Ostia Portus

The Portus Library
Portus Documentation

v4.3.1

Notice

Notice

Topics:

- *Trademarks*

This supersedes all previous notices.

COPYRIGHT LICENSE:

Trademarks

The following terms are trademarks of Ostia Software Solutions Limited in the United States, other countries, or both:

Portus®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Contents

Preface: Preface.. ix

Chapter 1: Concepts and Facilities..11

Why a Standards Based Service Oriented Architecture ?.. 12
The Portus approach..13

Chapter 2: Release Notes...15

Compatibility with other products.. 16
Known issues...16
New features...16
Features to be discontinued in future versions.. 17

Chapter 3: Installation.. 19

Installation Prerequisites...20
Overview... 20
Portus Control Centre prerequisites.. 20
Portus Server Operating System Prerequisites...20
Additional software prerequisites...21
Installing the Portus Control Centre.. 22
Prerequisites.. 22
Download and install Eclipse...22
Install the Portus Control Centre from an Update Site...22
Next Steps...23
Windows Server installation... 23
Introduction...23
Installation using the Deployment Wizard...24
Post install steps.. 29
z/OS Server installation...30
Using the Deployment Wizard to install on a z/OS host..30
z/OS Server installation...33
z/VSE Server installation... 36
Using the Deployment Wizard to install on a z/VSE host...36
Operating the Portus Server... 38
Linux/Unix Server installation.. 41
Using the Deployment Wizard for Linux/Unix type systems..41
Server Installation..47
Control Centre and Server Update..48
Using the Eclipse Wizard to find and install updates...48
Update Server - Windows.. 48
Update Server - non-Windows Platforms... 49
Server uninstallation guide..50
Uninstall Portus Server - Linux, z/Linux, AIX and Solaris...50
Uninstall Portus Server - Windows..50

Chapter 4: Administration...51

Introduction to the Portus Control Centre.. 52
The Control Centre..53
The Portus View..53

Drag-and-drop in the Portus View....59
Portus Service Configuration....61
The Monitoring View....66
The Portus Control Centre Action Log....67
Portus BusinessDataViews....68
Overview / Concepts....69
Using the BusinessDataView WebService....72
Working with BusinessDataView Objects....72
Additional features....74
Summary....76
Creating BusinessDataViews....76
BusinessDataView Editor hints....98
Server administration....105
Starting / stopping a Portus Server on Windows....106
Starting / stopping a Portus Server on z/OS....106
Starting / stopping a Portus Server on z/VSE....106
Starting / stopping a Portus Server on Linux, AIX and Solaris....106
Updating an expired license....106
WebService Creation....107
Initiating Service creation....107
Creating Service(s) from Adabas....107
Assigning a different DataView to existing service....109
Creating Service(s) from SQL databases....110
Creating Service(s) from Stored Procedures....111
Creating Service(s) from a SYSOBJH extract file....112
Creating Service(s) from Natural....115
Creating Service(s) from 3rd party web services....119
Using Portus with 122
Portus Statistics....123
Server Setup....123
Service Setup....125
Driver Setup....125
Monitoring....126
Reports....127
XML....127
WSDL....129
SOAP....130
URL....130
Example....131
Service Import....133
Governance....136
Usage Governance....136
Lifecycle Governance....139
Portus Security....142
Advanced Topics....143
How to install a server on a remote Windows....143
Running multiple instances of Portus....143
IPv6 Considerations....144
Reducing memory footprint using streaming....145

Chapter 5: Reference....147

Tutorials / Samples....148
Adabas Tutorials....148
MySQL Tutorials....204
SOAP over IBM MQ Series Tutorial....217
Usage Governance Tutorials....222

Natural Samples....226
SOA Resource Access....228
Creating a Stylesheet for your Portus Data....237
Create HTML page from City XML....238
Data Masking....243
Example....244
Internationalization....244
Setting codepages....244
Which codepage do I use ?....244
SOAP versus REST differences....245
Troubleshooting....245
Performance Hints....246
Frequently Asked Questions....247
How do I active the Software AG sagenv file post Portus Installation ?....247
How do I modify the machine identifier in the JESMSGLG?....247
Introduction to Eclipse....247
Elements of the Workbench....250
Creating Resources....251

Chapter 6: Troubleshooting....257

Portus Client side issues....258
Portus Server side issues....259
Portus Control Centre (Eclipse) issues....260
Portus Installation issues....263

Chapter 7: Appendix....265

Tut_02_List.java....266
First steps (ex01_SoaGatewayFirst.php)....267
List Employees (ex02_SoaGatewayEmpList.php)....268
List Employees descending (ex02a_SoaGatewayEmpListDescending.php)....270
List Employees sorted (ex02a_SoaGatewayEmpListSorted.php)....272
List by Sub Descriptor (ex06_SoaGatewaySpecial_SubDescriptor.php)....274
List by Super Descriptor (ex06_SoaGatewaySpecial_SuperDescriptor.php)....276
Add an Employee (ex03_SoaGatewayEmpAdd.php)....277
Get an Employee (ex04_SoaGatewayEmpGet.php)....278
Delete an Employee (ex05_SoaGatewayEmpDel.php)....279
A simple PHP form for accessing Adabas (ex10_SoaGatewaySimpleForm.php)....280
All-in-one PHP form accessing the Employees file (ex15_SoaGatewayUpdateForm.php)....281
empMiniList.rb (Ruby)....285
ASGDemo.cs (C#)....285
Get a record, save the LOB to a file (wsf_lobsGet.php)....286
MySQL_City.cs (C#)....287
SOAP over IBM MQ Series (SOAPEntireXTut.java)....288

Preface

Preface

This book explains each tasks separately. You can jump in anywhere, get the information you need, and complete your work.

You can use this book to master tasks if you wish or just keep it handy as a resource so you can focus on other issues.

Chapter 1

Concepts and Facilities

Topics:

- *Why a Standards Based Service Oriented Architecture ?*
- *The Portus approach*

A guide to Portus concepts

Portus will enable an organization to save a large percentage of their integration budgets by using a productized approach to what has, until now, been an expensive, people intensive, repetitive process to get at existing data and business logic.

Portus enables access to these assets from all of the most popular languages, products and Enterprise Service Buses (ESBs) available today. This capability has been promised a number of times over the years and has not been delivered, however, with the advent of W3C standards such as WSDL and SOAP, and specifications such as the REST architecture, this promise can now be delivered.

If you are a little skeptical because you have heard all this before, read on to see how Portus achieves this.

Why a Standards Based Service Oriented Architecture ?

A Service Oriented Architecture (SOA) can mean different things to different people. Ultimately, in its loosest form, if a program can be called in some way and give a result, it is providing a service and thus the infrastructure could be called ‘Service Oriented’. At the end of the day, many organizations have been doing this for a long time by implementing the business logic and screen logic separately so that the business logic may be called from multiple places so a SOA is nothing new apart from the terminology.

The nirvana that most organizations wish to get to is where their products and applications can interface seamlessly with each other on multiple platforms. This is sometimes achieved by what are called ‘Web Services’. However, this term is often also applied loosely on the basis that if a service is delivered via the Web it is a ‘Web Service’. These are generally proprietary, non standard Web Services which can only be integrated once you know the proprietary mechanism they are used.

What does offer the integration that organizations desire are Web Services that are implemented using W3C standards like WSDL and SOAP or specifications such as the REST architecture. The implementation of interfaces using these standards will enable applications to seamlessly interface with each other without any programming requirement. Portus specifically only offers access to data and business logic using recognized standards to enable simpler integration. Any reference to Web Services in Portus documentation set will mean standards based Web Services implementing using WSDL and SOAP or using the REST Architecture.

The Standards Used

The Simple Object Access Protocol or SOAP emerged as a mechanism to enable applications to access objects on any platform without any knowledge of the platform. Along with the Web Services Definition Language or WSDL, it forms the basis for the provision of standards based Web Services. It has been accepted as the best protocol available for accessing data and applications on heterogeneous platforms without any knowledge of the platform or the operating system where the accessed object resides. It is emerging as the agreed standard for future application development and integration efforts. In addition, Portus enables access to data and business logic using the REpresentational State Transfer or REST architecture which enables access to resources using a simple URL based approach.

Open and Simple Protocol

The protocol is open and thus is not controlled or owned by any vested interest. It has been designed to work with any software or hardware. Its greatest asset, like TCP/IP and HTTP before it, is its simplicity. It is possible to read SOAP requests and responses with the human eye and understand what they are trying to achieve. The SOAP Standard itself is extremely light particularly when compared to other standards for linking programs together such as CORBA, COM, COM+ and so on.

Platform and Operating System Agnostic

The test for any standard that claims to link software together is how easily it can be run on new platforms. In the case of SOAP, it is already in use on all of the leading platforms and operating systems available today. While there is a large amount of code available today that uses or implements SOAP in some way, it is the standard itself and not necessarily a set of code that is important. The SOAP standard can be used to solve each problem in the best possible way instead of relying on lots of infrastructure to enable SOAP. This flexibility enables the standard to be used on any platform or operating system.

Zero client footprint

As all of the leading technologies have an integrated understanding of how to call SOAP Services, there is absolutely no software footprint for the SOAP Service on the client platform. This leads to easier deployment and maintenance of SOAP based clients.

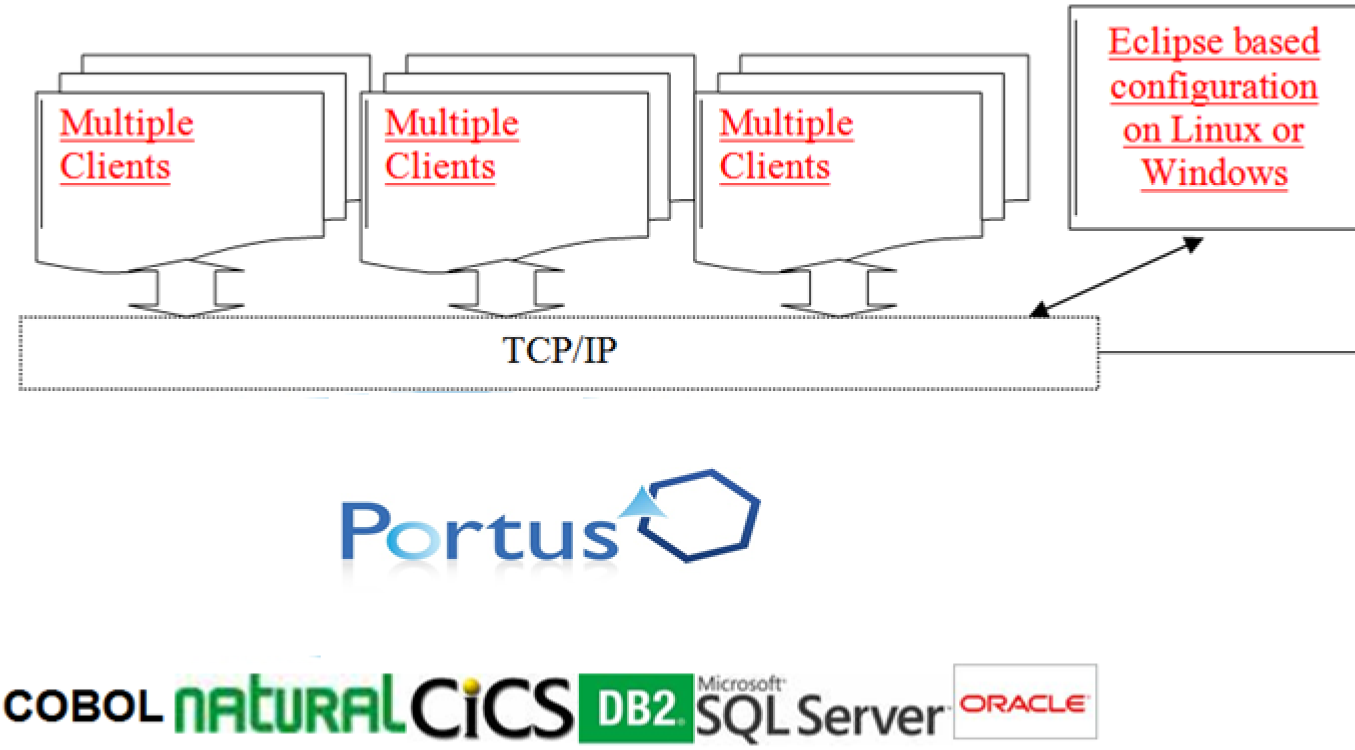
IBM, Microsoft, BEA, SAP and others agree

All the leading software manufacturers believe that standards based Web Services are the way of the future. All new and upcoming releases of software will include some support for SOAP. For applications that run on the desktop, this generally means that they can issue SOAP requests. For heavier duty 'server' applications, they will generally have the ability to issue SOAP requests and to service SOAP requests in some form or another.

The Portus approach

Portus adopts the view that data and business logic can be exposed in a simple, open and logical fashion that requires no coding on the platform where the data and business logic resides. This leads to the point where access to existing data and business logic on any existing platforms can be achieved by installing software and configuring it. This task can be undertaken by the system administrators who are normally available on site. Looking to the future, later projects can then continue to reuse the software with only a configuration requirement to make additional data available for other application development or integration projects.

Overview



Less Points of Failure Leads to a More Resilient Solution

With this architecture, there are less points of failure. The application talks directly to the software that accesses the database or the program containing the business logic, thus the additional hop from client to middleware and middleware to server is avoided. Less points of failure leads to a more stable and reliable implementation.

Only Configuration - No Programming Effort

Once Portus has been installed on a platform, the only additional effort that must be made to make data or business logic available is to configure the product to access the data or business logic. This can be done by the administrator or systems programmer for the existing platform instead of having to get programmers with the appropriate knowledge to develop code on that platform. Portus provides a discovery interface for each supported database or programming language to automatically create the Web Services for you. This means that your programmers or users can focus on the business issue in hand instead of spending time on getting to the data or business logic.

Chapter 2

Release Notes

Topics:

- *Compatibility with other products*
- *Known issues*
- *New features*
- *Features to be discontinued in future versions*

This document gives an overview of the features of Portus that have been introduced or modified since the previous releases. It also contains information about know restrictions and problems.

Compatibility with other products

The following describes Portus compatibility with other software, including prior releases of Portus itself. You may need to upgrade your installation of the software if co- and prerequisites listed here are not met.

- There is an issue if Portus is run from an authorized dataset on z/OS, and will be using the Adabas driver. Please contact the Adabas support group for an optional zap to prevent a S0C4 in ADALNKR, for Adabas 7.4.3 this is fix number AY743119.
- When using the Portus Natural driver, the required minimum levels are
  - Natural 6.3.12 on OpenSystems (Windows / Linux / Unix)
  - Natural 4.2 on z/OS and z/VSE
- Relational databases are accessed via ODBC, therefore the necessary ODBC libraries must be installed. The interface module must be same architecture as the Portus server. For example, Portus running on 64-bit Linux accessing MySQL, needs a 64-bit install of unixODBC and MyODBC. On Windows, a 32-bit install of MyODBC is required, and the 32-bit ODBC configuration dialog must be used : %WINDIR%\SysWOW64\odbcad32.exe

Known issues

Currently there are no known issues

New features

The following new features are introduced with this release of Portus

1. Enhanced support for Statistics

   Metrics can be now gathered at the web service level and at the driver level.
2. New Lifecycle Governance

   Support for Web Service Governance, allowing multiple versions of the same web service to co-exist, and track the lifecycle of services from Test, to Production, to Deprecation.
3. New Usage Governance

   Support for Usage Governance which enables real-time monitoring of web services through logging of data such as input, output, user, location and time.
4. Message Oriented Middleware

   Support for messaging systems IBM WebSphere MQ and Software AG EntireX Broker to log and process usage governance data.
5. Enhanced REST functionality

   Enhancements in REST support, allowing complex web service structures to be passed in a concise manner as part of the URL.
6. Business Data View Enhancements

   Enhancements in BDV support, allowing grouping, nesting and versioning of all BDV Web Services.
7. Web Service Driver

   A new driver which allows web services to be called from Portus , opening up new connectivity routes for legacy infrastructure.
8. Employees Demo Enhancements

Support for the nested keys on database resources, including support for the latest version of Software AG's Employees Demo Files.

9 Xerces v31

Performance and functional improvements in XML processing, including integration of the latest and greatest version of the XML parsing library, Xerces.

10 Control Centre

The Control Centre has been greatly enhanced, by introducing a new, more compact Eclipse perspective, for easier maintenance.

11 Data Masking

The ability to hide sensitive data by applying data masking algoritms.

Features to be discontinued in future versions

The following features will be discontinued in one of the next versions of Portus

The Control Centre "Legacy Perspective"

This version of Portus introduces a new Eclipse perspective into the Portus Control Centre, the administration and monitoring component.

Portus servers at previous levels can still be administrated with the "Legacy Perspective" in the current version of the Portus Control Centre without restrictions, but support for new features (like, in this version, usage and lifecycle governance) are, and will continue to be, introduced into the new "Portus Perspective" only. So you are strongly encouraged to familiarise yourself with the new "Portus Perspective" as soon as possible, the "Legacy Perspective" will be dropped with one of the next releases of the Portus Control Centre.

Chapter 3

Installation

Topics:

- *Installation Prerequisites*
- *Installing the Portus Control Centre*
- *Windows Server installation*
- *z/OS Server installation*
- *z/VSE Server installation*
- *Linux/Unix Server installation*
- *Control Centre and Server Update*
- *Server uninstallation guide*

Portus Installation overview

The Portus installation process uses the Eclipse based "Find and Install" dialog to download and deploy the required installation files. If you are new to Eclipse, see the Introduction to Eclipse section.

Installation Prerequisites

Overview on page 20
Portus Control Centre prerequisites on page 20
Portus Server Operating System Prerequisites on page 20
Additional software prerequisites on page 21

Overview

The Portus software consists of two components:

- The Portus Server component runs locally to your "resource". In this instance a resource is defined as an Adabas, MySQL, PostgreSQL or SQL Server database. It can also be defined as a Natural program, an LE program, a shared library or a DLL.
- The Portus Control Centre component is a central GUI used to configure and monitor one or more Portus Servers. This component runs within the Eclipse IDE, and must be run on Linux or Windows.

Portus Control Centre prerequisites

- On Linux, the GTK2 Windowing system and its dependencies.
- A Java Runtime Environment (minimum: JRE 1.6)

Portus Server Operating System Prerequisites

Portus is currently supported on the following platforms:

- Windows 2003/2008/2012/2016 Server (standard edition and enterprise edition)
- Windows 2000 Professional, Server and Advanced Server
- Windows 7/8/10

Note: All Windows 64-bit platforms are supported by running the Portus server in 32-bit compatibility mode.

- Linux x86-64 (64-bit)
  - SuSE Linux Enterprise 11-SP3 or SP4
  - Red Hat Enterprise 6.4 or above
- Linux s390x (64-bit)
- AIX 5.3 / 6.1 (64-bit)
- Solaris 9/10 SPARC (64-bit)
- HP-UX 11i v2 Itanium (64-bit)
- z/OS
- z/VSE

z/OS

- A userid and password that can be used on the z/OS system.
- Appropriate RACF/ACF2/TOP-SECRET access to create datasets.
- Portus installation requires that server files be sent to the z/OS host, therefore an accessible ftp server must be running on this host.
- A free TCP/IP port that can be used (One per Portus server)
- The ability to add a started task or submit a long running job.
- Adabas databases accessed via Portus must be UES-enabled.
- The Adabas WAL813.L003 as well as WAL813.LOAD (or above) libraries must be in the STEPLIB chain, otherwise no access to Adabas will be possible from the Portus server.

- Under certain circumstances, the Adabas link routines may not pick up the SVC and DBID values specified via DDCARD. In this case the SVC number and Database Id must be zapped into ADALNKR, as described in the Adabas installation documentation, and the modified ADALNKR made available in a STEPLIB library ahead of the one containing the original, unzapped version.

Windows

No additional pre-reqs required.

Linux

- The Portus installation scripts require the bash shell.
- The Portus installation requires that server files be sent to the target machine, therefore an accessible ftp or SSH server must be running on the target machine.
- PHP CGI and Soap modules are required.

z/Linux

- The Portus installation scripts require the bash shell.
- The Portus installation requires that server files be sent to the target machine, therefore an accessible ftp or SSH server must be running on the target machine.

Solaris

- The Portus installation scripts require the bash shell.
- The Portus installation requires that server files be sent to the target machine, therefore an accessible ftp or SSH server must be running on the target machine.

AIX

- At least 100MB of free disk space
- The installation files require the "tar" and "gunzip" programs to be extracted.
- The Portus installation scripts require the bash shell.
- The Portus installation requires that server files be sent to the target machine, therefore an accessible ftp or SSH server must be running on the target machine.

Additional software prerequisites

Portus requires the following software versions

- Minimum Adabas v5 on OpenSystems platforms (Windows / Linux / Unix); Adabas 6.1.3 + Adabas Client (ACL) 6.2.1.8 required for LOBs (large objects) access
- Any supported UES-enabled Adabas version on the mainframe (v7.4 and higher); Adabas 8.1 or higher required for LOB access
- iTrac ADAOS-3371 must be applied
- Only unixODBC 2.2.13 or above is supported.
- To use any relational database driver (MySQL/Oracle/Postgres/etc) ODBC must be installed, and the appropriate driver interface for your database (e.g. MyODBC/Oracle ODBC/psqlodbc/etc) must also be installed. On Windows this must be a 32-bit driver, and should be configured via %WINDIR%\SysWOW64\odbcad32.exe
- If accessing MSSQL Server resources, it must be version 2008 or later
- As Portus is a server-based application, Windows-based ODBC DSNs must be created using the "System" tab

Hardware Specifications

The following specifications for standard Windows and Linux installations are recommended:

- 2 CPUs
- 4GB RAM
- 2 GB Disk space for installation

Installing the Portus Control Centre

Prerequisites on page 22
Download and install Eclipse on page 22
Install the Portus Control Centre from an Update Site on page 22
Next Steps on page 23

Prerequisites

Java is required to run the **Eclipse SDK**

If you already have Java Runtime Environment (JRE, v1.6 or higher) installed, you may skip this step.

To find out what level of java you have installed, run the command `java -version` from either a shell (Unix/Linux) or command box (Windows) depending on your system.

If Java is not installed already in your environment, follow these steps

- Install the latest version from *here*.
- Find the "Java Runtime Environment (JRE)" and download it.
- Accept the license agreement, and choose the correct package for your platform.
- Start the Java installer (for example double-click) and follow the instructions.

  The Java download site provides detailed install instructions.
- Once the installer has finished, Java is installed in your environment.

Download and install Eclipse

The Eclipse OpenSource framework forms the base for the Portus administration and monitoring tool, the 'Control Centre'.

Eclipse 3.7 (or higher) is required to run the Portus Control Centre. If you have a valid copy of Eclipse installed already, you may skip this step.

Steps required to install the Eclipse framework:

1 Download Eclipse version 3.7 for your environment :
   - for *Windows x86*
   - for *Windows x86_64*
   - for *Linux x86 GTK2*
   - for *Linux x86-64 GTK2*

2 Unzip the downloaded packages, this will create a directory structure under a top-level directory named 'eclipse', at the selected file system location. for example C:\eclipse on Windows when unzipping into C:\, or /user/eclipse on Unix/Linux if the selected location was /user

3 Start Eclipse - 'eclipse.exe' on Windows, './eclipse' on Unix/Linux, from the top-level eclipse directory.

If you are new to Eclipse, please take the '*Getting started with Eclipse*' tour before continuing.

Install the Portus Control Centre from an Update Site

- Start the Update Manager, which can be found on the Eclipse main menu under **Help -> Install new software**.

Note: As the Eclipse Update Manager may need to get to an external site, you may have to specify a HTTP proxy in order to access it from an intranet. Instructions on how to set the HTTP proxy can be found in the Troubleshooting section.

- In the **Work with** text box, enter the URL

  ```
  http://cloud.ostiasolutions.com/Portus/brazilUpdateSite
  ```

  OR

  If you have previously downloaded the archived update site, select the Add button, then Archive, and choose the zip file

  ```
  portus-brazil-updateSite.zip
  ```
- The Update Manager will now contact the update site and display the list of available downloads.
- An empty box next to the features name indicates that the feature is not selected
- Choose a feature by clicking the checkbox beside it. A tick appears in the box beside the features
- You should always select the Portus Control Centre' catagory
- You should also select the install kit(s) which correspond to the system that will run the Portus Server
- In some cases, you may also drill down by expanding the feature menu.
- Once finished, click **Next** and the required features will be downloaded.
- Depending on the amount of features selected, and the speed of your connection, this may take some time.
- Once the download has finished, you will be presented withe Legal Notices dialog. Read and Accept the license agreement before continuing.

  Check **I accept the terms in the license agreement** and click **Next**
- Review the features to be installed, and click **Finish**
- The selected features will now be installed.
- If the features have not been signed, you will be asked if you wish to continue with the installation.

  If the features have been signed, you will be asked if you want to install a digitally signed feature.

  Either way, click **OK**
- Restart Eclipse when prompted.

Next Steps

Now the Portus Control Centre has been installed. Proceed to the relevant server installation section choosing a link below.

Windows Server installation

Introduction on page 23
Installation using the Deployment Wizard on page 24
Post install steps on page 29

Introduction

This section outlines how to install the Portus Server on Windows

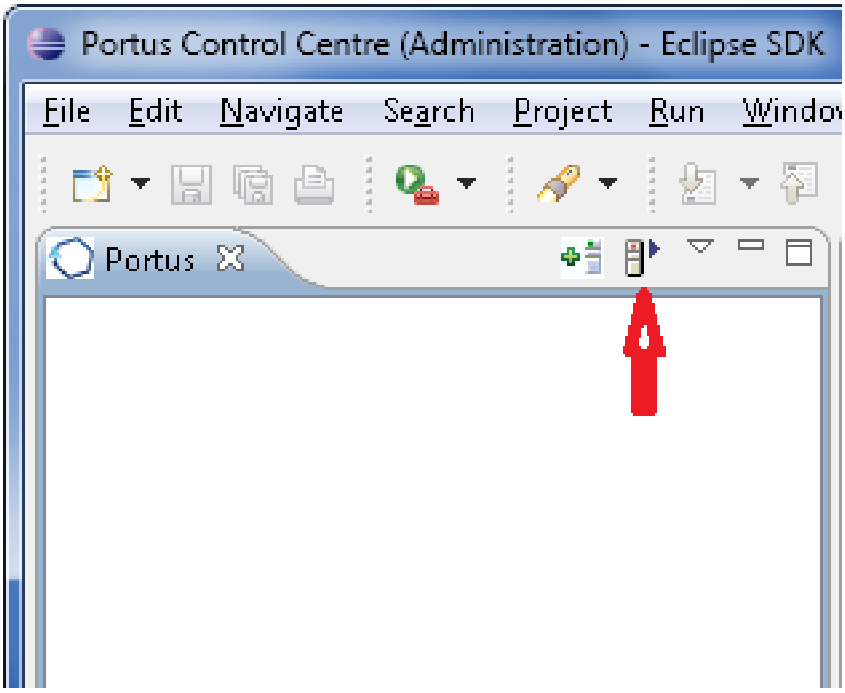
See here for installation troubleshooting hints.

Important: If the version you are installing is not Portus v4.3.1, then your installation may display a version number different from the screenshots below. There is no functional installation difference between versions, so this can be safely ignored.

Installation using the Deployment Wizard

On Windows, the Portus install is done via the Deployment Wizard. This is a wizard which runs within Eclipse, and guides you through the steps required to install and configure your Portus server.

- A dialog will be presented the first time the Portus Control Center is started after its installation. Specify a project name and click **Continue**
- To start the deployment wizard at any time, click the following button

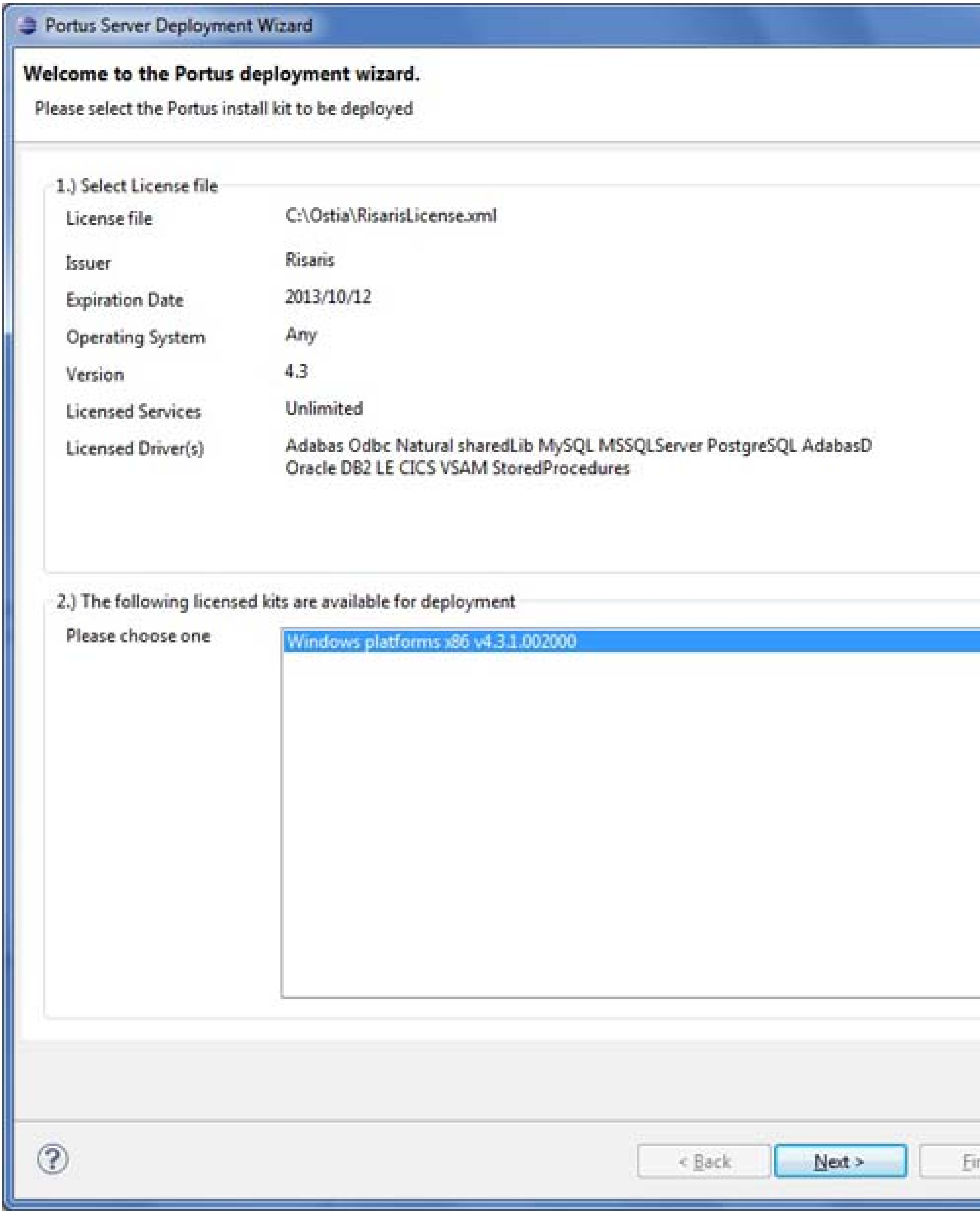

Portus Control Centre (Administration) - Eclipse SDK
File Edit Navigate Search Project Run Windo
Portus

The Deployment Wizard will appear

- Click "**Browse**" to select your license, choose your (Windows) installation kit from the list of available and licensed kits.

Important: This license should reside on a local HDD (Hard Disk Drive), and not on a network drive or SSD (Solid State Drive).

Important: Please make sure you are not selecting a license file from C:\ProgramData... If your license file is located in the programData folder pleace make a temporary copy of it to another location and select the file forom there.



Portus Server Deployment Wizard

Welcome to the Portus deployment wizard.

Please select the Portus install kit to be deployed

1.) Select License file

License file	C:\Ostia\RisarisLicense.xml
Issuer	Risaris
Expiration Date	2013/10/12
Operating System	Any
Version	4.3
Licensed Services	Unlimited
Licensed Driver(s)	Adabas Odbc Natural sharedLib MySQL MSSQLServer PostgreSQL AdabasD Oracle DB2 LE CICS VSAM StoredProcedures

2.) The following licensed kits are available for deployment

Please choose one: Windows platforms x86 v4.3.1.002000

Click **Next**

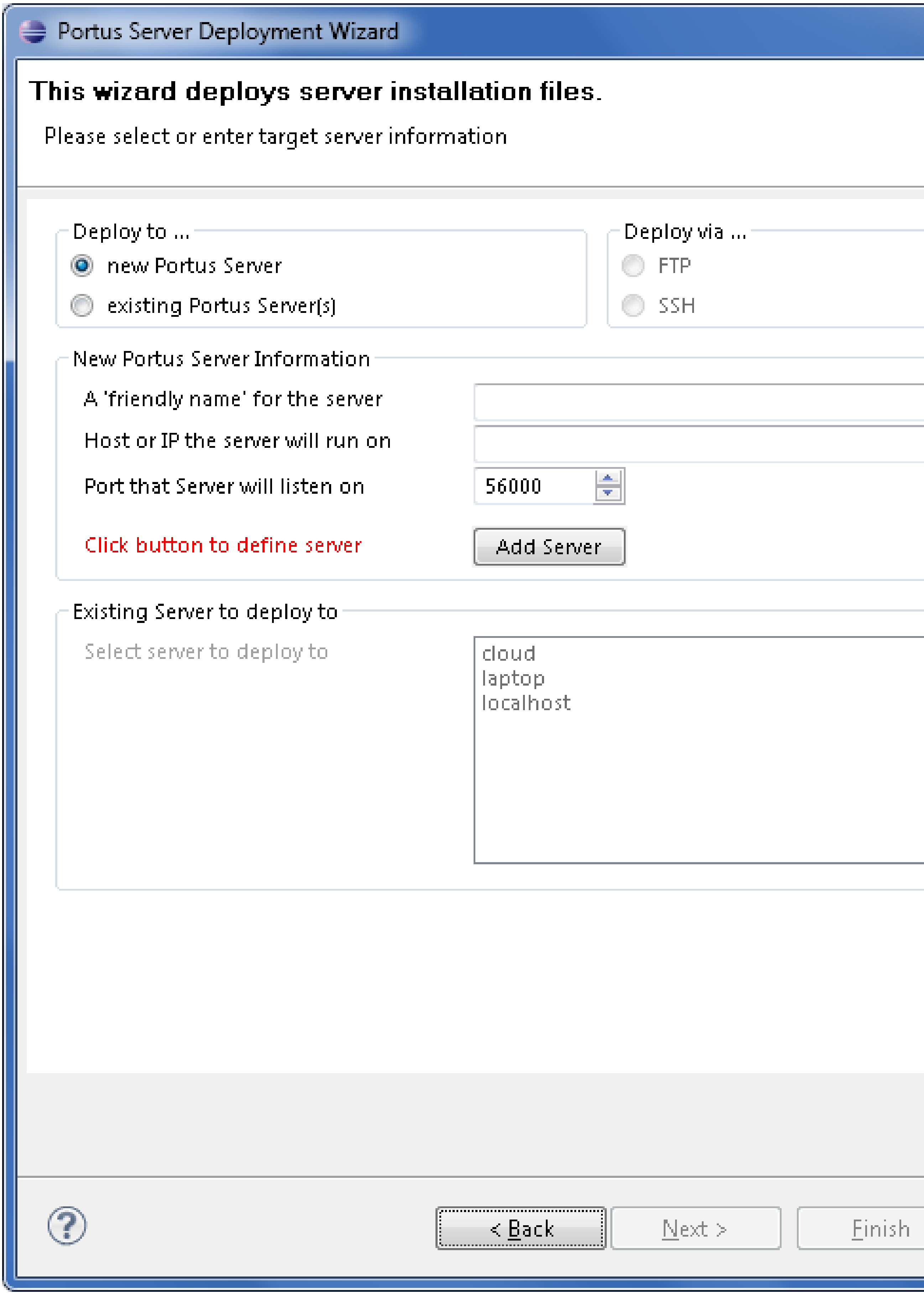
- The following screen allows you to either select an existing Portus server to deploy to, or to define a new Portus server to the Control Center. This server definition will be used for server administration later on.

For Windows, you should

1 Enter a 'friendly name'. This name will be used to refer to a Portus Server without needing to enter the hostname (or IP) later on.

2. Enter the hostname or IP of the machine on which the Portus server will run. This host/IP will be the machine you are currently working on.
3. Enter the port which you would like Portus to listen for requests on. A new Apache web server will be installed, you must ensure that the port choosen here is available for use and not blocked by a firewall. The port entered here will utimately be the port that Portus uses to service requests.

Important: Once you have filled in all of the above, click 'Add Server'



Portus Server Deployment Wizard

This wizard deploys server installation files.

Please select or enter target server information

Deploy to ...

● new Portus Server

○ existing Portus Server(s)

Deploy via ...

○ FTP

○ SSH

New Portus Server Information

A 'friendly name' for the server

Host or IP the server will run on

Port that Server will listen on: 56000

Click button to define server

Existing Server to deploy to

Select server to deploy to

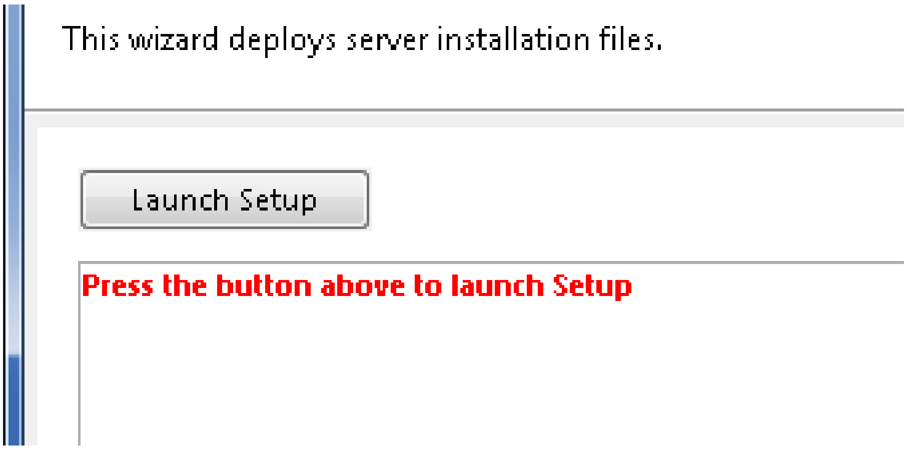
cloud
laptop
localhost

The server will now be added to your Portus Control Center 'Servers View' for later use.

Click **Next**

- Now the Deployment Wizard must launch the Windows installer program.

Press the **Launch Setup** button



This wizard deploys server installation files.

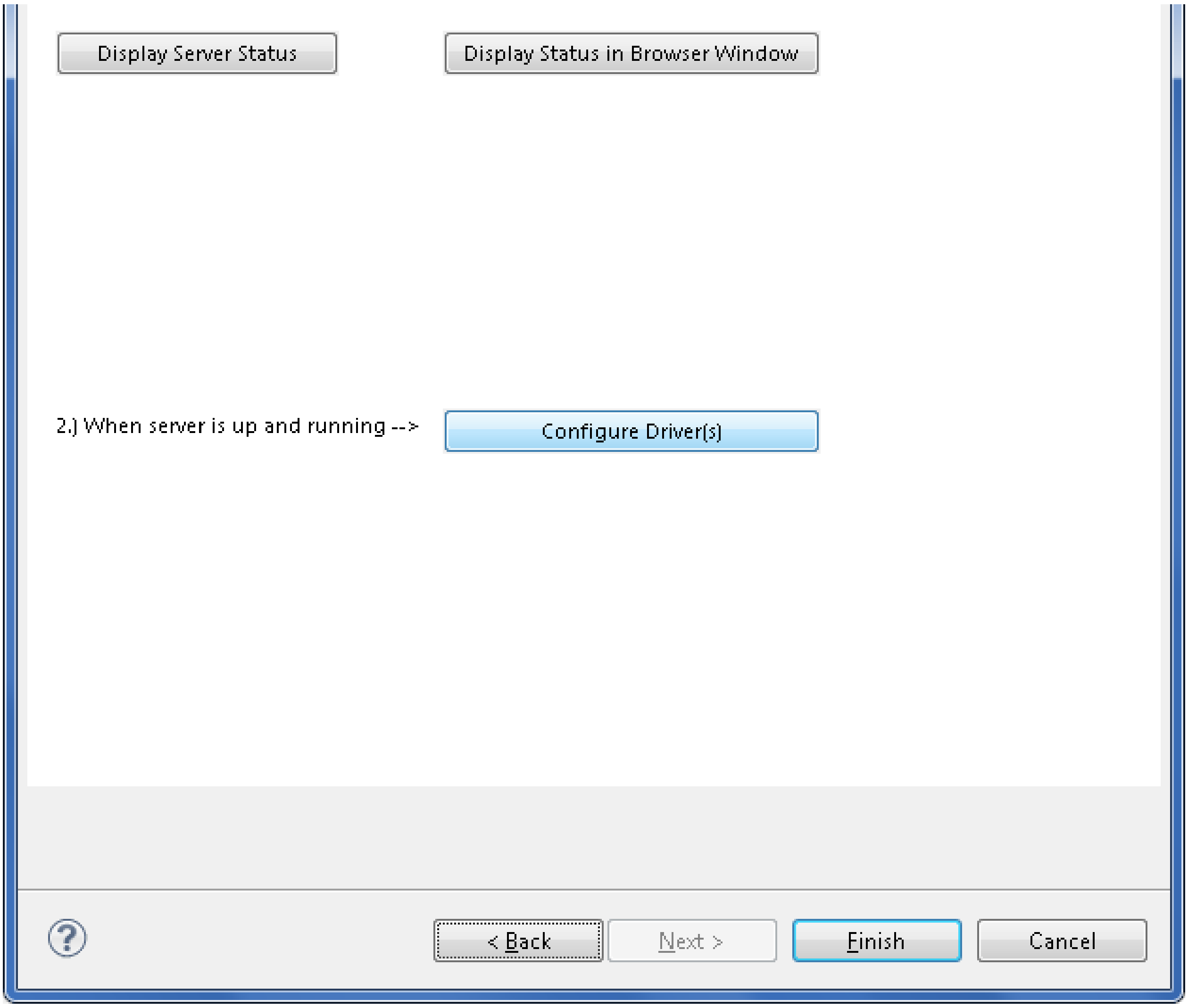
Press the button above to launch Setup

- A dialog will pop up alerting that the process needs to check for an existing installation.
- **Note:** If there is an existing install present please skip to the documentation for an upgrade.
- The deployment wizard now starts an MSI installer to install the Portus Server component on the system. When the first page appears, click **Next**
- If you require Portus to be installed in a non-default directory, click **Browse** and select your directory. Otherwise click **Next**
- Click **Install** to continue
- Please wait while Portus is installed
- When the installation is complete, click **Finish**
- Return to the Deployment Wizard, the message log will indicate either success or failure of the setup process.

