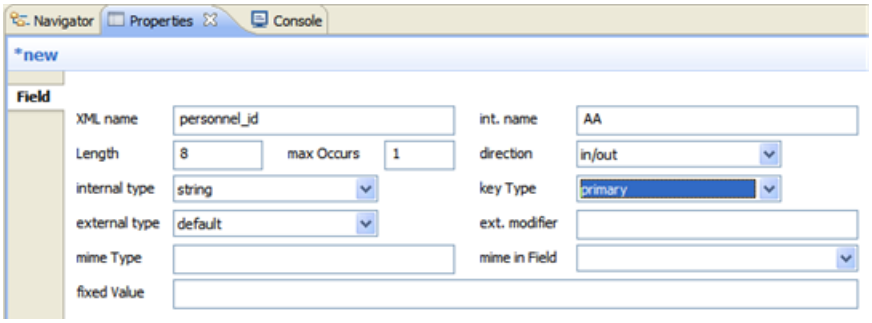



4. We can start adding the actual DataView Definition elements now, right-click on the root element in the editor window, from the context menu appearing now select **add field**.

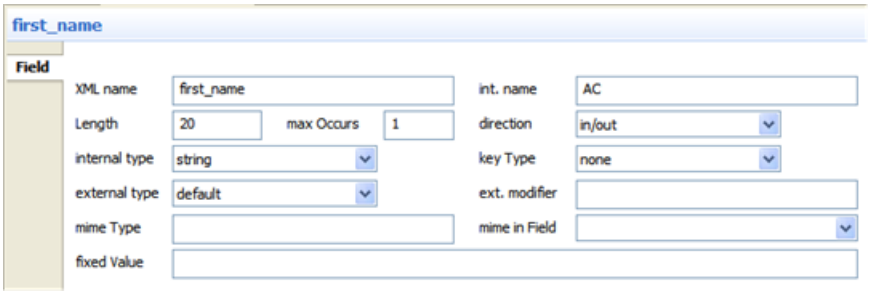


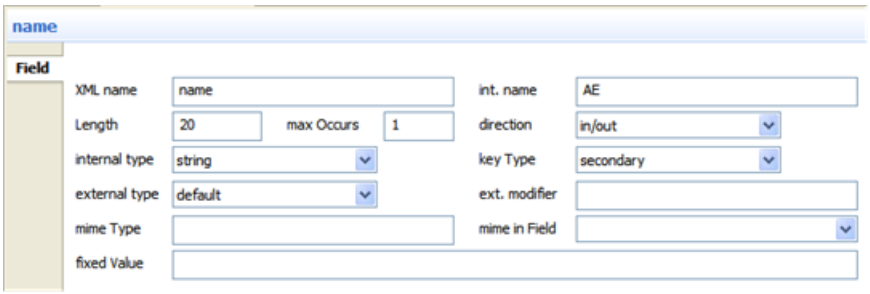

5. A new element ("Field") has been added with all properties set to initial values.

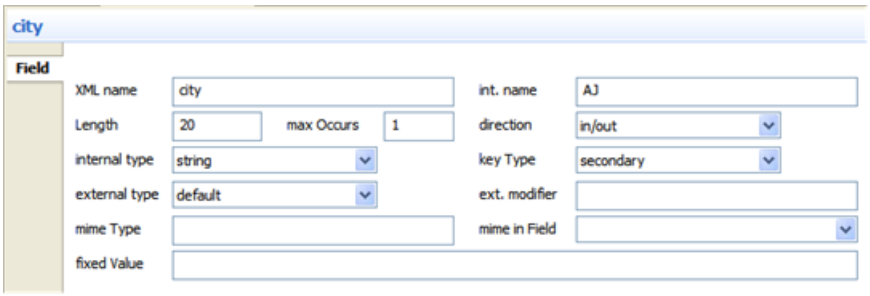
   The first field we are going to add is the "personnel Id", set the properties as follows:



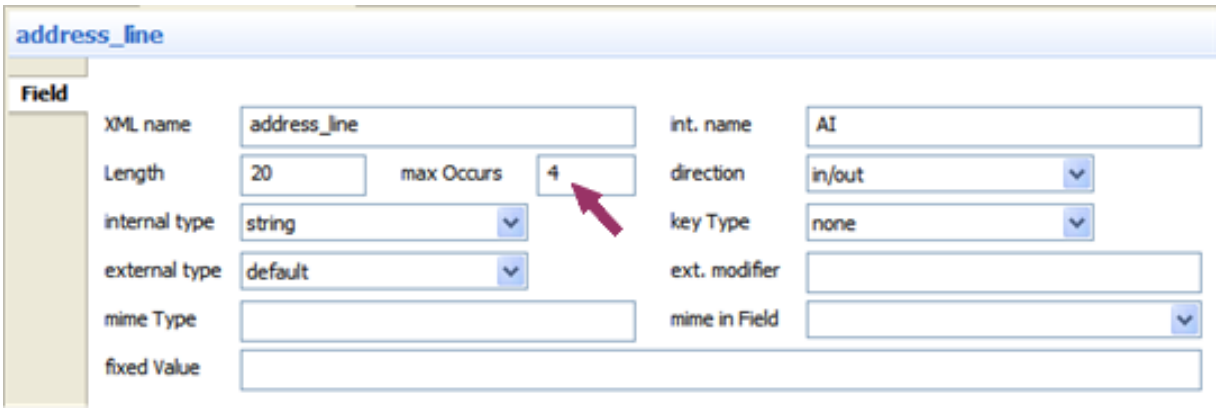
6. Add a few more fields with the following attributes:



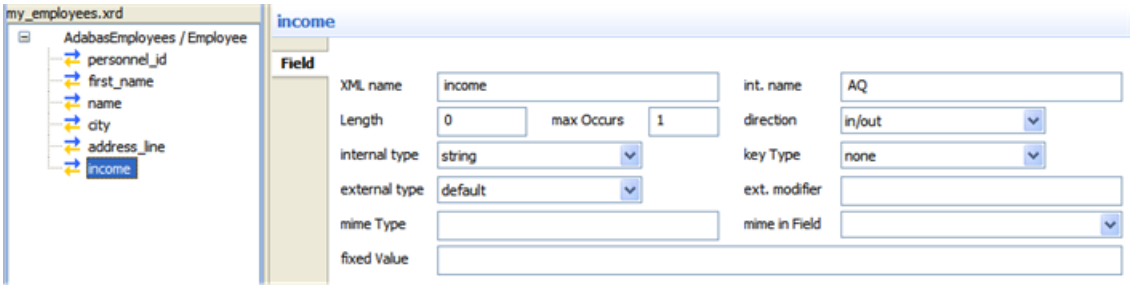

Next we are going to define the "address line" field, which is a MU (multiple value) field. MU fields are defined like a "flat" field, with the exception of the max Occurs Property being set to a value >0



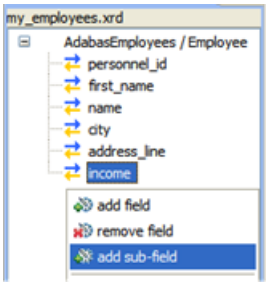
7. Lastly, we will define a PE (periodic group), the structure of the "income" group of the "Employees" file:

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

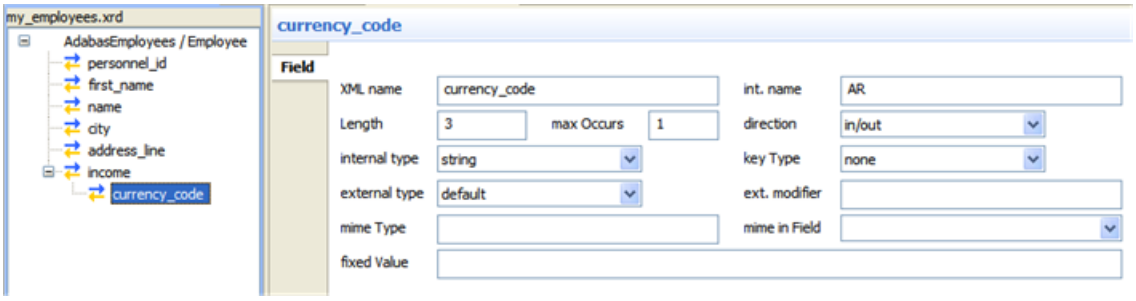
   So we first add the "group field" income. Only the xml Name, int. name (internal name = Adabas field name) and max Occurs properties are relevant here.

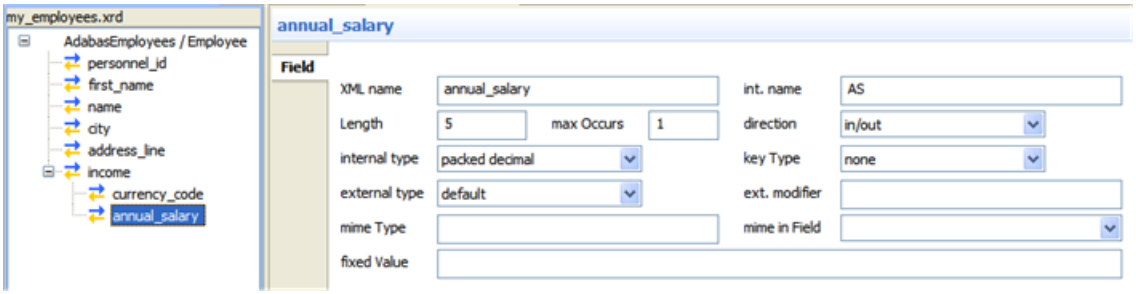



   To actually turn the income field to a PE group field, right-click it, select **add subfield**

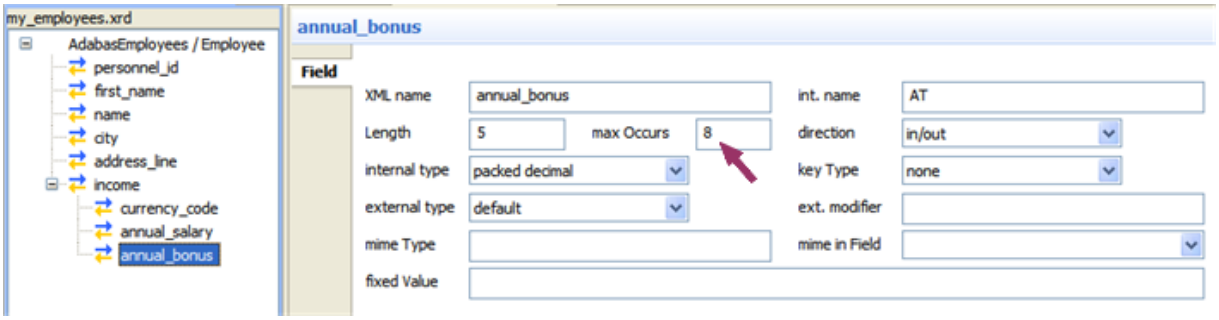



   PE Sub Field properties are equivalent to those of "regular" fields. Define the three PE-fields as follows:

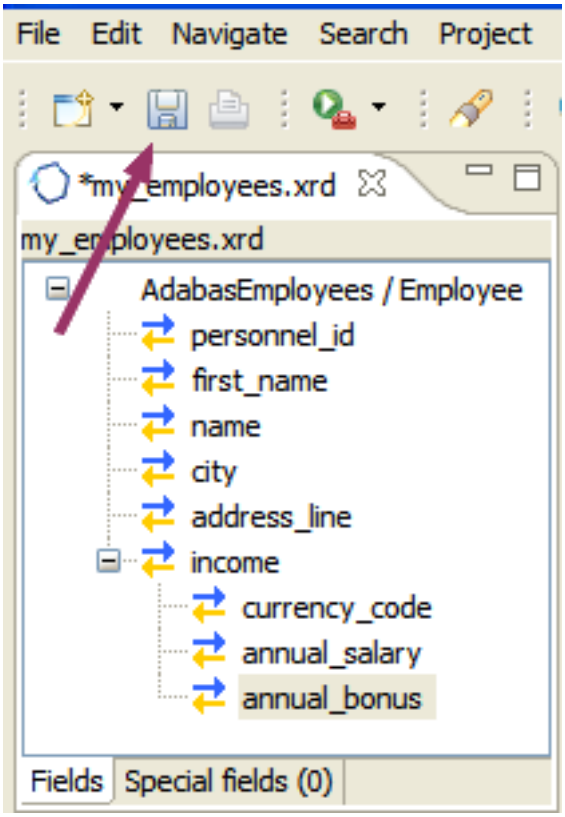





max Occurs >0 on the PE-field level denotes a MU within a PE.




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



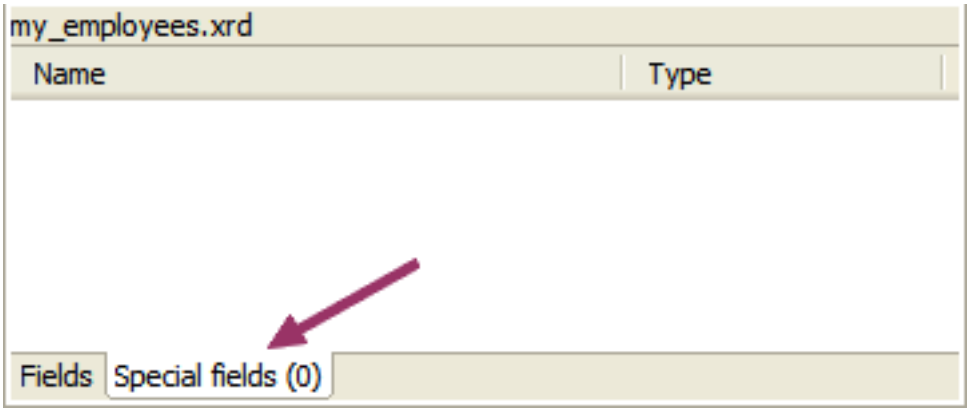

9. **Export** the DataView to the target SOA Gateway server.