Click **Next**

- The next screen is used to verify that your Portus is up and running, and then to configure it based on your license.

1. **Display Server Status**: This displays this a short summary of the server status in the deployment wizard
2. **Display Status in Browser Window**: This displays more information about the servers status in a browser.
3. **Configure Driver(s)**: Allows you to automatically create drivers based on your license.


Display Server Status
Display Status in Browser Window
2.) When server is up and running -->
Configure Driver(s)
?
< Back
Next >
Finish
Cancel

- You should now click Configure Driver(s) to create Portus drivers that are enabled in your license. If you choose to not add drivers now, they can also be created at a later stage, more information about defining drivers can be found in the *Portus View* section of this documentation.

  The "Driver Wizard" will present a list of licensed drivers, and come up with all drivers selected and ready to be defined. You may now just click the **Configure** button to define all of them, or deselect those you do not need or want to set up at that stage.

  Click "**Configure**".

  In case the driver does not require any additional parameters to be set, it will be defined now, otherwise the Driver Definition Dialog will ask for the additional parameter(s) to be set e.g. set the "ListMaxRecords" parameter to a value of "0" (zero) for an Adabas Driver. Click the **Save** button.

  The Driver Wizard Dialog indicates the success of the operation
- When **Show (post-setup) Driver Information** is selected, and information is available for the driver(s) just installed, it will be displayed (cumulatively) after the driver(s) have been configured
- Click **Done** to close the Deployment Wizard

Post install steps

Now Portus is installed - what's next ?

- New to Eclipse ? Take an *Eclipse getting started* tour.
- Configure Portus using the *Portus Control Centre*
- Proceed directly to *define Web Services* using Portus.

z/OS Server installation

Using the Deployment Wizard to install on a z/OS host on page 30
z/OS Server installation on page 33

Using the Deployment Wizard to install on a z/OS host

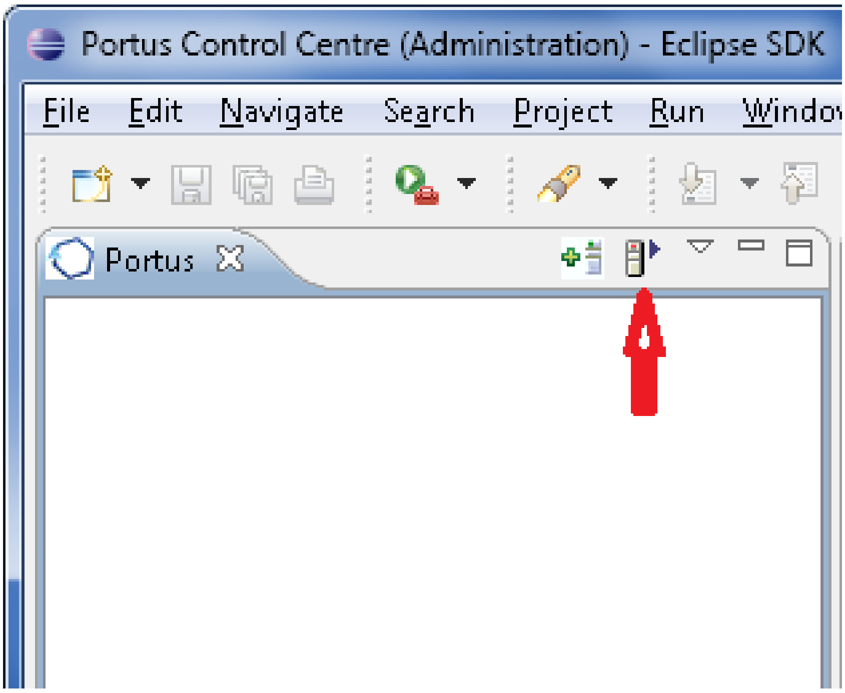
Note: If the version you are installing is not Portus v4.3.1, then your installation may display a version number different from the screenshots below. There is no functional installation difference between versions, so this can be safely ignored.

In preparation of running the Portus server on z/OS the installation datasets have to be deployed to the z/OS host. The deployment wizard on your Portus Control Centre on Windows or Linux can be used to deploy the required files to z/OS.

- If this is the first time you've started the Control Centre, the Portus Perspective will be activated automatically.

  If, for some reason, the Portus Control Centre perspective has not started, click **Window** -> **Open Perspective** -> **Other** and choose **Portus Control Centre** from the list. Click **OK**

  You will be asked to specify a project name or accept the default. Usually you will now simply click the 'Continue' button, which will then start the 'Deployment Wizard' to guide you through the process of defining your server within the Portus Control Centee and transfer (FTP) the installation files to the Portus Server target machine.
- If you, for whatever reason, opt to NOT run the Deployment Wizard at that time, you can start the wizard anytime later on by clicking the Deployment action button in the title bar of the Portus View


Portus Control Centre (Administration) - Eclipse SDK
File Edit Navigate Search Project Run Windo
Portus

  The Deployment Wizard will appear
- Click "**Browse**" to select your license, choose your (z/OS) installation kit from the list of available and licensed kits.

  Important: This license should reside on a local HDD (Hard Disk Drive), and not on a network drive or SSD (Solid State Drive).

  Click **Next**

- The following screen allows you to either select an existing Portus server to deploy to, or to define a new Portus server to the Control Center. This server definition will be used for server administration later on.
- To define a new server

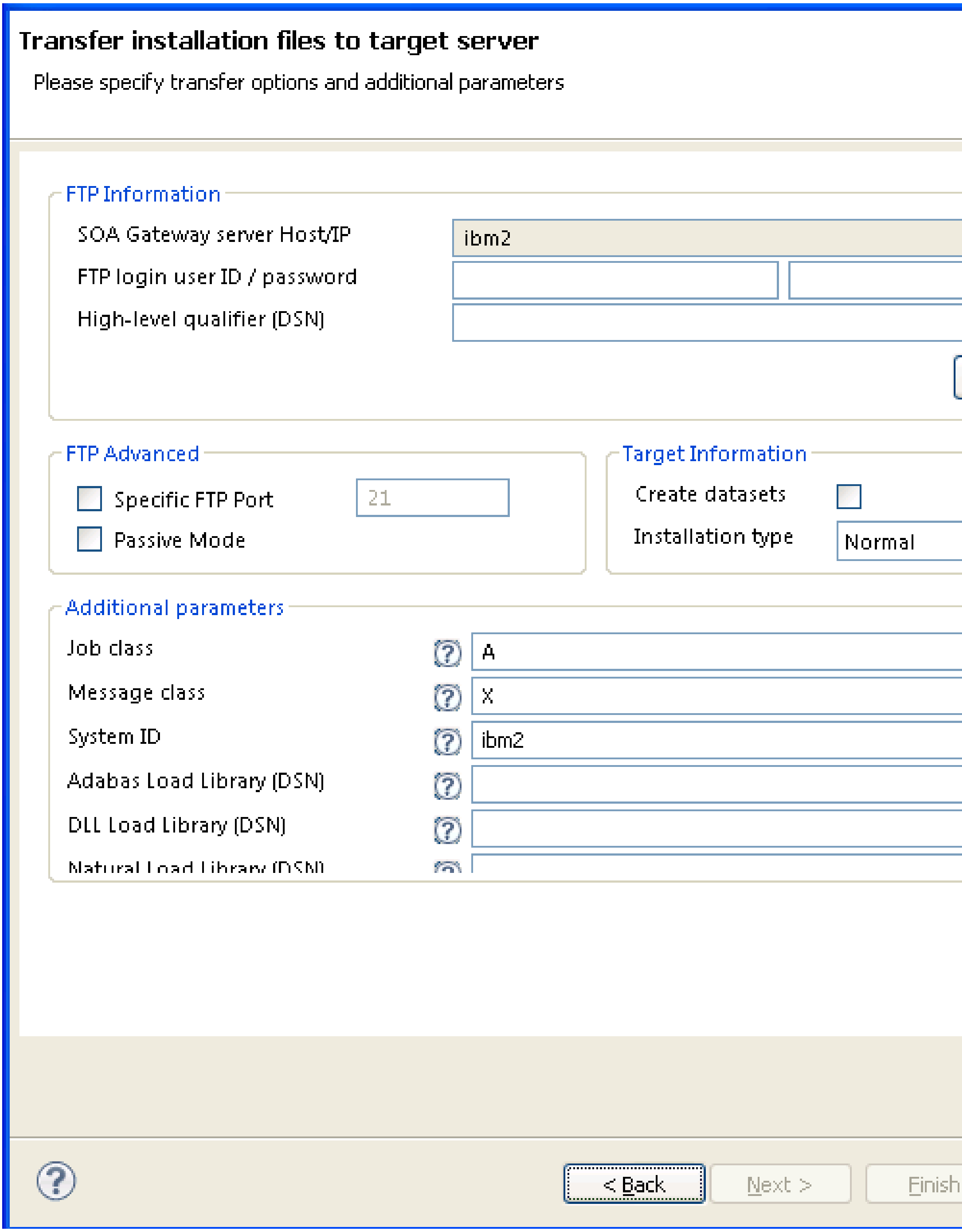
  1 Enter a 'friendly name'. This name will be used to refer to a Portus Server without needing to enter the hostname (or IP) later on.

  2 Enter the hostname or IP of the machine on which the Portus server will run. This host/IP will be used to send the install files (via FTP) and will become the host/IP that you use when issusing requests to Portus .

  3 Enter the port which you would like Portus to listen for requests on. A new Apache web server will be installed, you must ensure that the port choosen here is available for use and not blocked by a firewall. The port entered here will utimately be the port that Portus uses to service requests.

  Important: Once you have filled in all of the above, click 'Add Server'

  The server will now be added to your Portus Control Center Portus View' for later use.
- Alternatively, if an already defined server is to be used (i.e. installation files deployed to it), check 'Deploy to existing Portus Server' and select a server from the list in the 'Existing Server Information' section.
- Click **Next**
- The next screen is used to deploy the installation files to the target machine, i.e. the machine is where the Portus server will run.



Transfer installation files to target server

Please specify transfer options and additional parameters

FTP Information

SOA Gateway server Host/IP: ibm2

FTP login user ID / password:

High-level qualifier (DSN):

FTP Advanced

□ Specific FTP Port: 21

□ Passive Mode

Target Information

Create datasets □

Installation type: Normal

Additional parameters

Job class: A

Message class: X

System ID: ibm2

Adabas Load Library (DSN):

DLL Load Library (DSN):

Natural Load Library (DSN):

- The hostname / IP will have already been filled by what has been specified on the previous page.
- Enter the username which is used to login (via FTP) to the server machine (if required)
- Enter the password (if required)
- Enter a high-level qualifier to send the installation files to.

Important:

Check the **Create datasets** box (in the target information group) if you want the wizard to create the necessary datasets (recommended). Otherwise, these must be already allocated. For more information, see *here*

- Installation type. In most cases, this should be left as **Normal** . If the datasets that are being sent are patch datasets, you should select **Patch** from the drop-down menu.
- The *Additional parameters* group will allow you to enter the DSN(s) required for the operation of the driver(s) to be installed. Parameters related to all licensed drivers are queried here, you may omit those for all drivers you do not intend to use.

  Note: Click the Help button(s) next to the DSN name input fields to view information about each datasets purpose.
- If your FTP server requires a port other than 21, you can specify that by selecting **Specfic FTP Port** and entering the FTP port number.
- If your FTP server requires passive mode transfers, you can enable this by selecting the **Passive Mode** checkbox
- --> Click the Start FTP button to begin the file transfer.
- Once the FTP has completed, you need to logon to the target machine, and run the *server-specific installation steps*.
- Now that the server has been installed, you can query the server status from the deployment wizard.

  You can choose to view this status information in the Deployment Wizard, in a browser, or both.
- Click the **Display Server Statu**s or the **Display Status in Browser Window** buttons to return the status of the server.
- You should now click **Configure Driver(s)** to create Portus drivers that are enabled in your license. If you choose to not add drivers now, they can also be created at a later stage, more information about defining drivers can be found in the *Portus view* section of this documentation.

  The "Driver Wizard" will present a list of licensed drivers, and come up with all drivers selected and ready to be defined. You may now just click the **Configure** button to define all of them, or deselect those you do not need or want to set up at that stage.

  Click "Configure".

  In case the driver does not require any additional parameters to be set, it will be defined now, otherwise the Driver Definition Dialog will ask for the additional parameter(s) to be set. e.g. we set the "ListMaxRecords" parameter to a value of "0" (zero) for an Adabas driver. Click the **Save** button.
- When **Show (post-setup) Driver Information** is selected, and information is available for the driver(s) just installed, it will be displayed (cumulatively) after the driver(s) have been configured
- Click **Done** to close the Deployment Wizard
- Configure Portus using the *Portus (Eclipse) Control Centre*

z/OS Server installation

Dataset Creation

Note: This step can be ignored if you chose the **Create Datasets** option in the Deployment Wizard. If so, continue to *here*

The Portus Installation on z/OS uses 4 datasets, which must be created in advance. The datasets can be any name you wish, so long as they each use the same high-level qualifier. This allows the user to specify one dataset name prefix in the Java wizard, and the installation is based upon this qualifier.

The 4 datasets to create are RECV, JCL, CONF, and ZIP.

For example, if the high-level qualifier is USER.INST001, then the following would be the datasets that need to be created.

- USER.INST001.RECV
- USER.INST001.JCL
- USER.INST001.CONF
- USER.INST001.ZIP

The required dataset attributes are as follows:

Name	Type	RECFM	LRECL	BLKSIZE	SIZE
RECV	Sequential	FB	80	3120	see below
JCL	PDS	FB	80	3120	
CONF	PDS	VB	508	512	
ZIP	Sequential	VB	16380	16384	see below

These local files will be located in the directory where the wizard was started.

XMIT File: AdabasSOAGateway/zOS/ASG.XMIT
ZIP File: AdabasSOAGateway/zOS/asg.zip

The size attribute for the RECV and ZIP datasets to be created on the zOS should be calculated as follows:

RECV File: Divide the size of ASG.XMIT in bytes by 3120. Round the resultant size upwards to the nearest integer to obtain the file size in blocks. We recommend a size of 9700 blocks.

ZIP File: Divide the size of asg.zip in bytes by 16380. Round the resultant size upwards to the nearest integer to obtain the file size in blocks. We recommend a size of 700 blocks.

After having transfered the files using the Depl,oyment Wizard the datasets will contain the following

`RECV`		Sequential file containing the Portus load modules. This will be TSO RECEIVEd as part of the installation process.
`JCL`	SOAGINS1	A job to perform a TSO receive.
	SOAGINS2	A job which deletes, creates, initializes, and unzips the Portusfile system into the file system available to Apache in this environment. The DEL step of this job will finish with CC=8 if the PFS dataset does not exist. Likewise, if the PFS dataset does exist, it will be deleted by this job.
	SOAGINS3	A job which copies a Apache's configuration file from a PDS member to the PortusEnvironment.
	SOAGINS4	A job which copies the license file the SOA Gateway Environment.
	SOAGPROC	PROC to be used for running all other jobs.
	SOAGSTRT	Job which starts the Portus Server.
	PFSPROC	PROC to be used for various utilities.
	CPLOGS	A job which copies Apache's error_log to an OUTPUT DDCARD if a failure occurs during Portusinitialization. Once the Portusserver is initialized, this log will be written to the JES Spool.
	PFSLS	A job which lists the contents of the Portusfile system.
	CLIINI	A DB2 CLI Initialization File. If using DB2 Driver this will need to be modified to reflect the Sub-System ID and location name. Your DBA can provide this information.
`CONF`	CONFIG	File used by Portusto declare environment variables
	HTCONF	A copy of the Portus Apache configuration file. This can be edited and copied into the Portus file system using SOAGINS3.
	GROUP	Run-time configuration file
	HOSTS	Run-time configuration file

	PASSWD	Run-time configuration file
	DXRTRACE	Run-time parameters for debugging. Only to be used at the direction of support
	DDCARD	Adabas Run parameters
	SYSPARM	Run-time parameters.
	ASGLIC	License File.
`ZIP`		A Sequential zip-file containing the Portus file system.

If you do not wish to install the Portus (Eclipse) Control Centre, proceed to z/OS Server Setup

You can either, logon to the z/OS host and setup the Portus Server, or install the client pieces now (if selected) and run the server setup at a later time.

Submitting JCL and Starting Server

Review and modify the following jobs prior to submitting them to reflect your installation standards and the names of the datasets created during your installation.

1. Review and submit SOAGINS1 to TSO RECEIVE the Portus load library.
2. Review and submit SOAGINS2 to create, init, and populate the Portus file system.
3. Review and submit SOAGINS3 to copy the Apache configuration file into the Portus filesystem.
4. Review and submit SOAGINS4 to copy the Portus license into the Portus filesystem.
5. If using the Adabas Driver review the DDCARD member and specify the correct Adabas SVC number. Under certain circumstances, the Adabas link routines may not pick up the SVC and DBID values specified via DDCARD. In this case the SVC number and Database Id must be zapped into ADALNKR, as described in the Adabas installation documentation, and the modified ADALNKR made available in a STEPLIB library ahead of the one containing the original, unzapped version.
6. If using the DB2/ODBC Driver review the CLIINI member and specify the correct MVSDEFAULTSSID and Location name.

   Refer to you DBA for this information.
7. Submit SOAGSTRT to bring up the Portus (Apache) server. The server is ready to be used when you see the "XMiddLe ... is ready for requests" message in the XMIDCARD DD.

   Note: Restarting and stopping Portus: Due to the nature of the z/OS Apache configuration it is not possible to restart the server with a single command. The server must be stopped with "/p jobName" and started by submitting SOAGSTRT again. In this scenario, Apache may issue a warning about the httpd.pid being overwritten. This warning can safely be ignored.
8. If the Portus Server is to be run as a started task (STC), copy SOAGSTRT to a procedure library.

Starting the Server with HTTPS support

To enable HTTPS connections, a sample SSL certificate and SSL sample key have been provided. These are for development only, and should not be used in a production environment.

1. If you wish the server to listen on a non-default HTTPS port, review ...CONF(SSLCONF) and change all instances of 443 to the port you wish to listen for HTTPS connections.
2. Edit the ...CONF(HTCONF) member and uncomment the **Include conf/extra/httpd-ssl.conf** directive
3. Ensure the server has been stopped
4. Review and submit SOAGINS5 to copy SSLCONF into the Portus file system.
5. Review and submit SOAGINS3 to copy HTCONF into the Portus file system.
6. Edit SOAGPROC and change the server parameters to -X -DSSL

   E.g. **EXEC PGM=HTTPD,PARM='-X -DSSL',REGION=0M**
7. Submit SOAGSTRT to bring up the Portus (Apache) server. The server is ready to be used when you see the "XMiddLe ... is ready for requests" message in the XMIDCARD DD.

HTTPS access can be tested by checking (change host and the HTTPS port to suit your system)

https://myServer:8443/SoaGatewayStatus

For more information on configuring Apache for HTTPS see *here*

z/VSE Server installation

Using the Deployment Wizard to install on a z/VSE host on page 36
Operating the Portus Server on page 38

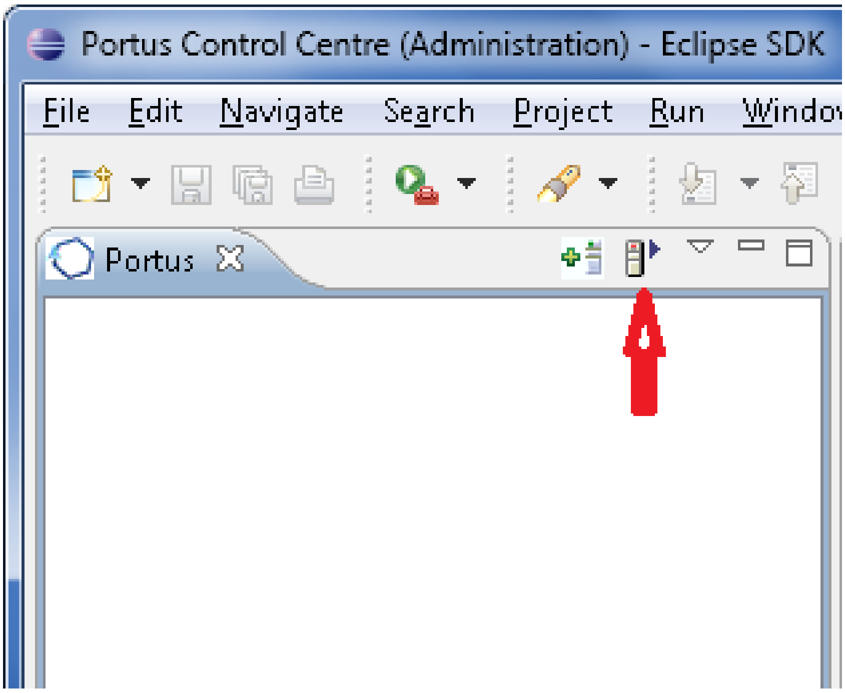
Using the Deployment Wizard to install on a z/VSE host

In preparation of running the Portus server on z/OS the installation datasets have to be deployed to the z/VSE host. The deployment wizard on your Portus Control Centre on Windows or Linux can be used to deploy the required files to z/VSE.

- If this is the first time you've started the Control Centre, the Portus Perspective will be activated automatically.

  If, for some reason, the Portus Control Centre perspective has not started, click **Window -> Open Perspective -> Other** and choose **Portus Control Centre** from the list. Click **OK**

  You will be asked to specify a project name or accept the default. Usually you will now simply click the 'Continue' button, which will then start the 'Deployment Wizard' to guide you through the process of defining your server within the Portus Control Centee and transfer (FTP) the installation files to the Portus Server target machine.
- If you, for whatever reason, opt to NOT run the Deployment Wizard at that time, you can start the wizard anytime later on by clicking the Deployment action button in the title bar of the Portus View



The Deployment Wizard will appear

- Click "**Browse**" to select your license, choose your (z/VSE) installation kit from the list of available and licensed kits.

  Important: This license should reside on a local HDD (Hard Disk Drive), and not on a network drive or SSD (Solid State Drive).

  Click **Next**

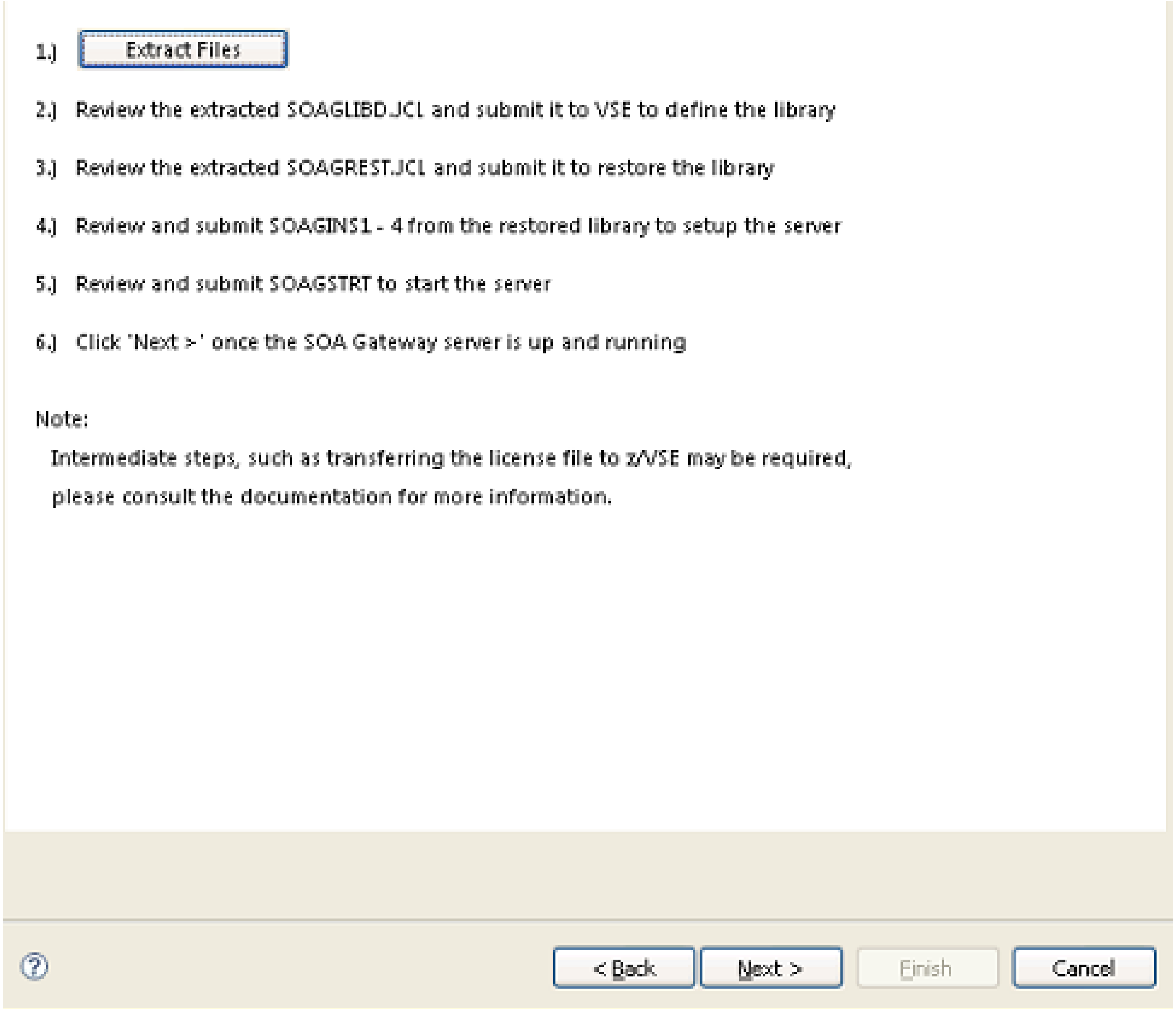
- The following screen allows you to either select an existing Portus server to deploy to, or to define a new Portus server to the Control Center. This server definition will be used for server administration later on.
- To define a new server

1. Enter a 'friendly name'. This name will be used to refer to a Portus Server without needing to enter the hostname (or IP) later on.
2. Enter the hostname or IP of the machine on which the Portus server will run. This host/IP will be used to send the install files (via FTP) and will become the host/IP that you use when issusing requests to Portus.
3. Enter the port which you would like Portus to listen for requests on. A new Apache web server will be installed, you must ensure that the port choosen here is available for use and not blocked by a firewall. The port entered here will utimately be the port that Portus uses to service requests.

👉 **Important:** Once you have filled in all of the above, click 'Add Server'

The server will now be added to your Portus Control Center Portus View' for later use.

- Alternatively, if an already defined server is to be used (i.e. installation files deployed to it), check 'Deploy to existing Portus Server' and select a server from the list in the 'Existing Server Information' section.
- Click **Next**
- The next screen allows you to extract the z/VSE specific files to your local environment.



1.) Extract Files

2.) Review the extracted SOAGLIBD.JCL and submit it to VSE to define the library

3.) Review the extracted SOAGREST.JCL and submit it to restore the library

4.) Review and submit SOAGINS1 - 4 from the restored library to setup the server

5.) Review and submit SOAGSTRT to start the server

6.) Click 'Next >' once the SOA Gateway server is up and running

Note:

Intermediate steps, such as transferring the license file to z/VSE may be required, please consult the documentation for more information.

- Click the "Extract Files" button and extract the z/VSE files to your local system.

- Using FTP, send your Portus license file into the SAGLIB.ASGvvv library. This file should be named `ASGnn.LIC` where nn is the version of Portus to be installed.

  Important: This file must be not be translated during the transfer, therefore ensure it is transferred in binary.

  Important: In case you did not restore the Portus sublibraries into a library named SAGLIB, modify member CPLIC.P and update accordingly

  Review SOAGINS3 and adjust the library/volume/extent settings. Submit this job to copy your license file to the Portus filesystem.
- If you wish to change the port that Portus will listen on (default: 56000) transfer the HTTPD.CONF file from z/VSE to your PC and edit it there. The directive to change is `Listen`.

  Important: This file must be not be translated by the FTP, therefore ensure it is transferred in binary.

  Important: In case you did not restore the Portus sublibraries into a library named SAGLIB, modify member CPHTTPD.P and update accordingly

  Review SOAGINS4 and adjust the library/volume/extent settings. Submit this job to copy the HTTPD.CONF file to the Portus filesystem.
- In case you did not restore the Portus sublibraries into a library named SAGLIB, modify member CONFIG.P to point the SAG_RTS_ETC=FILE: setting from

  ```
  ///SAGLIB/ASG43100/
  ```

  to

  ```
  ////<yourlib>/ASG43100/
  ```
- Review SOAGSTRT and adjust the library/volume/extent settings.

  Ensure that an Adabas (or WAL (Adabas Limited)) v8 library is included in the LIBDEF, it is essential for the operation of the Portus server that the level of Adabas is 8.1.3.02 or above .

  If you plan to use the CICS Driver, ensure that Portus runs in OS390 emulation mode. Add the "OS390" to the JCL, e.g.

  ```
  // EXEC   HTTPD,SIZE=AUTO,PARM='-DONE_PROCESS',OS390
  ```

  Submit SOAGSTRT.JCL to start your Portus server.

  In the Portus Server Deployment Wizard dialog click the **Display Server Status** or the **Display Status in Browser Window** buttons to return the status of the server.

  Click **Finish** to close the Deployment Wizard

  Configure Portus using the *Portus (Eclipse) Control Center*

Operating the Portus Server

- *Sizing the partition for the Portus server* on page 38
- *Using a disk file for the ADARUN parameters* on page 40

Sizing the partition for the Portus server

The Portus server, started with the parameters as delivered, will require a partition with ca. 500 KB 24-bit and 36 MB 31-bit storage, plus ca. 20 KB SVA-24 and 160 KB SVA-31 storage.

To find out how much free SVA space is available on the system issue the AR command GETVIS SVA.

The output will look like this:

```
AR 0015 GETVIS USAGE   SVA-24    SVA-ANY                    SVA-24
   SVA-ANY
AR 0015  AREA SIZE:    1,872K    14,728K

AR 0015  USED AREA:    1,312K     6,788K MAX. EVER USED:    1,320K
    6,800K
AR 0015  FREE AREA:      560K     7,940K LARGEST FREE:        560K
    7,380K
AR 0015 1I40I  READY
```

To get more information about SVA allocation and usage run the LIBR utility function LISTDIR SDL.

The output will look like this:

```
STATUS DISPLAY       SDL  AND  SVA                  DATE:
 2009-07-12
                                                    TIME: 22:25

--------------------------------------------------------------------
SDL       TOTAL ENTRIES :    908   (100%)

          USED  ENTRIES :    514   ( 57%)

          FREE  ENTRIES :    394   ( 43%)


SVA(24)   TOTAL SPACE   :   2188K  (100%)

          USED  SPACE   :   1558K  ( 71%)

           - PFIXED AREA:    165K  (  8%)  START AT: 002C7928

          FREE  SPACE   :    630K  ( 29%)


SVA(31)   TOTAL SPACE   :   7620K  (100%)

          USED  SPACE   :   6685K  ( 88%)

           - PFIXED AREA:    680K  (  9%)  START AT: 051C6F00

          FREE  SPACE   :    935K  ( 12%)

--------------------------------------------------------------------
DIRECTORY DISPLAY    SDL SORTED BY PHASE NAME       DATE:
 2009-07-12
                                                    TIME: 22:25

--------------------------------------------------------------------
 M E M B E R    ORIGIN SVA/MOVE  LOADED  PHASE  ADDRESS  ENTRY
 POINT
NAME     TYPE   SYSLIB   MODE   INTO SVA SIZE   IN SVA   IN SVA

--------------------------------------------------------------------
$$BACLOS PHASE    YES    MOVE      31      554  04BE6C48 04BE6C48

$$BATTNA PHASE    YES    MOVE      31     2216  04BE6E78 04BE6E78

$$BATTNK PHASE    YES    MOVE      31     1104  04BE7720 04BE7720
```

```
$$BATTNR PHASE    YES    MOVE      31      389  04BE7B70 04BE7B70

$$BCASI3 PHASE     NO    MOVE      31      838  04BEF880 04BEF880

$$BCLOSE PHASE    YES    MOVE      31     1192  04BE7CF8 04BE7CF8

$$BCLOS2 PHASE    YES    MOVE      31      624  04BE81A0 04BE81A0

$$BCLOS5 PHASE    YES    MOVE      31     1032  04BE8410 04BE8410

$$BCLRPS PHASE    YES    MOVE      31      712  04BE8818 04BE8818

$$BCVSAM PHASE    YES    MOVE      31      768  04BE8AE0 04BE8AE0

$$BCVS02 PHASE    YES    MOVE      31      326  04BE8DE0 04BE8DE0

$$BDYD$$ PHASE     NO    MOVE      31      104  04BEFBC8 04BEFBC8

...
...
```

In /apache2/conf/extra/httpd-mpm.conf there is a section entitled worker MPM - this details the number of threads, maxclients etc that is available for Apache to use. The httpd-mpm.conf worker MPM settings that we provide as default is presented below. Please refer to Apache documentation for detailed information on these parameters.

```
# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept
 spare
# MaxSpareThreads: maximum number of worker threads which are kept
 spare
# ThreadsPerChild: constant number of worker threads in each server
 process
# MaxRequestsPerChild: maximum number of requests a server process
 serves
<IfModule mpm_worker_module>
    ServerLimit         1
    StartServers        1
    MaxClients          8
    MinSpareThreads      2
    MaxSpareThreads     8
    ThreadsPerChild     8
    MaxRequestsPerChild 0
</IfModule>
```

Please note that should you wish to adjust these parameters there will need to be adjustments in the partition size also. Installation job cpHttpd.jcl and cpHttpd.p can be modified to copy files from the pfs.

Using a disk file for the ADARUN parameters

The Portus start job, SOAGSTRT.JCL, as distributed uses inline ADARUN parameters.

It may prove advantageous to put these parameters on a disk file instead, the following is sample JCL to do this:

```
* $$ JOB JNM=DITTOCS,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
```

```
// JOB DITTIOCS    CARD TO SEQUENTIAL DISK FILE
// UPSI 1
// DLBL CARD,'SOAGATE.ADARUN.PARAMETERS',0,SD
// EXTENT SYS010,vvvvvv,1,0,ssss,1
// ASSGN SYS010,DISK,VOL=vvvvvv,SHR
// EXEC DITTO
$$DITTO CS FILEOUT=CARD,RECFMOUT=F,BLKSIZE=80
ADARUN PROGRAM=RENTUSER
ADARUN SVC=<your_ADABAS_SVC_number>
ADARUN DATABASE=<your_ADABAS_dbid>
/*
/&
* $$ EOJ
```

After having created the disk file replace the inline ADARUN parameters in SOAGSTRT.JCL with the appropriate sequence of DLBL CARD, EXTENT and ASSGN statements.

Linux/Unix Server installation

The following will guide you through the process of installing and starting the Portus server.

Using the Deployment Wizard for Linux/Unix type systems on page 41
Server Installation on page 47

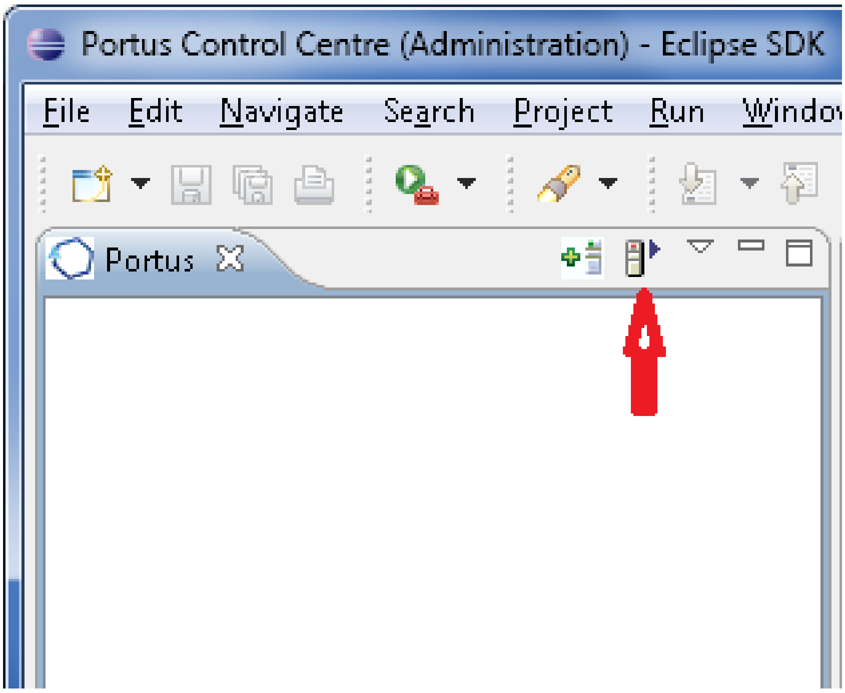
Using the Deployment Wizard for Linux/Unix type systems

In preparation of running the Portus server on the target system installation files have to be deployed to a remote maching. The deployment wizard on your Portus Control Centre on Windows or Linux can be used to deploy the required files to the target server.

- If this is the first time you've started the Control Centre, the Portus Perspective will be activated automatically.

  If, for some reason, the Portus Control Centre perspective has not started, click **Window -> Open Perspective** -> **Other** and choose **Portus Control Centre** from the list. Click **OK**

  You will be asked to specify a project name or accept the default. Usually you will now simply click the 'Continue' button, which will then start the 'Deployment Wizard' to guide you through the process of defining your server within the Portus Control Centee and transfer (FTP) the installation files to the Portus Server target machine.
- If you, for whatever reason, opt to NOT run the Deployment Wizard at that time, you can start the wizard anytime later on by clicking the Deployment action button in the title bar of the Portus View


Portus Control Centre (Administration) - Eclipse SDK
File Edit Navigate Search Project Run Windo
Portus

The Deployment Wizard will appear

- Click "**Browse**" to select your license, choose your server installation kit from the list of available and licensed kits.

Important: This license should reside on a local HDD (Hard Disk Drive), and not on a network drive or SSD (Solid State Drive).

Click **Next**

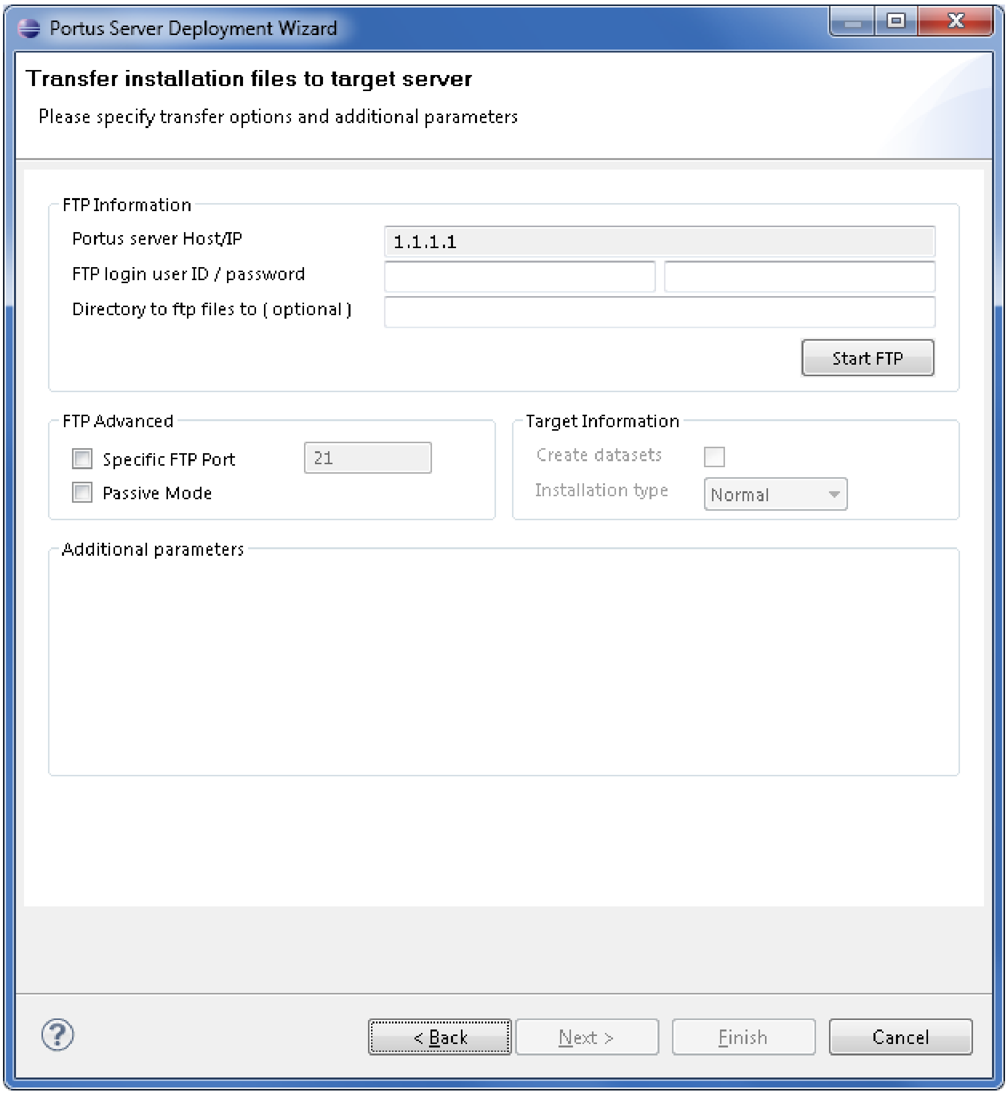
- The following screen allows you to either select an existing Portus server to deploy to, or to define a new Portus server to the Control Center. This server definition will be used for server administration later on.
- To define a new server

1. Enter a 'friendly name'. This name will be used to refer to a Portus Server without needing to enter the hostname (or IP) later on.
2. Enter the hostname or IP of the machine on which the Portus server will run. This host/IP will be used to send the install files (via FTP) and will become the host/IP that you use when issusing requests to Portus.
3. Enter the port which you would like Portus to listen for requests on. A new Apache web server will be installed, you must ensure that the port choosen here is available for use and not blocked by a firewall. The port entered here will utimately be the port that Portus uses to service requests.
4. **Note:** Once you have filled in all of the above, click 'Add Server'
5. The server will now be added to your Portus Control Center Portus View' for later use.