## Editing "special" fields

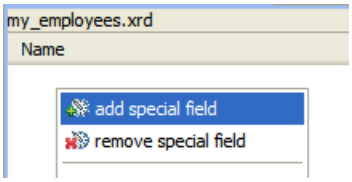
The following "special" fields can be defined for a DataView

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

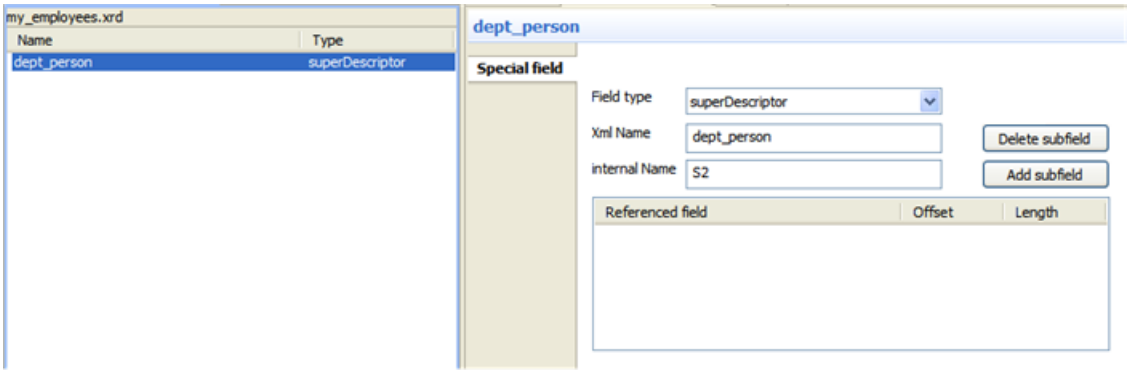
1. In the DataView editor window, click on the "Special fields" tab to add special fields



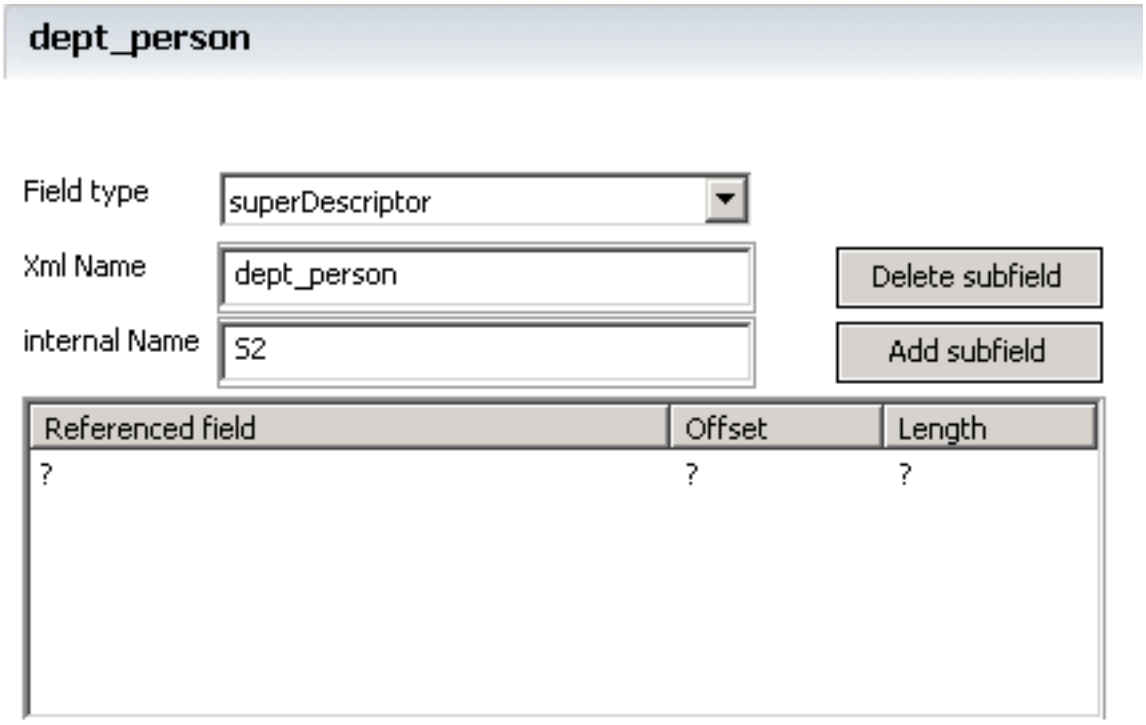

2. Right-click into the empty editor area, select "add special field"



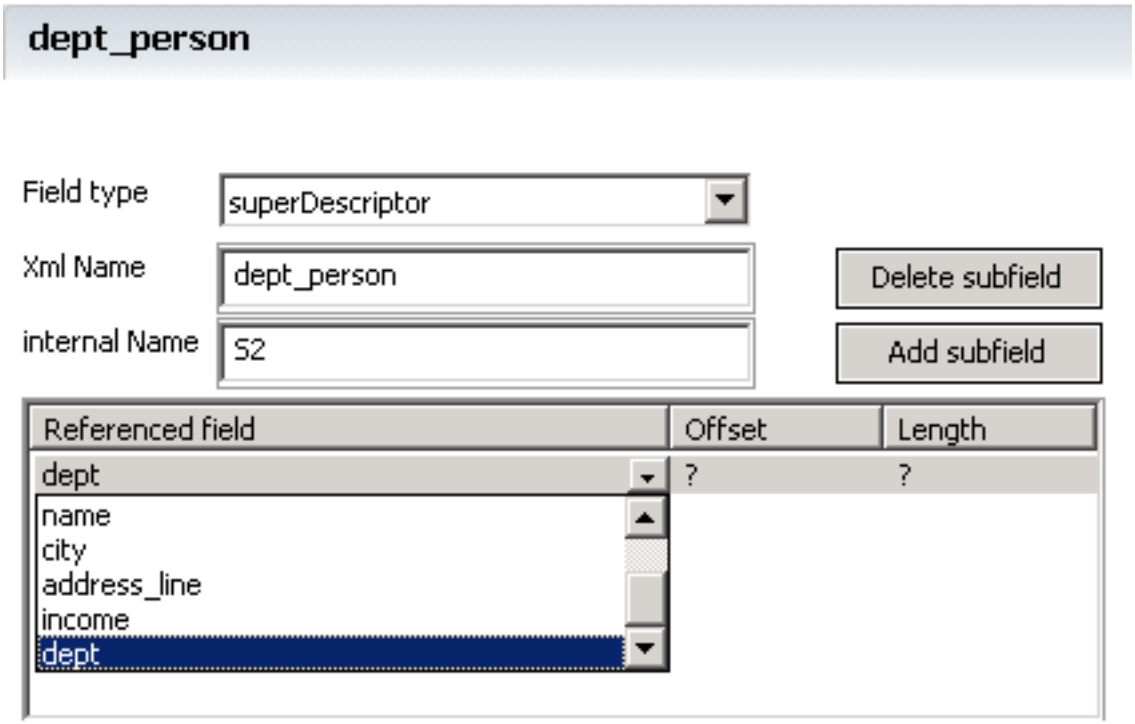

3. For example we define a special field called "dept_person" with an "internal name" (the Adabas short name) of "S2", being of type "superDescriptor"




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

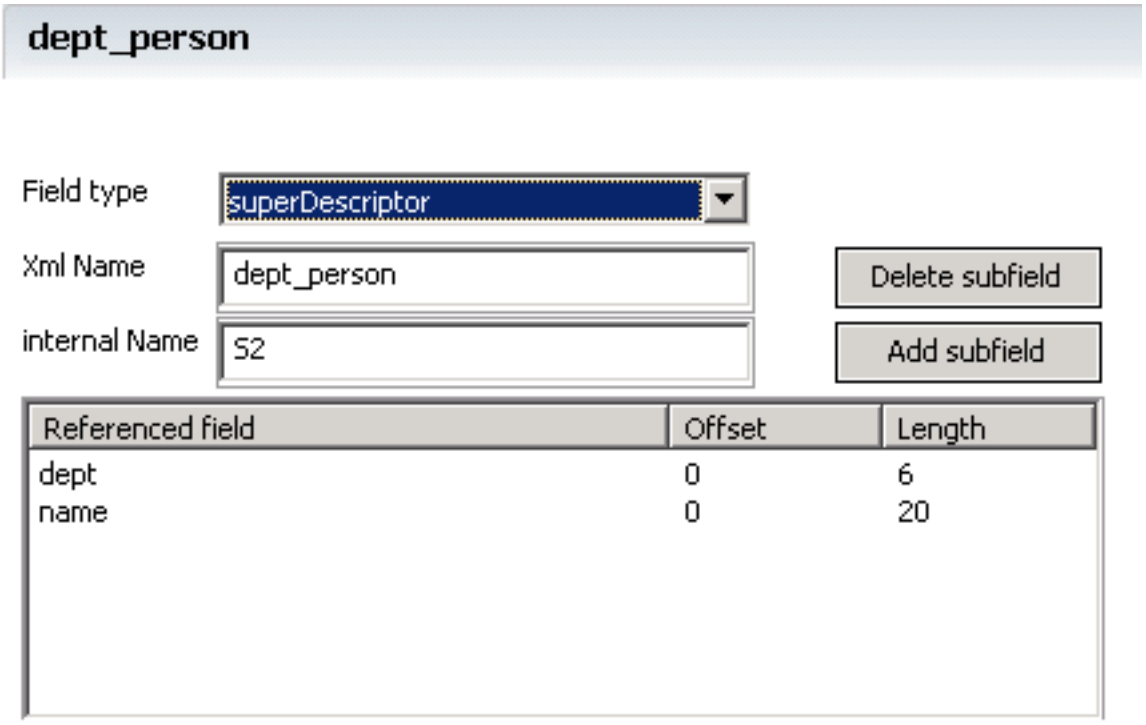



5. Click on the field value under the "Referenced field" heading, select the field to be added from the list of fields in the dropdown-box. Here we select the "dept" field



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




## Enhanced Type Conversion

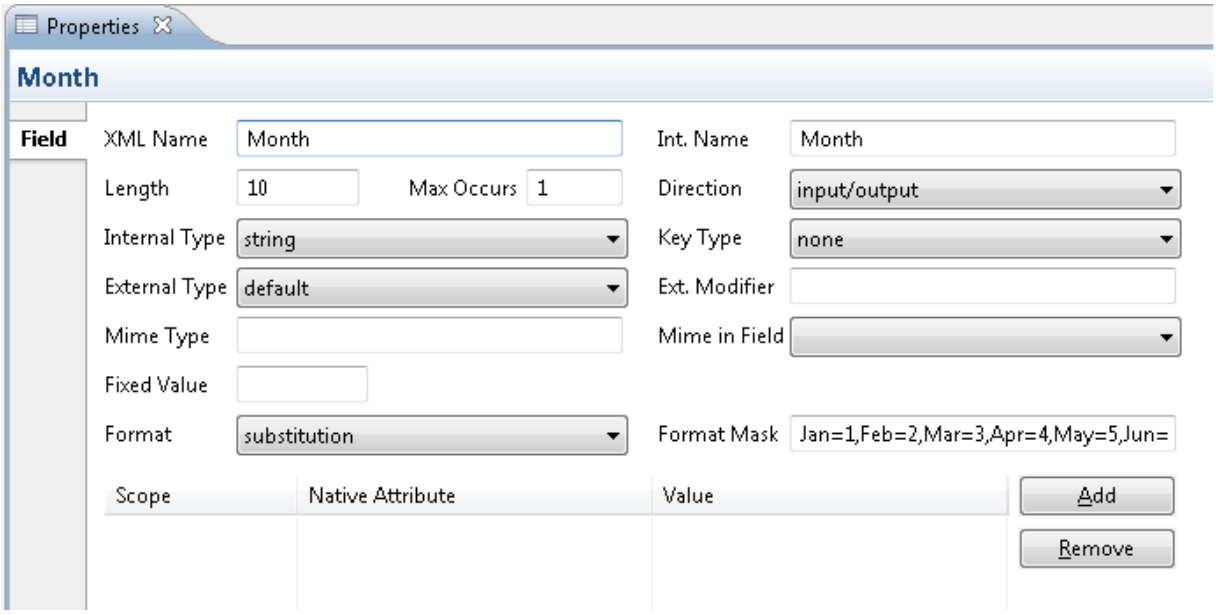
This facility allows the conversion of a string value to an integer equivalent and visa versa. This is analogous to an enumeration i.e. for Jan substitute 1, Feb substitute 2 etc.

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




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

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

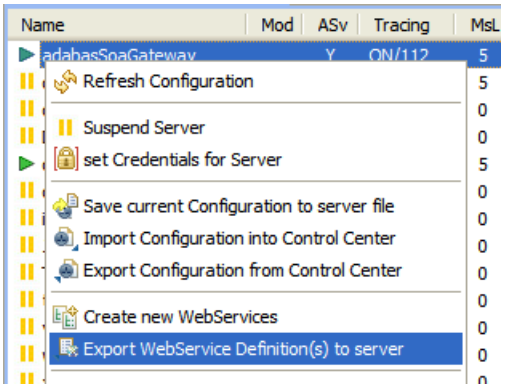
## Exporting a DataView to a SOA Gateway server

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

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

### Using the server based export function

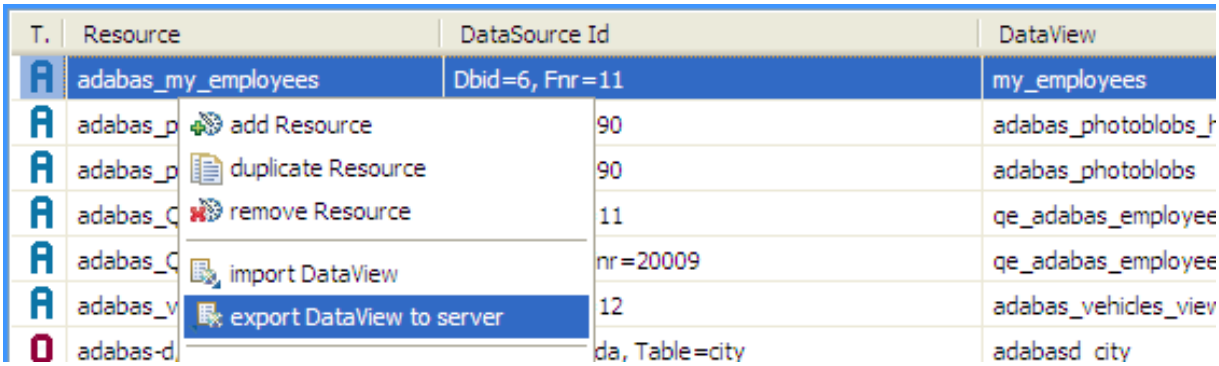
1. Select **Export Resource Definitions** from the context menu of the server you wish to export to.




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.




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.

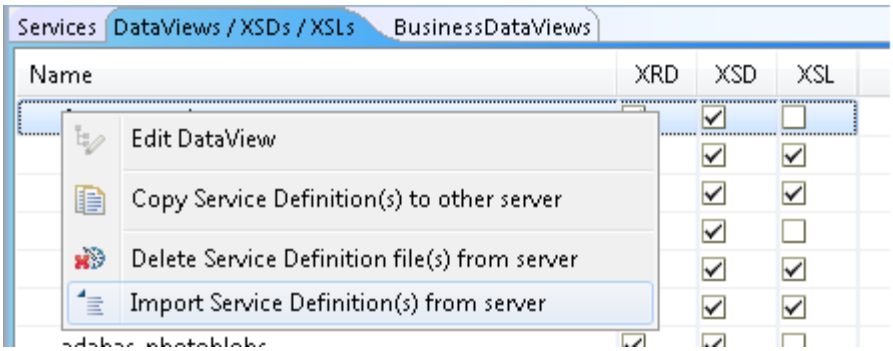
   Reply **Yes** if you want to use the newly exported DataView immediately, that is as soon as all requests currently using that DataView have completed, the new copy will be used for all subsequent requests.

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

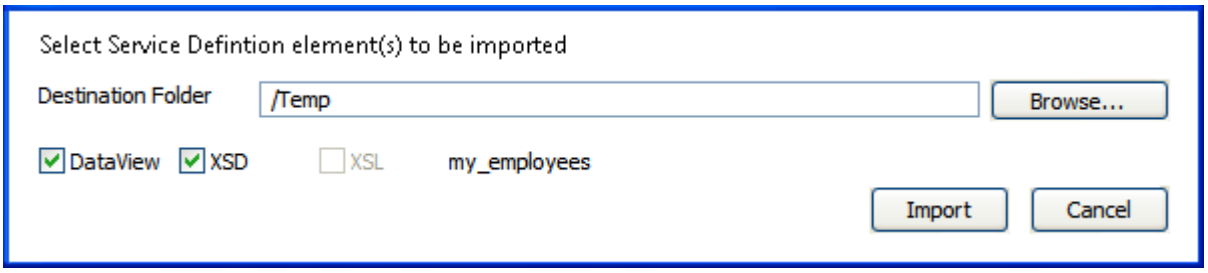
## Importing an existing SOA Gateway DataView

To import an existing DataView:

1. From the SOA Gateway 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"



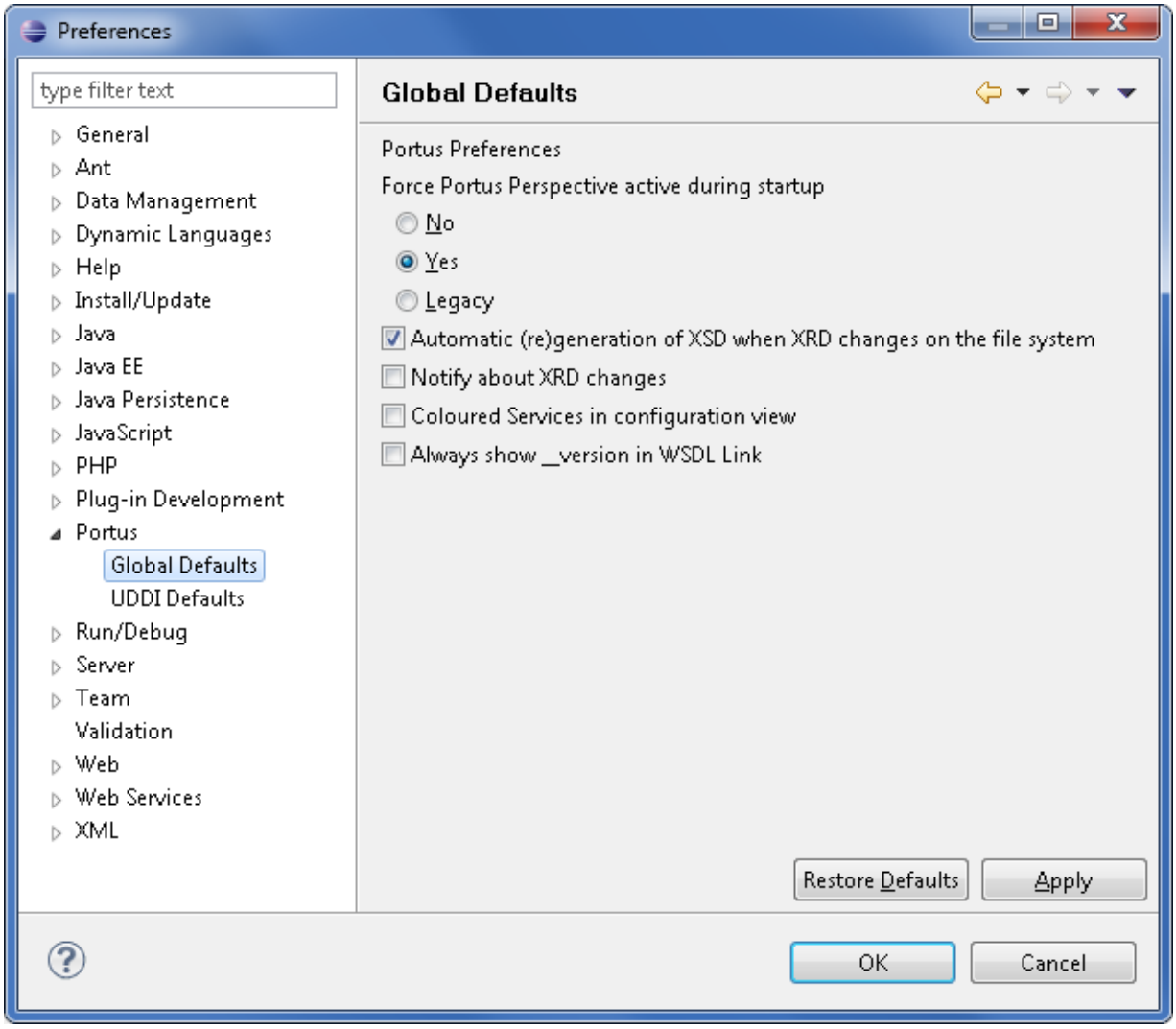

## Creating a XML Schema for a DataView

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 SOA Gateway 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 SOA Gateway 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.

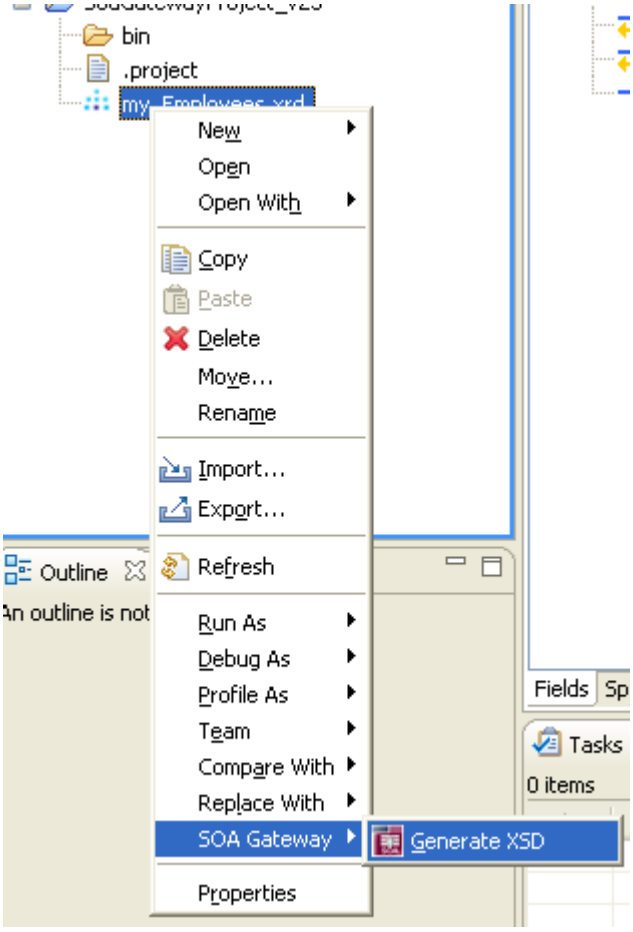
A XML Schema for a DataView can be created

- 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 -> SOA Gateway -> Global Defaults.




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

  Select **SOA Gateway** 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 extenstion ( xrd versus xsd )

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

# 22 Using SOA Gateway with Adabas


- Creating a SOA Gateway DataView from an Adabas FDT ..... 213
- Create DataView(s) from a SYSOBJH unload file ..... 213

SOA Gateway 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 SOA Gateway 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 SOA Gateway Service can be used, the mapping between the physical layout (the Adabas FDT) and the SOA Gateway (XML) representation must be in place. This mapping is called the SOA Gateway '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 SOA Gateway DataViews for Adabas resources

1. **Dynamically create a DataView from an Adabas FDT**
2. **Create DataView(s) from a SYSOBJH unload file**

## Creating a SOA Gateway 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 SOA Gateway 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 SOA Gateway Adabas resources can be generated/imported from SYSOBJH extracts.