- Alternatively, if an already defined server is to be used (i.e. installation files deployed to it), check 'Deploy to existing Portus Server' and select a server from the list in the 'Existing Server Information' section.
- Select the transport mechanism, one of
  - FTP: only transfer, actual installation has to be carried out by logging on to the target system and starting the installation script manually
  - SSH: this method optionally allows for automatic ("silent") installation, no separate login to the remote system required. After the files have been transferred the install script will be executed by the deployment wizard, the Portus server (optionally) be started.
- Click **Next**

- The next screen is used to deploy the installation files to the target machine, i.e. the machine is where the Portus server will run. Which of the following paths will be taken depends on the choice made regarding the transport mechanism above.

FTP



Portus Server Deployment Wizard

Transfer installation files to target server

Please specify transfer options and additional parameters

FTP Information

Portus server Host/IP: 1.1.1.1

FTP login user ID / password:

Directory to ftp files to (optional):

FTP Advanced

□ Specific FTP Port: 21

□ Passive Mode

Target Information

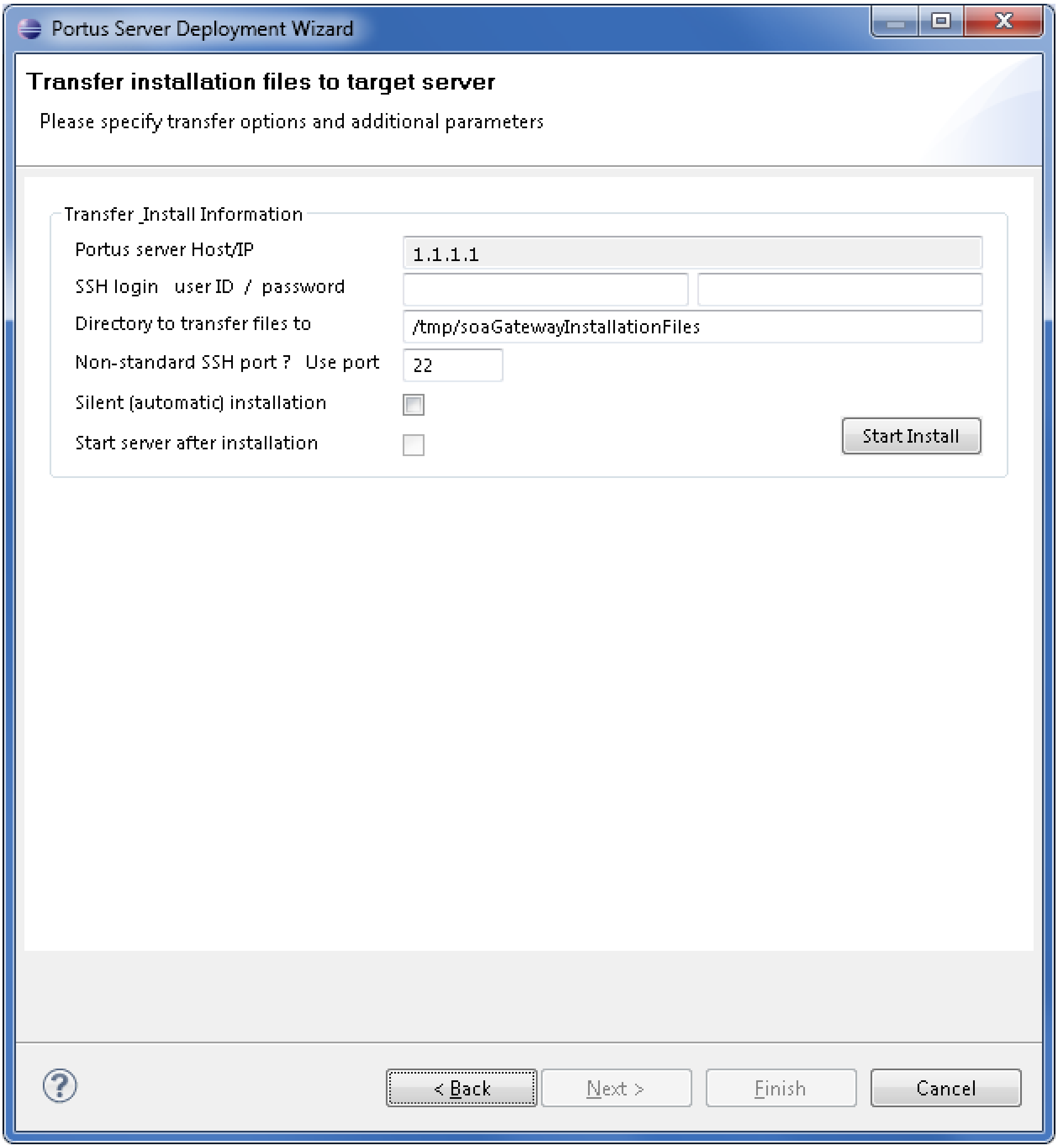
Create datasets □

Installation type: Normal

Additional parameters

- The Host / IP will have already been filled by what has been specified on the previous page.
- Enter the username which is used to login (via FTP) to the server machine (if required)
- Enter the password (if required)
- Enter a directory to send the installation files to. This directory may be relative to the login directory, or can be a fully qualified path name. The directory will be created if it does not exist.
- If your FTP server requires a port other than 21, you can specify that by selecting **Specfic FTP Port** and entering the FTP port number.
- If your FTP server requires passive mode transfers, you can enable this by selecting the Passive Mode checkbox
- **Note:** Click the **Start FTP** button to begin the FTP.
- The *Additional parameters* group will allow you to enter (if required) additional information required for the operation of the driver(s) to be installed. Parameters related to all licensed drivers are queried here, you may omit those for all drivers you do not intend to use.
- Once the FTP has completed, you need to logon to the target machine, and run the *server-specific installation steps*.

SSH



Transfer installation files to target server

Please specify transfer options and additional parameters

Transfer Install Information

Portus server Host/IP: 1.1.1.1

SSH login user ID / password:

Directory to transfer files to: /tmp/soaGatewayInstallationFiles

Non-standard SSH port ? Use port: 22

Silent (automatic) installation □

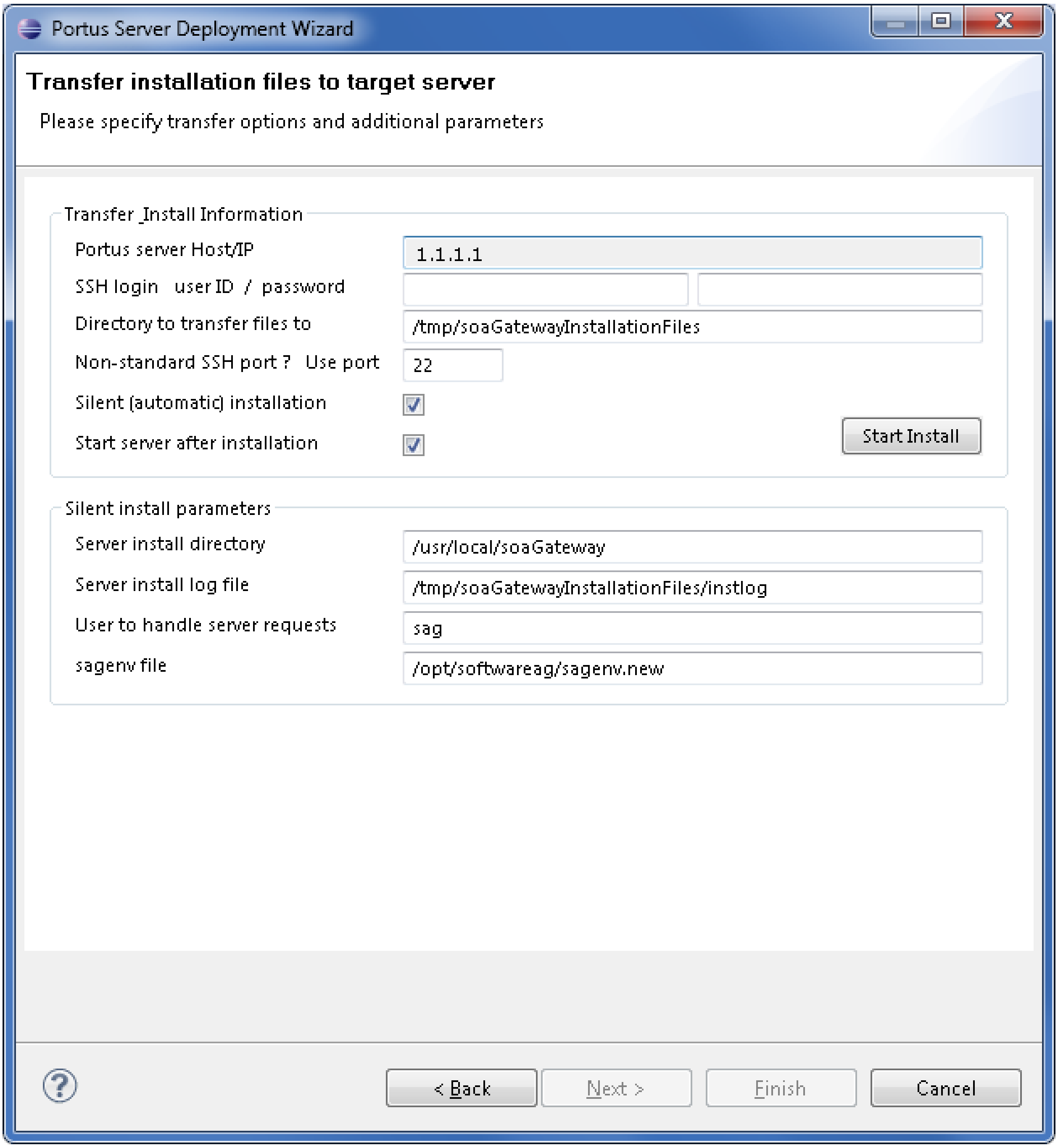
Start server after installation □

Once you have started the transfer you will prompted for your confirmation that it is really the host you want to send the files to.

- If you want the Deployment Wizard to also schedule the actual installation remotely, check the "**Silent (automatic) installation**" option, set the various directories, then click **Start Install**.

  Important: If you do not want to do the installation silently, follow the instructions *here*

  Important: If you are updating an existing installation, the "Silent (automatic) installation" should **not** be checked.



Portus Server Deployment Wizard

Transfer installation files to target server

Please specify transfer options and additional parameters

Transfer _Install Information

Portus server Host/IP: 1.1.1.1

SSH login user ID / password:

Directory to transfer files to: /tmp/soaGatewayInstallationFiles

Non-standard SSH port ? Use port: 22

Silent (automatic) installation ☑

Start server after installation ☑

Silent install parameters

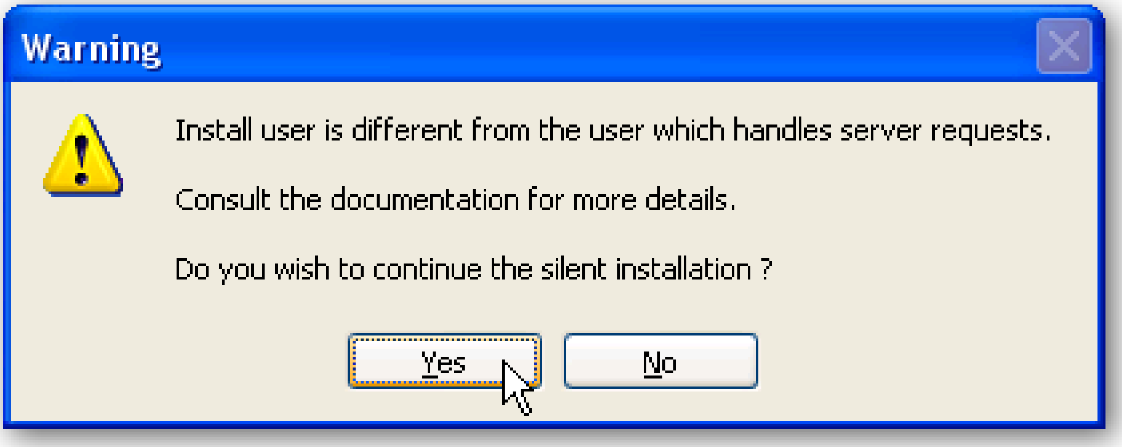
Server install directory: /usr/local/soaGateway

Server install log file: /tmp/soaGatewayInstallationFiles/instlog

User to handle server requests: sag

sagenv file: /opt/softwareag/sagenv.new

- When different User IDs are specified for the "login" and "handle server requests", the following will prompt you for confirmation, please check if the authorization level for the user running the Portus server is sufficient to access and execute files created by the "login user".



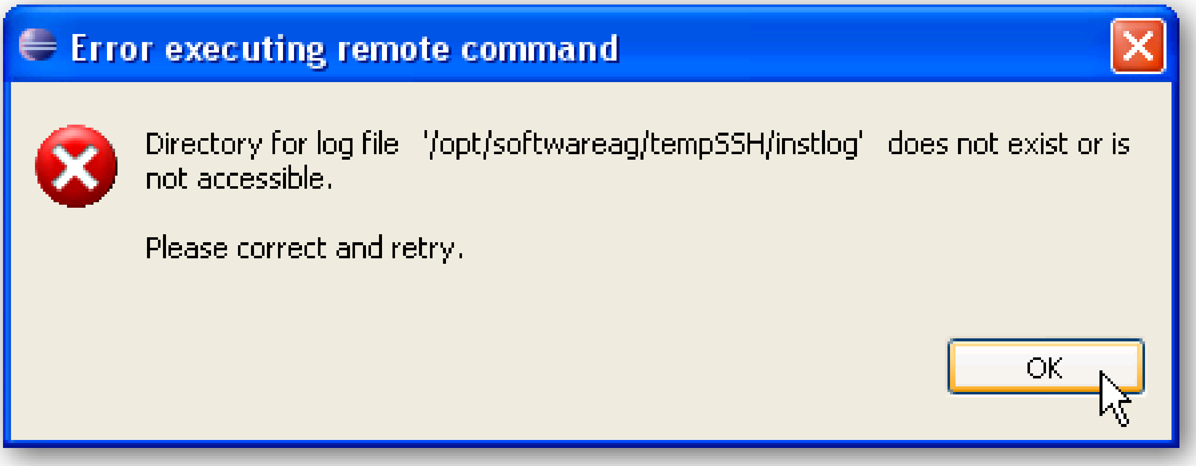
Warning

Install user is different from the user which handles server requests.

Consult the documentation for more details.

Do you wish to continue the silent installation ?

- If the directory specified to contain the install log file does not exist you will be asked to create it before the installation can continue.



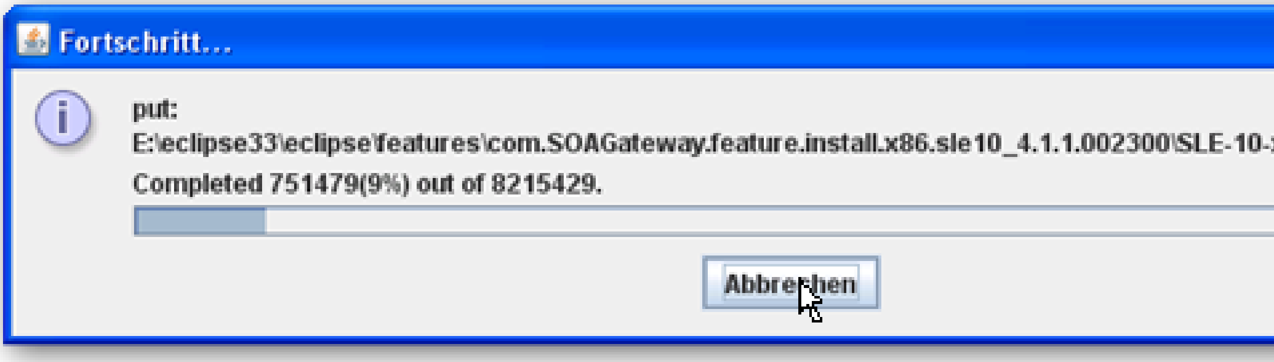
Error executing remote command

Directory for log file '/opt/softwareag/tempSSH/instlog' does not exist or is not accessible.

Please correct and retry.

- All other files will be created by the installation scripts.

  The overall transfer status will be shown directly within the Deployment Wizard window, for larger files an external window will pop up to indicate the progress.



Fortschritt...

put:
E:\eclipse33\eclipse\features\com.SOAGateway.feature.install.x86.sle10_4.1.1.002300\SLE-10-
Completed 751479(9%) out of 8215429.

- **Note:** To enable "passwordless logins" execute the following steps:

1. Run the command : ssh-keygen -t dsa

   e.g.

```
Generating
public/private dsa key pair. Enter file in which to save the
 key
(/home/myUser/.ssh/id_dsa): Enter passphrase (empty for no
 passphrase): Enter
same passphrase again: Your identification has been saved in
/home/myUser/.ssh/id_dsa. Your public key has been saved in
/home/myUser/.ssh/id_dsa.pub. The key fingerprint is:
7d:a9:f9:44:31:c3:d8:6c:d8:c1:d0:5f:39:f8:75:79
 myUser@192.168.135.51
```

2. Copy your public key to the remote server

   e.g.

```
>> scp .ssh/id_dsa.pub
myUser@192.168.135.99:.ssh/ Password: id_dsa.pub 100% 600
 0.6KB/s
00:00
```

3. Login to the remove server and add your public key to the authorized keys.

   e.g.

   ```
   myUser@192.168.135.51:~> ssh -l myUser
   192.168.135.99 Password: Last login: Tue Sep 29 17:02:47
    2009 from
   v-br.vpn.risaris >> cd .ssh/ >> cat id_dsa.pub >>
   authorized_keys2
   ```

   - You should now be able to login to the remote server without a password.

     In the case of problems, SSH information will appear in /var/log/messsages or /var/log/lastlog

- Now that the server has been installed, you can query the server status from the Deployment Wizard.

  You can choose to view this status information in the Deployment Wizard, in a browser, or both.
- Click the **Display Server Status** or the **Display Status in Browser Window** buttons to return the status of the server.
- You should now click **Configure Driver(s)** to create Portus drivers that are enabled in your license. If you choose to not add drivers now, they can also be created at a later stage, more information about defining drivers can be found in the *Portus View* section of this documentation.

  The "Driver Wizard" will present a list of licensed drivers, and come up with all drivers selected and ready to be defined. You may now just click the **Configure** button to define all of them, or deselect those you do not need or want to set up at that stage.

  Click "**Configure**".

  In case the driver does not require any additional parameters to be set, it will be defined now, otherwise the Driver Definition Dialog will ask for the additional parameter(s) to be set. e.g. for an Adabas driver set the "ListMaxRecords" parameter. Click the **Save** button.
- When **Show (post-setup) Driver Information** is selected, and information is available for the driver(s) just installed, it will be displayed (cumulatively) after the driver(s) have been configured
- Click **Done** to close the Deployment Wizard
- Configure Portus using the *Portus (Eclipse) Control Centre*

Server Installation

This section outlines machine specific installation steps.

Host-type: Linux / *ix type

1. On the target machine, in the installation directory, execute the script ./installServer.sh.

   Important: It is recommended that this script runs using the **root** user.

   Important: If you wish to update an existing installation, run this script with the **-update** option.
2. The script will check if the prerequisites are met.
3. Enter the full path of the Portus installation. Example: /usr/local/soaGateway
4. If required, specify the Software AG sagenv file that is to be used (an absolute path must be used). Example: /opt/softwareag/sagenv.new

   Note: If you do not have a sagenv file available, enter "skip".
5. The next prompt asks what user Portus should be started as. If you included a valid SAGENV file in the previous step, you should choose the user who runs Software AG products, normally the "sag"

user. Otherwise, if you started the script as root, lesser privileged user will be chosen, normally www or nobody. If you did not start the script as root, the current user will be chosen.

6 The script now has all the information to commence the installation. Follow the on-screen instructions to start Apache.

Control Centre and Server Update

This will guide you through the process of updating your existing PortusControl Centre and Server.

Note: The information here is for minor updates, e.g. hotfixes, patches, etc. For all major updates (e.g new software release), Portus should be installed anew, and existing services should be copied using the Control Centre. See *here* for more details.

The Portus updates are provided via an Eclipse wizard. When you update your Control Centre (the "Base" feature), the updates are applied after the Find and Install process has finished, and the Eclipse IDE restarts.

Your Portus server will not be updated until you re-deploy the server.

Using the Eclipse Wizard to find and install updates

It is assumed that the Portus Control Centre has already been installed, as detailed in the respective section of this installation guide.

This section describes how to update the Control Centre and also the server installation kits to a level higher than the one currently installed

There are two ways to do the update

1) If you have Internet Access -

- Choose Help -> Install New Software.
- Choose the Portus Update Site URL as provided by product support.
- Uncheck "Contact all update sites during install ... "

2) If you have been provided with a .zip file containing Portus updates -

- Choose Help -> Install New Software.
- Click Add..
- In the Add Repository Dialog box - In the Name field enter a name for this update meaningful to you
- Click the Archive... button and browse to the location of the .zip file and select it.
- Click OK

If newer versions are found, they will be presented as available, select the item(s) you wish and Click Next and you will be asked to restart eclipse.

If you followed the steps above, you now have successfully updated your Control Centre, and have downloaded the updated server installation files.

Note: The Portus Server has not been changed at this stage. You need to re-deploy to update the server.

Update Server - Windows

From the Portus Deployment Wizard click the "Launch Setup" button. The wizard will check for an existing version and if located, will proceed with the update.

Note: The check for an existing install may take 30+ seconds on some machines that have a large amount of data in the registry.

Once an the wizard has located the install it will display a confirmation dialog. When confirmed the wizard will begin the process of backing up configuration files, updating and restarting the server. The Windows UAC (if enabled) will prompt you for Administrator privileges a number of times during this process.

Note: The server will be unable to restart if the updated server is installed to a different disk to the previous installation. E.g. If an existing server is installed on C: and is uninstalled and reinstalled to D: it will fail. If this occurs a manual restart of the Apache server is required. The server must be stopped then started.

After the process completes return to the control center view and refresh the server.

Note: If you have changed the Portus Windows Service to start as a user *other* than SYSTEM, you need to run "services.msc" and re-apply this modification

Manual Backup for Failed Install

If the MSI encounters an error during the reinstall process the automated restoration of config files will not complete and a manual backup is required or the data will be lost.

Files will be backed up to the OS default temp directory. The temp directory can be located by clicking start->run and entering %temp% .

Inside this folder you will find the product license file, sampleConfig-win.xml, portusHttpd.conf and portusSSL.conf. Copy these files to a new location (eg. Desktop).

Follow the steps to install a new server. Once the install is complete go to the ProgramData (hidden by default) and copy the 4 files you backed up to this directory.

C:\ProgramData\Portus 4.3.1\configuration

Update Server - non-Windows Platforms

z/OS

You can run a new Portus server side-by-side with the existing server, so long as the datasets names and the port that the server listens on are different.

Alternatively, you can re-deploy a new server, and simply use the new LOAD library in the STEPLIB of your existing installation.

Follow the instructions *here* to deploy the updated server.

Note: Ensure you choose new dataset names.

Follow the instructions *here* to copy web services between servers.

*nix Platforms

You can run a new Portus server side-by-side with the existing server, so long as the installation location and server port are different.

Alternatively, you can transfer the updated installation files, and provide the -update option when starting the installServer.sh script.

Follow the instructions *here*.

Note: If you want to create a new installation, ensure you choose a new installation directory, and a server port different from the existing.

Follow the instructions *here* to copy web services between servers.

If you have updated an existing installation, then the update is now complete.

Server uninstallation guide

- *Uninstall Portus Server - Linux, z/Linux, AIX and Solaris* on page 50
- *Uninstall Portus Server - Windows* on page 50

Uninstall Portus Server - Linux, z/Linux, AIX and Solaris

1 Ensure the Portus server is not running, if it is running change to the <SERVER_install_dir>/apache2/bin directory, replacing <SERVER_install_dir> with the location where you installed the Portus server.

Run the command ./apachectl stop

2 The <SERVER_install_dir> can now be removed. Use the command rm -rf <SERVER_install_dir>.

Important: Once you run this rm command, all files and directories in "<SERVER_install_dir>" will be permanently deleted.

3 This completes the uninstall process.

Example:

```
# cd /usr/local/server/apache2/bin
# ./apachectl stop
# cd /usr/local
# rm -rf server
#
```

Uninstall Portus Server - Windows

1 Start the Windows Control Center by clicking **Start -> Control Panel**

2 Select the **Add or Remove Programs** or **Programs and Features** (depending on your version of Windows)

3 Select the Portus version you want to uninstall and click **Change/Remove**

4 Select **Remove** radio button and click **Next**

5 Click **OK** to uninstall Portus

Chapter 4

Administration

Topics:

- *Introduction to the Portus Control Centre*
- *The Control Centre*
- *Portus BusinessDataViews*
- *Server administration*
- *WebService Creation*
- *Using Portus with ...*
- *Portus Statistics*
- *Governance*
- *Portus Security*
- *Advanced Topics*

Portus Administration overview

Portus administration.

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

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

Introduction to the Portus Control Centre

The Portus Control centre provides a single-point-of-control facility for managing Portus server configurations.This information was developed for ...

If the Portus perspective is not active already open it by selecting Window, ->Open Perspective -> Other -> Portus.

First steps with the Portus Control Centre - Administration Perspective

1 The primary elements of the Portus Perspective are

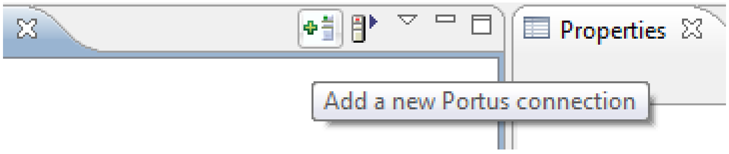
- **Portus View -** lists defined servers and their defined drivers and services.
- **Portus Action Log** - displays informational and error messages
- **Properties View -** lists specific detail about a specific service / resource
- **Navigator View -** displays the contents of workspace in use. The files listed here are on the local disk.
- **Editor View -** this opens the required resource in the appropriate editor.

Important: At any time, you can choose **Window, Reset Perspective** to restore the original

layout.

2 The first action to be carried out is to define a server.

Important: The Deployment Wizard may have created a server already, so this step may be skipped in this scenario.



Click on the server icon to the right of the **Portus** tab to bring up the **add New Server Dialog**.

- **ServerName**: Enter a symbolic name for the server to be defined.
- **Host/IP**: Enter the servers host name / IP address.
- **Port**: Enter the port the server is listening on.
- **Timeout** (sec): In case the server is slow in responding to requests from the Control Centre, the timeout may need to be set to a value higher than the default of 10 seconds.
- **Autodisconnect**: Check this box if you want each of the defined servers to be put into a suspended mode when Eclipse exits. This is be useful as it speeds up Eclipse initialization. Although, once the Control Centre starts, you will have to resume each of your servers before working with them again.

Click "**Save**".

The server will appear in the Portus View. Servers whch are reachable are indicated by a green arrow, unreachable servers are denoted by a red circle, a message in the Properties area indicates the reason why the server cannot be contacted. Clicking on a server entry in the Portus View will trigger the display of the server's configuration in the Properties View. The PropertiesView gives comprehensive information about each server's configuration. The side tabs are:

- General
- Tracing
- Global
- Drivers
- Governance
- Statistics

3 More information about the various views and their functions can be found in the Portus Control Centre section.

The Control Centre

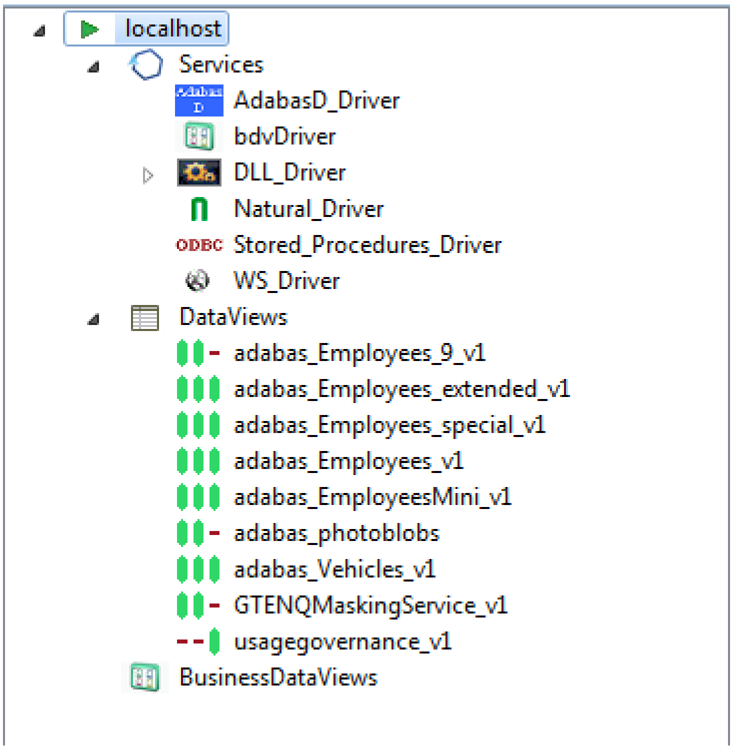
Portus Control Centre overview

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

The Portus View

The Portus View lists, and allows operation on

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



Operations at the *Server* level

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

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

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

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

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

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

  1 **Refresh Configuration**: When a Portus Server configuration is modified "on-the-fly", for example by modifying, adding or deleting resources, these changes are volatile, they are not

preserved over a server restart. Additionally, a server configuration may have been modified by someone else. In either case, your local representation of the Portus server configuration may no longer be accurate, issue a Refresh to synchronize it by selecting the "Refresh Configuration" option on the context menu. A message will be written to the Action Log as well as the status line after the refresh operation is completed.

2. **Disconnect from Server**: When many servers are defined, "inactive" servers will still be contacted when you start the Control Centre. This may take quite a while until the TCP/IP timeout is reached, so the connection to knowingly "inactive" servers can be "suspended", which will cause them to be ignored until the connection is reestablished. For disconnected servers the "Disconnect" context menu item will change to "Connect to Server".
3. Set Credentials for Server: Set the login credentials (HTTP and SOAP User Id and password) for secured Portus servers.

   Important:

   To use HTTP credentials you must have a <Location /configurationService> directive enabled.
4. **Add driver** to Server:
5. **Add new Portus connection**: Define a new Portus server to the Control Centre
6. **Remove Server definition**: Deletes the selected entry from the Servers View.
7. **Hide this Server** Hides the selected entry from the Servers View. Can be unhidden if required.

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

  Here you can

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

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

Server status display / parameter modification

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

To modify the Portus server entry:

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

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

Modify tracing and message level options

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

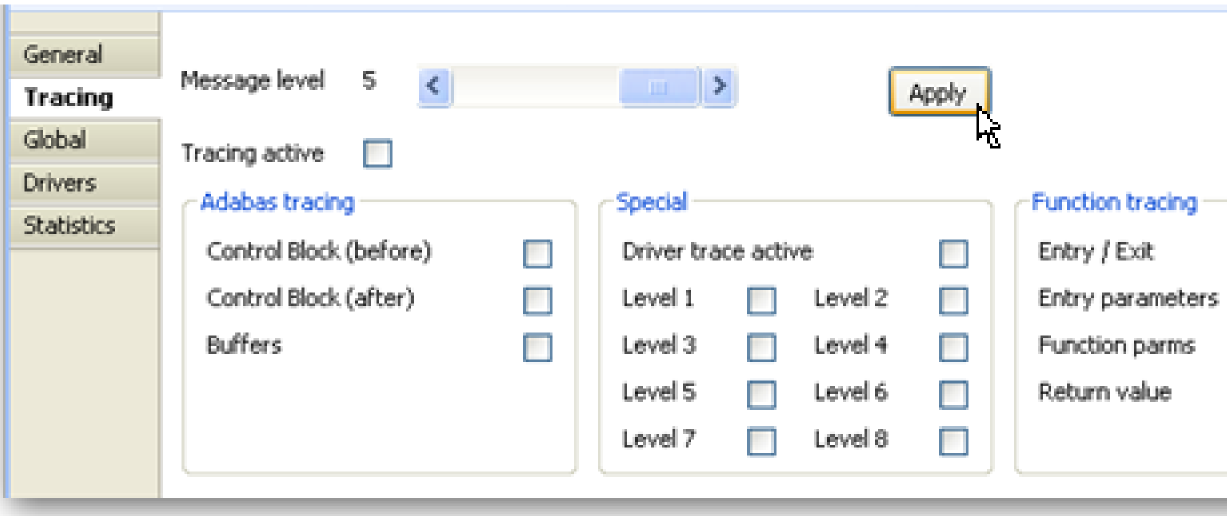
Important:

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

To change either the message level or trace options:

1. Select the "Tracing" tab from the Properties view

2 Modify the message level or tracing option(s), as appropriate, click "Apply" to send the changes to the Portus Server.



Message level 5

Tracing active □

Adabas tracing

Control Block (before) □

Control Block (after) □

Buffers □

Special

Driver trace active □

Level 1 □ Level 2 □

Level 3 □ Level 4 □

Level 5 □ Level 6 □

Level 7 □ Level 8 □

Function tracing

Entry / Exit

Entry parameters

Function parms

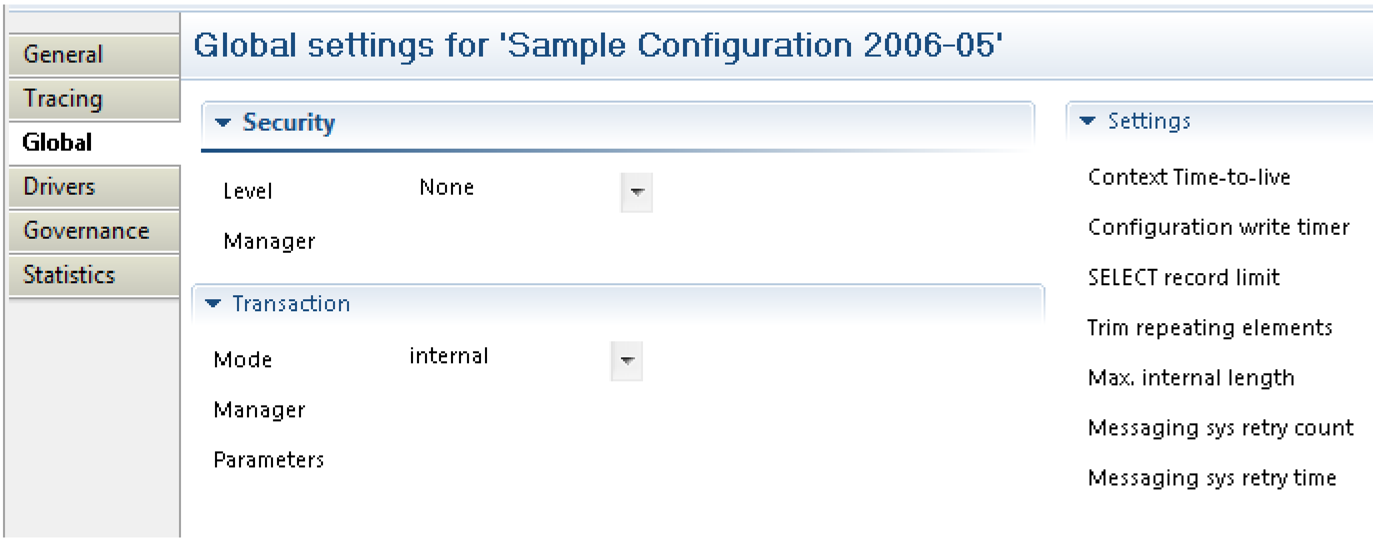
Return value

Modify global settings

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

Important:

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



Global settings for 'Sample Configuration 2006-05'

Security

Level None

Manager

Transaction

Mode internal

Manager

Parameters

Settings

Context Time-to-live

Configuration write timer

SELECT record limit

Trim repeating elements

Max. internal length

Messaging sys retry count

Messaging sys retry time

Security Settings

There are 4 security levels that Portus can run at

1 **None**: No Portus security.
2 **Userid**: All requests must have a user ID.
3 **Password**: All requests must have a user ID and password.
4 **SSLCert**: All requests must have a SSL 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 Portus uses to handle data integrity.

Server Settings

You can set specific options for the server

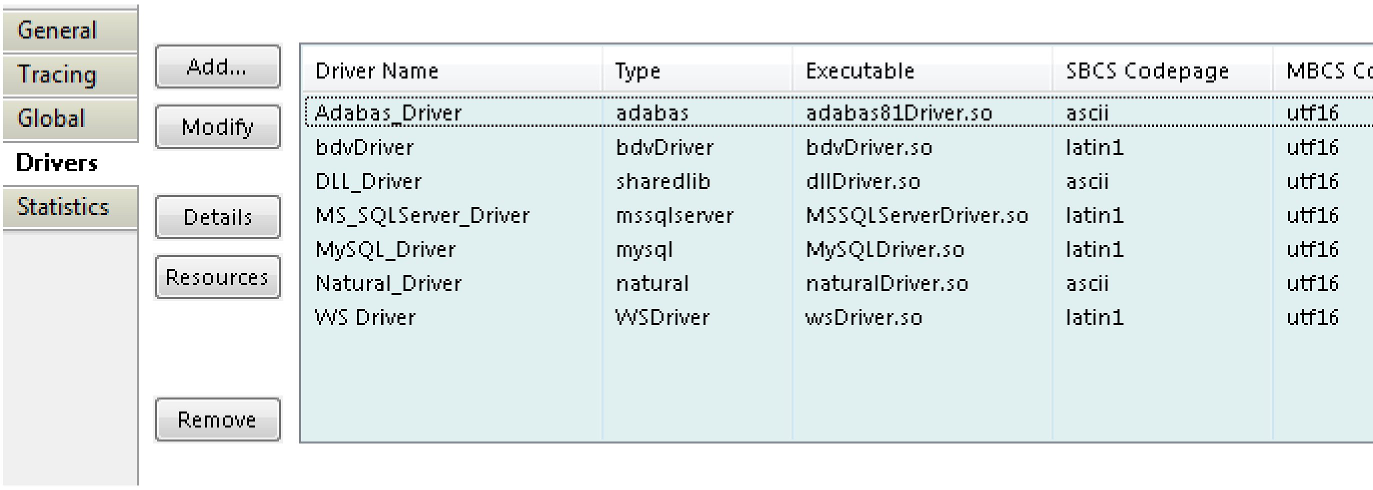
1 Context Time-to-live: specifies the time (in seconds) after which orphaned contexts (for example due to timeouts or broken connections) will be considered "dead" and cleaned up.
2 Configuration write timer: The in-core copy of the Portus configuration will be automatically saved to file after the specified interval (in seconds) elapsed. A value of 0 means the current configuration will never be saved back to the configuration file.
3 Select record limit: Limit the amount of records returned from a "select" or "select next" call.
4 Trim repeating elements. This option only applies to Natural-based web services. Repeating array elements will be trimmed based on the first empty field found. For example, if the service returns . . . <myArray>ONE</myArray><myArray>TWO</myArray><myArray/><myArray/><myArray/> . . ., when this option is turned on the output will be trimmed to . . .<myArray>ONE</myArray><myArray>TWO</myArray> . . ..
5 Max internal length: An upper limit on the size of a field in the XRD.
6 Messaging sys retry count: If using Portus to connect to a messaging system such as EntireX or MQ, it may be that a failure occurs attempting to connect due to a database not being started. A retry count can be set to allow the database to be started and the connection to succeed.
7 Messaging sys retry time: This sets the time to wait until another attempt is made to connect to whatever system has initially caused the messaging system to throw an error on connection.

Add / modify drivers

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

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

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



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

- You may left-click a driver entry and select **Details** to display more information about this driver.
- To modify an existing driver, left-click the driver,and select **Modify**
- To add a new driver, click the **Add...** button.

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

Preset values are presented , modify as appropriate.

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

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

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

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

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

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

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

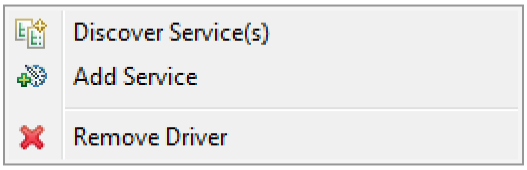
Suggested Name	Library Name	Suggeste SBCS	Suggeste MBCS	Notes / Other Options / Default values	More Info
Adabas_Driver	ADADRVR	CP1141	utf16	ListMaxRecords is optional	
Natural_Driver	NATDRVR	CP1141	utf16	Natural Batch Pgm : NATBATCH Init Params : ETID=OFF Pre-Init Sessions : 2 Max. Sessions: 8 Natural Library Name : SYSSOA Natural Steplibs: SYSSOAEX,SYSEXT Natural Security: No **Note:** Natural libraries are required in the Portus server STEPLIB, the Natural Batch Nucleus used MUST NOT	

Suggested Name	Library Name	Suggeste SBCS	Suggeste MBCS	Notes / Other Options / Default values	More Info
				be LE-enabled (specify LE370=NO).	
CICS_Driver	CICSDRVR	CP1141	utf16	CICS APPLID: specify target CICS system application id CICS EXCI library is required in the Portus steplib.	
VSAM_Driver	VSAMDRVR	CP1141	utf16		
LE_Driver	LEDRVR	CP1141	utf16	Runtime Options: TRAP(OFF,NOSPIE),RPTOPTS(ON),RPTSTG(ON) envMax: 50 preEnvInit: 5	
DLL_Driver	DLLDRVR	CP1141	utf16		
DB2_Driver	DB2DRVR	CP1141	utf16		

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

Operations at the *Driver* level

The following context functions are available at the driver level



1 Discover Service(s) - This starts the Discovery Wizard which can be used to create web services.

2 *Add Service*

3 Remove Driver, which is only possible when no service exists for the driver

Add a Service

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

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

1 Enter the name of the Service. This will have an impact on the URL used to access the WSDL, and the endpoint the service uses.