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

# 23 SOA Gateway Service Creation

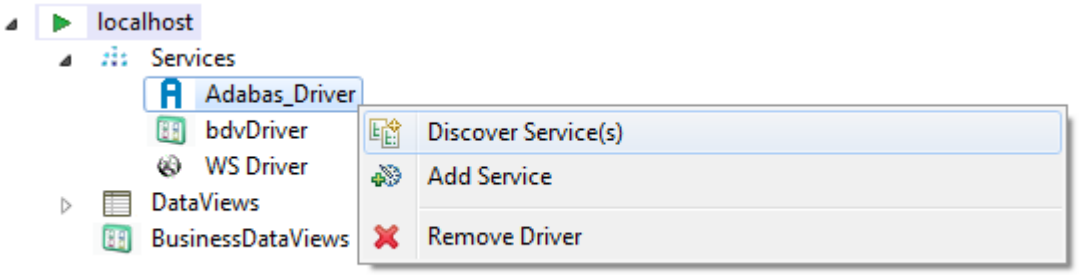

- Initiating Service creation ..... 216
- Creating Service(s) from Adabas ..... 218
- Assigning a different DataView to existing service ..... 219
- Creating Services from a SYSOBJH extract file ..... 221
- Creating Service(s) from 3rd party web services ..... 224

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 SOA Gateway, 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:

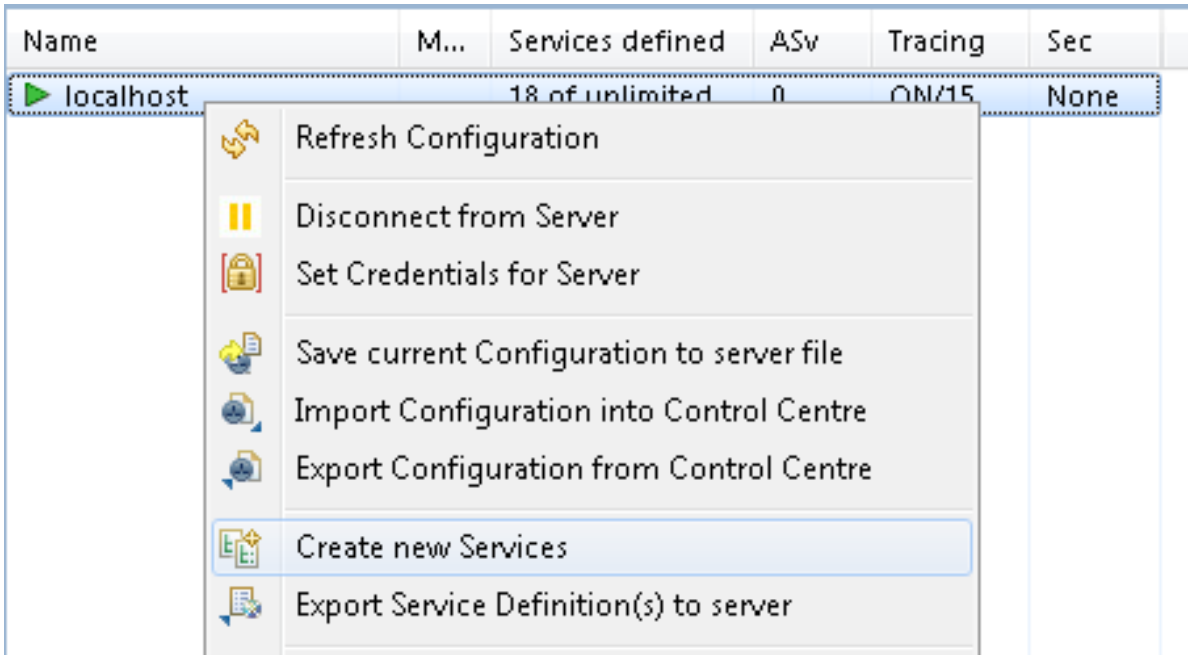



   Discover Service(s) will start the discovery wizard and request the specific parameters required for the selected driver type.

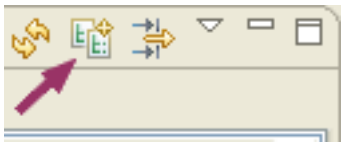
   Add Service will ask you to name the service and you can then manually set the various parameters as per your particular system.

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:

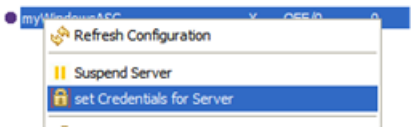


- The next step is to select the driver to be used as the source.

  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:

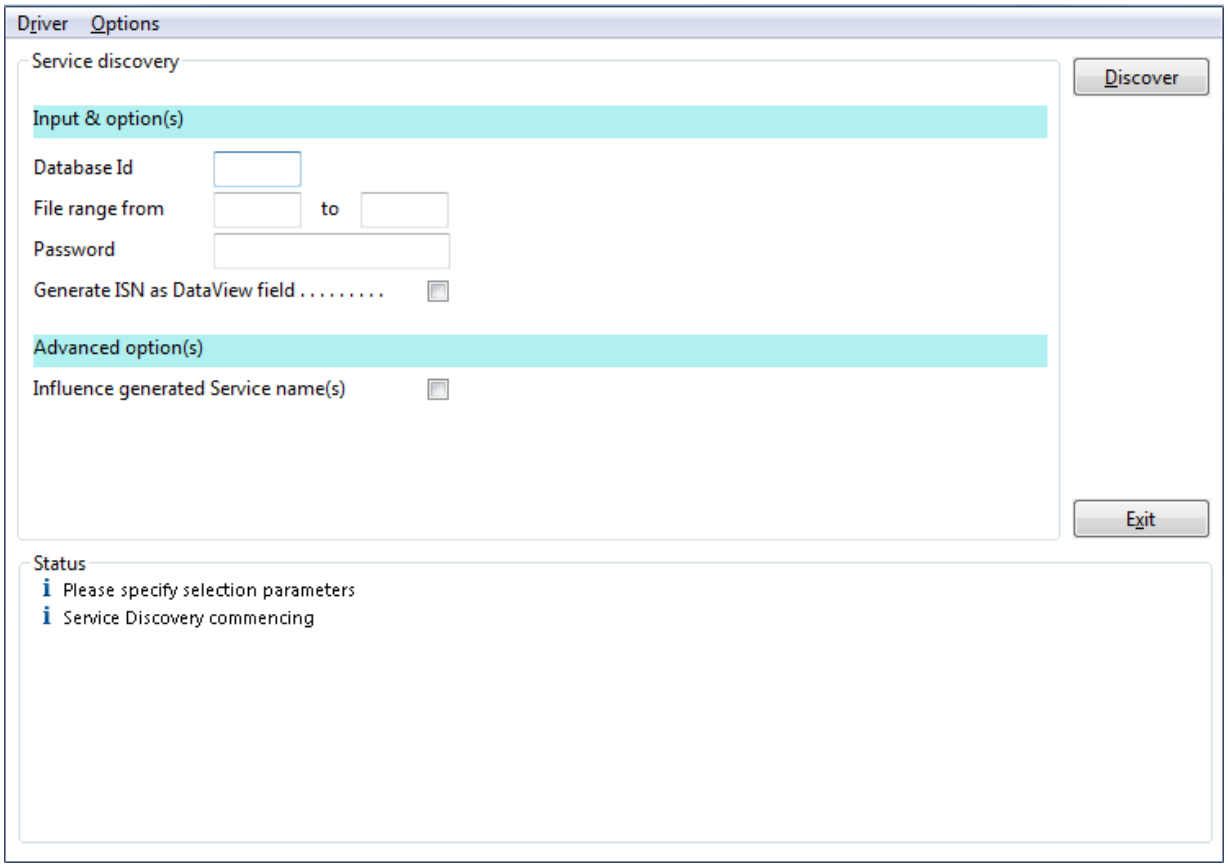



Specify the credentials - User ID and Password - in the *SOAP Security* section, click *Set*

Status and error information can be retrieved later on from the SOA Gateway Action Log

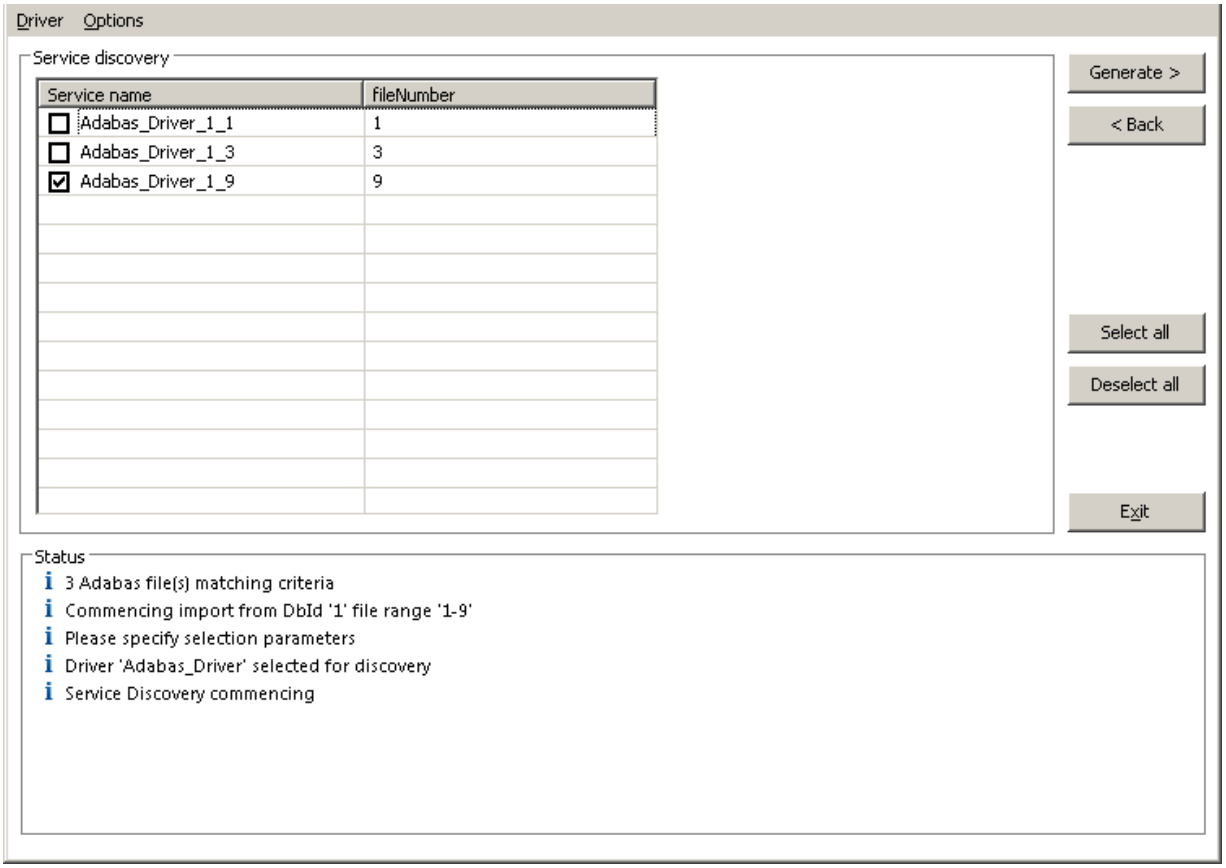
## Creating Service(s) from Adabas

- 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:




**Note:** Discovering the full range of possible file numbers (1 - 5000) may take substantial time and adversely affect performance of both the SOA Gateway 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.

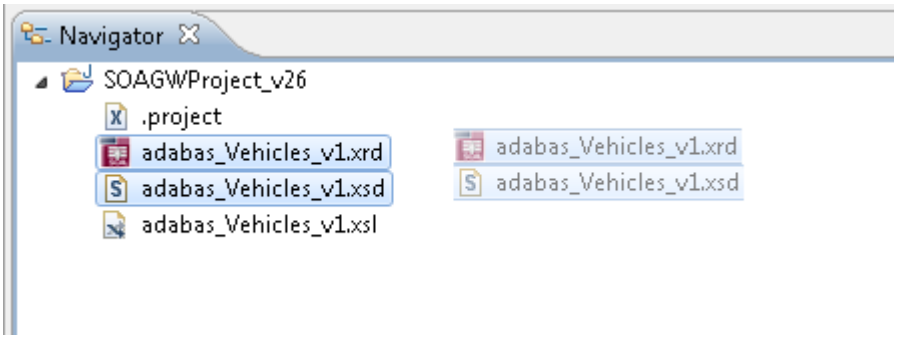
**Notes:**

1. The Adabas "long names" DataViews are available *on the SOA Gateway Server* where Adabas is licensed.
2. These can be imported using the Legacy Perspective here
3. These can be imported using the SOA Gateway Administration Perspective here.

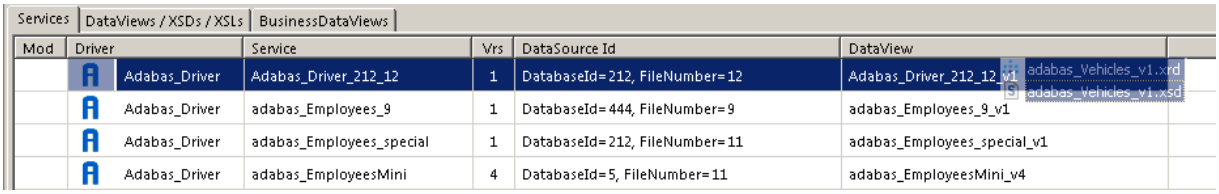
- OR the pre-created (Business)DataView is available on the server already
- The web service you wish to modify has already been created.

If the above assumptions are OK, follow these steps:

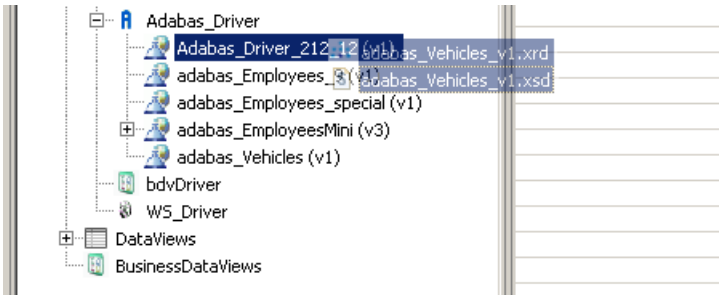
1. Drag the XRD and XSD (if available) files from the Navigator View.




2. In either perspective, drop the files onto the service you wish to modify




OR




In the SOA Gateway Perspective a new (Business)DataView can be assigned by dragging it from the respective Views section and drop it onto the service.

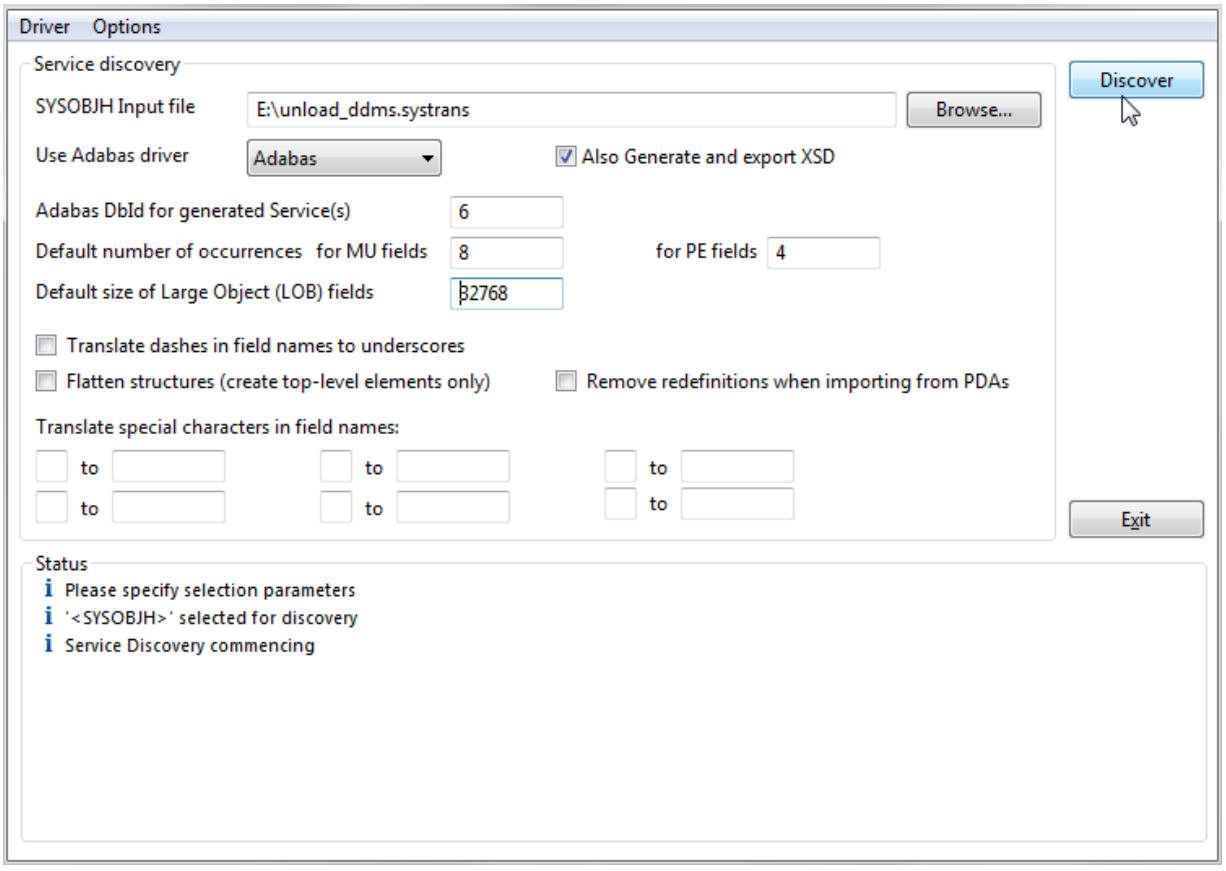
3. Right-click the serverice and select the *Refresh* function

4. The WSDL should now reflect the DataView.

## 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).



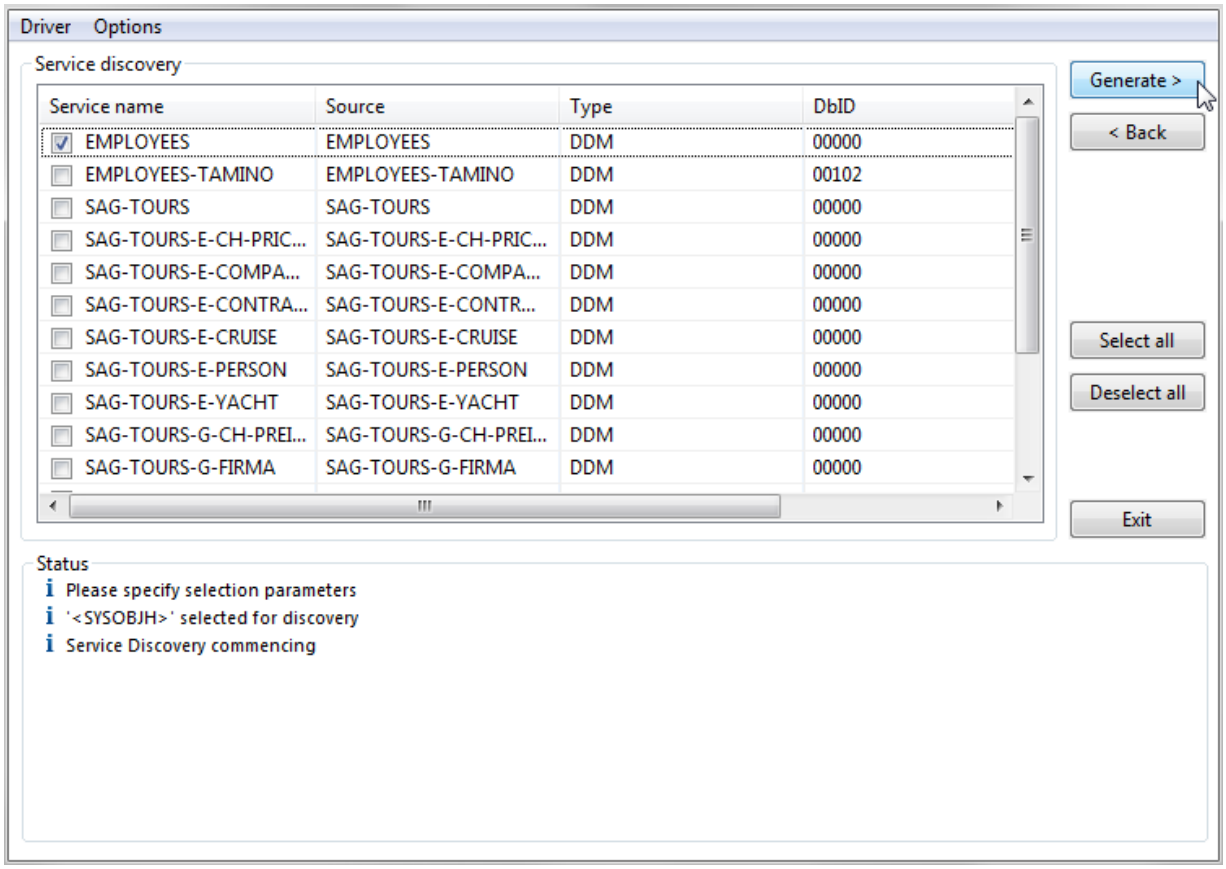

A number of options can be specified here

- 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

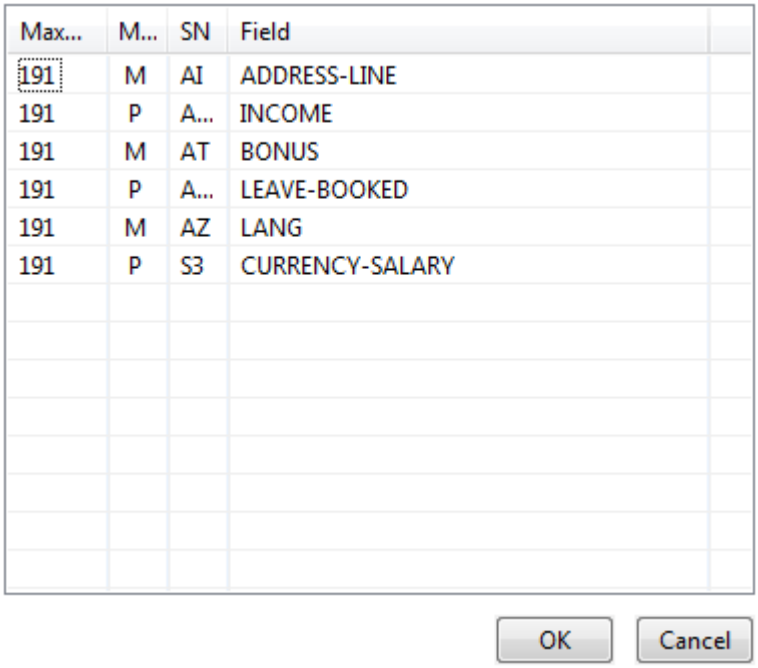
- if you wish dashes ('-') in file and element names to be translated to underscores ('_'). This may be desirable where the target programming language used to access SOA Gateway Resources do not accept dashes in field names, an example being PHP.

- Click the *Discover* button
- Select the DDM(s) to be turned into Services, click the '*Generate* >' button:

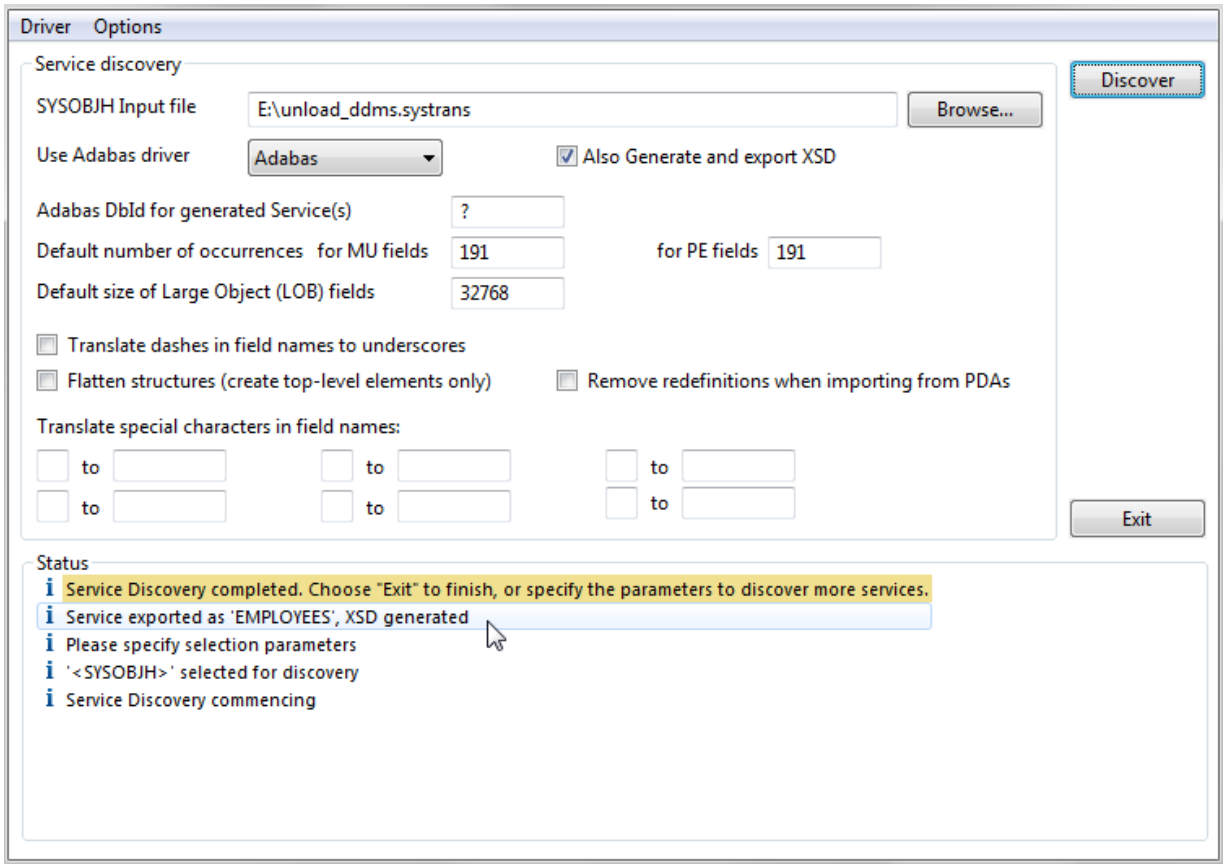



- For each DDM selected, which contains MU and or PE fields, a dialog will request you to specify the maximum number of occurrences expected.

**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).




- The status pane indicates what has been generated, you can now start the process again or exit the wizard.