2 The Control Centre will automatically create an "empty" DataView or BusinessDataView for you, based on the type of the driver, and will open the editor on the newly created view.

3 A tutorial on creating a DataView from scratch is available here

4 Once the file is saved, the (Business)DataView will be automatically exported back to the server and a XSD be generated., if the respective option is enabled

5 Select the new service, and edit the Service Properties to suit your back-end resource.

Operations at the *Service* level

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

Operations at the *DataView / BusinessDataView* levels

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

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

Note:

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

Drag-and-drop in the Portus View

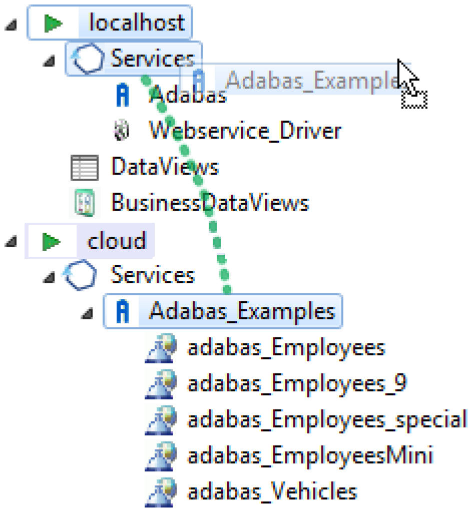
The Portus View incorporates extensive drag-and-drop support to copy between servers, and even to rearrange elements within a server environment.

The following operations are possible

- *Driver definition as a drag source* on page 59
- *Service as a drag source* on page 59

Driver definition as a drag source

Dragging a driver to the Services element of another server



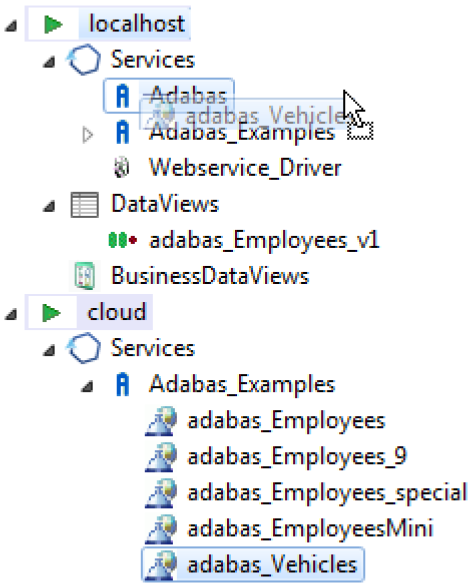
will copy the driver definition as well as all services defined using this driver, along with all definition files ((Business)DataViews, XSDs, XSLs)

Service as a drag source

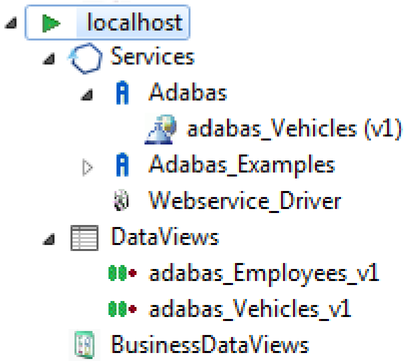
- *Dragging services from one server to another* on page 59
- *Dragging services from one driver to another on the same server* on page 60

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 the service as required.

DataView and BusinessDataView as a drag source

(Business)DataViews can be dropped as follows

- *Dropping a (Business)DataView on a Service* on page 60
- *Dropping a (Business)DataView on a Server definition* on page 61

Dropping a (Business)DataView on a Service

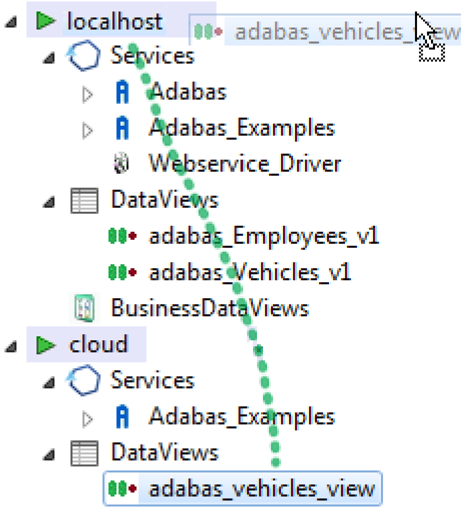
A DataView or BusinessDataView dropped on a service will replace the respective view currently used by that service.

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



Portus Service Configuration

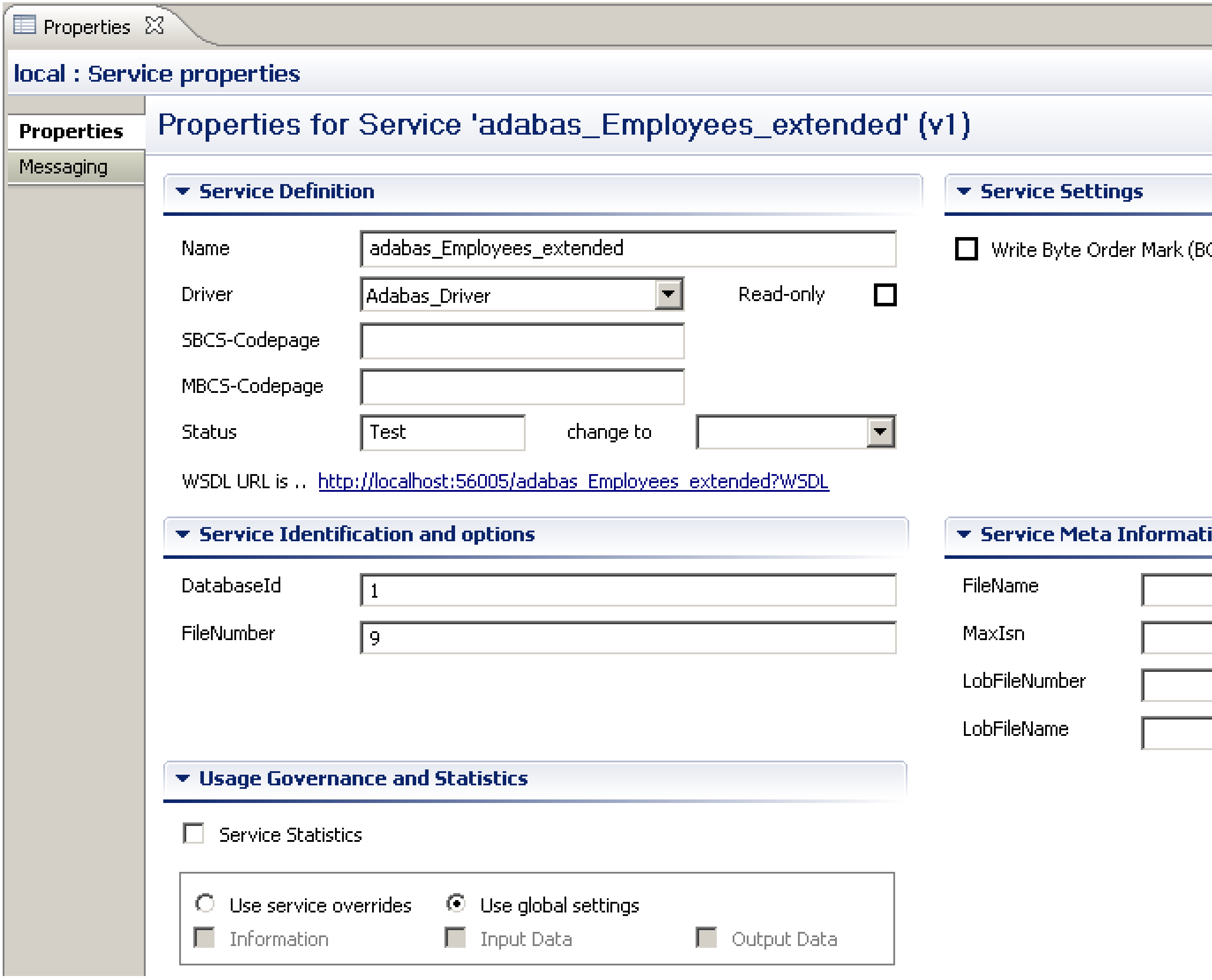
When a driver entry is expanded, the list of services associated with this driver is also displayed.

Left-clicking the web service name brings up the *Service Properties* on page 62 in the Properties view.

Right-clicking the web service name brings up the *Service context menu* on page 63 Menu

This section also provides information about *Lifecycle Governance* on page 66 (LG)

Service Properties



Properties

local : Service properties

Properties
Messaging

Properties for Service 'adabas_Employees_extended' (v1)

Service Definition

Name: adabas_Employees_extended

Driver: Adabas_Driver

Read-only □

SBCS-Codepage:

MBCS-Codepage:

Status: Test

change to:

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

Service Settings

□ Write Byte Order Mark (BO

Service Identification and options

DatabaseId: 1

FileNumber: 9

Service Meta Informati

FileName:

MaxIsn:

LobFileNumber:

LobFileName:

Usage Governance and Statistics

□ Service Statistics

○ Use service overrides ● Use global settings

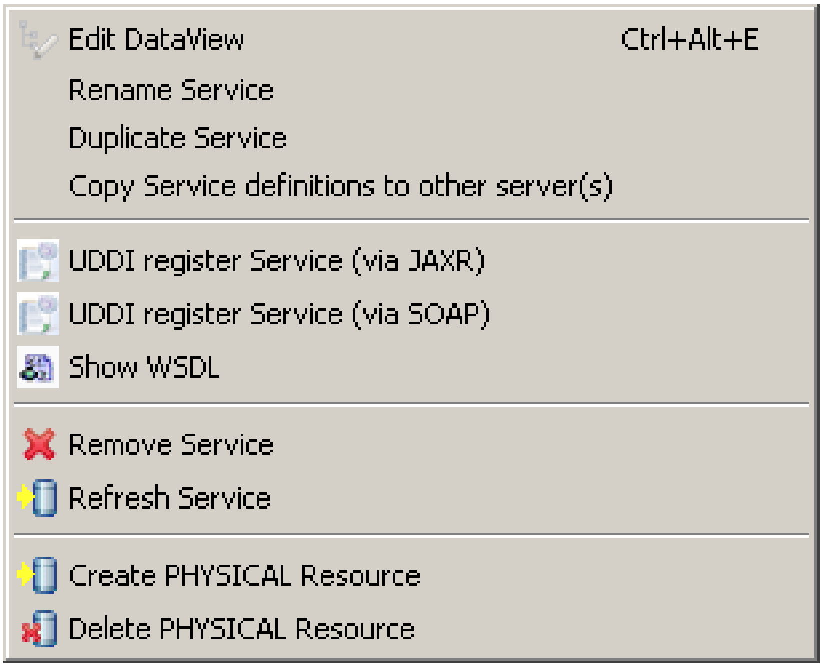
□ Information □ Input Data □ Output Data

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

- **Name** : The name of the web service. This will determine the WSDL URL and service endpoint.
- **Driver**: The driver in use for this web service.
- **SBCS/MBCS** : The SingleByte/MultiByte codepages used for data conversion with this service.
- **Service Status**: For more details see here
- **Service Identificatio**n. 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

Note: 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* on page 63: Rename the service. This will have consequences on the URL of the WSDL, and the endpoint of the web service.
- *Duplicate Service* on page 63: Create a new web service that is a copy of the existing. This service needs to have a different name from the original.
- Refresh Service: clears the internal cache on the Portus server to pick up changes to the (Business)DataView, XSD or XSL
- *Copy Service definitions*
- *UDDI Register* Service via JAXR or SOAP
- Show WSDL : Display the WSDL for this service. This is the starting point of for any web service clients that wish to consume this web service.
- Remove Service: Remove this service from the Portus Server

  Note: If the Remove orphaned dataviews box is checked, the internal configuration files (XRD, XSD and XSL) will also be deleted from the server.
- Create Physical Resource: Create a back-end resource (E.g Adabas file/MySQL Table/etc), physically, based on the Portus service.
- Delete Physical Resource: Physically delete a back-end resource described by the web service.

Rename Service

1. Select "Rename Service" from the context menu
2. 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.
3. 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 service parameters as needed

Copy Service definitions to other server(s)

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

- using the *Copy Wizard*
- using *drag-and-drop*

Copy Service(s) using the Copy Wizard

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

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

  Process:

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

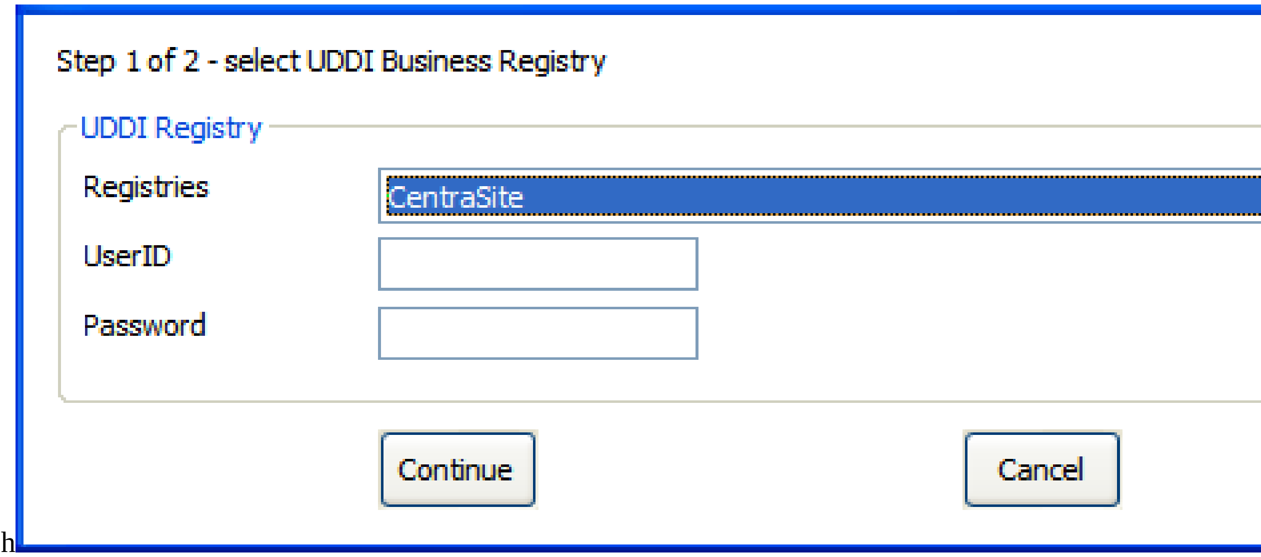
Register a Service with an UDDI Business Registry (UBR)

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

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

1. Define a UDDI Business Registry to the Portus Control Centre (from the Eclipse Preferences dialog)
2. Select the **UDDI register Service** function from the context menu

3 Select one of the predefined UDDI Business Registries, enter the user credentials required to be able to publish



Step 1 of 2 - select UDDI Business Registry

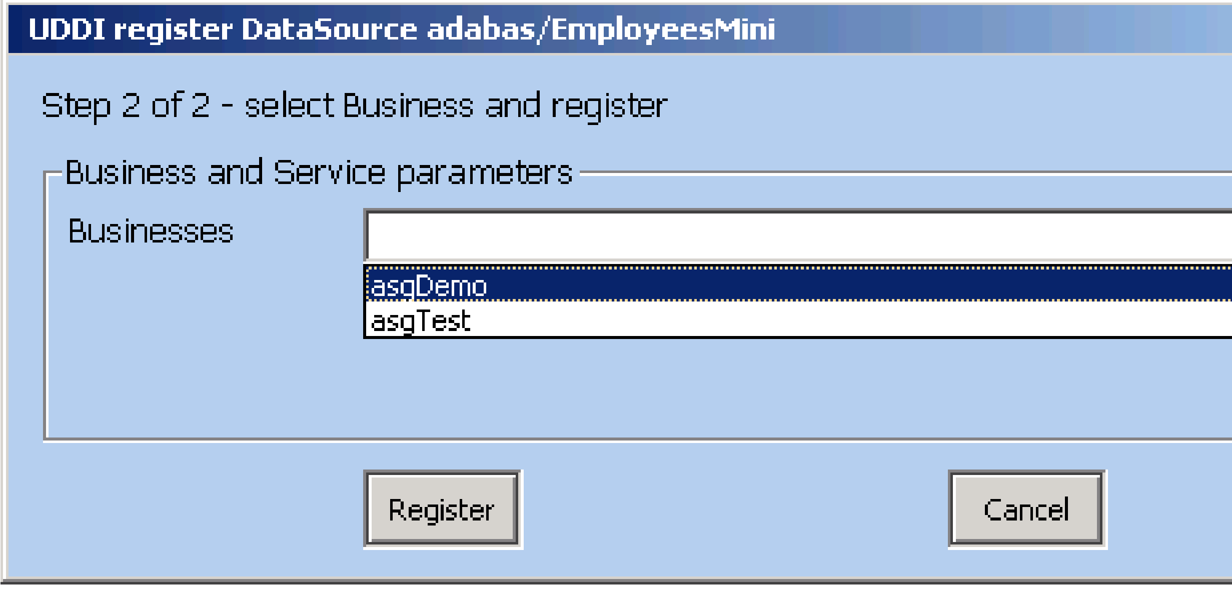
UDDI Registry

Registries: CentraSite

UserID:

Password:

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



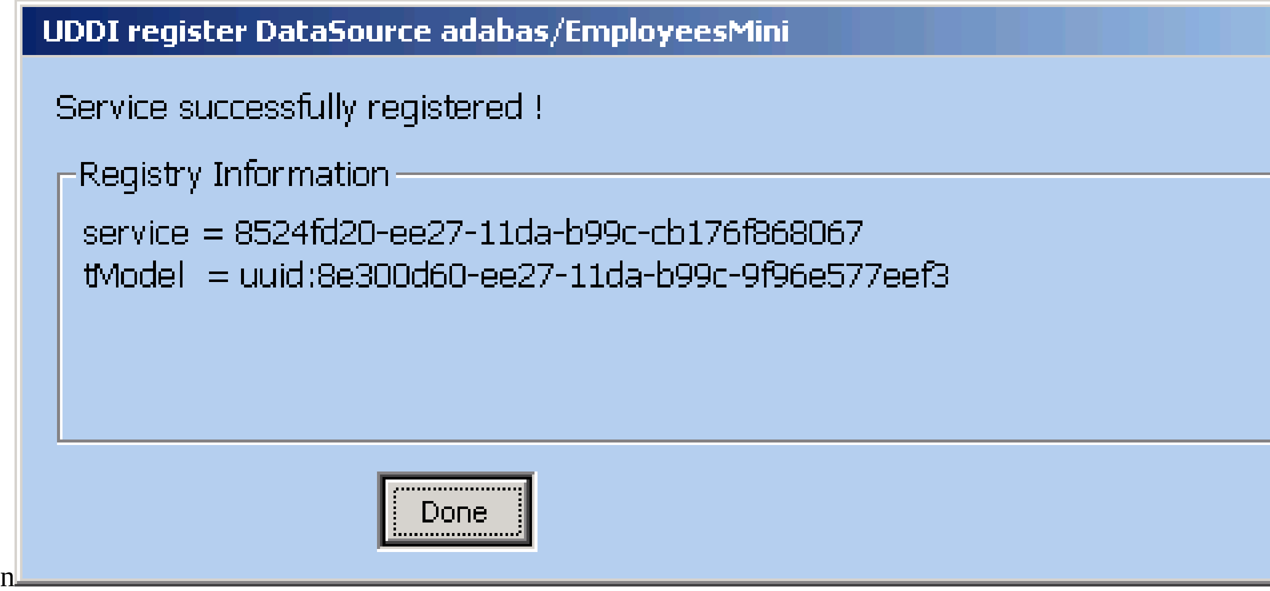
UDDI register DataSource adabas/EmployeesMini

Step 2 of 2 - select Business and register

Business and Service parameters

Businesses: asgDemo
asgTest

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



UDDI register DataSource adabas/EmployeesMini

Service successfully registered !

Registry Information

service = 8524fd20-ee27-11da-b99c-cb176f868067
tModel = uuid:8e300d60-ee27-11da-b99c-9f96e577eef3

6 Click **Done**

Lifecycle Governance

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

Service Version

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

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

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

  The WSDL will contain a service endpoint which reflects the service version.
- When working with REST requests, the same __version argument can be used.

  ```
  http://host:port/myService?LIST&__version=5&ID=4*
  ```
- Internally, a new DataView is created for every service version. The DataView name will have the version number appended on the end of the file name.

Service Status

There are 4 statuses a Portus service can be in:

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

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

   When the Administrator edits the dataView of a frozen service, a new version of this service will be created, it's status set to test, and it's version number incremented by 1.
3. **Deprecated**: Clients will still be able to call a service with this status, but they will receive a warning message in the SOAP headers. A warning message will also be written to the Portus log, and the client should be urged to upgrade.
4. **Historical**: The service is no longer active, and clients who attempt to call this service will receive a SOAP fault.

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

Current Status	Next Status
Test	Frozen
Frozen	Test, Deprecated
Deprecated	Test, Historical
Historical	Deprecated

The Monitoring View

The Control Centre Monitor View allows the server log files to viewed from within the Control Center.

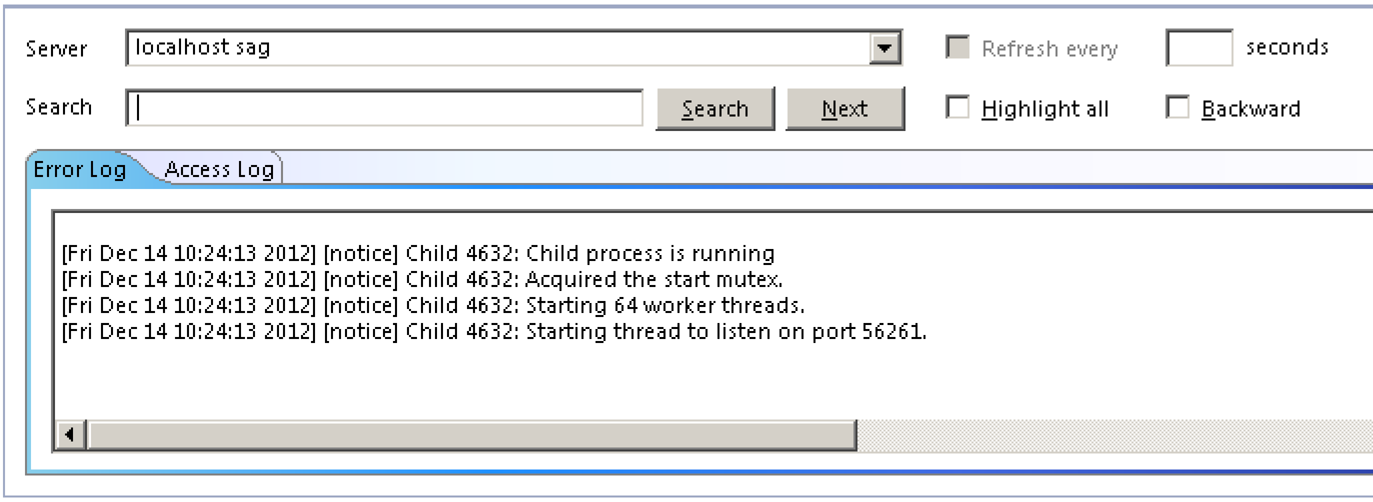
The Monitor log is not available on mainframe server installations

To start the Monitor View, select Window -> Show View -> Other -> Portus Monitoring

Select your server from the drop-down menu. The Control Centre will connect to this server and download the server logs (Apache Error and Access logs).

Attention: If the logs are very large, this may take some time

The **Refresh** button will download the latest log changes



Server: localhost sag

Search:

```
[Fri Dec 14 10:24:13 2012] [notice] Child 4632: Child process is running
[Fri Dec 14 10:24:13 2012] [notice] Child 4632: Acquired the start mutex.
[Fri Dec 14 10:24:13 2012] [notice] Child 4632: Starting 64 worker threads.
[Fri Dec 14 10:24:13 2012] [notice] Child 4632: Starting thread to listen on port 56261.
```

The Portus Control Centre Action Log

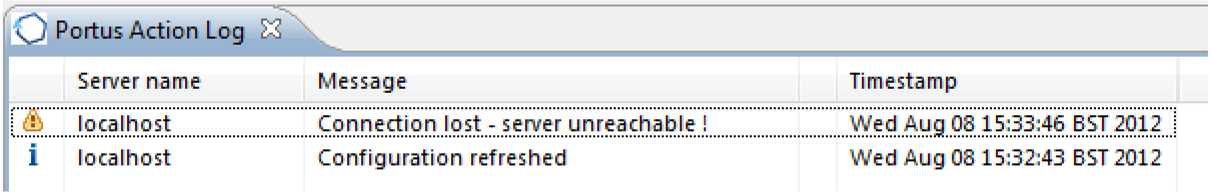
The Action Log View displays information and error messages related to the 'client' actions carried out within your Control Centre session.

The action log does NOT show any server messages, these are available through the Apache log-files on your server(s).

Working with the Action Log

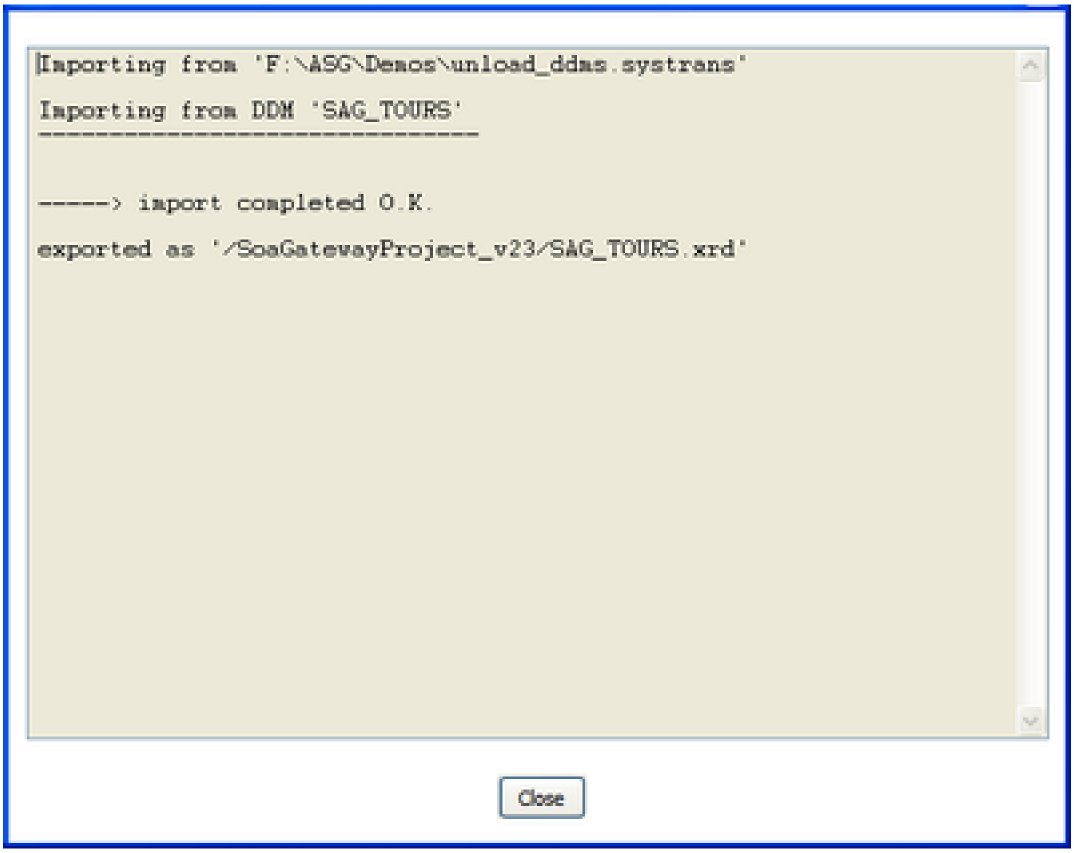
These are the columns visible in the Action Log View

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



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

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



```
Importing from 'F:\ASG\Demos\unload_ddms.systrans'

Importing from DDM 'SAG_TOURS'
--------------------------------


-----> import completed O.K.

exported as '/SoaGatewayProject_v23/SAG_TOURS.xrd'
```

Click the 'Close' button to dismiss the window.

Portus BusinessDataViews

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

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

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

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

- *Overview / Concepts* on page 69
- *Using the BusinessDataView WebService* on page 72
- *Working with BusinessDataView Objects* on page 72
- *Additional features* on page 74
- *Summary* on page 76
- *Creating BusinessDataViews* on page 76
- *BusinessDataView Editor hints* on page 98

Overview / Concepts

All businesses have enormous amounts of data which is exposed via various interfaces and applications, however, rarely, if ever, does one application, one database table or one file represent a complete piece of business related data.

The reason for this is straightforward in that most IT projects tend to focus on the ‘data model’ and the efficiency of getting at data as the primary driver when designing such systems.

This is still core to system design today to ensure that the physical architecture can handle the loads anticipated for a given system. However, it generally means that new projects must deal with data models that are sometimes hard to understand from a business perspective.

Portus technology now offers the ability to map the existing data and business logic assets into composite views that reflect the business view of the data. This means that newer projects can deal with business related concepts and objects when viewing or updating the core assets.

A Simple Example

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

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

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

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

2 The Vehicle Table or File

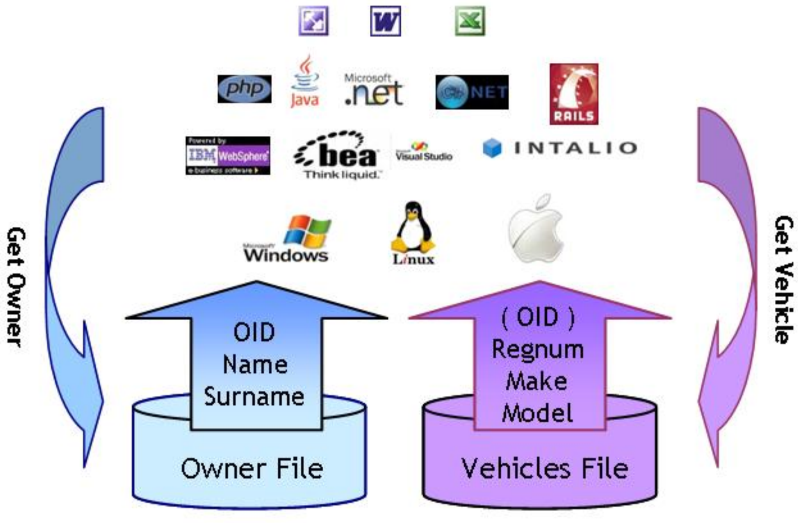
> **Note:** This will contain details about the vehicles that are owned by owners. For the purposes of this illustration, we will assume the following fields:

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

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

The traditional approach

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



Traditional Implementation

There are a number of issues with this scenario:

1 For each request to get the data there is a requirement to make two requests to the back end database across what will normally be a busy network

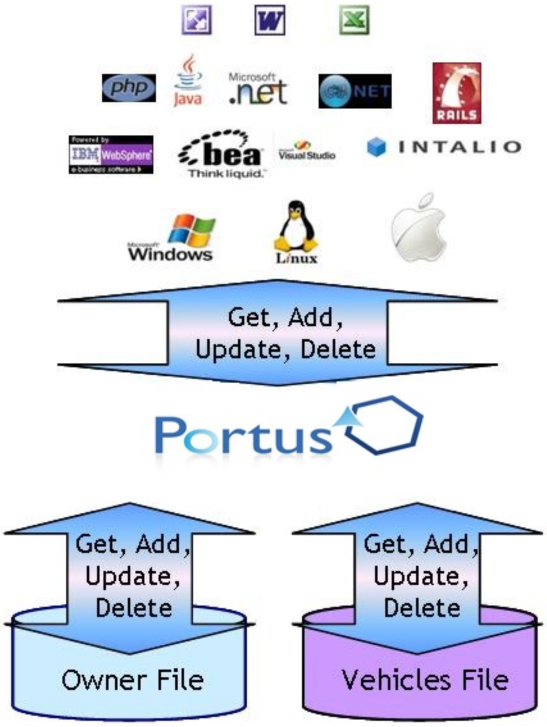
2 The application needs to understand that the ‘OID’ returned is used to get the associated entry in the Vehicles table/file. While in this case, the OID number may be of value to the client, it may not be necessary for the client to see or know this. In many cases, keys are used simply as a means to locate a unique record and have no value outside the database itself. However, these must still be understood and manipulated by the programmer.

3 For traditional access, software must be installed on the client system to access the back end databases. This costs time and money to keep up to date and, in many cases, has additional license costs associated with it.

4 If the data model is not clear to the programmers developing on the client side, the databases may end up with inconsistent data.

5 If the data is in multiple databases, there may be an issue with synchronizing updates to those databases unless an infrastructure supporting the two-phase-commit (2PC) protocol is available and in use.

The Portus approach



This scenario offers a number of substantial advantages:

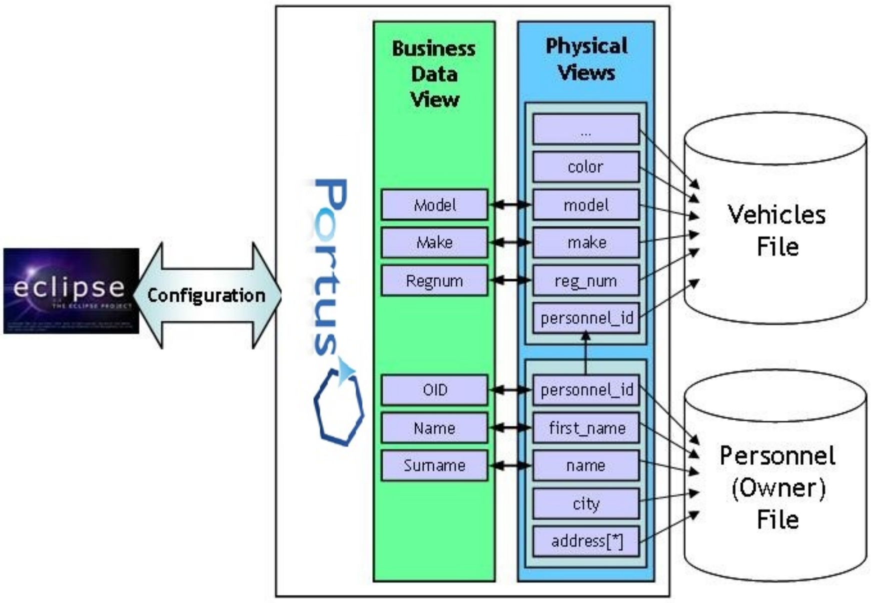
1. Only one trip to the back end system is required to retrieve or update the composite object.
2. The programmer does not have to know or understand internal links between tables or files and just has to deal with the fields and columns of relevance to the application.
3. Portus needs to be installed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

Creating a BusinessDataView Service

A BusinessDataView based WebService is created in Portus based on existing "physical" WebServices. The physical service is the service that is created to directly expose a table, file or piece of business logic.

This means that existing services defined to Portus may be reused in one or more Business Data Views.

The Eclipse Control Centre supplied with Portus is used to create the business data view as illustrated in the following diagram:



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

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

Using the BusinessDataView WebService

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

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

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

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

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

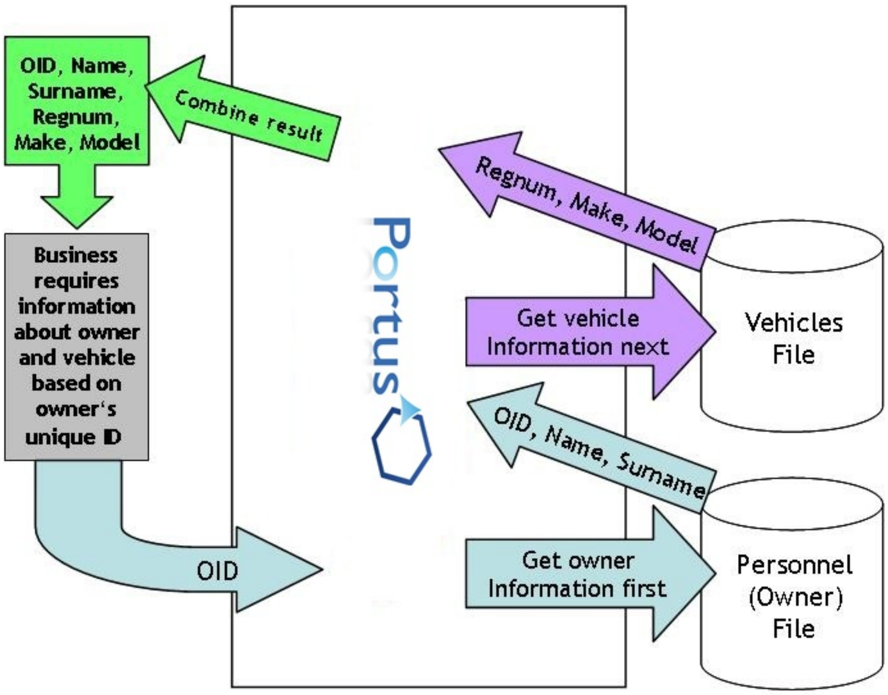
```
http://<host>:<port>/OwnerVehicle?F=LIST&OID=1234*
```

Working with BusinessDataView Objects

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

Accessing a BusinessDataView Object

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

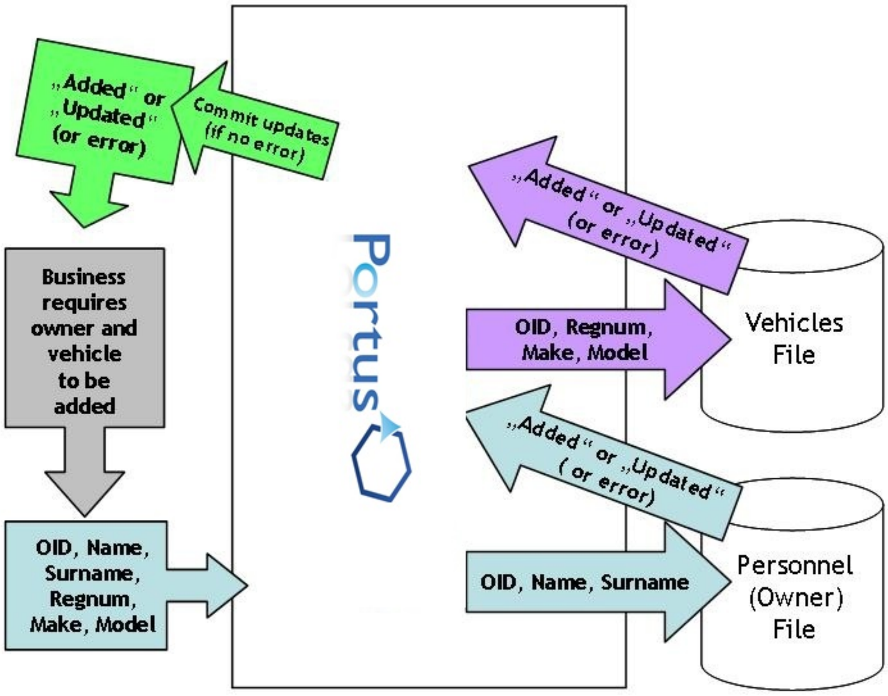

OID, Name, Surname, Regnum, Make, Model
Combine result
Business requires information about owner and vehicle based on owner's unique ID
OID
Portus
Regnum, Make, Model
Get vehicle Information next
Vehicles File
OID, Name, Surname
Get owner Information first
Personnel (Owner) File

Some notes about the above:

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

Adding or updating a BusinessDataView Object

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

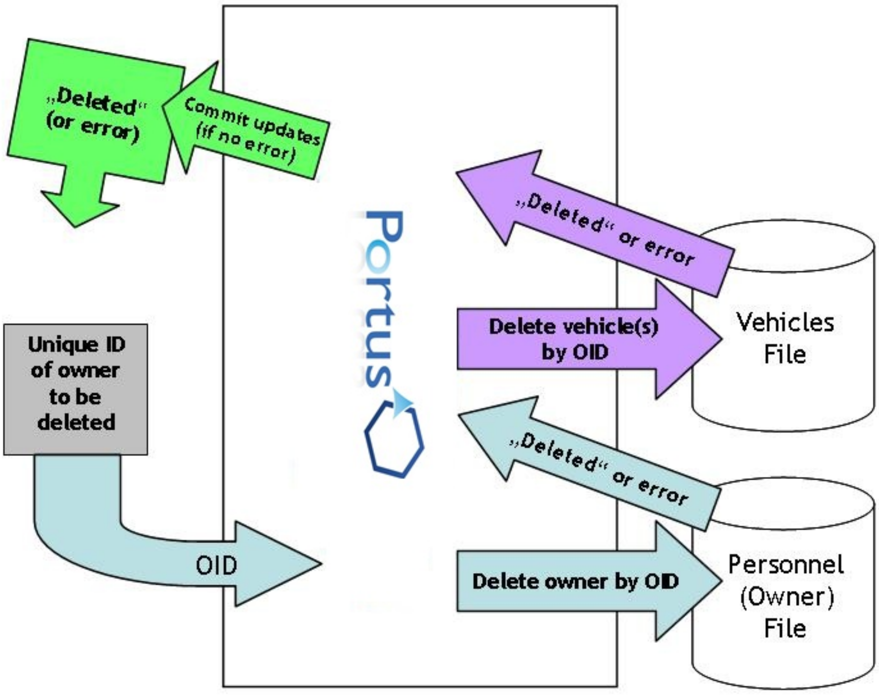

„Added" or „Updated" (or error)
Commit updates (if no error)
Business requires owner and vehicle to be added
OID, Name, Surname, Regnum, Make, Model
Portus
„Added" or „Updated" (or error)
OID, Regnum, Make, Model
Vehicles File
„Added" or „Updated" (or error)
OID, Name, Surname
Personnel (Owner) File

Some notes about the above:

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

Deleting a BusinessDataView Object

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



Some notes about the above:

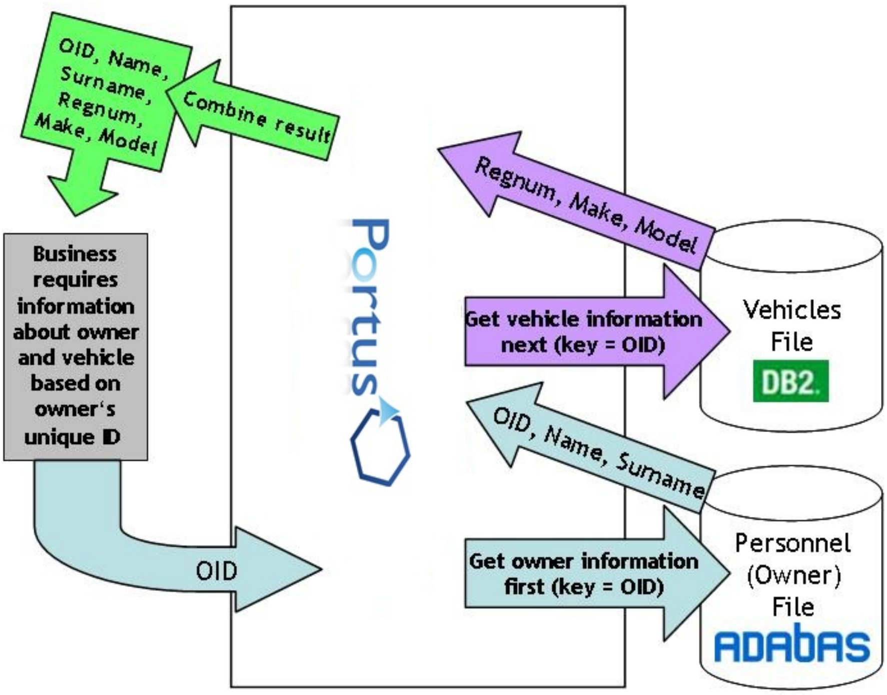
1. Either both (all) deletes from the physical database work or neither do using a transactional semantic and thus the database remains consistent.
2. All data is supplied in one request, this avoids the need for two round trips between the client and the server to delete the objects (physical DataViews).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

Additional features

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

Data in Multiple Databases

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

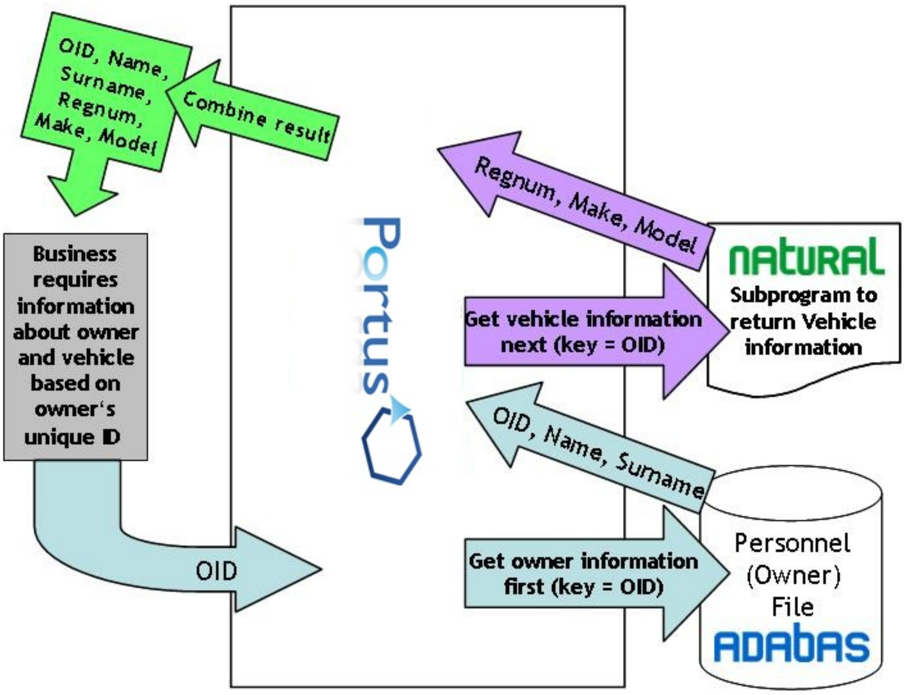

OID, Name, Surname, Regnum, Make, Model
Combine result
Regnum, Make, Model
Business requires information about owner and vehicle based on owner's unique ID
Portus
Get vehicle information next (key = OID)
Vehicles File
DB2
OID, Name, Surname
OID
Get owner information first (key = OID)
Personnel (Owner) File
ADABAS

Some notes about the above:

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

BusinessDataViews Including Business Logic

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


OID, Name, Surname, Regnum, Make, Model
Combine result
Regnum, Make, Model
Business requires information about owner and vehicle based on owner's unique ID
Portus
Get vehicle information next (key = OID)
NATURAL
Subprogram to return Vehicle information
OID, Name, Surname
OID
Get owner information first (key = OID)
Personnel (Owner) File
ADABAS

Some notes about the above:

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

Summary

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

- Avoid the need for new projects to understand legacy data models. Enable them to work with the business model which is more natural for them
- Create Business Data View models from multiple sources including different databases and business logic as required.
- Ensure consistency in your data by exposing services that prevent applications creating inconsistencies in the data.
- Create abstract business services that may be used again and again from any technologies including Java, c#.net, vb.net, php, Ruby, Excel, Word, Internet Explorer and so on.
- Secure your assets based on the business view of the data and not the physical view of the data which is where the security is really required.
- Using the Portus messaging features with MQ or EntireX, enable batch updates of your business objects that can run as part of your batch processing.