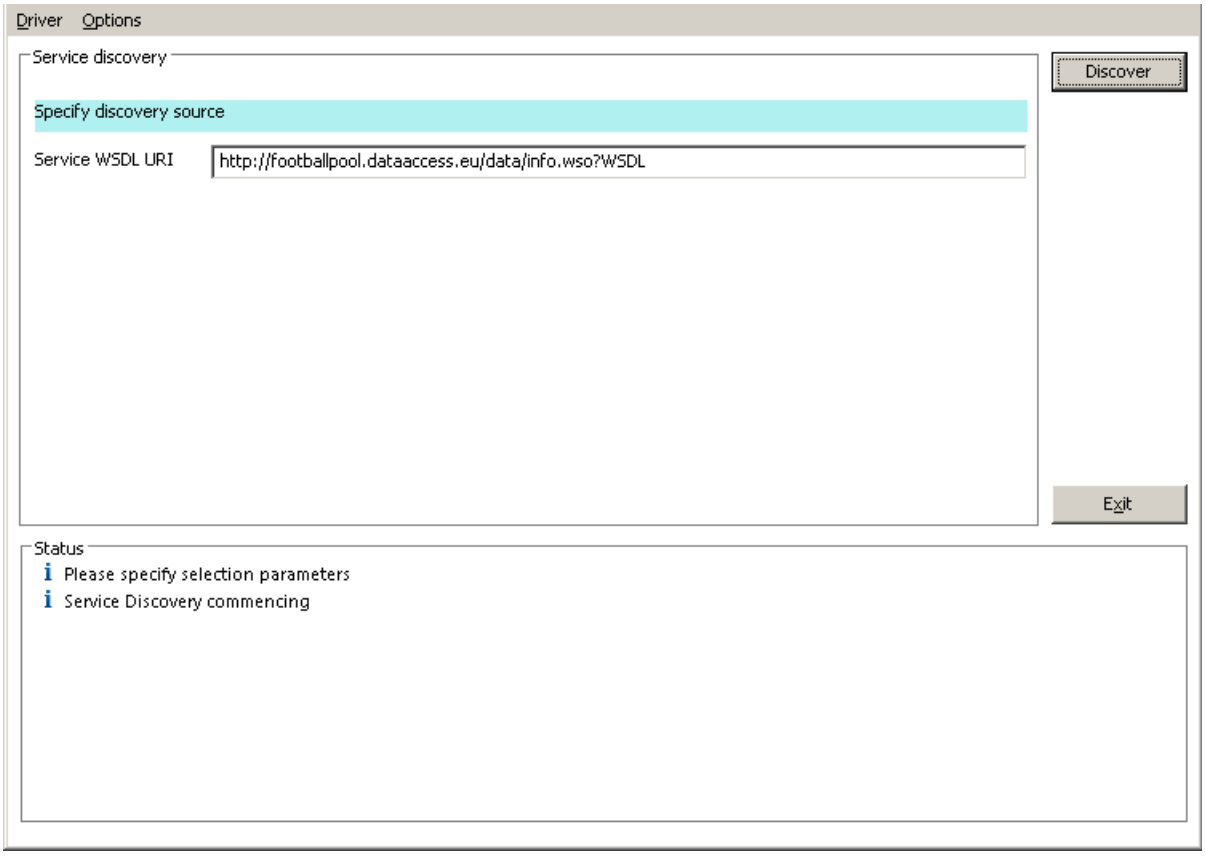



- The requested Services and Service definition files (DataView(s) and XSD(s)) have been added to your server configuration
- The Wizard is still active and ready for further discovery, click 'Discover' to start the process again, or 'Exit' to dismiss the wizard.

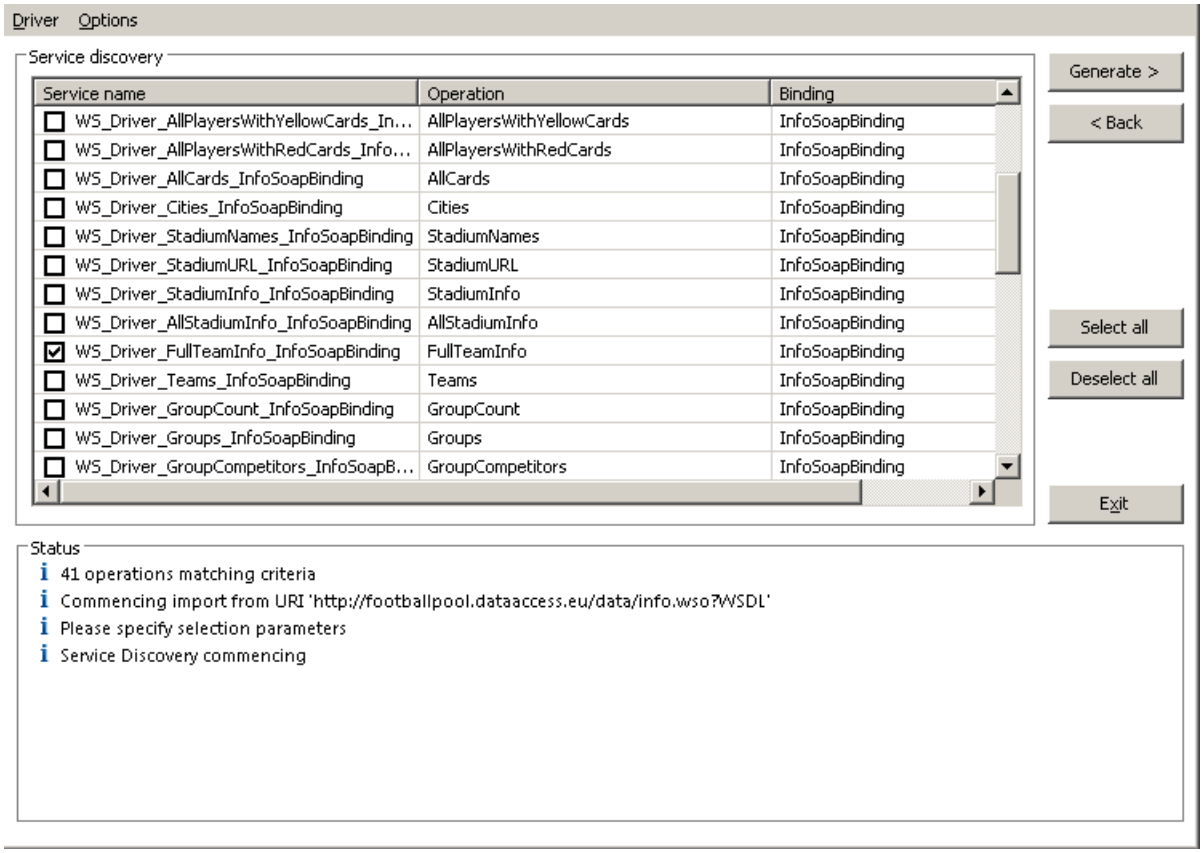
## Creating Service(s) from 3rd party web services

- 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:

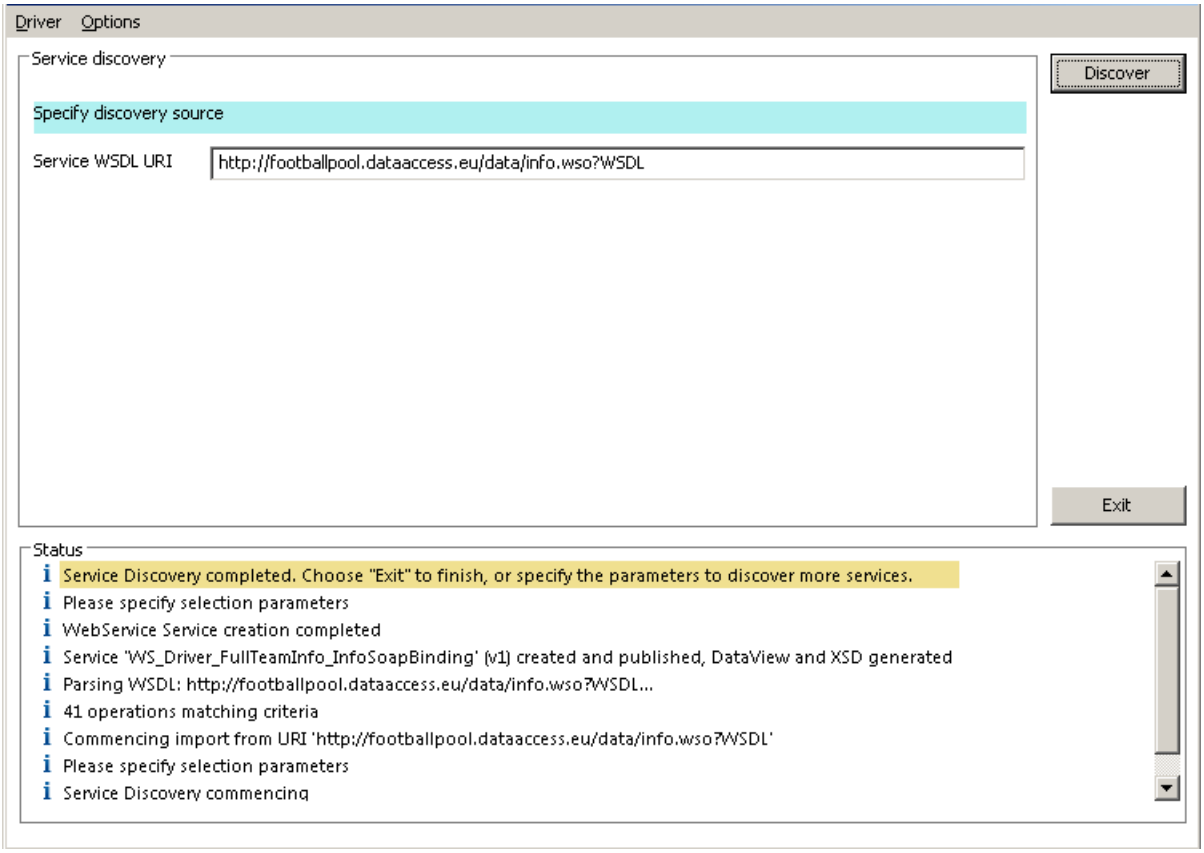



- The Status section will display the progress of each stage of the web service discovery and the Service discovery section will display the following information retrieved from the service description:




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.

# 24 Using SOA Gateway to access Web Services


- Introduction ..... 230
- Accessing Web Services ..... 230

Web services are Web based applications that use open, XML-based standards and transport protocols to exchange data with clients.

## Introduction

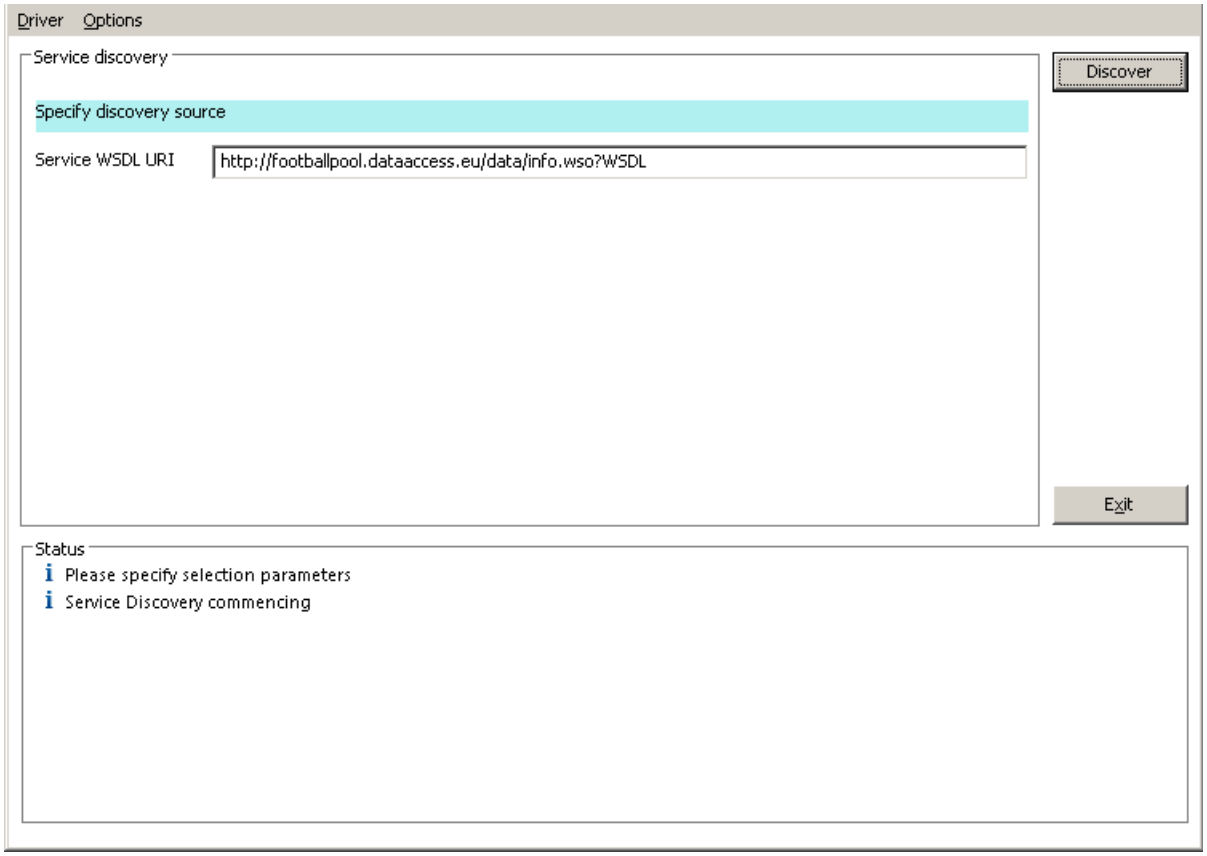
SOA Gateway can be used to access other web services. The web service driver queries the WSDL exposed by these services and creates a service to enable these to be called.

## Accessing Web Services

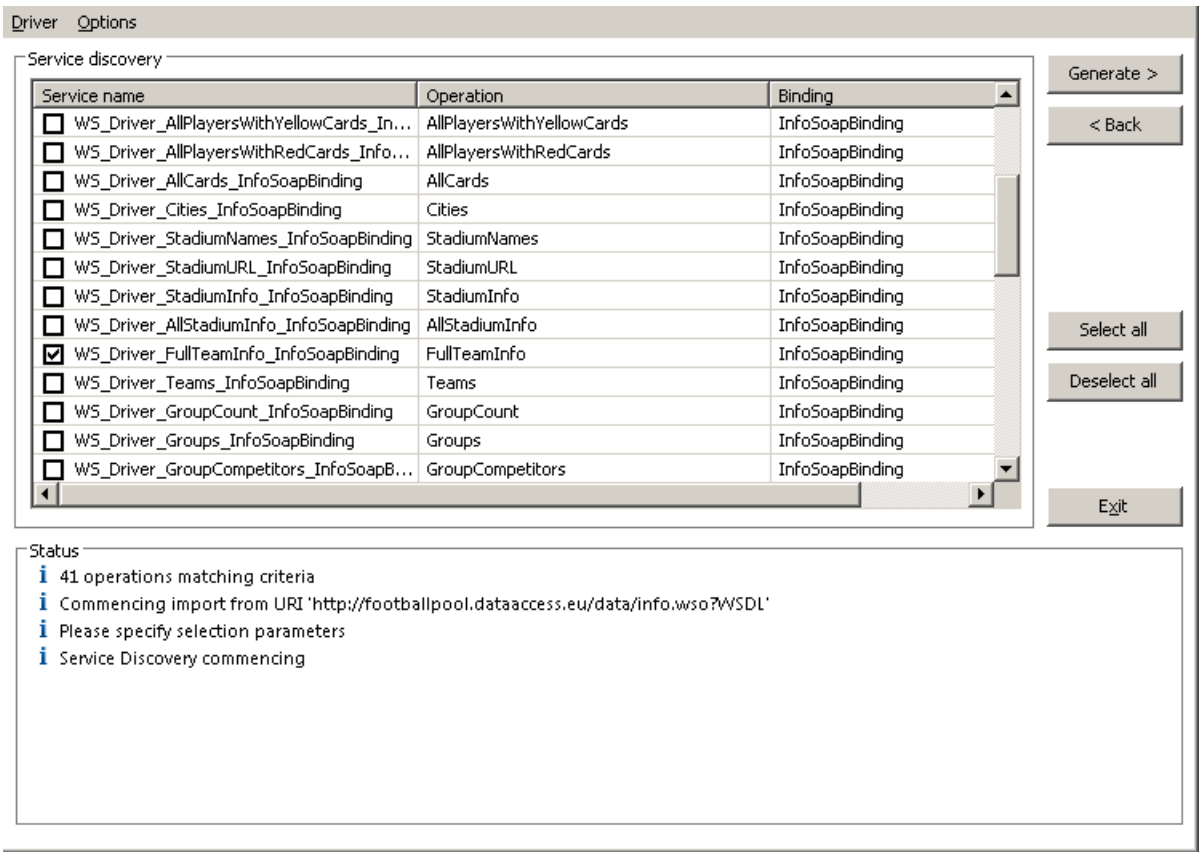
In this example we will use the web service http://footballpool.dataaccess.eu/data/info.wso.

- In the Control Centre SOA Gateway 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:

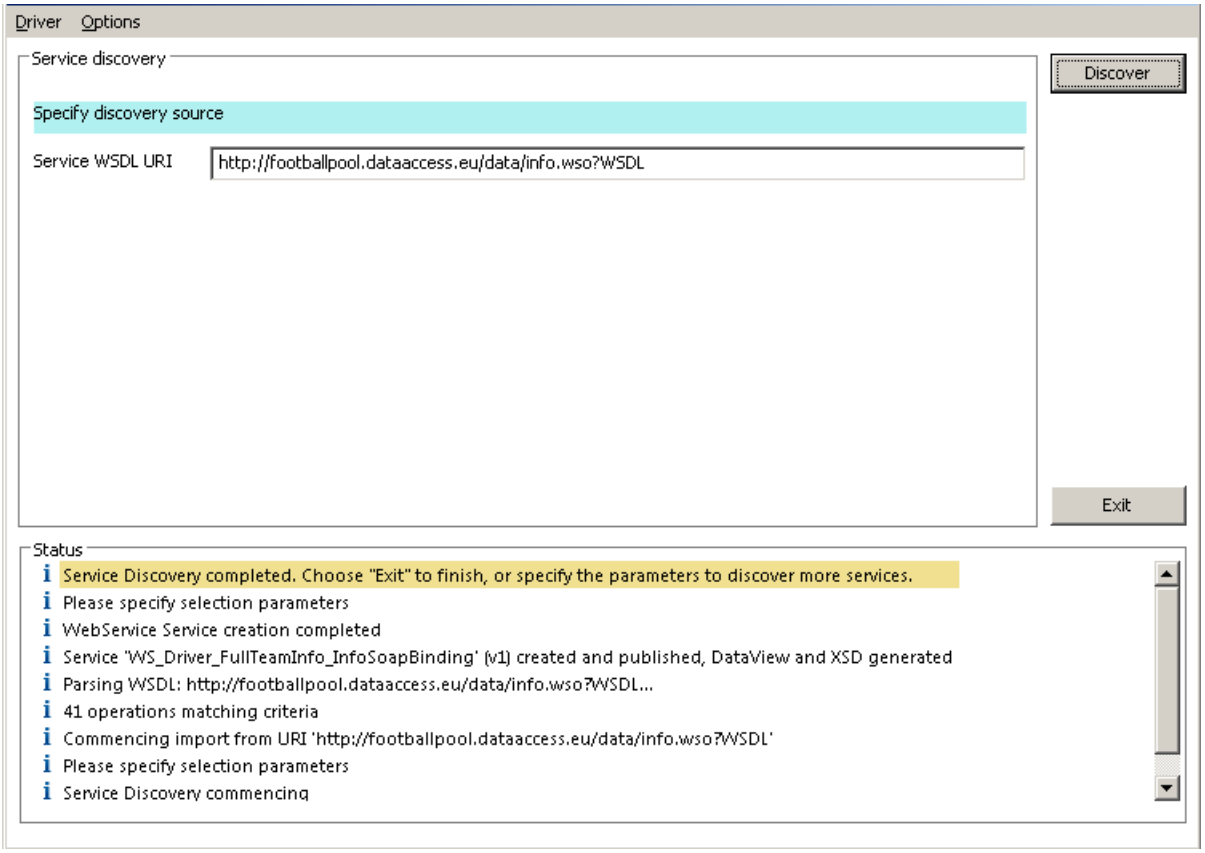


- The Status section will display the progress of each stage of the web service discovery and the Service discovery section will display the following information retrieved from the service description:

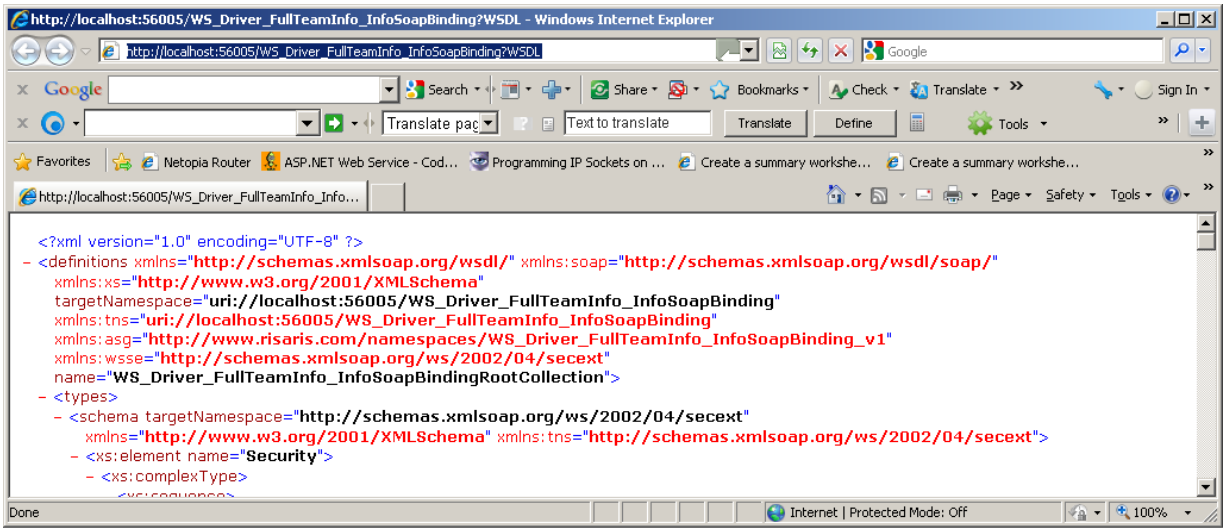



- Use Select all to choose all operations or select on an individual basis. For clarity we have chosen the FullTeamInfo operation only. When chosen click the Generate button.

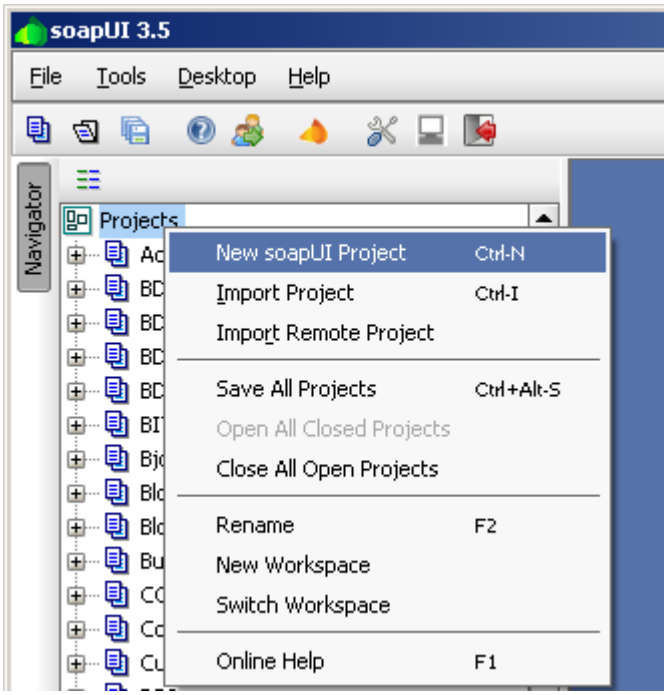
- The Status section will be updated with the progress of the generation process, listing the operations selected and the results for each one:



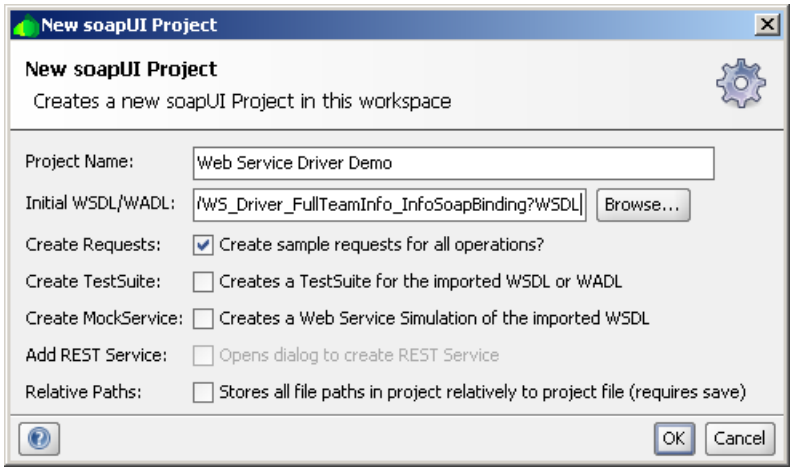

- The Wizard is still active and ready for further discovery, click Discover to start the process again, or Exit to dismiss the wizard.
- The generated Service(s) will be listed under the Web Service driver.
- To view the WSDL generated select 'WSDL URL is..' : which can be found in the Properties View for the service. This will open in your default browser e.g.