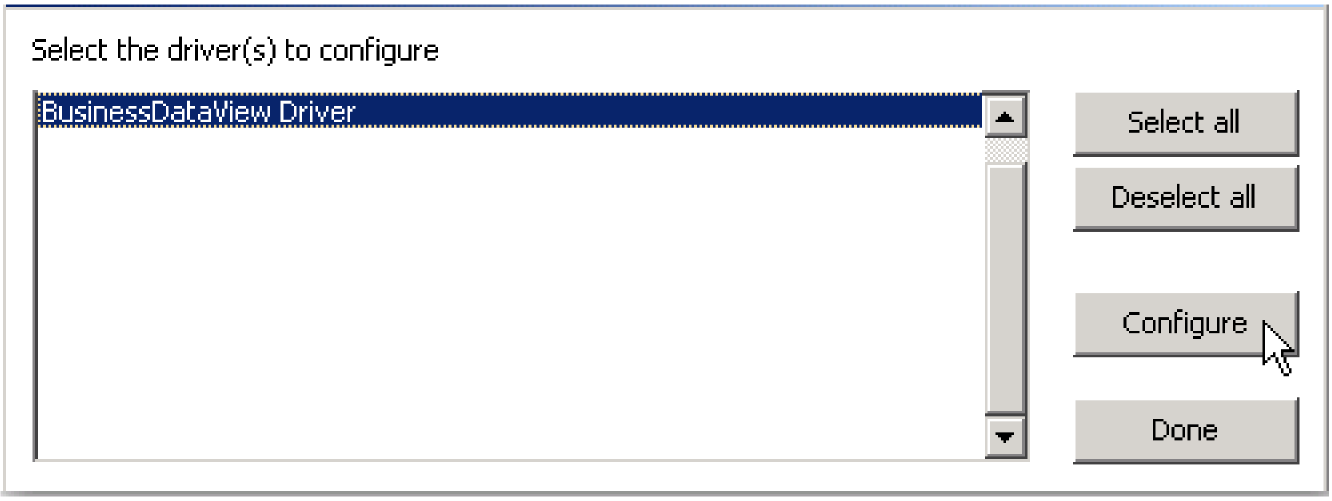
Creating BusinessDataViews

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

- Right-click on the server and select Add driver to server
- Select the BusinessDataView Driver and click Configure.



Select the driver(s) to configure

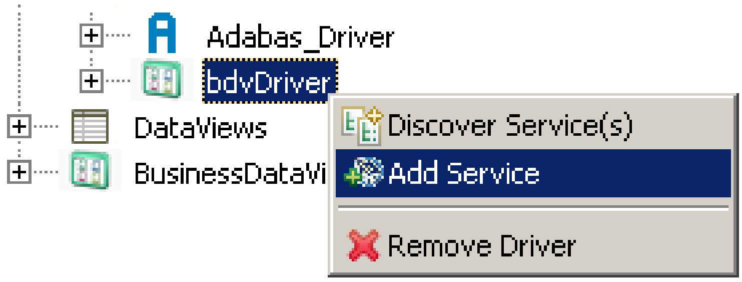
BusinessDataView Driver

Example 1

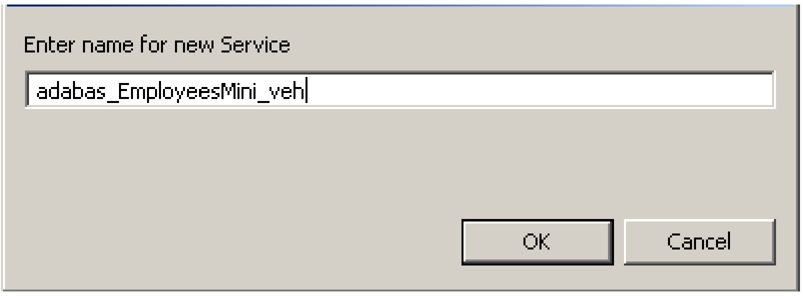
The Adabas demo file "Employees" is keyed on personnel number and the "Vehicles" file also has a personnel number which determines which car is owned by which employee. We wish to define a BusinessDataView which gives us personnel id and surname of the employee and, if present, one instance of their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

Creating the BusinessDataView

- Right-click on bdvDriver under Services and select 'Add Service'

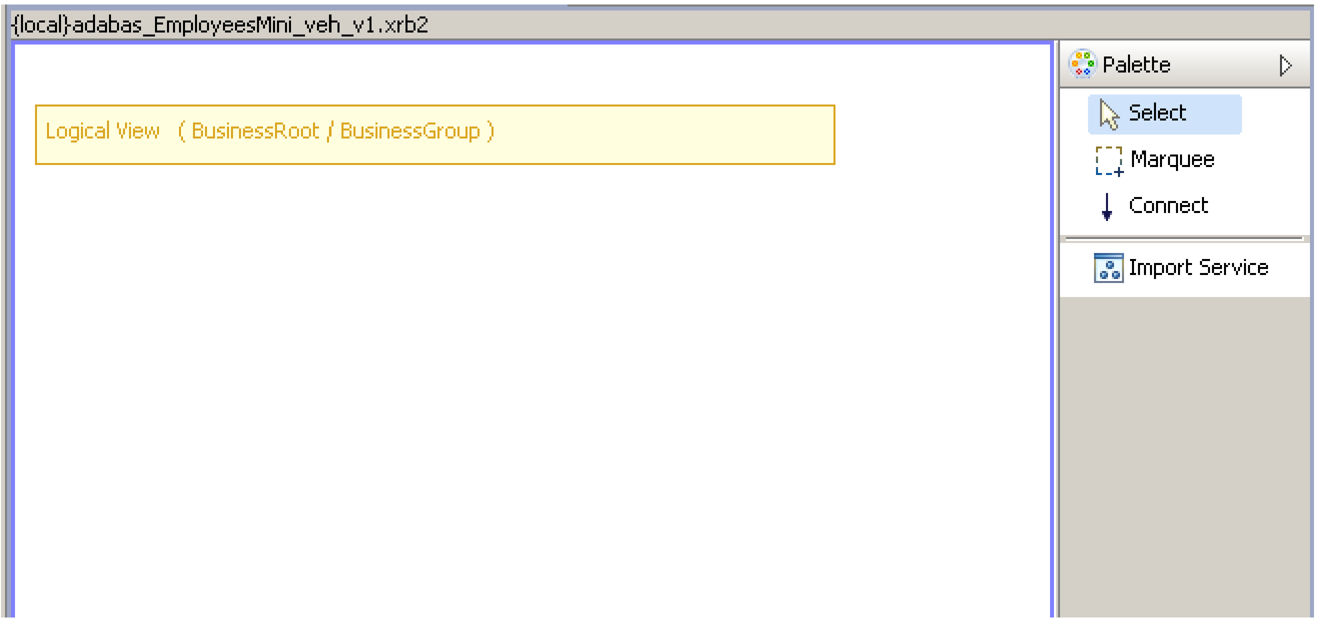


- Name it "adabas_EmployeesMini_veh", click OK

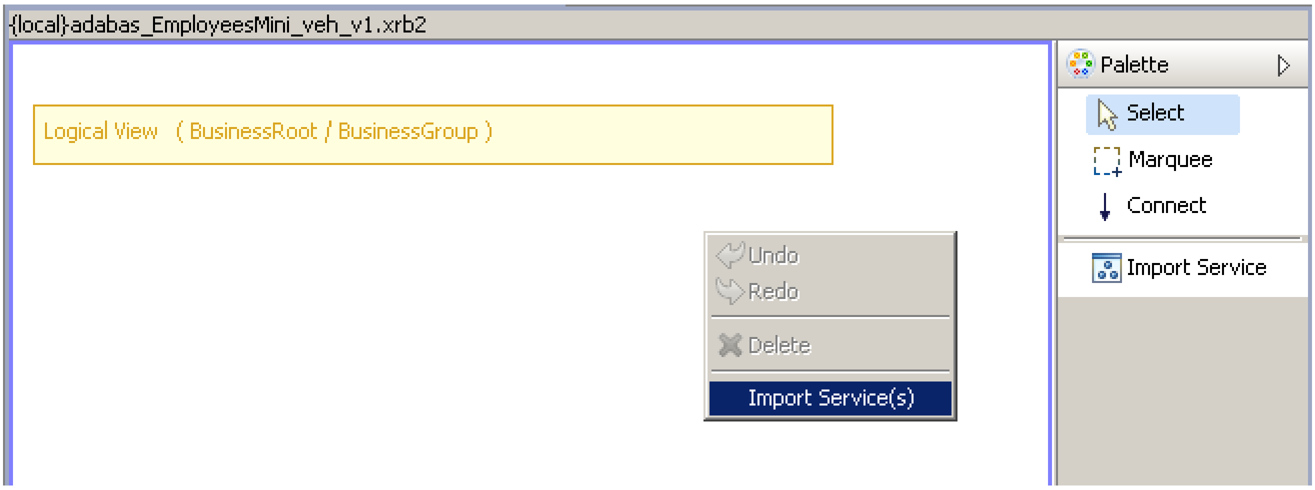

Enter name for new Service

adabas_EmployeesMini_veh

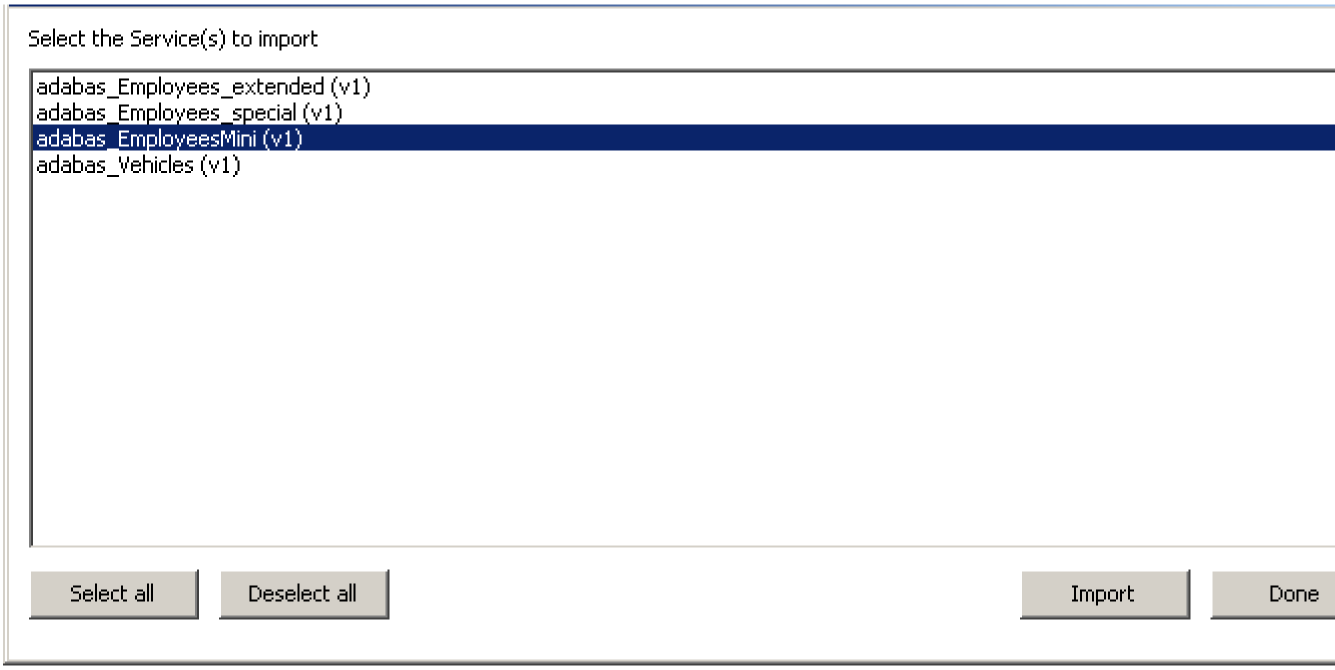
- This will open the BusinessDataView Editor with an initially empty "Logical View" created


{local}adabas_EmployeesMini_veh_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
Palette
Select
Marquee
Connect
Import Service

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

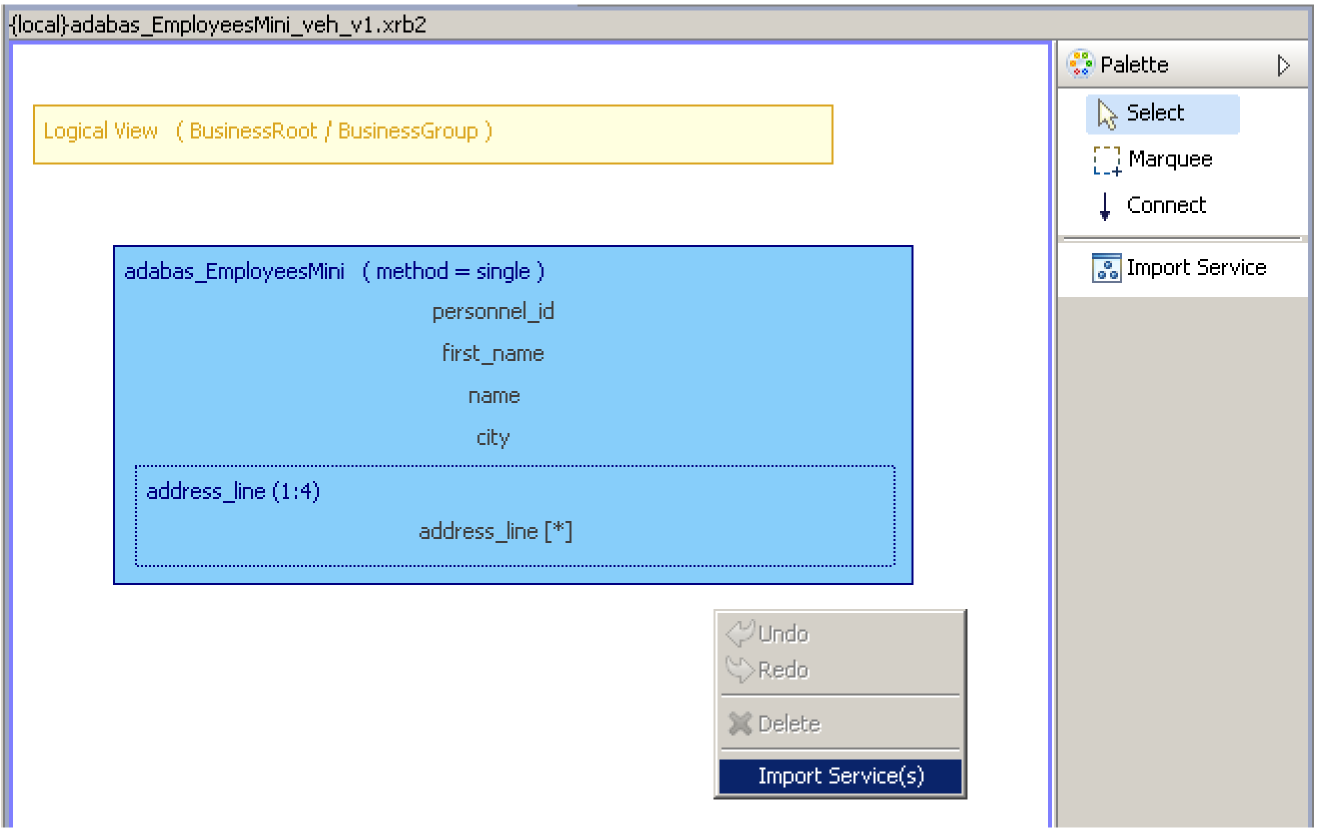

{local}adabas_EmployeesMini_veh_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
Undo
Redo
Delete
Import Service(s)
Palette
Select
Marquee
Connect
Import Service

- Select the adabas_EmployeesMini WebService service and click Import.

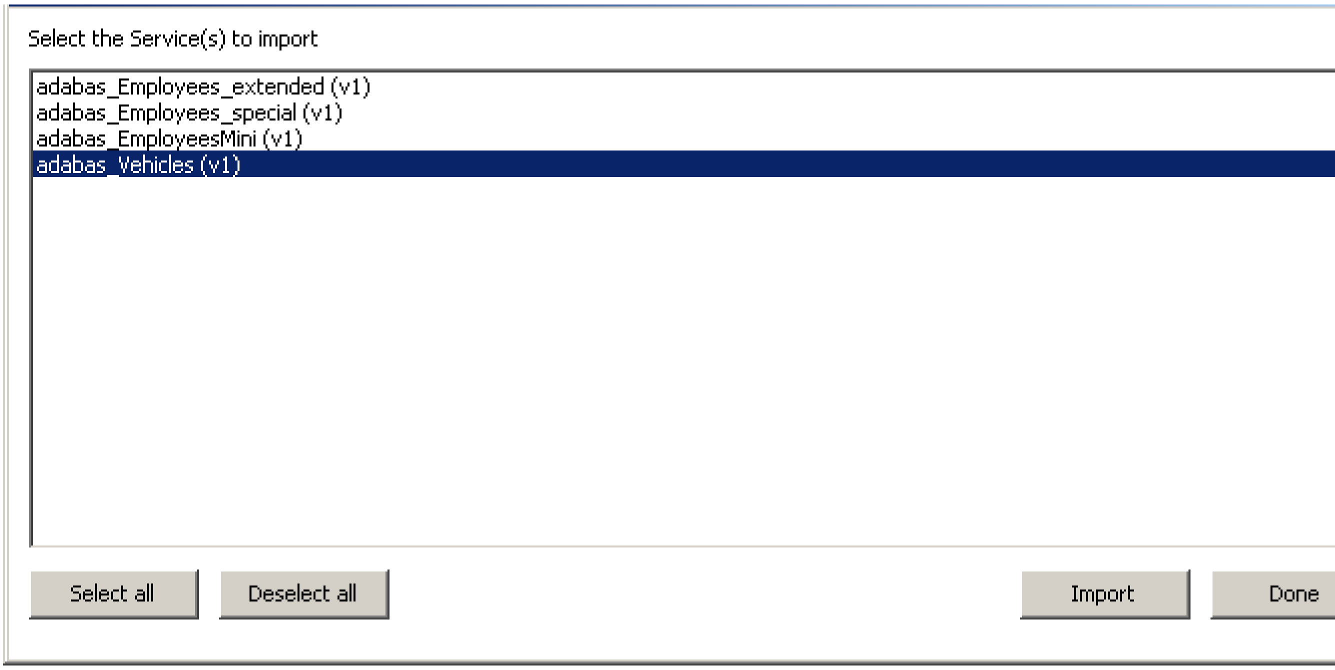

Select the Service(s) to import

adabas_Employees_extended (v1)
adabas_Employees_special (v1)
adabas_EmployeesMini (v1)
adabas_Vehicles (v1)

- 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

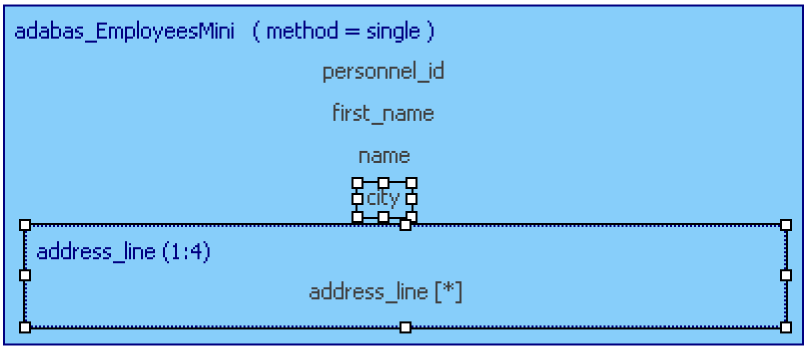


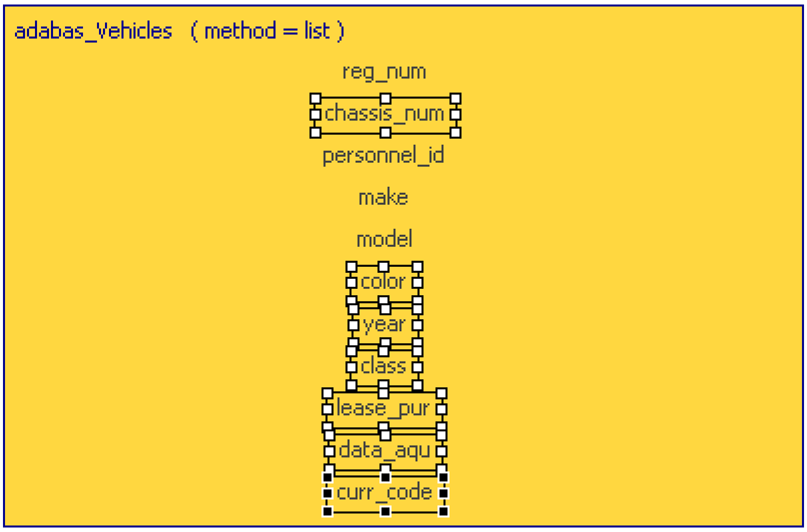
- Select the adabas_Vehicles WebService service and click Import.


Select the Service(s) to import
adabas_Employees_extended (v1)
adabas_Employees_special (v1)
adabas_EmployeesMini (v1)
adabas_Vehicles (v1)
Select all
Deselect all
Import
Done

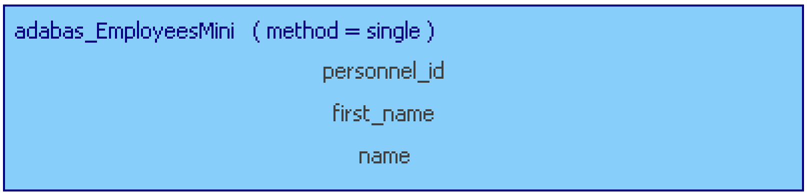
- Select the "Physical View" items not required for the time being (shown selected below) and hit the <DEL> key to remove them. There is no requirement whatsoever to use all of the "base WebService" fields within the BusinessDataView's "physical resource(s)". Note that there is also no requirement to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

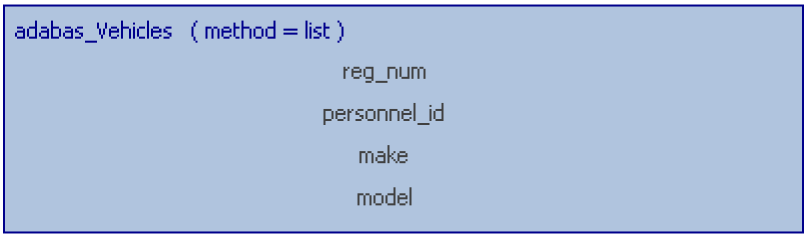
Note: Select the city item first and, holding down the Ctrl key, select the other items.



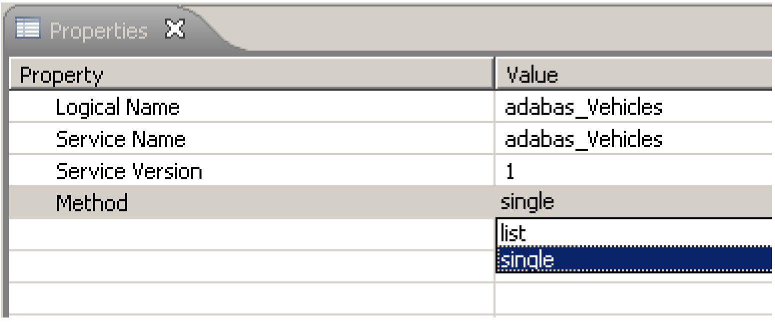


- This will leave you with the following items. For this example we are only interested in one vehicle per employee so we need to change the method from list to single. Our next example will demonstrate the use of the list method.



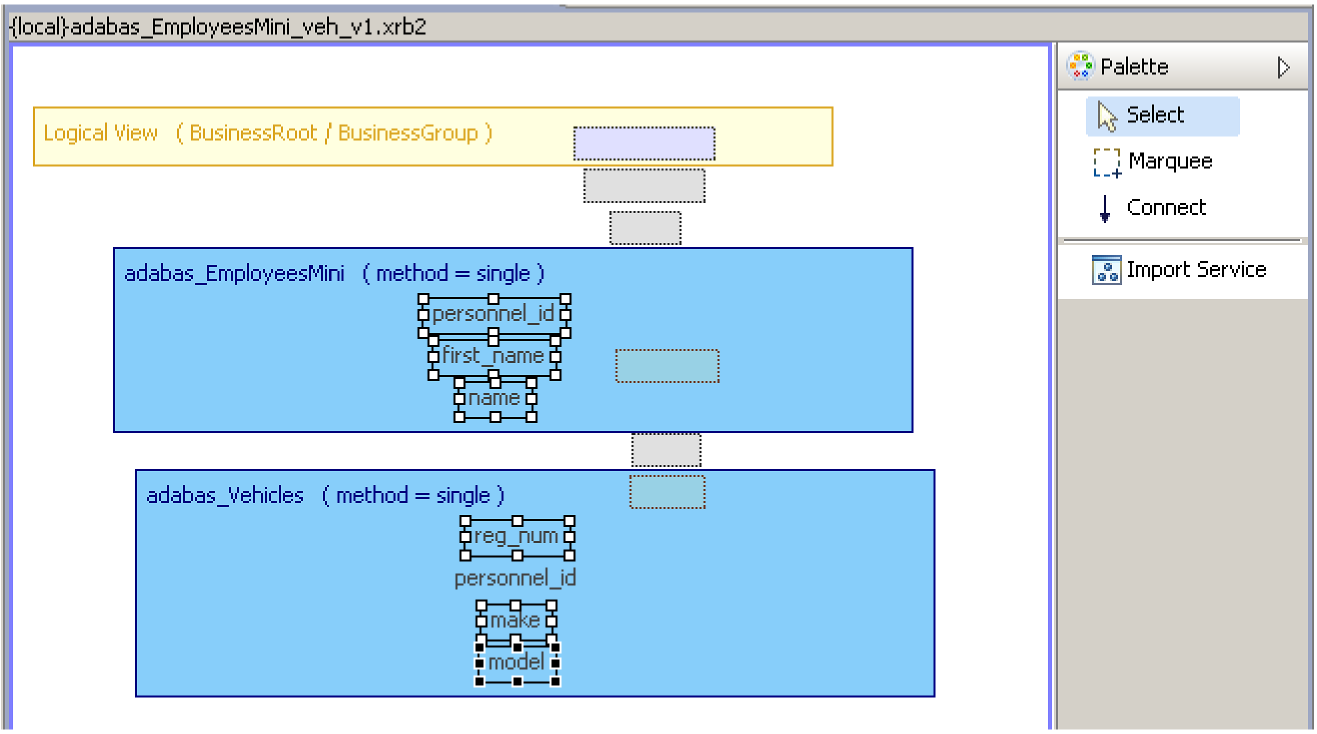


- Click on an empty area within the adabas_Vehicles physical view. The Properties view should display the properties for this Physical Resource. If the Properties view is not visible, select menu item Window -> Show View -> Properties. Change the Method to single as shown. Ctrl+S to save.



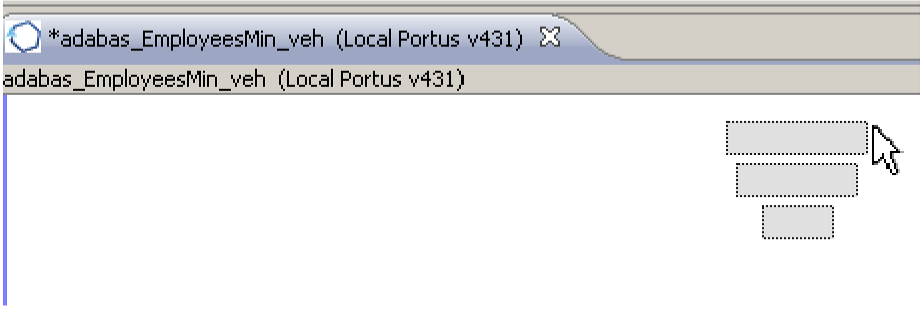
Property	Value
Logical Name	adabas_Vehicles
Service Name	adabas_Vehicles
Service Version	1
Method	single
	list
	single

- 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.



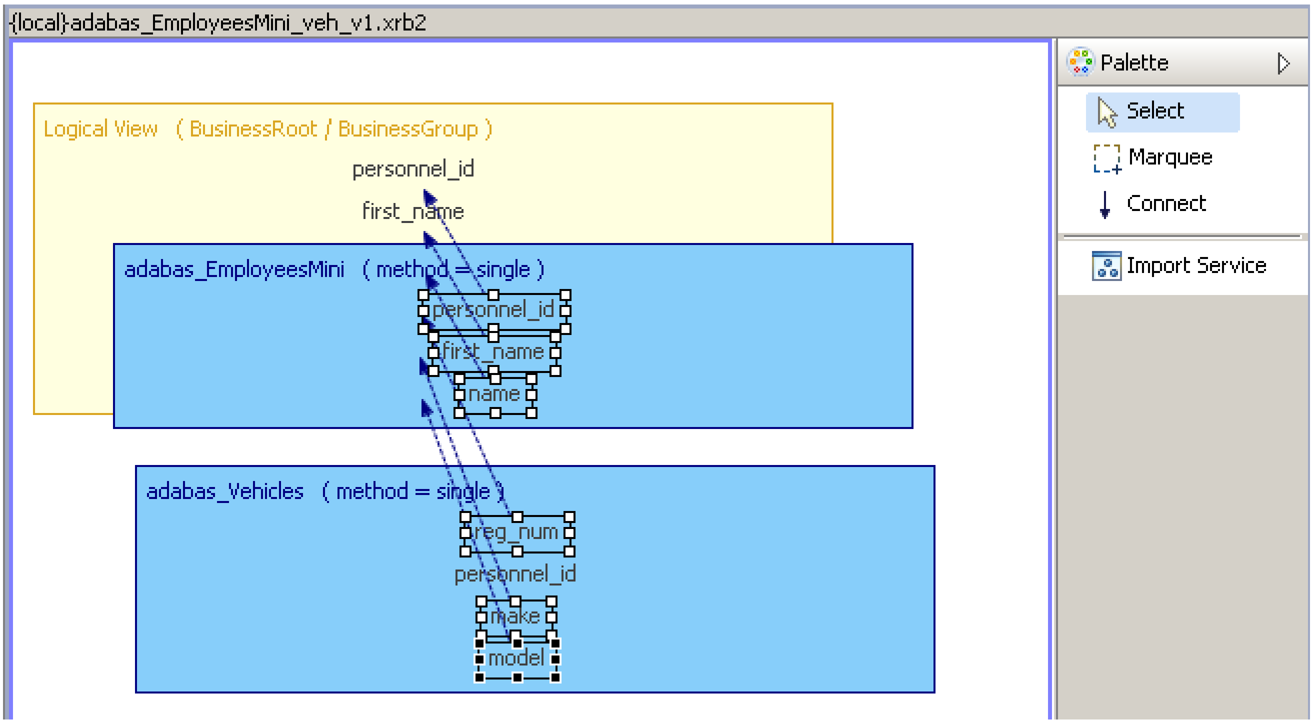
{local}adabas_EmployeesMini_veh_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
adabas_EmployeesMini (method = single)
personnel_id
first_name
name
adabas_Vehicles (method = single)
reg_num
personnel_id
make
model
Palette
Select
Marquee
Connect
Import Service

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

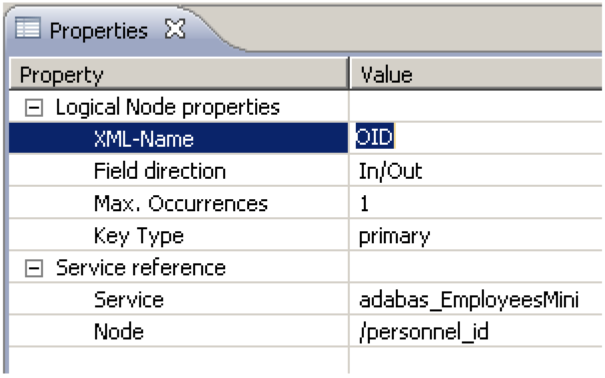


*adabas_EmployeesMin_veh (Local Portus v431)
adabas_EmployeesMin_veh (Local Portus v431)

- 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).