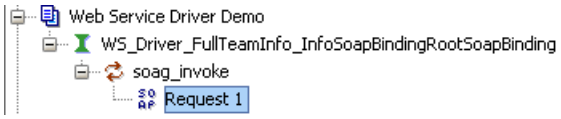

- Here we are using soapUI to demonstrate how this web service can be invoked. Open soapUI, right click on Projects and choose New soapUI Project :



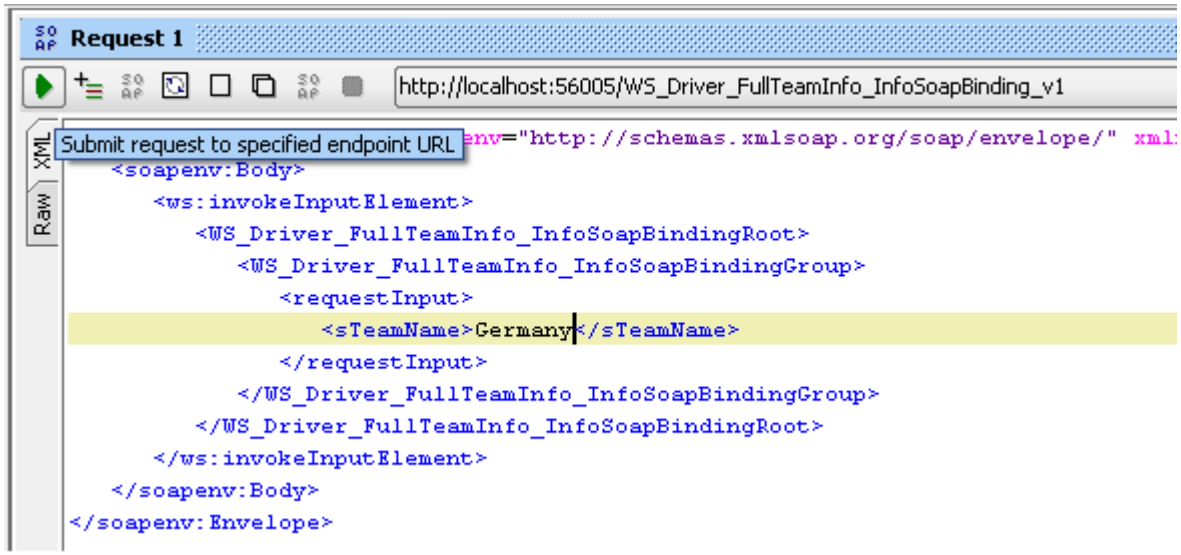

- Enter the name of the project and set the WSDL to the value copied previously :



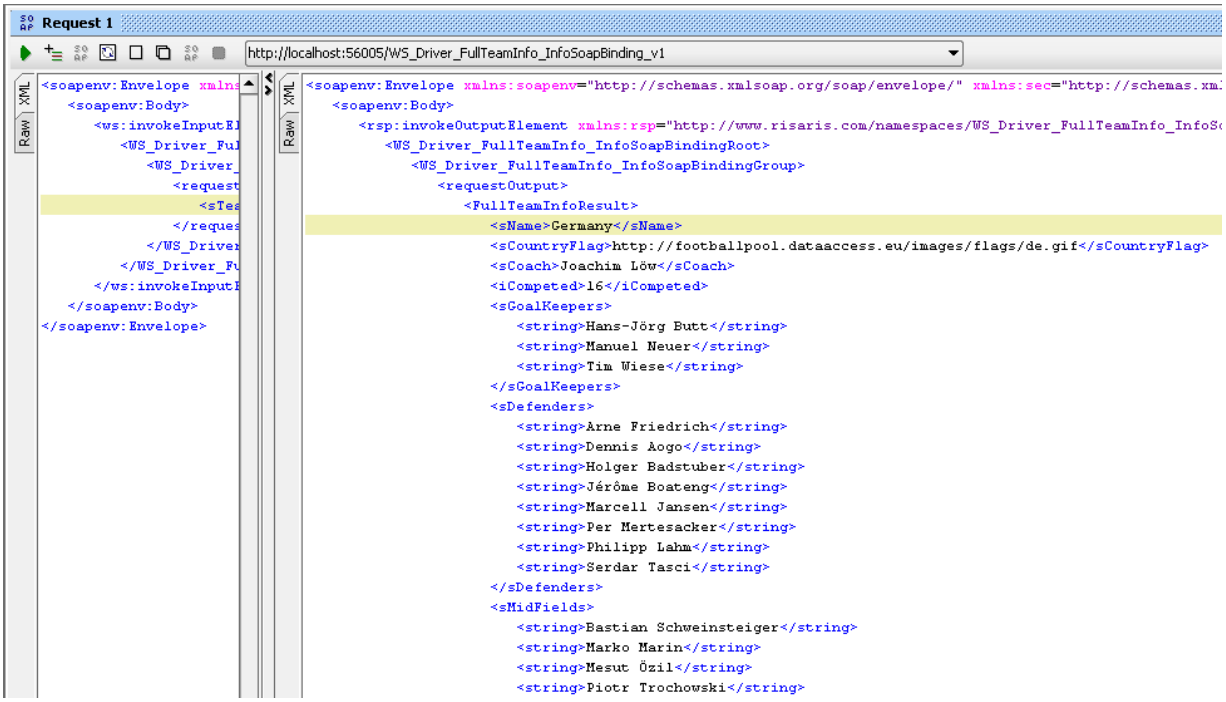

- When the import is complete double click on the Request 1 entry to open the dialog :




- 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:




Congratulations! You have now accessed a Web Service using SOA Gateway!

# 25 Message Oriented Middleware Support in SOA Gateway


- Introduction ..... 238

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 SOA Gateway supports.

In particular the systems supported by SOA Gateway are IBM WebSphere MQ and Software AG EntireX Broker.

## Introduction

There are 2 ways to integrate a messaging sytem with SOA Gateway.

1. To monitor for SOA Gateway specific requests, to process these and issue a response.
2. To log Usage Governance Data.

Tutorials are available which demonstrate these 2 capabilites.

The first tutorial uses SOA Gateway to write usage governance data to a messaging system and a second SOA Gateway 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 EntireX and call it using SOAP over EntireX. See here for details on this.

# 26 Usage Governance


- Introduction ..... 240
- Governance Data ..... 240
- Service collection ..... 241
- Controlling Collection ..... 242

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 SOA Gateway.

## 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

SOA Gateway 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 SOA Gateway server instance. The following information may be collected about each service.

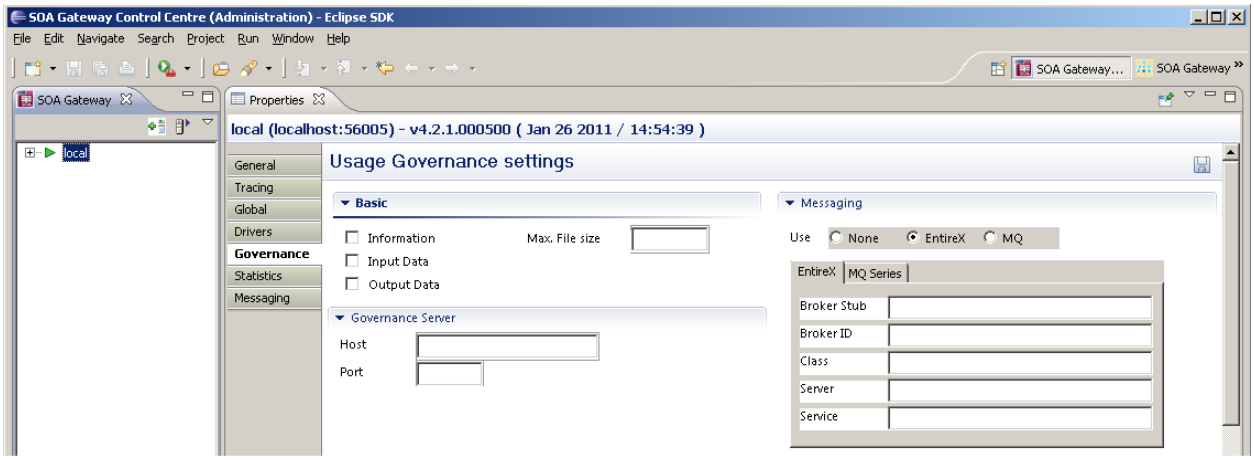
- No governance information about the service at all.
- Detailed access information about the service. This will record the access data (e.g. date and time of access, identity of caller, location of caller etc.) for each request made to the service.
- Request information provided to the service by the 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 SOA Gateway server instance. These may then be overridden at the service level.

### Global Collection

This is set on/off via the SOA Gateway 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 SOA Gateway 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 SOA Gateway 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.

# 27 Lifecycle Governance


- Concept ..... 246
- Web Service Versioning ..... 247
- Web Service Status ..... 248
- Practical Example ..... 248

This section describes how the SOA Gateway 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 SOA Gateway 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 SOA Gateway 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. SOA Gateway 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 “adabas_EmployeesMini” sample service that can be created when a new Adabas driver is defined (check “Create Sample Services” box). The process is the same for any other service defined in SOA Gateway.

**Important:** The SOA Gateway legacy perspective should not be used for configuring services with lifecycle governance. All further references to the Control Centre assume the SOA Gateway Administration perspective.

## Concept

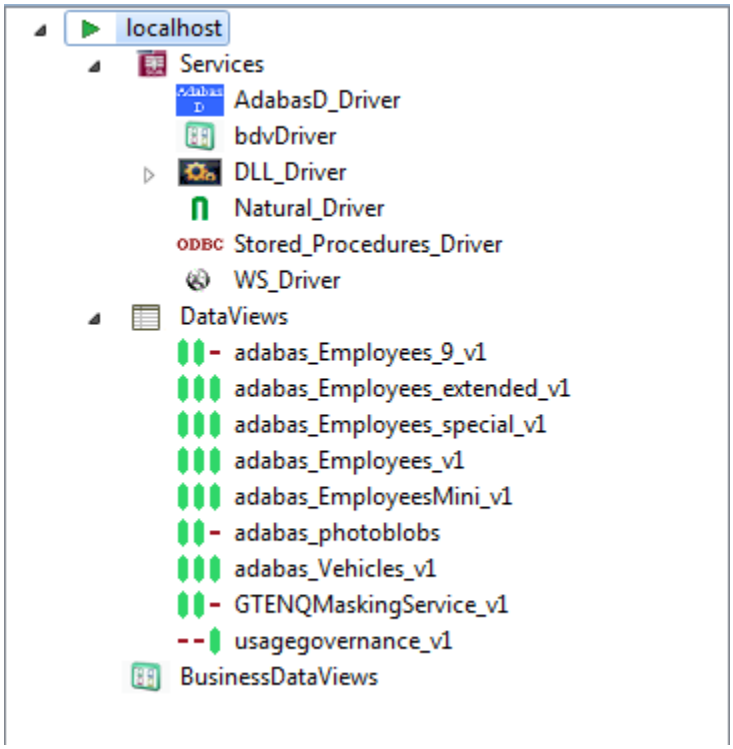
The basic concept of lifecycle governance is that it allows changes to be made to existing services without forcing existing clients to upgrade. Each client that wishes to consume a web service starts with retrieving the service WSDL. Thus, SOA Gateway will return different WSDLs based on the different 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 SOA Gateway 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, SOA Gateway will control the allocation and incrementation of service version numbers.

## Web Service Versioning

Every web service that is created in SOA Gateway 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, SOA Gateway 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/adabas_EmployeesMini?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 SOA Gateway 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

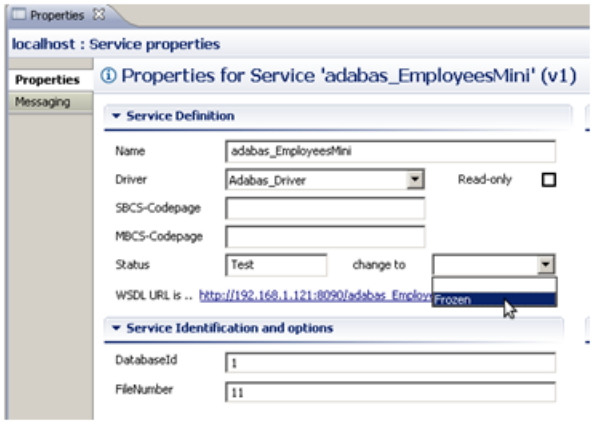
This status will be assigned to a service which is no longer active and has essentially been deleted. It is maintained in the configuration for historical purposes. If this service is invoked, the client will be returned a soap fault and a record logged that an attempt was made to use a historical version 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 SOA Gateway web services. Initially, the version number of the web service is “1”. This service can be modified has 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, bye 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

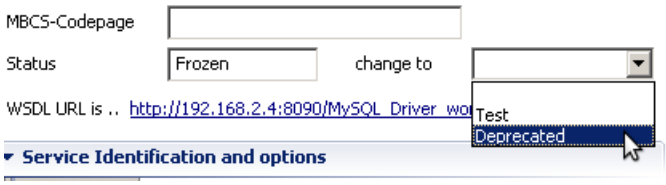
When a client requests the WSDL for this service, by default they will receive the version 2 WSDL. Thus new clients will get the most up to date version of the service. It is assumed that existing clients will only import the WSDL once (when version 1 was the most recent). To access the version 1 WSDL, provide the __version=1 argument.

```
http://host:port/adabas_EmployeesMini?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
```

# 28 HTTP Proxy Driver


- How to use ..... 252

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.

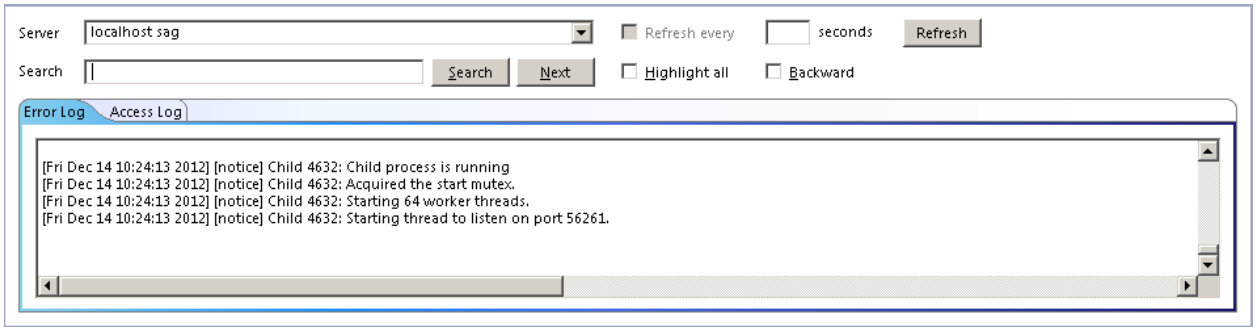
# 29 Monitor View

The Control Centre Monitor View allows the server log files to 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 -> SOA Gateway 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