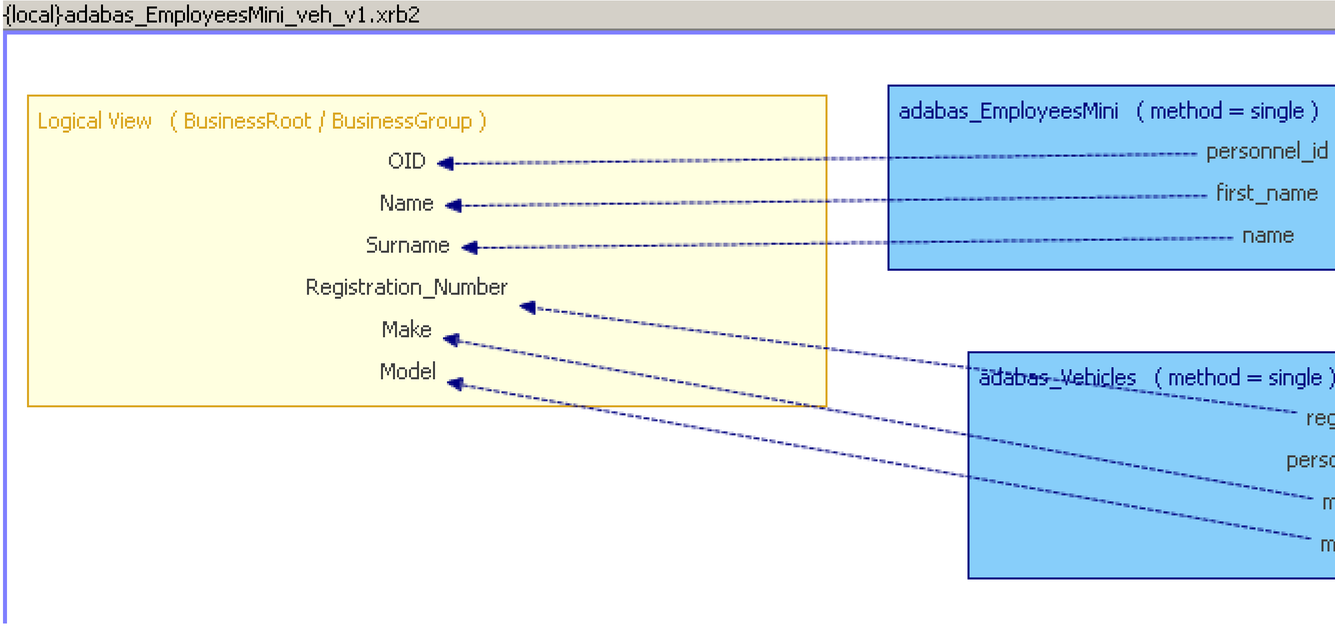

{local}adabas_EmployeesMini_veh_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
personnel_id
first_name
adabas_EmployeesMini (method = single)
personnel_id
first_name
name
adabas_Vehicles (method = single)
reg_num
personnel_id
make
model
Palette
Select
Marquee
Connect
Import Service

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

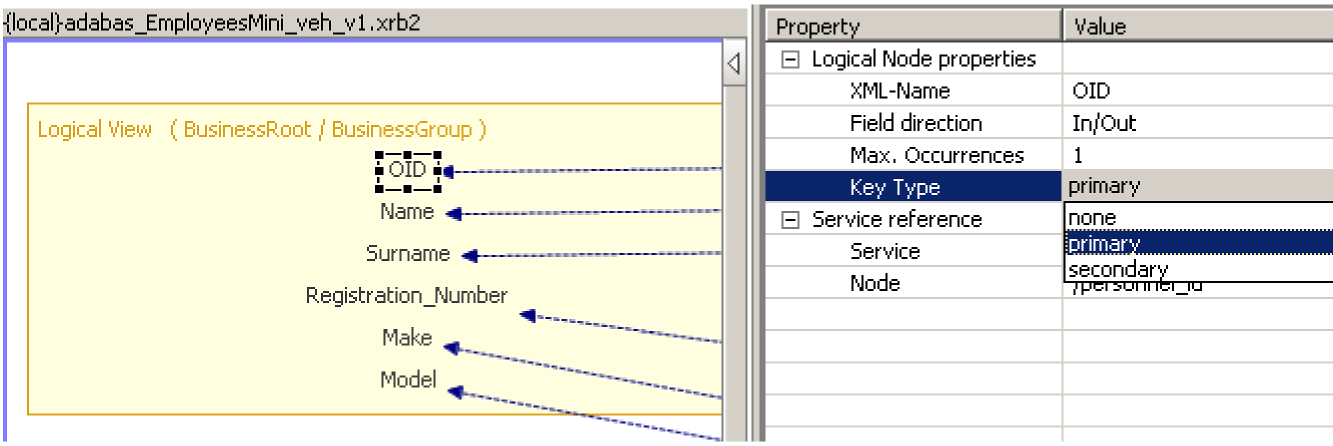


Property	Value
⊟ Logical Node properties	
XML-Name	OID
Field direction	In/Out
Max. Occurrences	1
Key Type	primary
⊟ Service reference	
Service	adabas_EmployeesMini
Node	/personnel_id

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

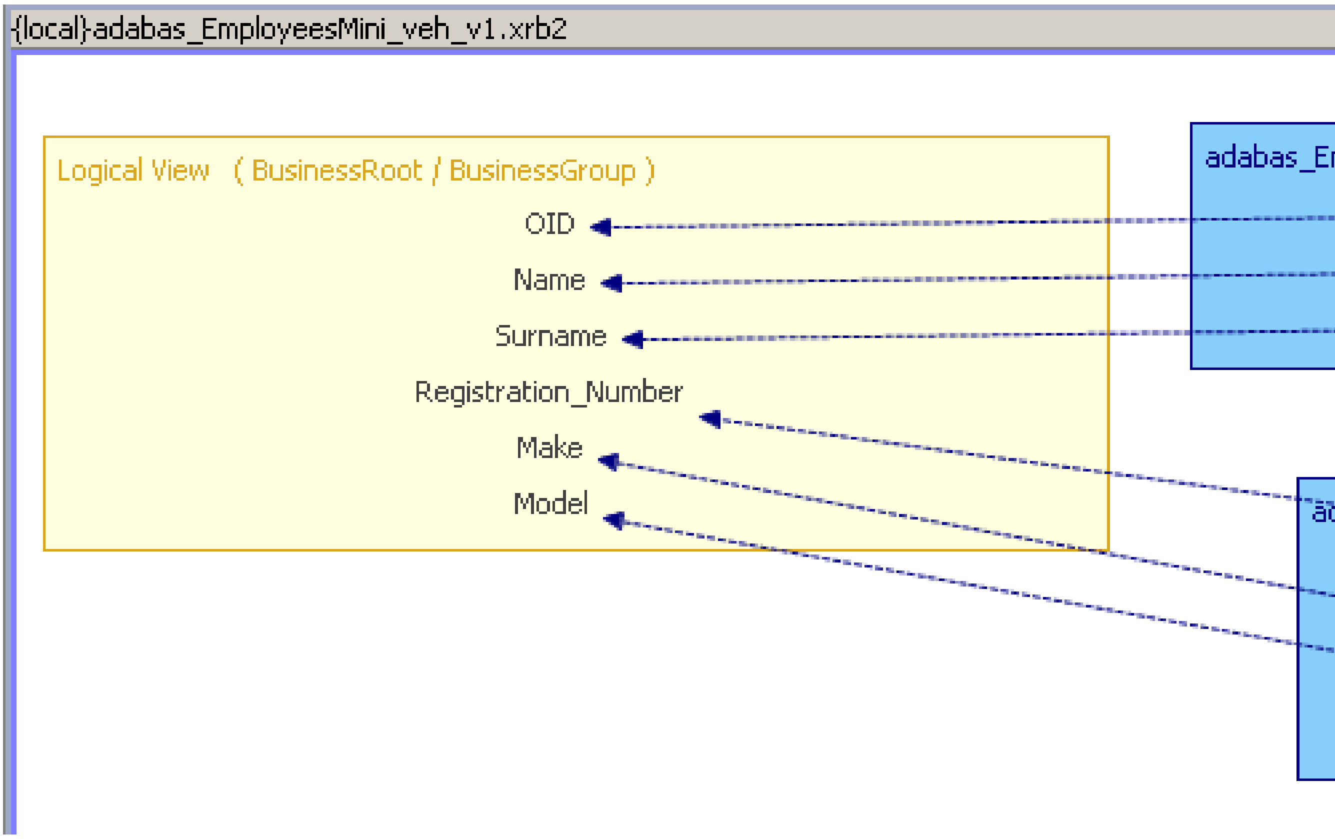

{local}adabas_EmployeesMini_veh_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
OID
Name
Surname
Registration_Number
Make
Model
adabas_EmployeesMini (method = single)
personnel_id
first_name
name
adabas_Vehicles (method = single)
reg
perso
m
m

- 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:

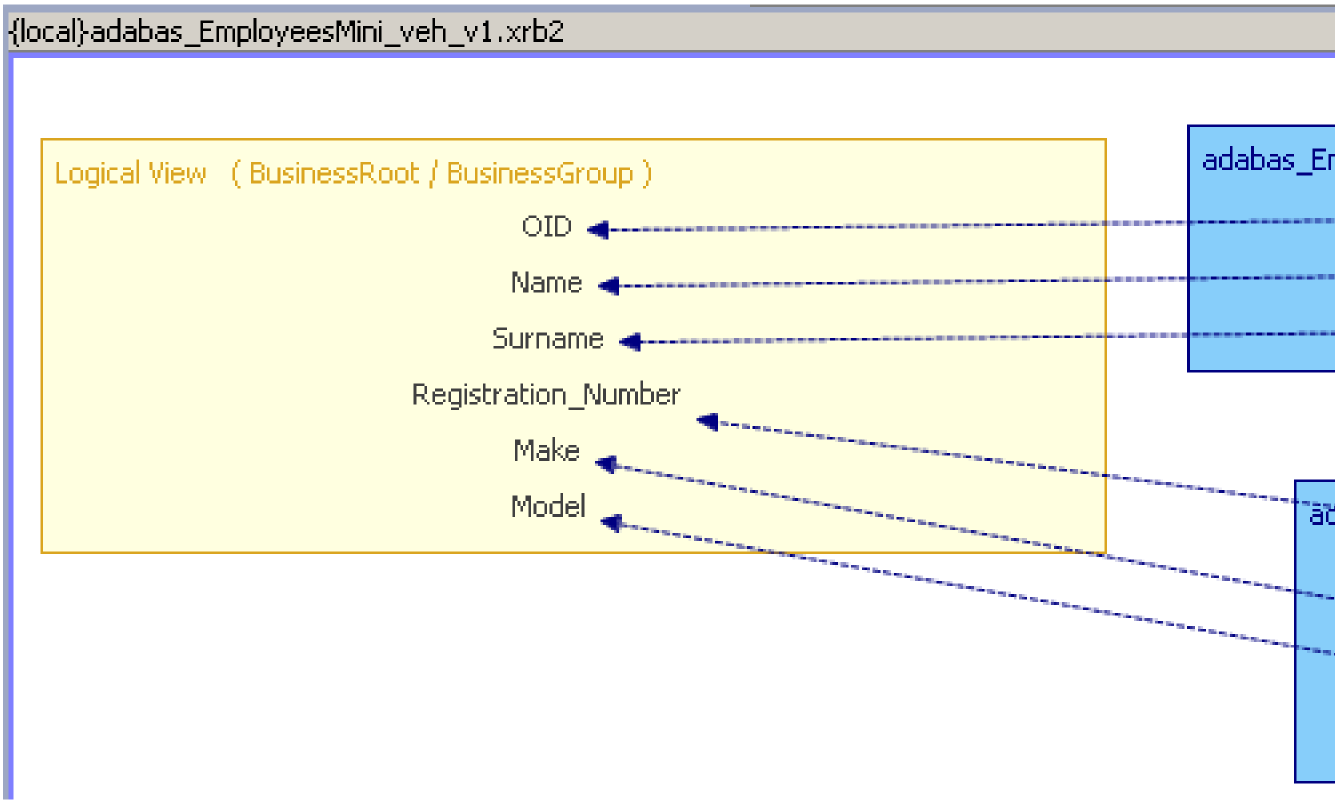

{local}-adabas_EmployeesMini_veh_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
OID
Name
Surname
Registration_Number
Make
Model

Property	Value
⊟ Logical Node properties	
XML-Name	OID
Field direction	In/Out
Max. Occurrences	1
Key Type	primary
⊟ Service reference	none primary secondary
Service	
Node	/personnel_id

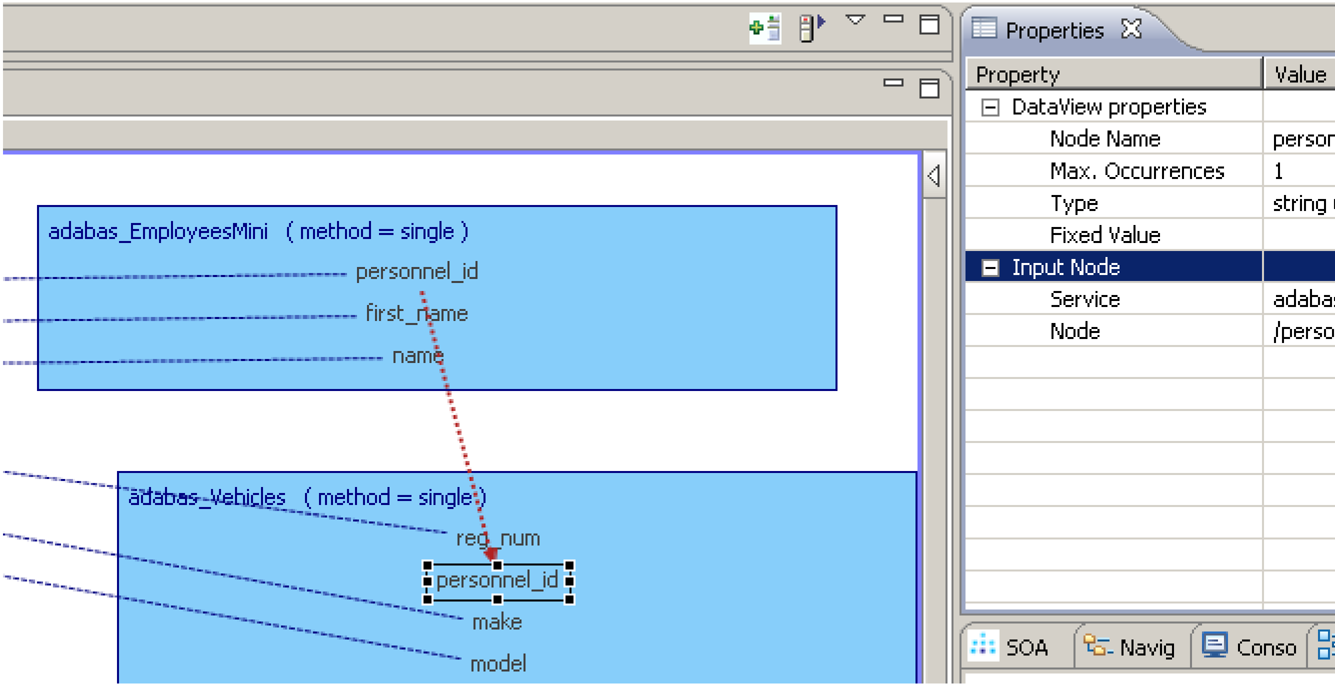
- The next step involves creating a connection or link between the appropriate field of one physical resource to a field in another physical resource. Select Connect item from the Palette. Select the personnel_id field of adabas_EmployeesMini view and move the cursor away. As you can see this marks the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


{local}-adabas_EmployeesMini_veh_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
OID
Name
Surname
Registration_Number
Make
Model
adabas_E
ac

- Now click the personnel_id field within the adabas_Vehicles view.


{local}adabas_EmployeesMini_veh_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
OID
Name
Surname
Registration_Number
Make
Model
adabas_Er
ad

- 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.

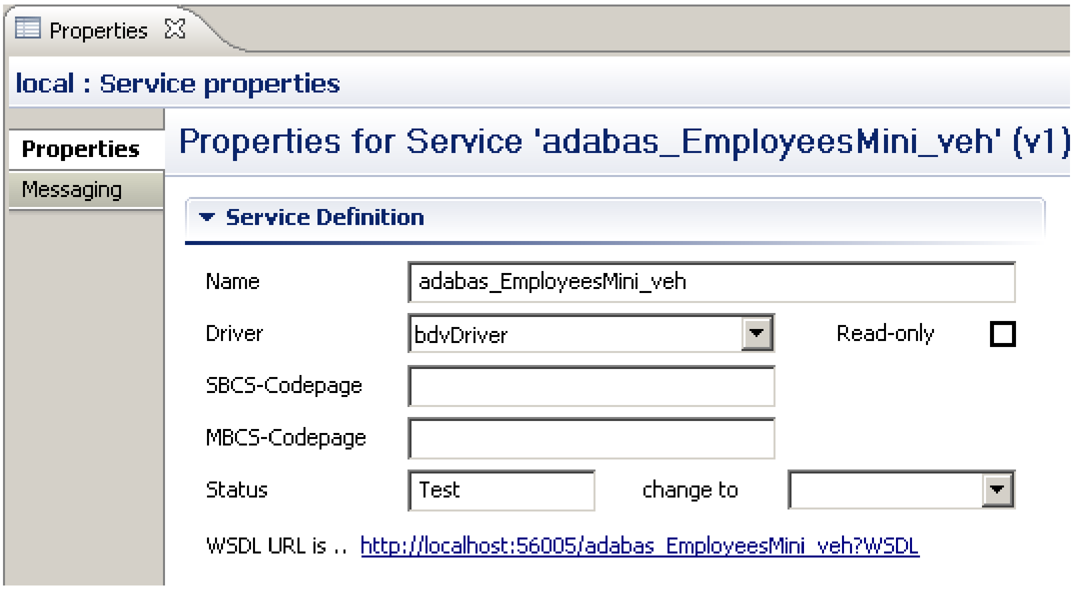


Property	Value
⊟ DataView properties	
Node Name	persor
Max. Occurrences	1
Type	string
Fixed Value	
⊟ Input Node	
Service	adaba
Node	/perso

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

Test the WebService

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



local : Service properties

Properties for Service 'adabas_EmployeesMini_veh' (v1)

Name: adabas_EmployeesMini_veh

Driver: bdvDriver

Read-only □

SBCS-Codepage:

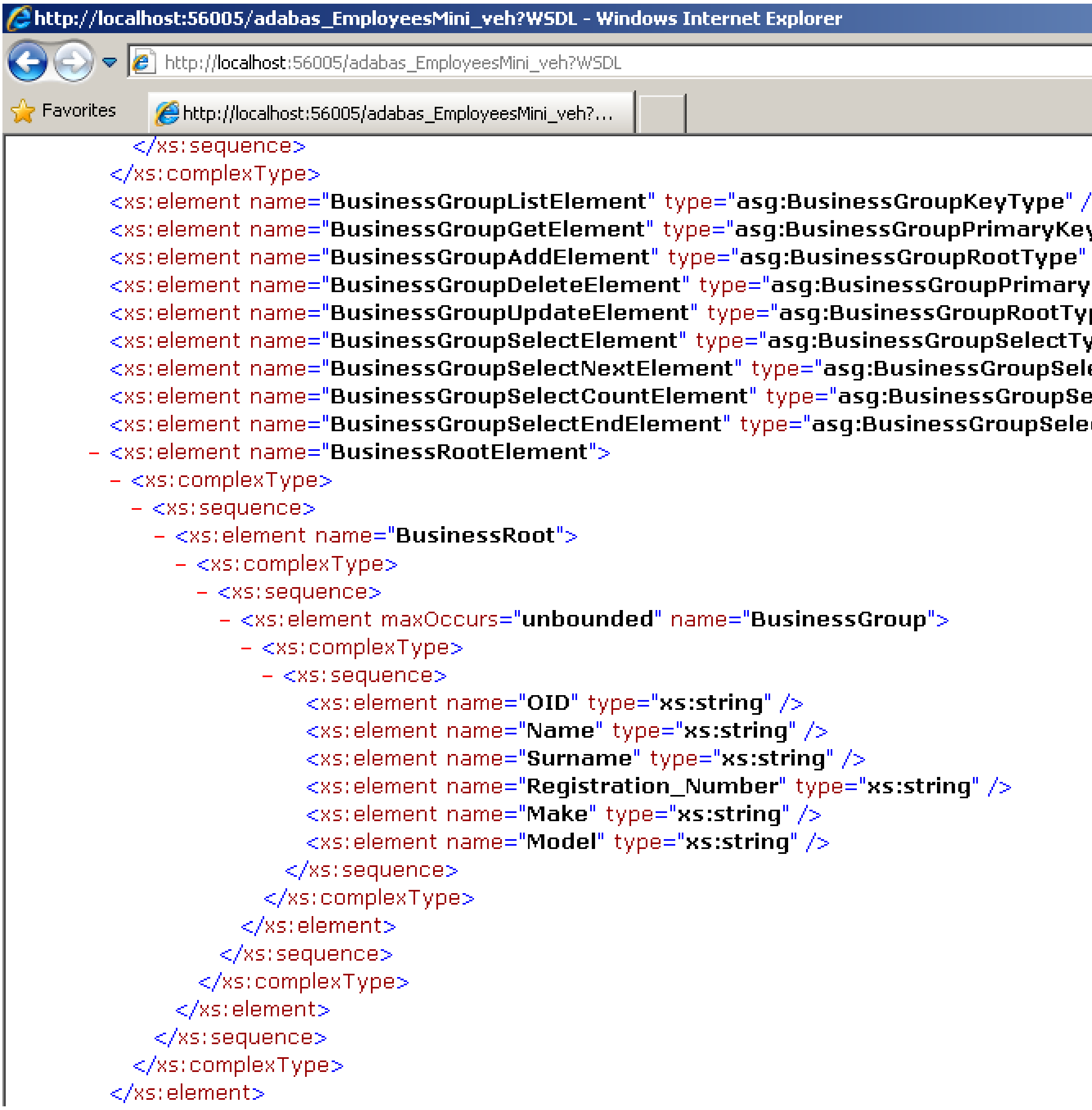
MBCS-Codepage:

Status: Test

change to:

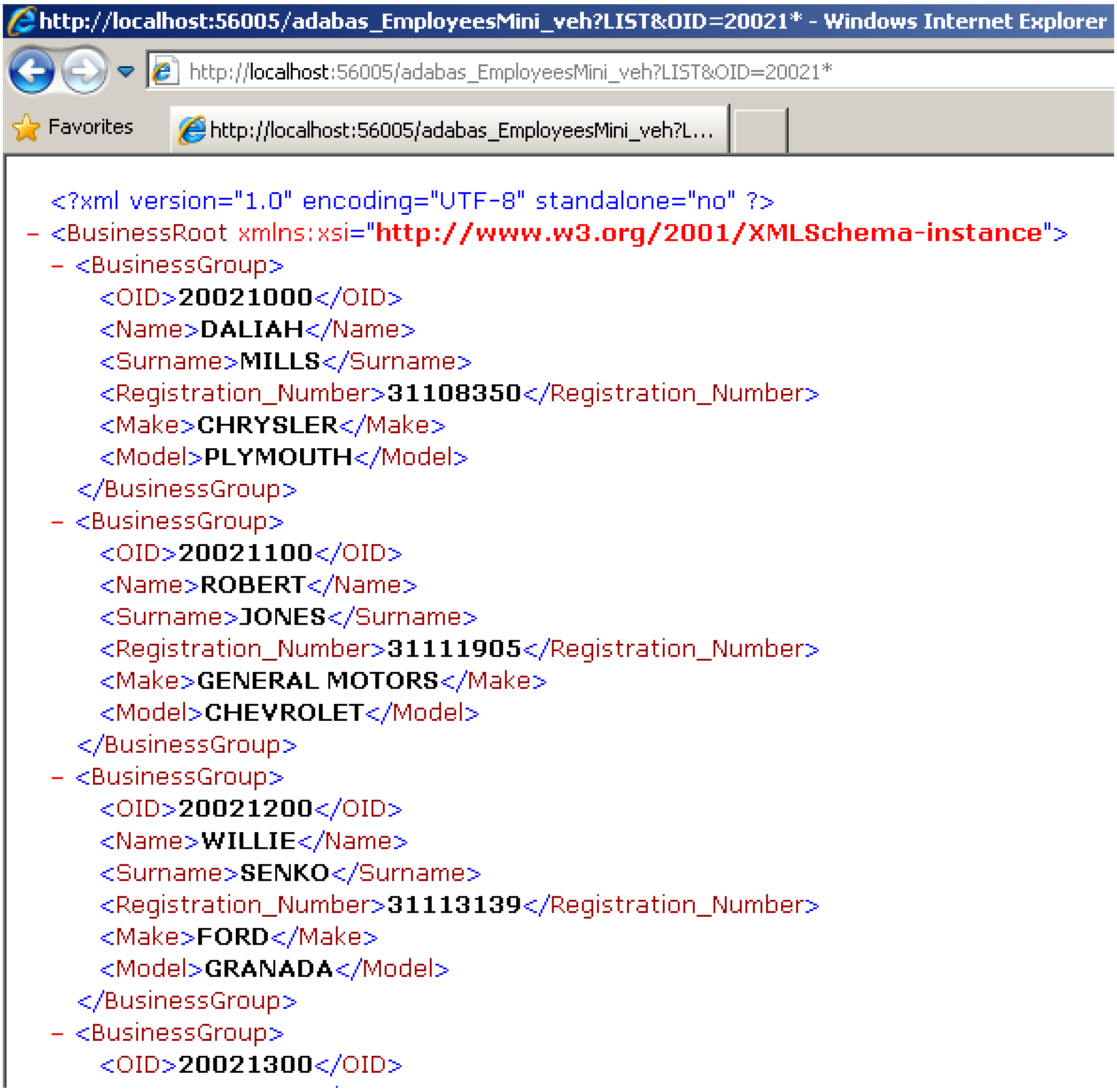
WSDL URL is .. http://localhost:56005/adabas_EmployeesMini_veh?WSDL

- 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).


http://localhost:56005/adabas_EmployeesMini_veh?WSDL - Windows Internet Explorer
http://localhost:56005/adabas_EmployeesMini_veh?WSDL
Favorites
http://localhost:56005/adabas_EmployeesMini_veh?...

```
          </xs:sequence>
        </xs:complexType>
        <xs:element name="BusinessGroupListElement" type="asg:BusinessGroupKeyType" /
        <xs:element name="BusinessGroupGetElement" type="asg:BusinessGroupPrimaryKey
        <xs:element name="BusinessGroupAddElement" type="asg:BusinessGroupRootType"
        <xs:element name="BusinessGroupDeleteElement" type="asg:BusinessGroupPrimary
        <xs:element name="BusinessGroupUpdateElement" type="asg:BusinessGroupRootTyp
        <xs:element name="BusinessGroupSelectElement" type="asg:BusinessGroupSelectTy
        <xs:element name="BusinessGroupSelectNextElement" type="asg:BusinessGroupSele
        <xs:element name="BusinessGroupSelectCountElement" type="asg:BusinessGroupSe
        <xs:element name="BusinessGroupSelectEndElement" type="asg:BusinessGroupSele
      - <xs:element name="BusinessRootElement">
        - <xs:complexType>
          - <xs:sequence>
            - <xs:element name="BusinessRoot">
              - <xs:complexType>
                - <xs:sequence>
                  - <xs:element maxOccurs="unbounded" name="BusinessGroup">
                    - <xs:complexType>
                      - <xs:sequence>
                          <xs:element name="OID" type="xs:string" />
                          <xs:element name="Name" type="xs:string" />
                          <xs:element name="Surname" type="xs:string" />
                          <xs:element name="Registration_Number" type="xs:string" />
                          <xs:element name="Make" type="xs:string" />
                          <xs:element name="Model" type="xs:string" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
```

- 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.


http://localhost:56005/adabas_EmployeesMini_veh?LIST&OID=20021* - Windows Internet Explorer
http://localhost:56005/adabas_EmployeesMini_veh?LIST&OID=20021*
Favorites
http://localhost:56005/adabas_EmployeesMini_veh?L...

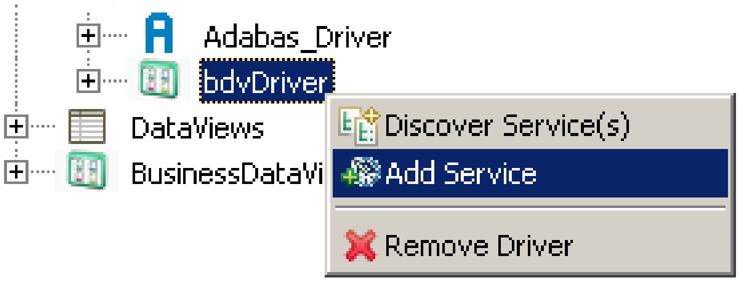
```
  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <BusinessRoot xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <BusinessGroup>
      <OID>20021000</OID>
      <Name>DALIAH</Name>
      <Surname>MILLS</Surname>
      <Registration_Number>31108350</Registration_Number>
      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

Example 2

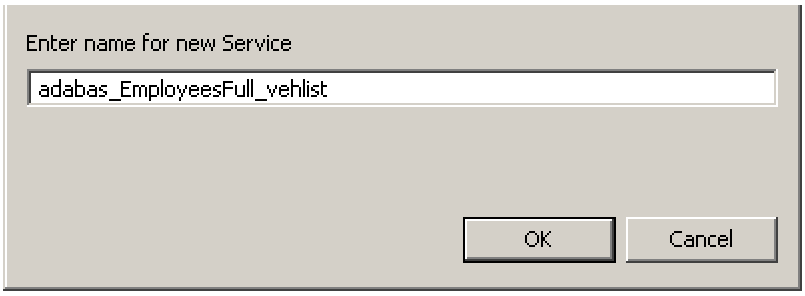
We now define a BusinessDataView which provides more employee details plus a list of vehicle(s) for the employee..

Create the BusinessDataView

- Right-click on bdvDriver under Services and select 'Add Service'



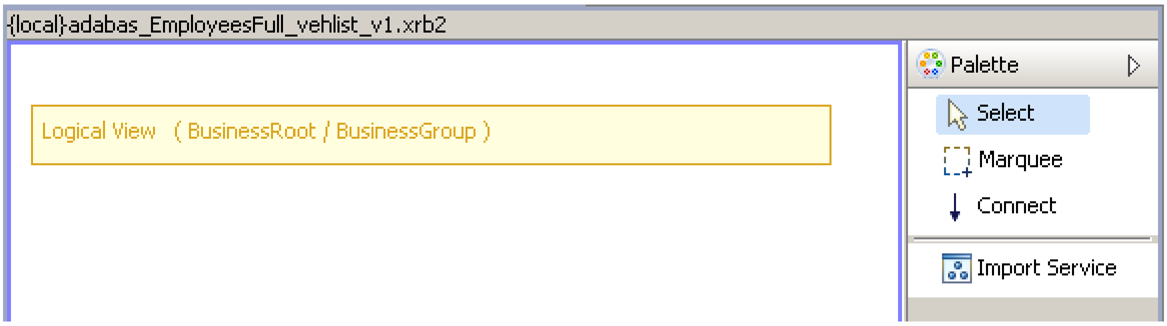
- Name it adabas_EmployeesFull_vehlist and click OK



Enter name for new Service

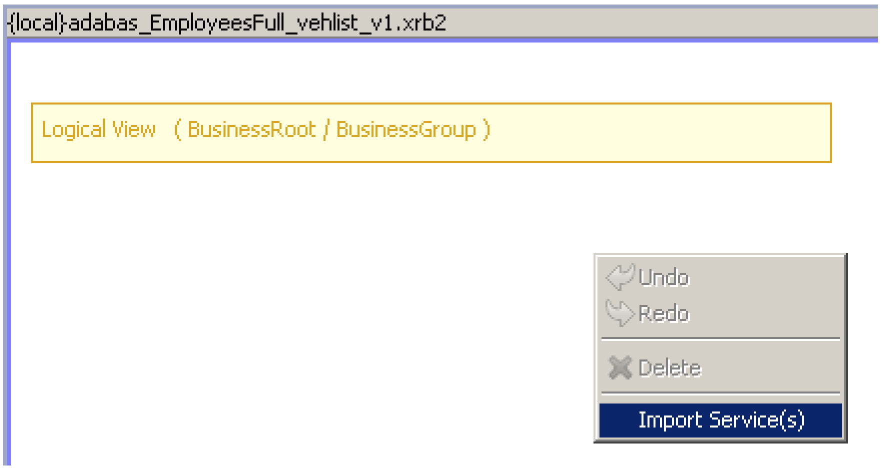
adabas_EmployeesFull_vehlist

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



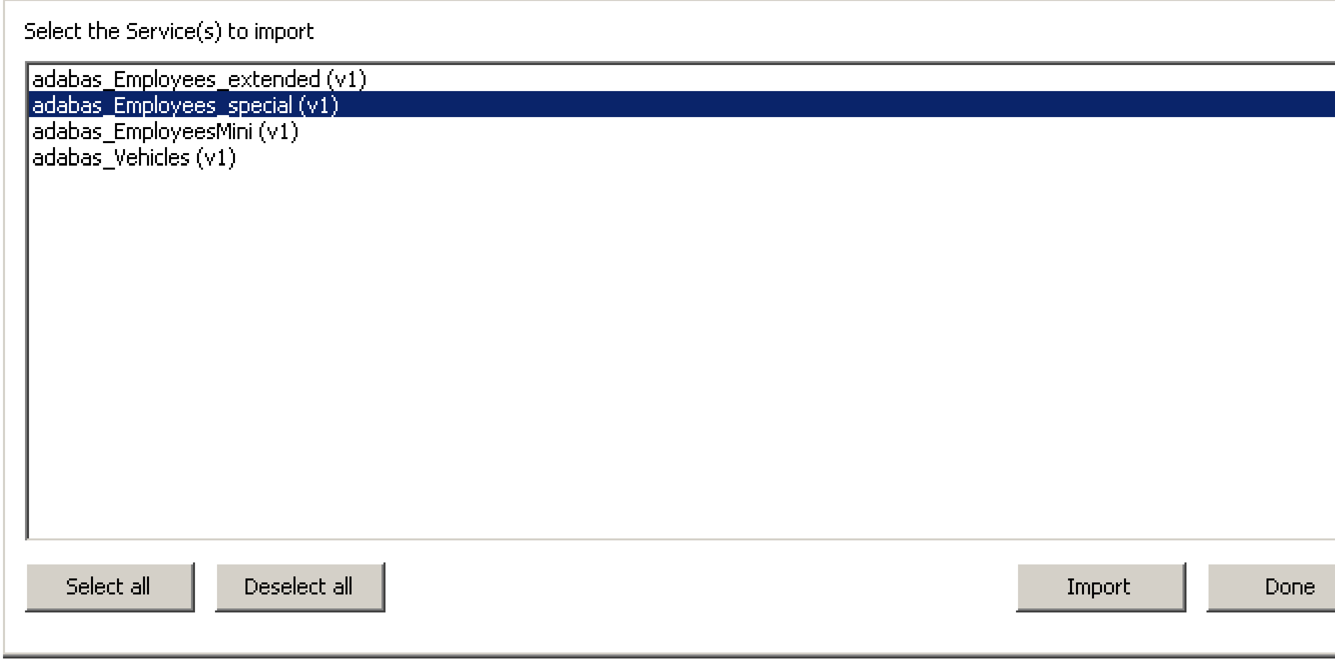
{local}adabas_EmployeesFull_vehlist_v1.xrb2

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



{local}adabas_EmployeesFull_vehlist_v1.xrb2

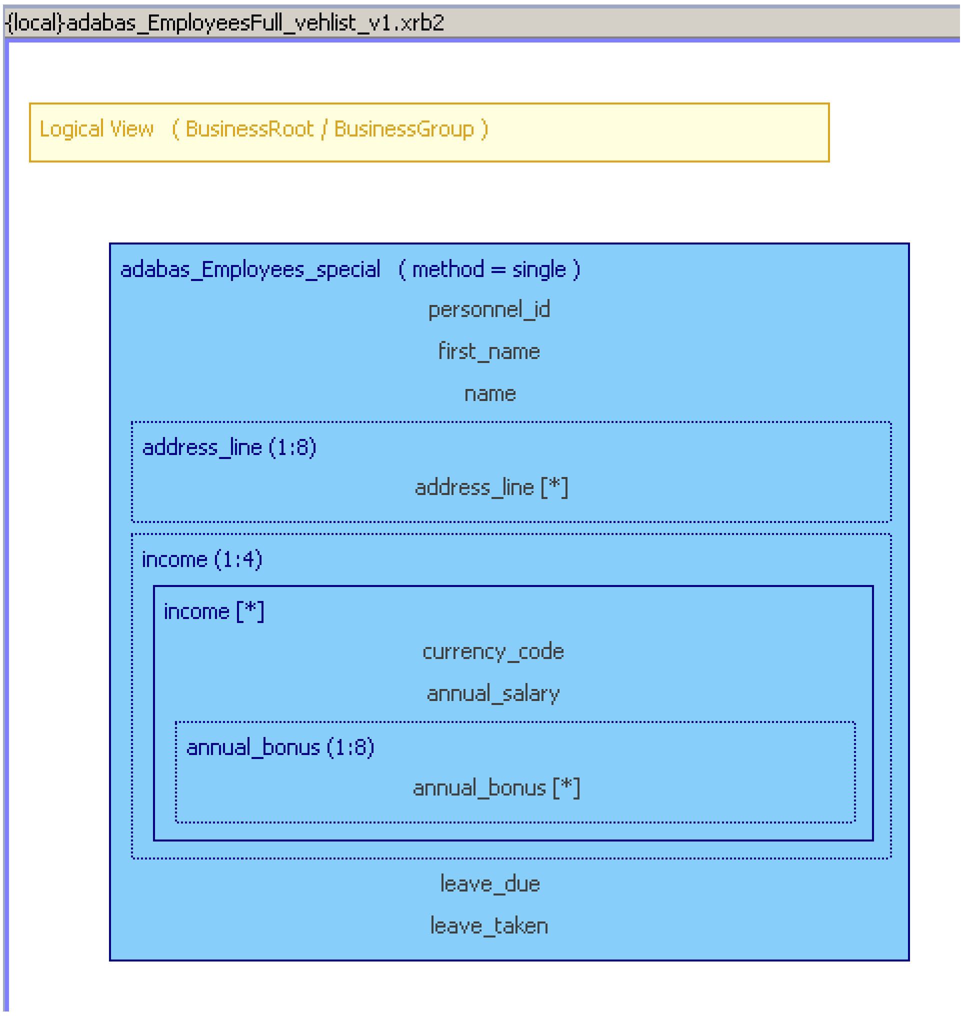
- Select the adabas_Employees_special service, click Import



Select the Service(s) to import

adabas_Employees_extended (v1)
adabas_Employees_special (v1)
adabas_EmployeesMini (v1)
adabas_Vehicles (v1)

- 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:

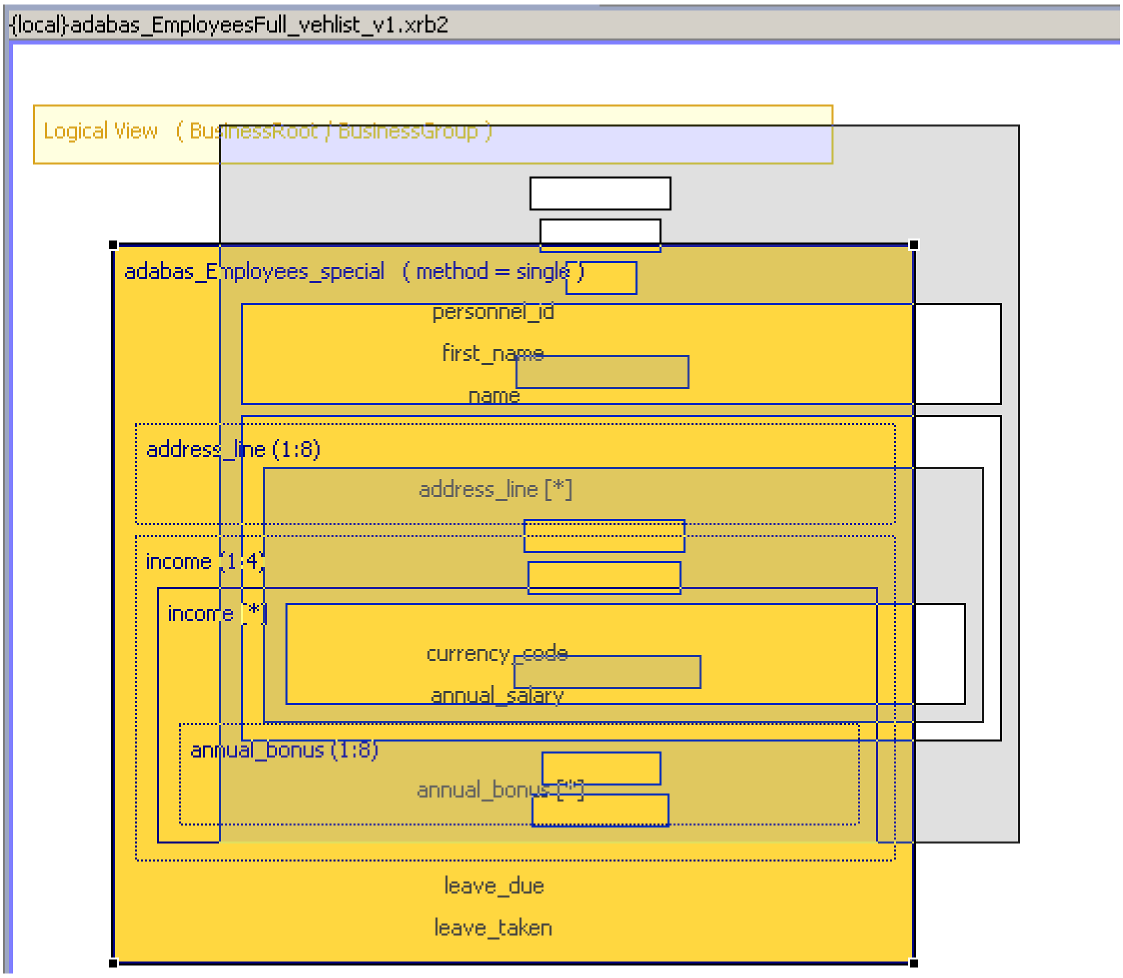


{local}adabas_EmployeesFull_vehlist_v1.xrb2

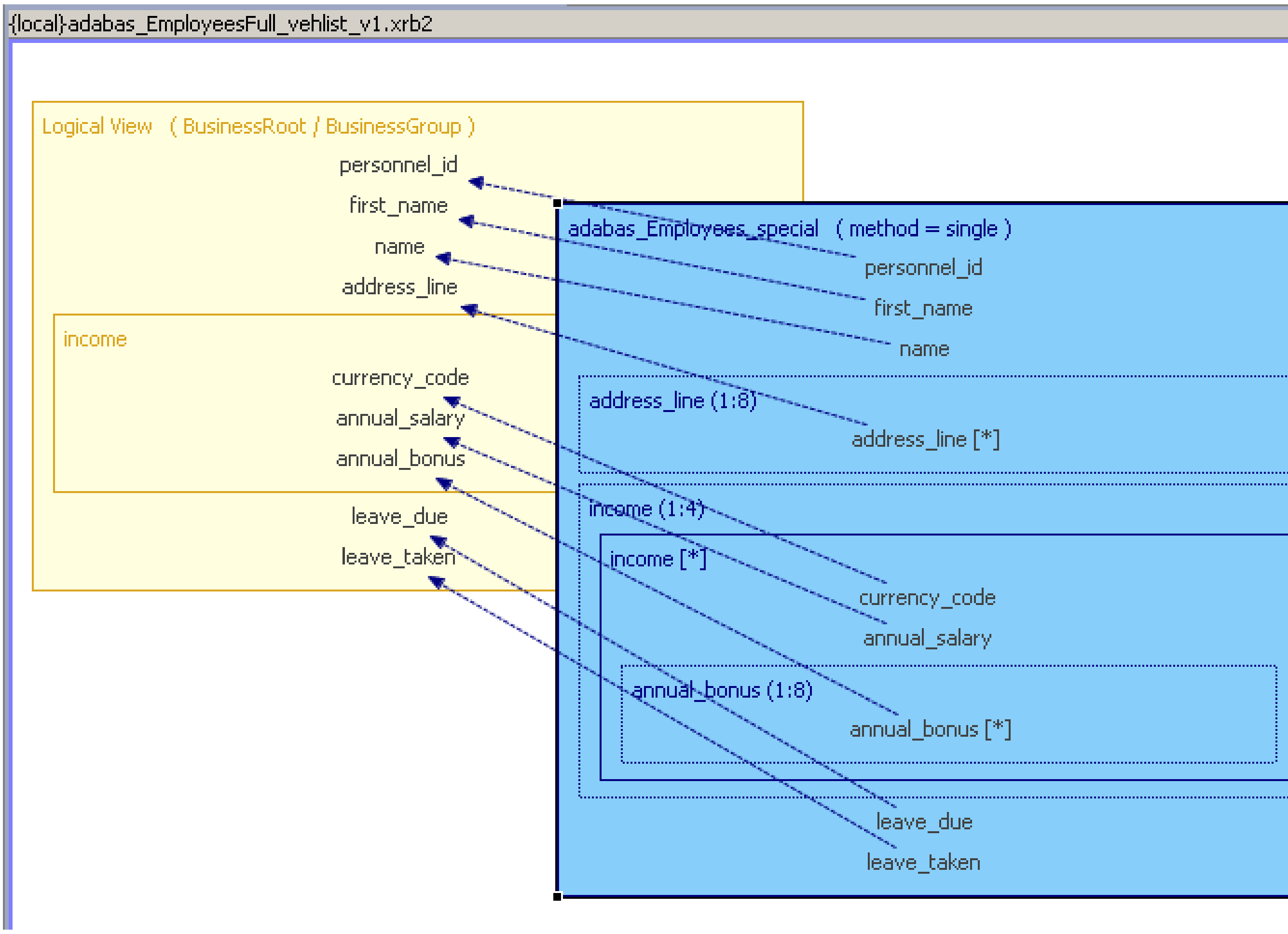
Logical View (BusinessRoot / BusinessGroup)

adabas_Employees_special (method = single)
personnel_id
first_name
name
address_line (1:8)
address_line [*]
income (1:4)
income [*]
currency_code
annual_salary
annual_bonus (1:8)
annual_bonus [*]
leave_due
leave_taken

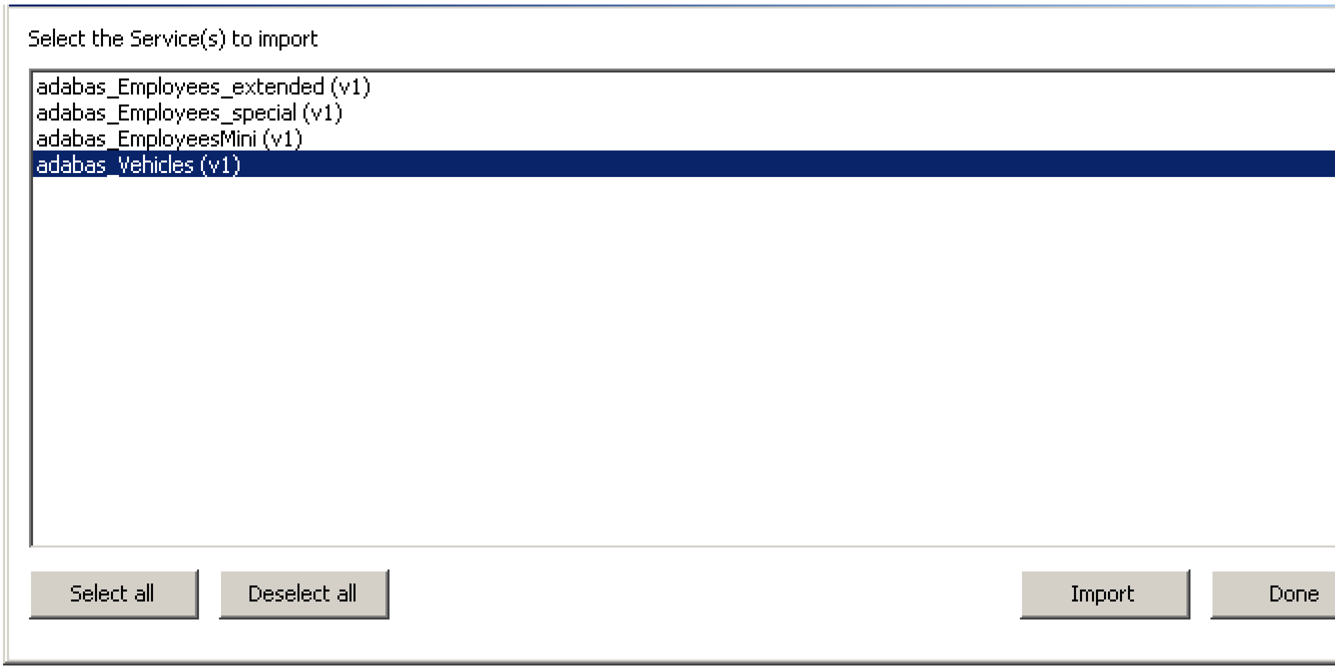
- It is good practice to save frequently (Ctrl+S, select the Save icon, File->Save). Select the entire adabas_Employees_special view, drag it to the LV and drop it there.



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

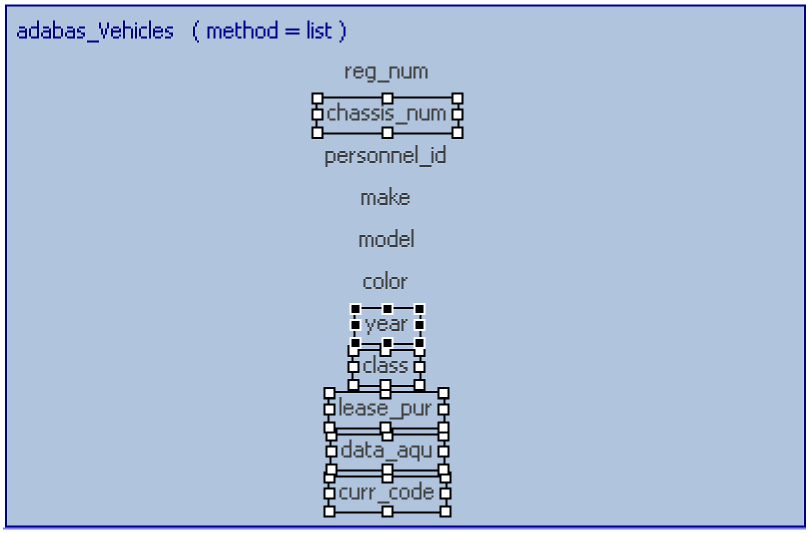


- Import the adabas_Vehicles WebService.

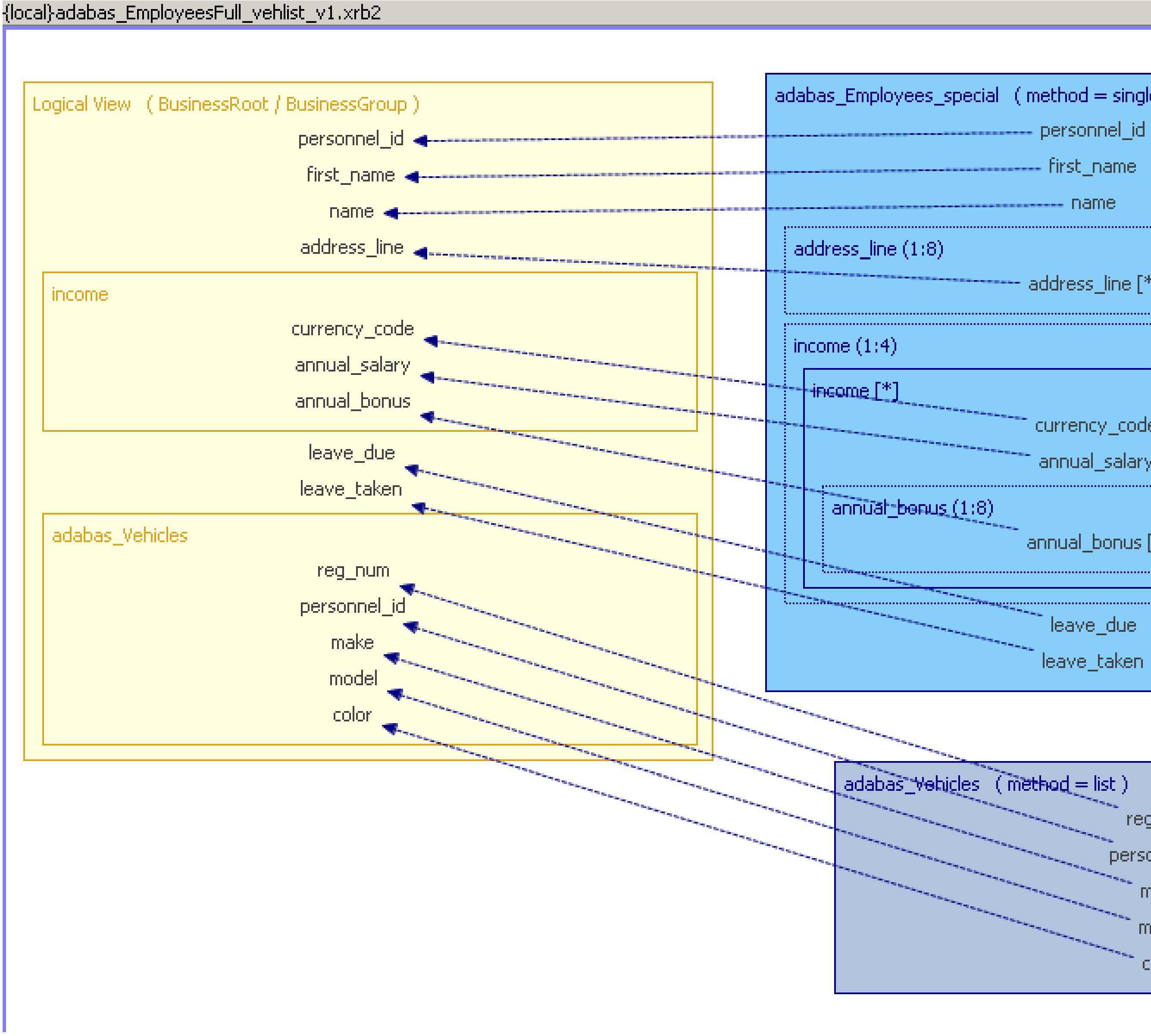

Select the Service(s) to import

adabas_Employees_extended (v1)
adabas_Employees_special (v1)
adabas_EmployeesMini (v1)
adabas_Vehicles (v1)

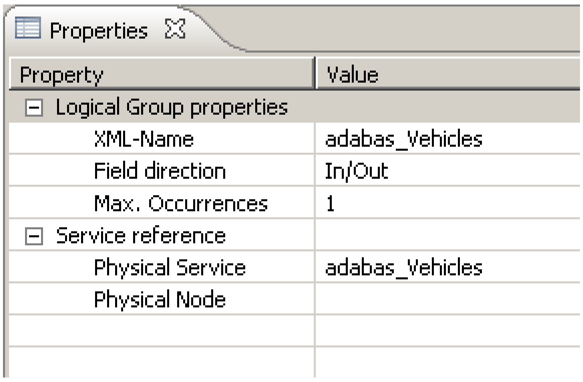
- Select the nodes which are NOT to be included in the "Logical View", as shown here, hit the <DEL> key or right-click and select Delete.



- 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".

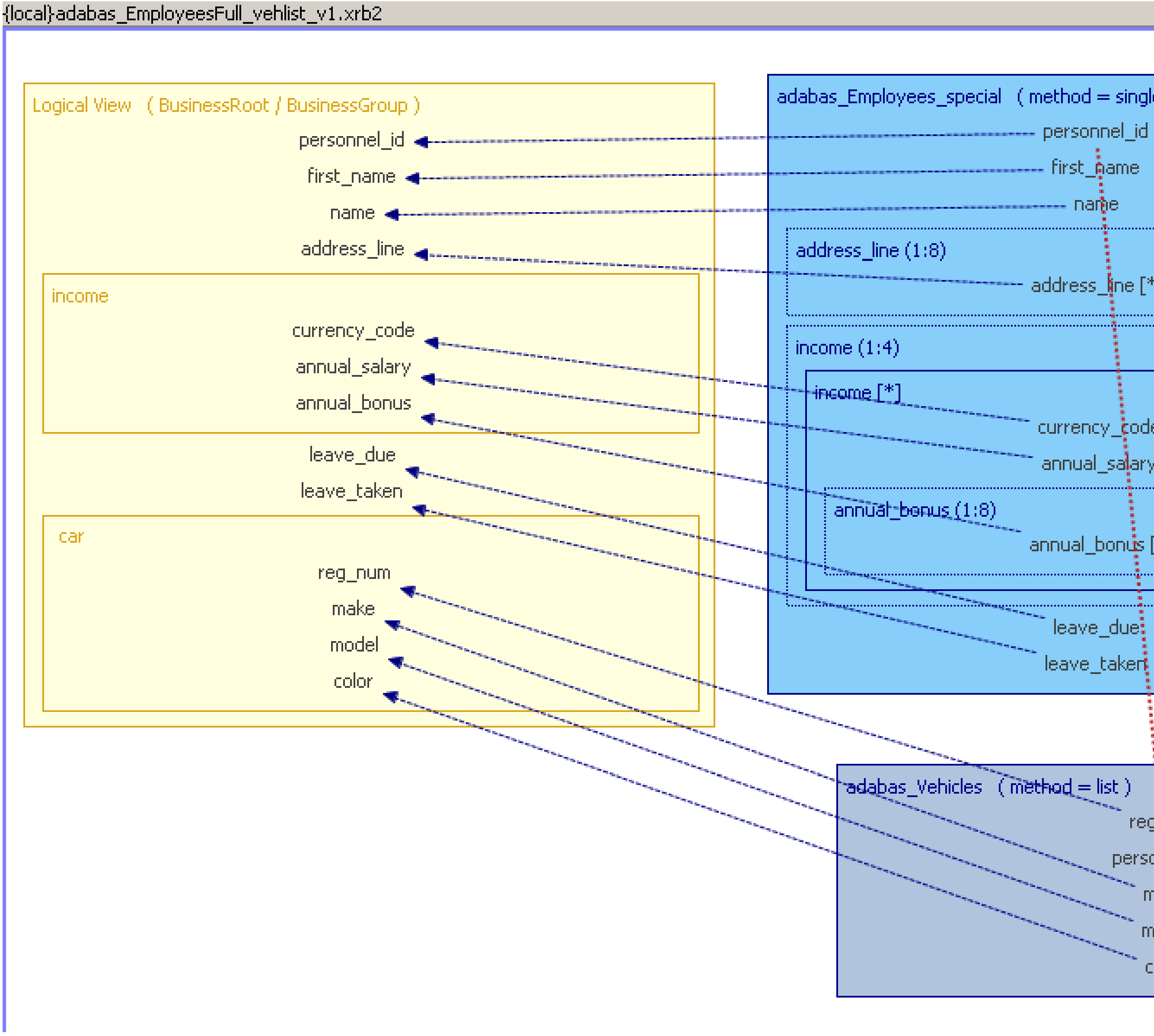


Properties

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:

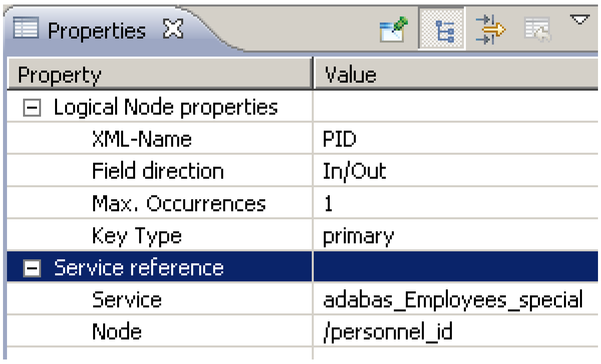


- We have already seen that the LV can be a subset of the underlying data. It can also represent that data in a different format e.g. the fields names can be changed and occurances of fields such as addreess_line can be reduced.

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

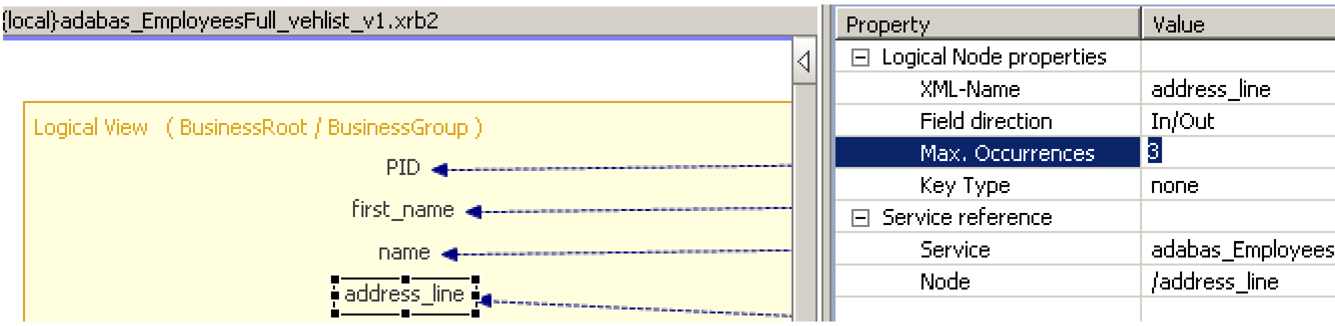
Rename the personnel_id field within the LV to PID (optional).

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



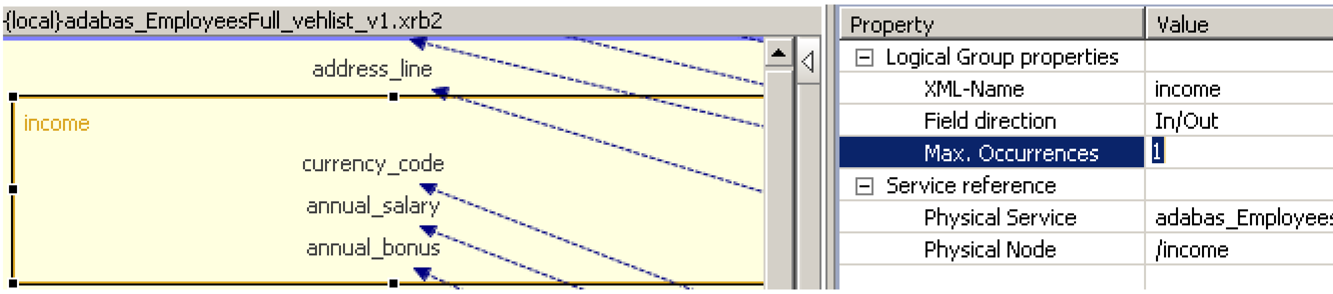
Property	Value
⊟ Logical Node properties	
XML-Name	PID
Field direction	In/Out
Max. Occurrences	1
Key Type	primary
⊟ Service reference	
Service	adabas_Employees_special
Node	/personnel_id

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


{local}adabas_EmployeesFull_vehlist_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
PID
first_name
name
address_line

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

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

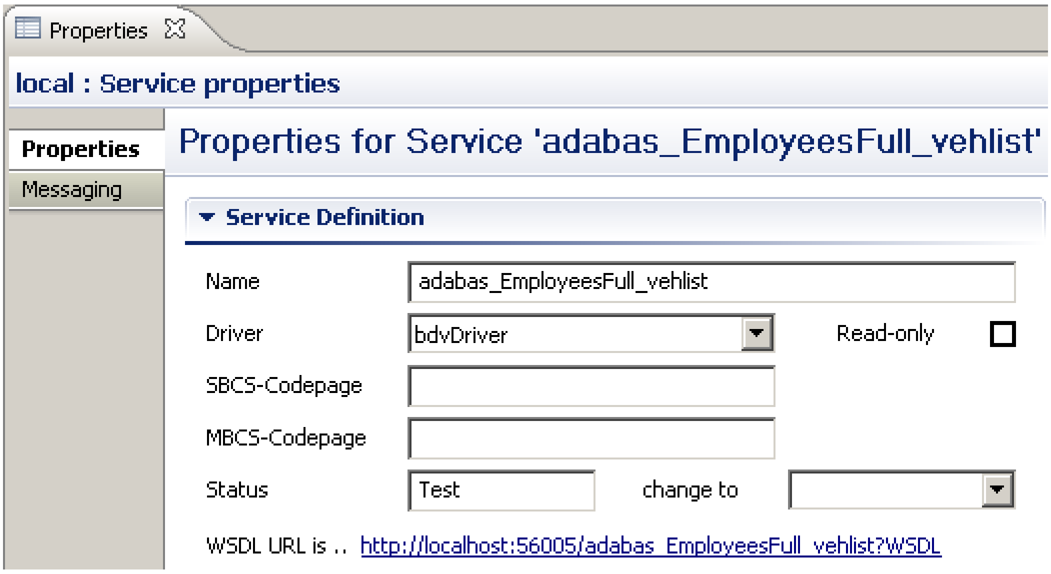


Property	Value
⊟ Logical Group properties	
XML-Name	income
Field direction	In/Out
Max. Occurrences	1
⊟ Service reference	
Physical Service	adabas_Employees
Physical Node	/income

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

Test the WebService

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


Properties
local : Service properties
Properties
Messaging
Properties for Service 'adabas_EmployeesFull_vehlist'
Service Definition

Name: adabas_EmployeesFull_vehlist

Driver: bdvDriver

Read-only ☐

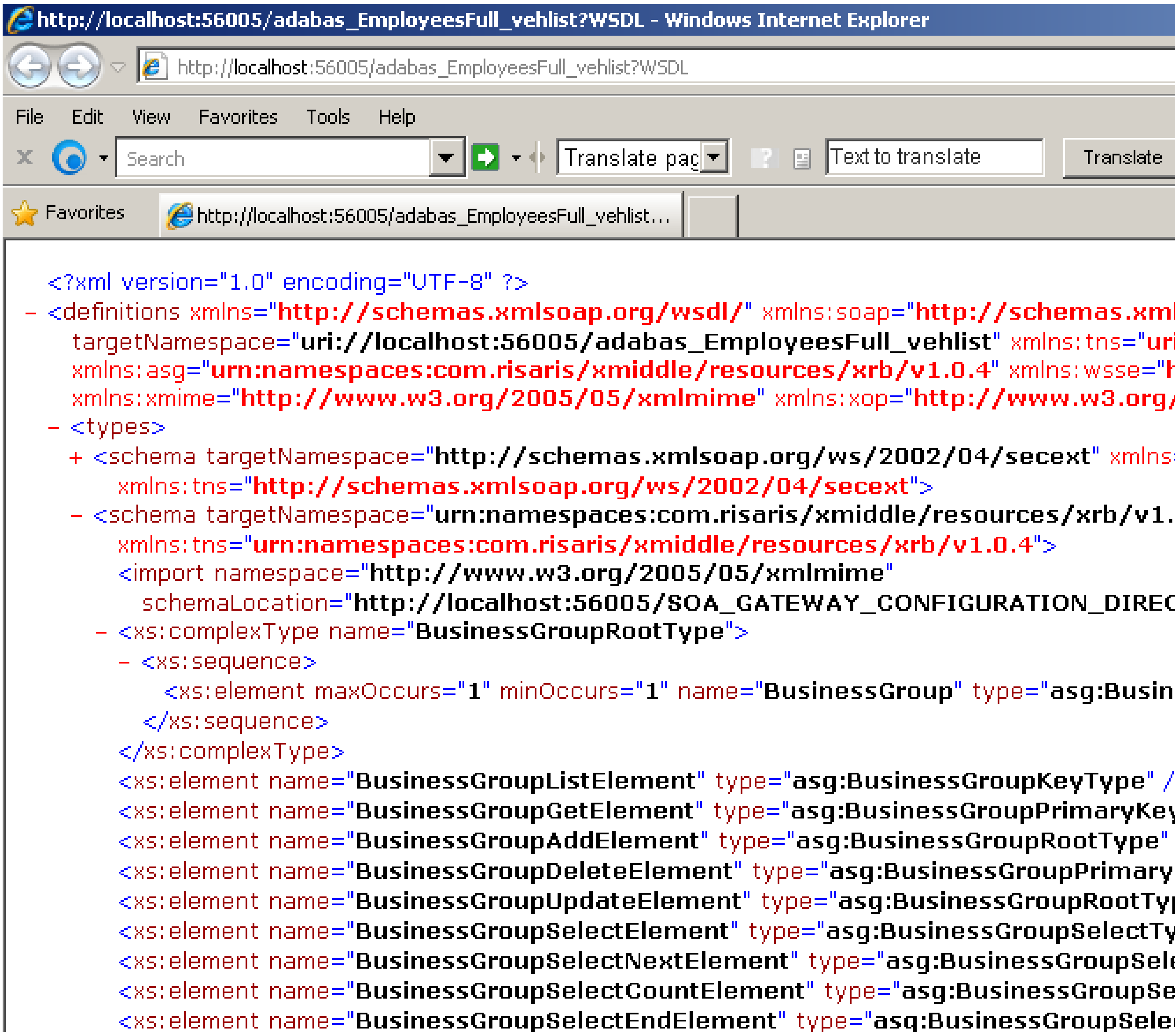
SBCS-Codepage:

MBCS-Codepage:

Status: Test change to:

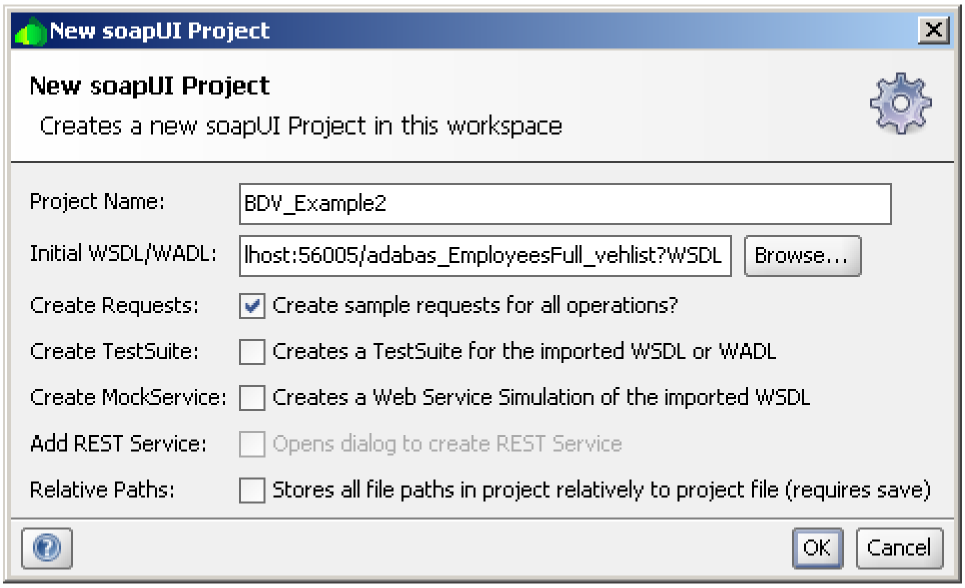
WSDL URL is .. http://localhost:56005/adabas_EmployeesFull_vehlist?WSDL

- 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).


http://localhost:56005/adabas_EmployeesFull_vehlist?WSDL - Windows Internet Explorer
http://localhost:56005/adabas_EmployeesFull_vehlist?WSDL
File
Edit
View
Favorites
Tools
Help
Search
Translate pag
Text to translate
Translate
Favorites
http://localhost:56005/adabas_EmployeesFull_vehlist...

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xml
    targetNamespace="uri://localhost:56005/adabas_EmployeesFull_vehlist" xmlns:tns="uri
    xmlns:asg="urn:namespaces:com.risaris/xmiddle/resources/xrb/v1.0.4" xmlns:wsse="h
    xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:xop="http://www.w3.org/
  - <types>
    + <schema targetNamespace="http://schemas.xmlsoap.org/ws/2002/04/secext" xmlns
        xmlns:tns="http://schemas.xmlsoap.org/ws/2002/04/secext">
    - <schema targetNamespace="urn:namespaces:com.risaris/xmiddle/resources/xrb/v1.
        xmlns:tns="urn:namespaces:com.risaris/xmiddle/resources/xrb/v1.0.4">
        <import namespace="http://www.w3.org/2005/05/xmlmime"
          schemaLocation="http://localhost:56005/SOA_GATEWAY_CONFIGURATION_DIREC
      - <xs:complexType name="BusinessGroupRootType">
        - <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="BusinessGroup" type="asg:Busin
          </xs:sequence>
        </xs:complexType>
        <xs:element name="BusinessGroupListElement" type="asg:BusinessGroupKeyType" /
        <xs:element name="BusinessGroupGetElement" type="asg:BusinessGroupPrimaryKey
        <xs:element name="BusinessGroupAddElement" type="asg:BusinessGroupRootType"
        <xs:element name="BusinessGroupDeleteElement" type="asg:BusinessGroupPrimary
        <xs:element name="BusinessGroupUpdateElement" type="asg:BusinessGroupRootTy
        <xs:element name="BusinessGroupSelectElement" type="asg:BusinessGroupSelectTy
        <xs:element name="BusinessGroupSelectNextElement" type="asg:BusinessGroupSele
        <xs:element name="BusinessGroupSelectCountElement" type="asg:BusinessGroupSe
        <xs:element name="BusinessGroupSelectEndElement" type="asg:BusinessGroupSele
```

- 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.



New soapUI Project

New soapUI Project

Creates a new soapUI Project in this workspace

Project Name: BDV_Example2

Initial WSDL/WADL: lhost:56005/adabas_EmployeesFull_vehlist?WSDL Browse...

Create Requests: ☑ Create sample requests for all operations?

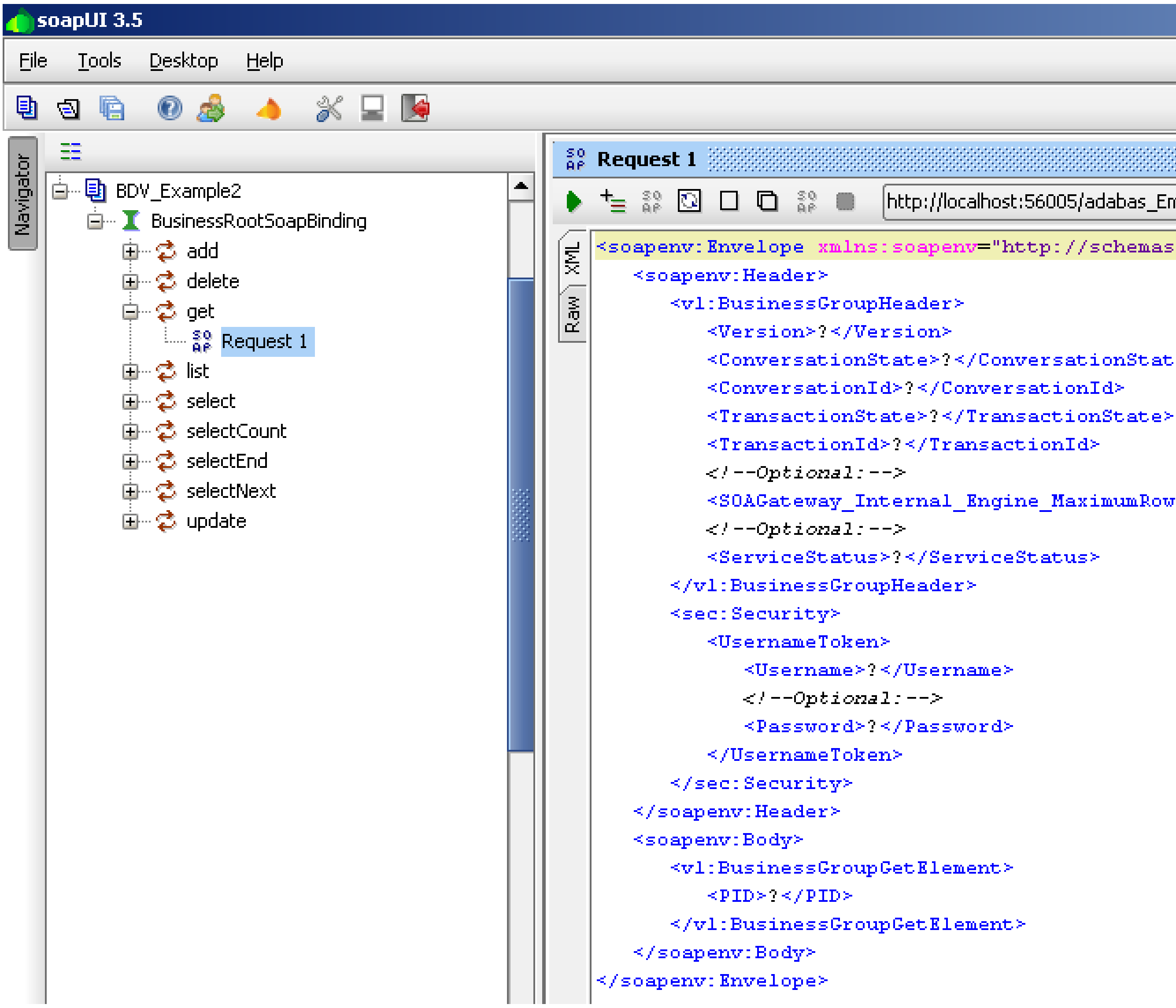
Create TestSuite: □ Creates a TestSuite for the imported WSDL or WADL

Create MockService: □ Creates a Web Service Simulation of the imported WSDL

Add REST Service: □ Opens dialog to create REST Service

Relative Paths: □ Stores all file paths in project relatively to project file (requires save)

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

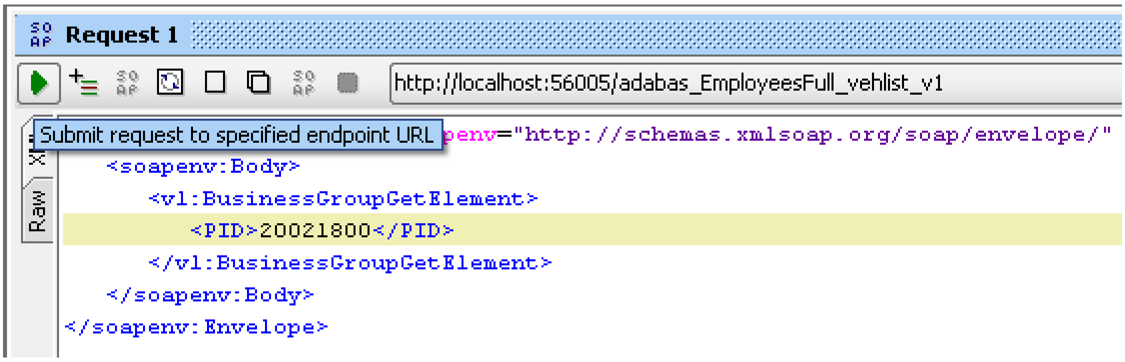


- BDV_Example2
  - BusinessRootSoapBinding
    - add
    - delete
    - get
      - Request 1
    - list
    - select
    - selectCount
    - selectEnd
    - selectNext
    - update

http://localhost:56005/adabas_Em

```
<soapenv:Envelope xmlns:soapenv="http://schemas
   <soapenv:Header>
      <vl:BusinessGroupHeader>
         <Version>?</Version>
         <ConversationState>?</ConversationStat
         <ConversationId>?</ConversationId>
         <TransactionState>?</TransactionState>
         <TransactionId>?</TransactionId>
         <!--Optional:-->
         <SOAGateway_Internal_Engine_MaximumRow
         <!--Optional:-->
         <ServiceStatus>?</ServiceStatus>
      </vl:BusinessGroupHeader>
      <sec:Security>
         <UsernameToken>
            <Username>?</Username>
            <!--Optional:-->
            <Password>?</Password>
         </UsernameToken>
      </sec:Security>
   </soapenv:Header>
   <soapenv:Body>
      <vl:BusinessGroupGetElement>
         <PID>?</PID>
      </vl:BusinessGroupGetElement>
   </soapenv:Body>
</soapenv:Envelope>
```

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

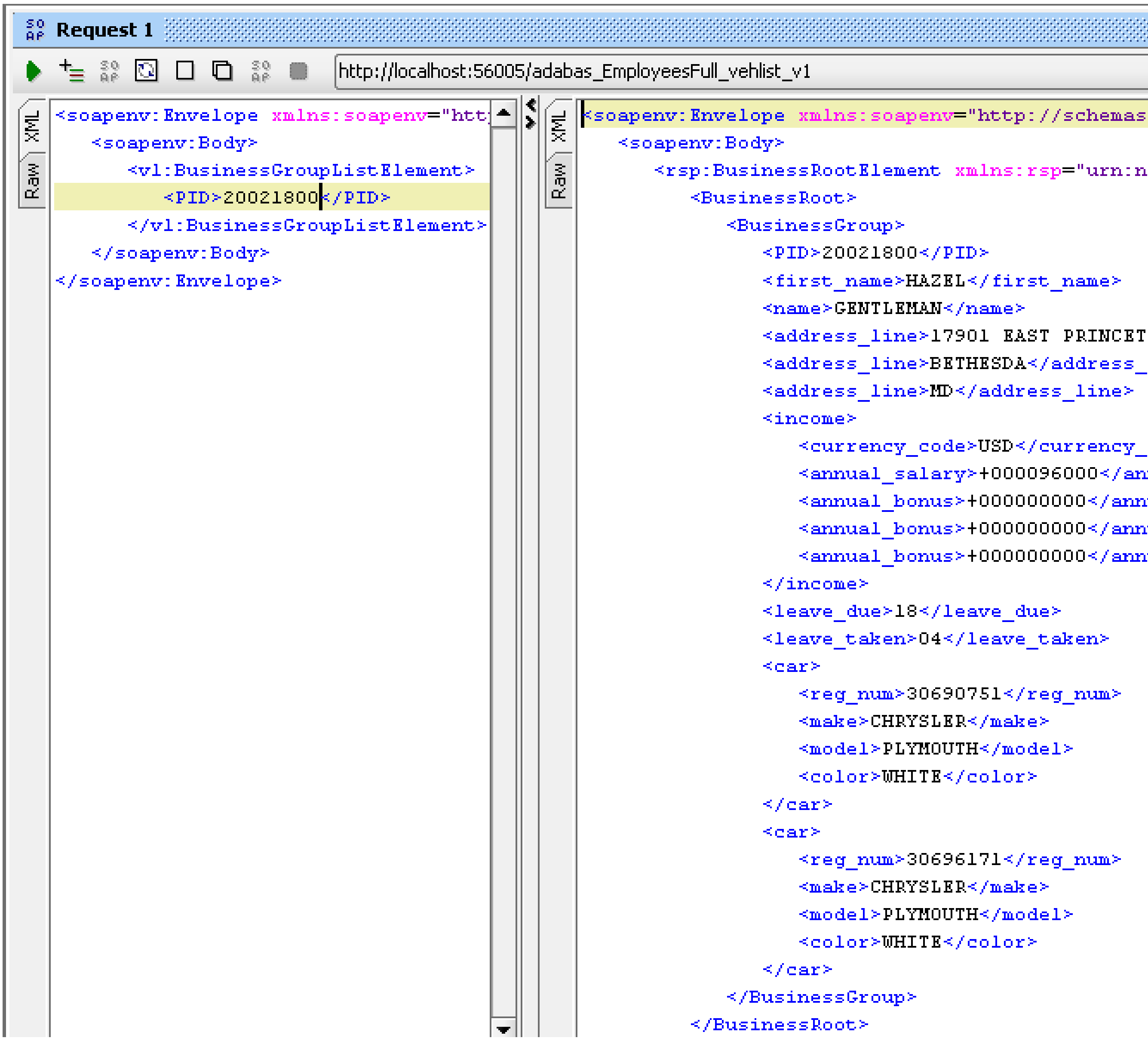


http://localhost:56005/adabas_EmployeesFull_vehlist_v1

Submit request to specified endpoint URL

```
                    penv="http://schemas.xmlsoap.org/soap/envelope/"
   <soapenv:Body>
      <vl:BusinessGroupGetElement>
         <PID>20021800</PID>
      </vl:BusinessGroupGetElement>
   </soapenv:Body>
</soapenv:Envelope>
```

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



http://localhost:56005/adabas_EmployeesFull_vehlist_v1

```
<soapenv:Envelope xmlns:soapenv="htt
   <soapenv:Body>
      <vl:BusinessGroupListElement>
         <PID>20021800</PID>
      </vl:BusinessGroupListElement>
   </soapenv:Body>
</soapenv:Envelope>
```

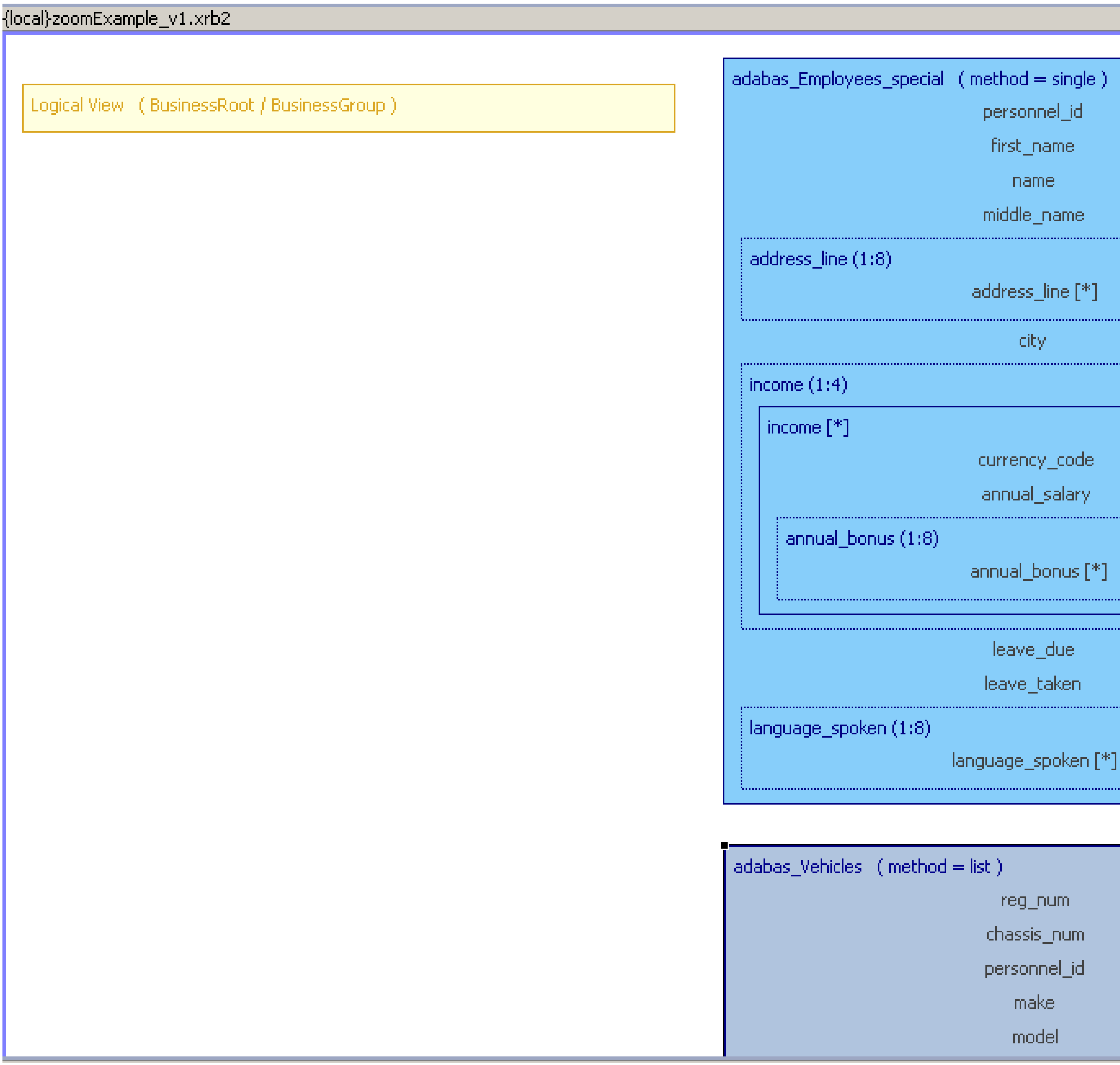
```
<soapenv:Envelope xmlns:soapenv="http://schemas
   <soapenv:Body>
      <rsp:BusinessRootElement xmlns:rsp="urn:n
         <BusinessRoot>
            <BusinessGroup>
               <PID>20021800</PID>
               <first_name>HAZEL</first_name>
               <name>GENTLEMAN</name>
               <address_line>17901 EAST PRINCET
               <address_line>BETHESDA</address_
               <address_line>MD</address_line>
               <income>
                  <currency_code>USD</currency_
                  <annual_salary>+000096000</an
                  <annual_bonus>+000000000</ann
                  <annual_bonus>+000000000</ann
                  <annual_bonus>+000000000</ann
               </income>
               <leave_due>18</leave_due>
               <leave_taken>04</leave_taken>
               <car>
                  <reg_num>30690751</reg_num>
                  <make>CHRYSLER</make>
                  <model>PLYMOUTH</model>
                  <color>WHITE</color>
               </car>
               <car>
                  <reg_num>30696171</reg_num>
                  <make>CHRYSLER</make>
                  <model>PLYMOUTH</model>
                  <color>WHITE</color>
               </car>
            </BusinessGroup>
         </BusinessRoot>
```

BusinessDataView Editor hints

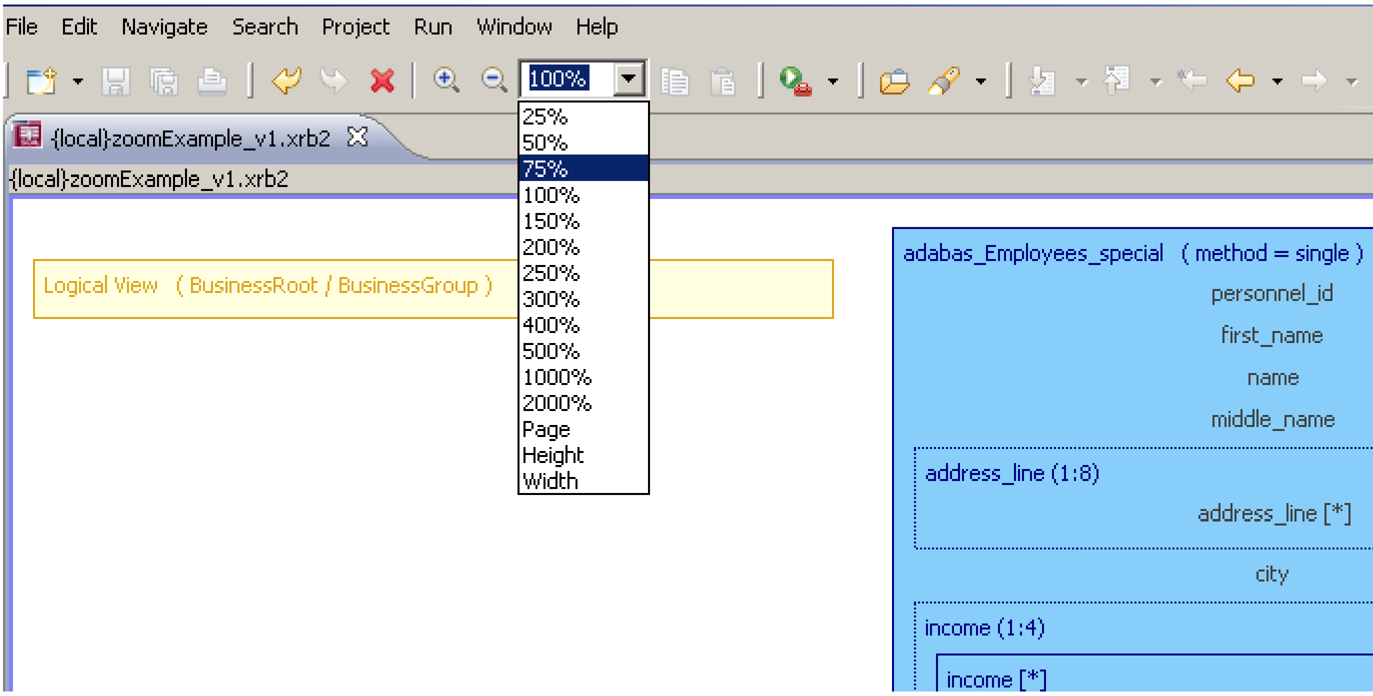
This section provides you with a number of general purpose hints for making your life easier in respect to editing BusinessDataViews.

Zooming

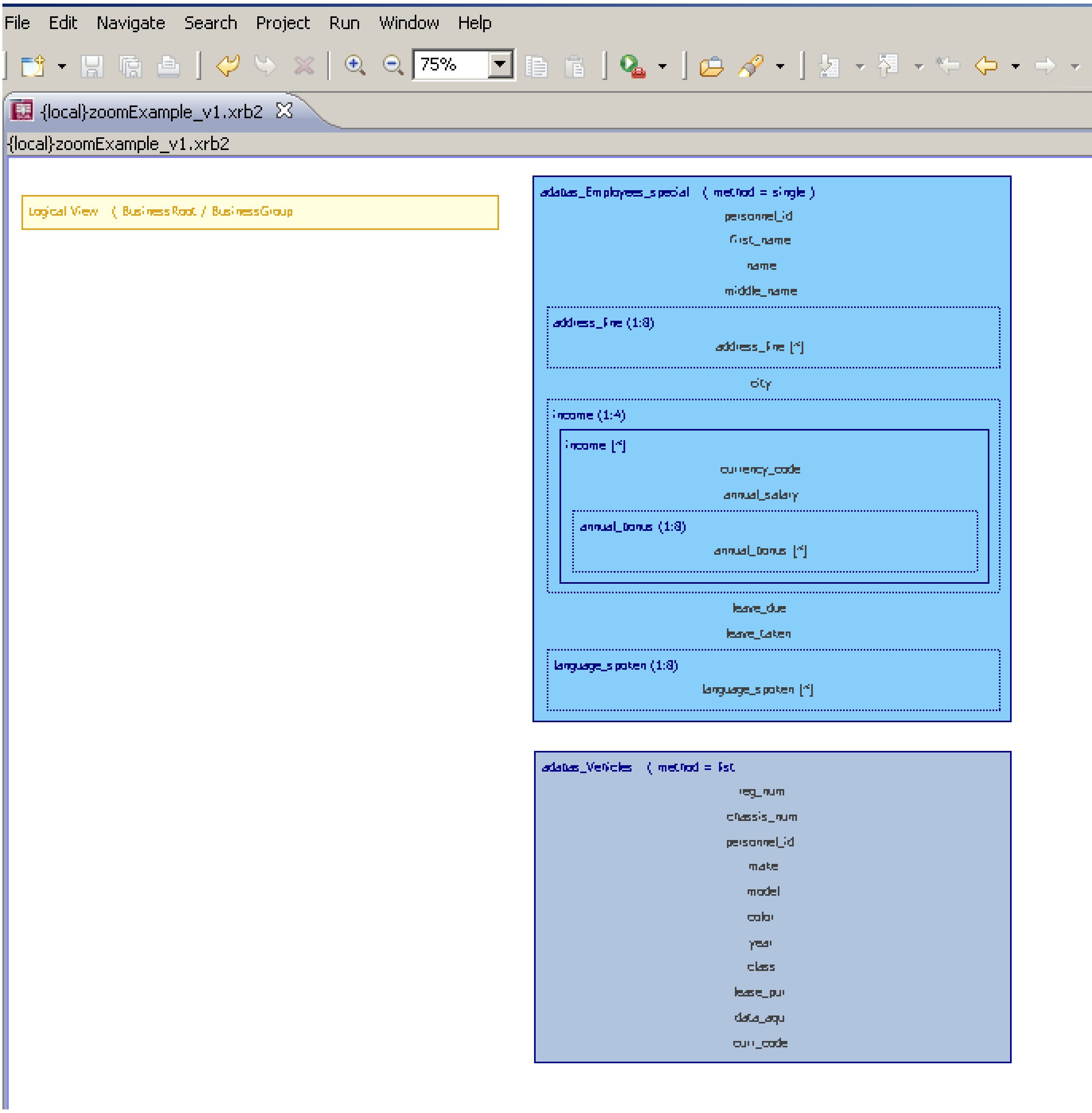
When the BusinessDataView is too big, or the screen is to small to display it in its entirety at a resolution of 100%, or if you want to see some element(s) in more detail, the zoom level can be changed accordingly.



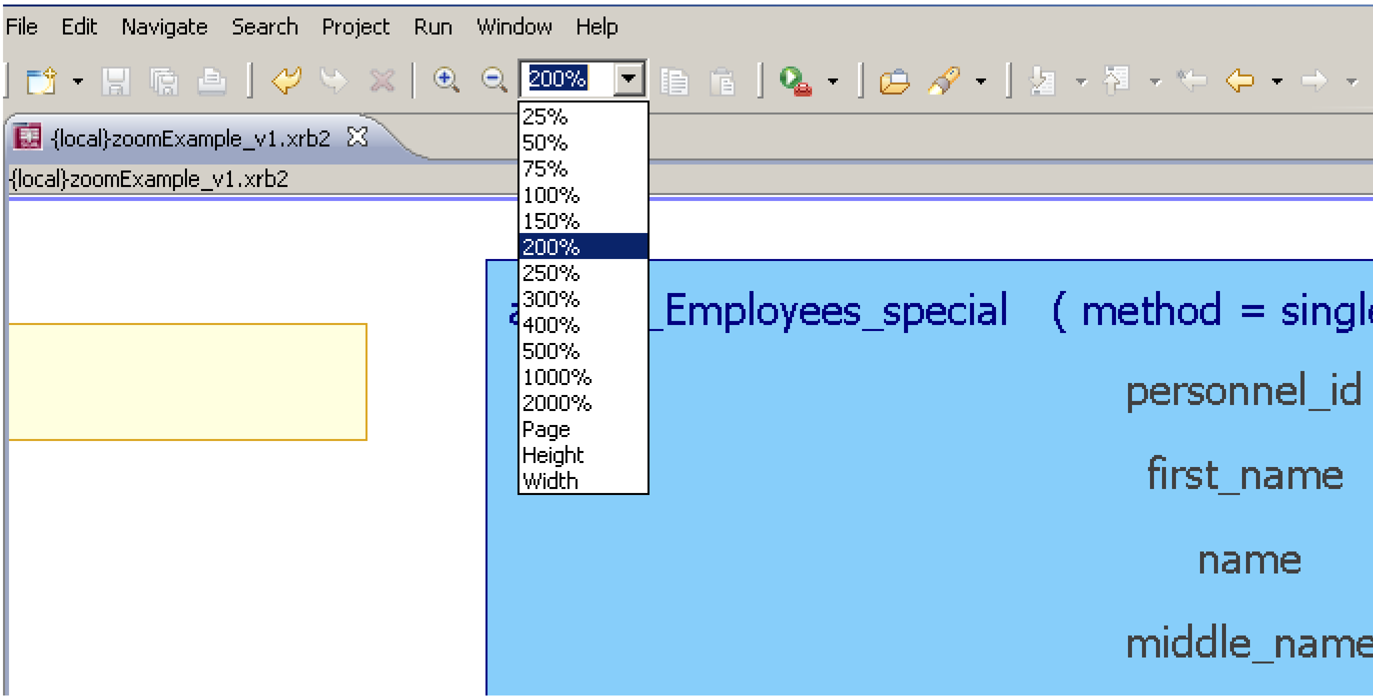
The initial display at 100%, not all details are visible.


File Edit Navigate Search Project Run Window Help
100%
25%
50%
75%
100%
150%
200%
250%
300%
400%
500%
1000%
2000%
Page
Height
Width
{local}zoomExample_v1.xrb2
{local}zoomExample_v1.xrb2
Logical View (BusinessRoot / BusinessGroup)
adabas_Employees_special (method = single)
personnel_id
first_name
name
middle_name
address_line (1:8)
address_line [*]
city
income (1:4)
income [*]

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



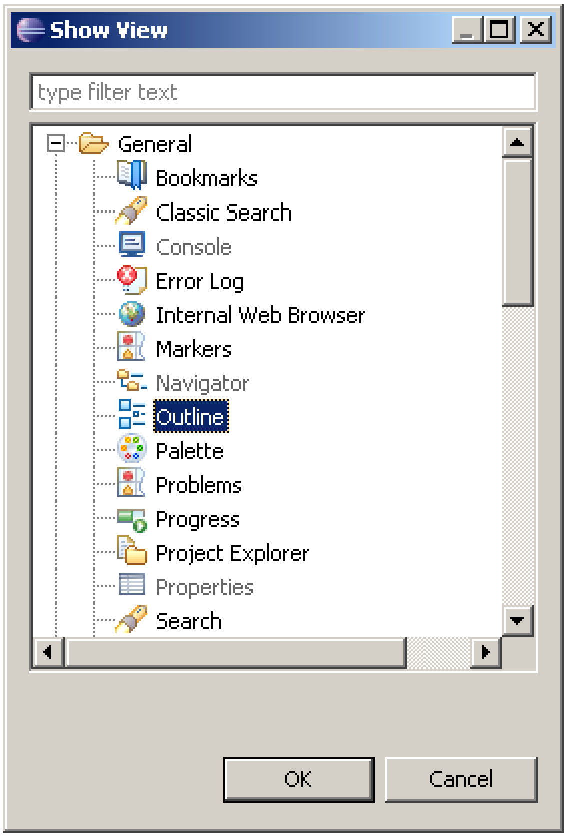
Increase the zoom level to a value above 100% to see more details.


File Edit Navigate Search Project Run Window Help
200%
25%
50%
75%
100%
150%
200%
250%
300%
400%
500%
1000%
2000%
Page
Height
Width
{local}zoomExample_v1.xrb2
{local}zoomExample_v1.xrb2
_Employees_special (method = singl
personnel_id
first_name
name
middle_name

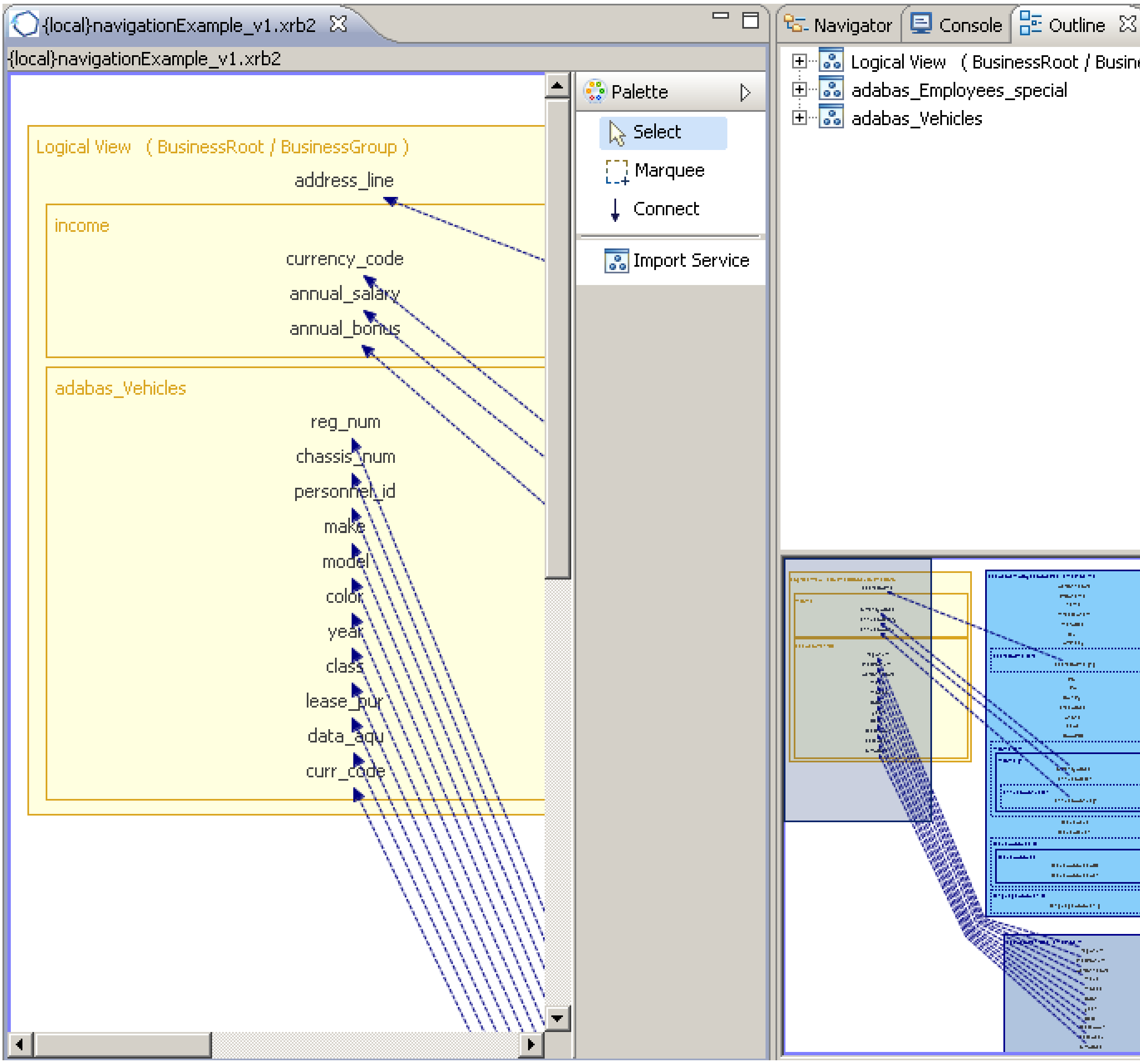
Making the BDV elements visible

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

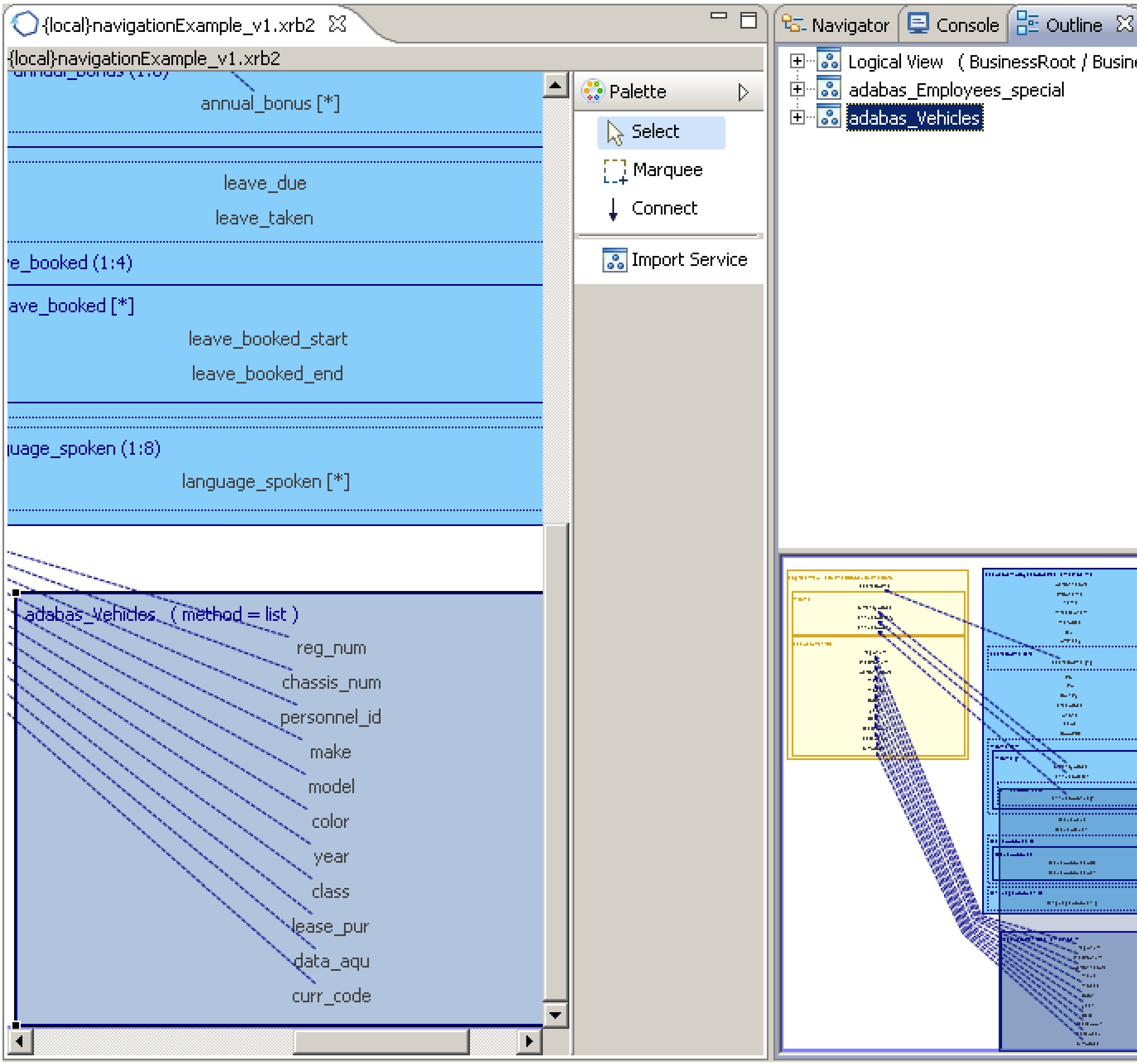
If the Outline view is not visible, from the menu select Window -> Show View -> Other..., select Outline in the General folder and hit OK.



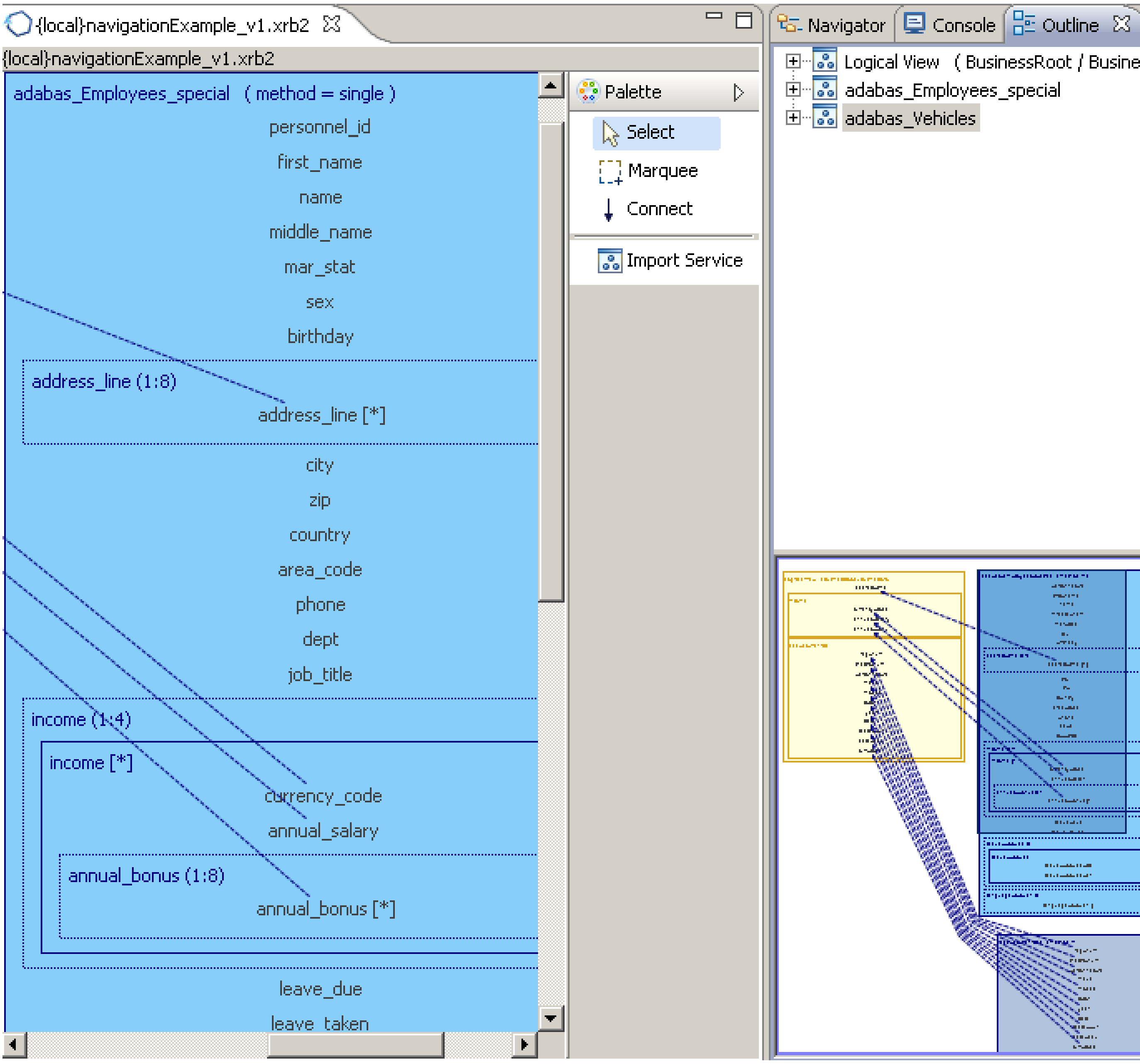
In this example we have a populated LV plus 2 PRs, with the employees PR partially visible. To make the vehicles, select the entry in the Outline view as



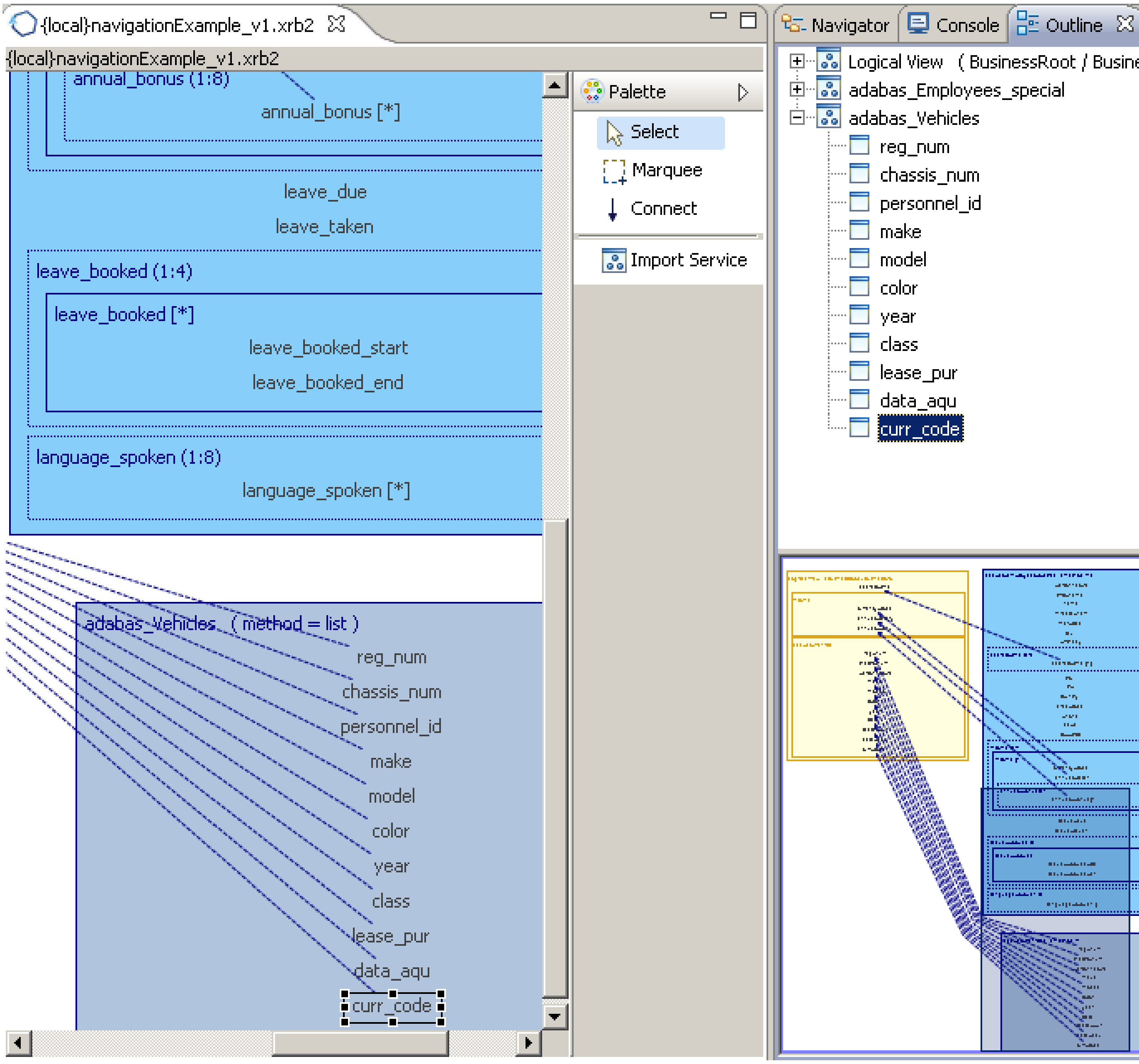
To make the vehicles PR visible, select the entry (adabas_Vehicles in this instance) in the Outline view as shown below. The view will align itself with the PR selected. Notice also how the aerial view in the bottom right-hand corner has also moved the focus to the PR, with the darker blue rectangle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



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



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



When a Physical View changes

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

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

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

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

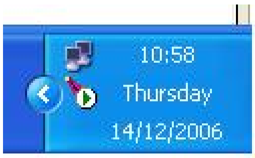
Server administration

- *Starting / stopping a Portus Server on Windows* on page 106
- *Starting / stopping a Portus Server on z/OS* on page 106

- *Starting / stopping a Portus Server on z/VSE* on page 106
- *Starting / stopping a Portus Server on Linux, AIX and Solaris* on page 106
- *Updating an expired license* on page 106

Starting / stopping a Portus Server on Windows

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



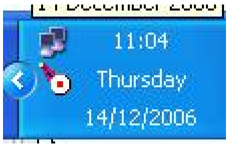
If not, start the program by clicking **Start -> Portus vN -> Apache -> Monitor Apache Servers**

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

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

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

If Portus is not running the icon looks like this



Starting / stopping a Portus Server on z/OS

Portus can be stopped by issusing "/p jobname" at the terminal.

To start Portus, re-submit the ASGSTRT job or restart the Portus started procedure.

Starting / stopping a Portus Server on z/VSE

Bring up the Attention Routine prompt for the Portus partition by issuing the AR command MSG <ASG_partition_ID>

Shut down the server with the command <ASG_partition_reply_id> EOJ

To start Portus, re-submit the ASGSTRT job or restart it with the respective PRELEASE command in case the ASGSTRT job has been submitted with DISP=K or DISP=L.

Starting / stopping a Portus Server on Linux, AIX and Solaris

As Portus runs within Apache, bring down Apache to stop the Portus. Use the apachectl command, which will be located in the [INSTALL_HOME]/apache2/bin directory of your Portus installation.

To start the server, issue the same command again with start as the first parameter.

Updating an expired license

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

- Linux/Solaris/HP/AIX : Replace the existing RisarisLicense.xml in *[INSTALL_HOME]/config*
- Windows : Replace the existing RisarisLicense.xml *%ALLUSERSPROFILE%\Portus 4.3.1/ configuration/*
- z/OS : Replace existing INSTALL.HOME.CONF(ASGLIC) and submit INSTALL.HOME.JCL(SOAGINS4).

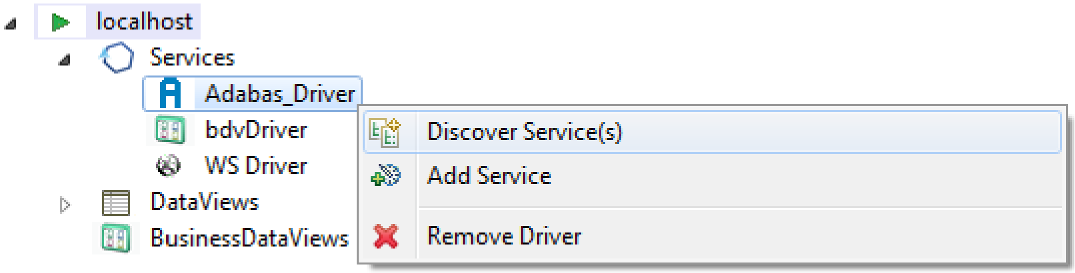
WebService Creation

Service definitions, and related mapping files (DataViews, XSDs) are created, in a semi-automatic to automatic process (also known as Service Discovery), from meta-data describing the characteristics of resources to be exposed as a Service through Portus, for example from an Adabas FDT, SQL Table description, Natural sources or even other Web Services.

- *Initiating Service creation* on page 107
- *Creating Service(s) from Adabas* on page 107
- *Assigning a different DataView to existing service* on page 109
- *Creating Service(s) from SQL databases* on page 110
- *Creating Service(s) from Stored Procedures* on page 111
- *Creating Service(s) from a SYSOBJH extract file* on page 112
- *Creating Service(s) from Natural* on page 115
- *Creating Service(s) from 3rd party web services* on page 119

Initiating Service creation

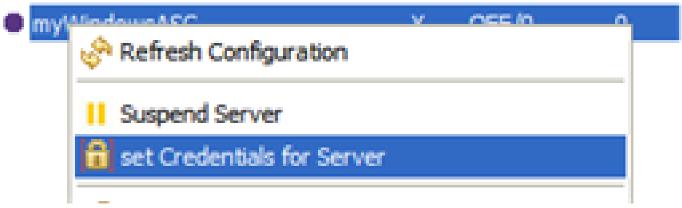
The Service creation process can be started by selecting the appropriate driver, right-clicking and selecting either Discover Service(s) or Add Service:



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.

It may be required to provide login credentials, for example when accessing a SQL database system, this can be done from the server's context menu as well:

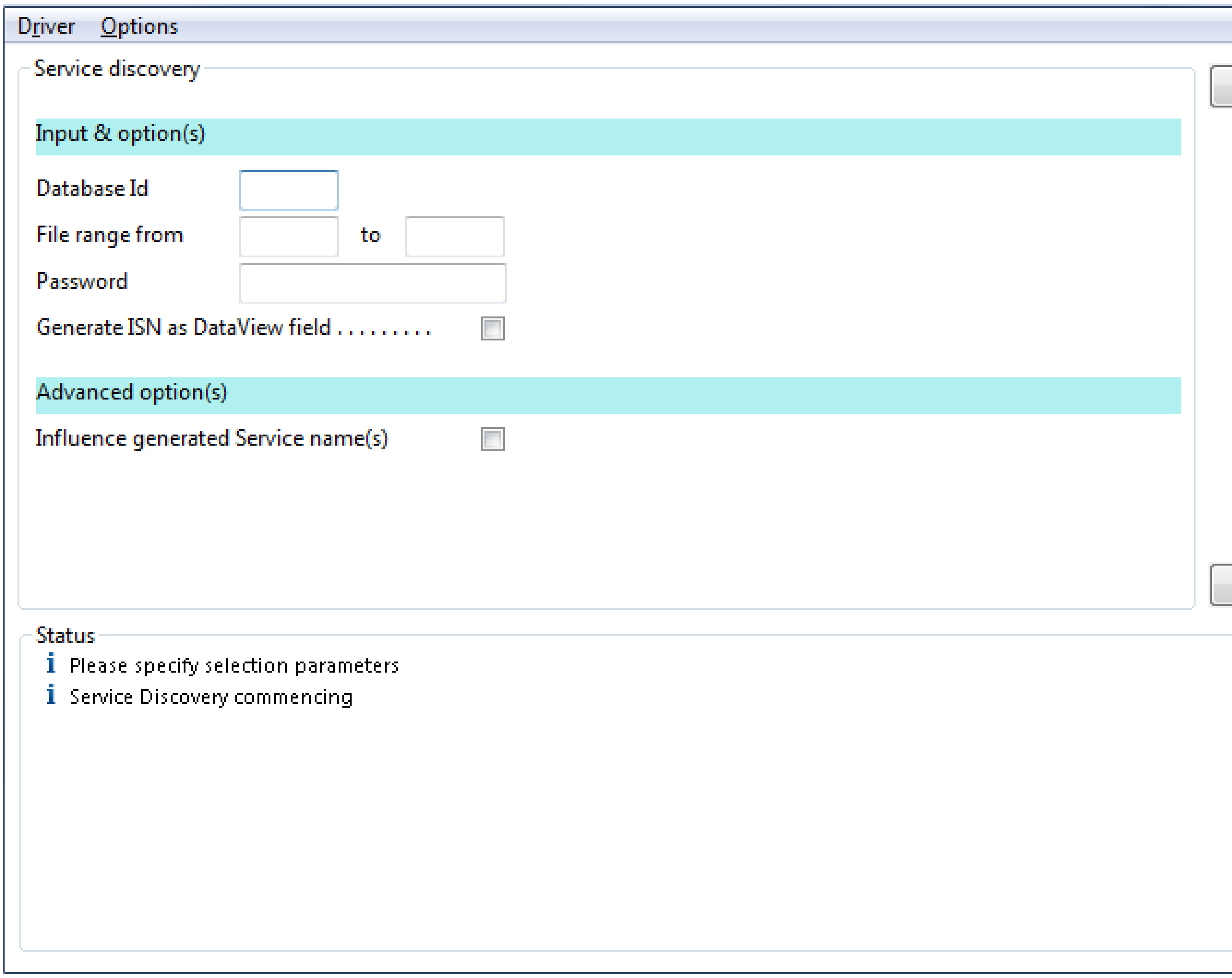


Specify the credentials - User ID and Password - in the SOAP Security section, click Set

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

Creating Service(s) from Adabas

- Specify the Database Id (range 1 - 255 on OpenSystems, 1 - 65535 on Mainframe platforms), plus 'File range from' and 'to' file numbers (1 - 5000) to limit the discovery to the selected range. Click the 'Discover' button:

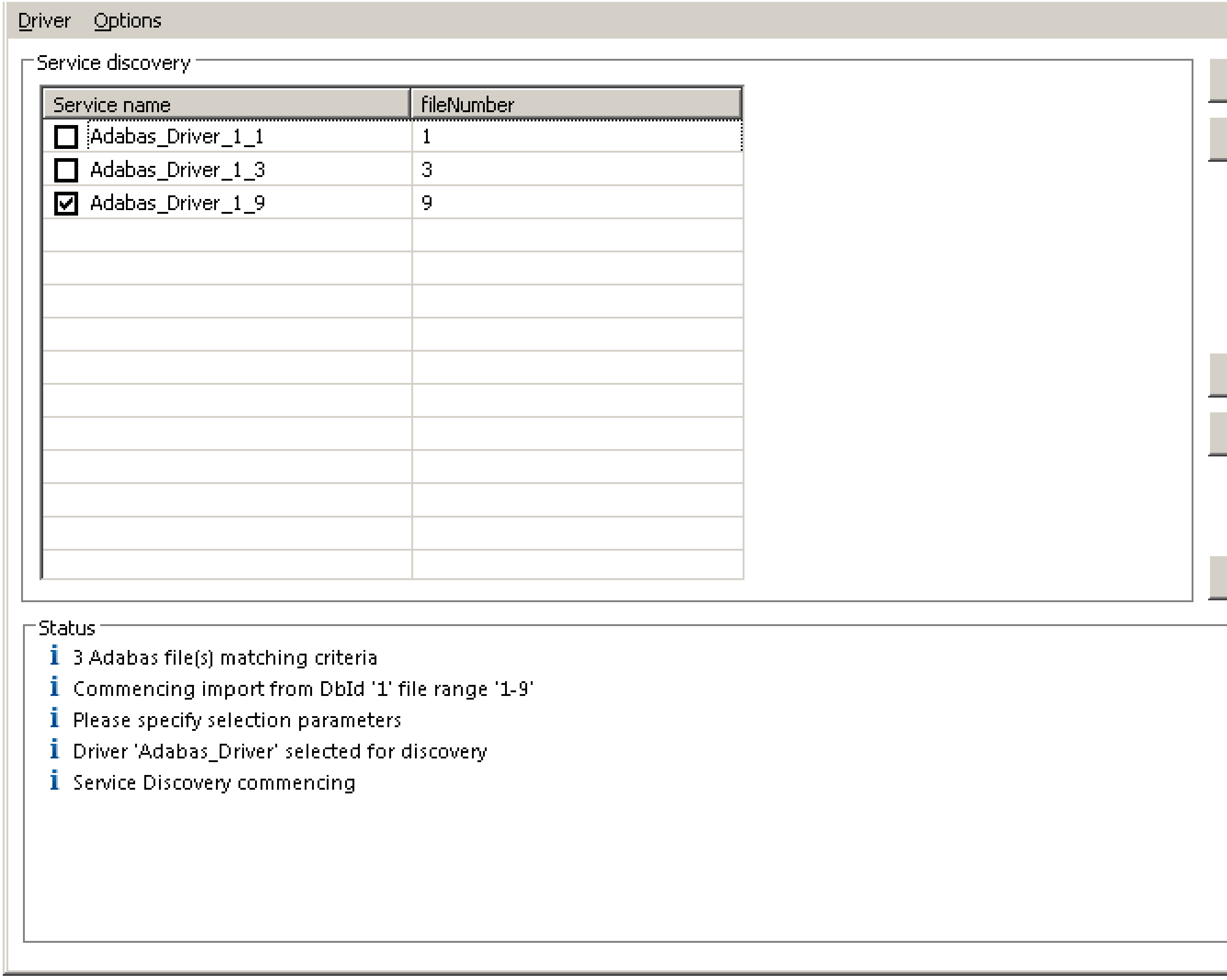


Please specify selection parameters

Service Discovery commencing

Note: Discovering the full range of possible file numbers (1 - 5000) may take substantial time and adversely affect performance of both the Portus server as well as the Adabas target database. Break the discovery process down to smaller range(s) of files.

- Select the file(s) to be turned into Services, click the 'Generate >' button:



Service name	fileNumber
□ Adabas_Driver_1_1	1
□ Adabas_Driver_1_3	3
☑ Adabas_Driver_1_9	9

i 3 Adabas file(s) matching criteria
i Commencing import from DbId '1' file range '1-9'
i Please specify selection parameters
i Driver 'Adabas_Driver' selected for discovery
i Service Discovery commencing

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.

  Note:

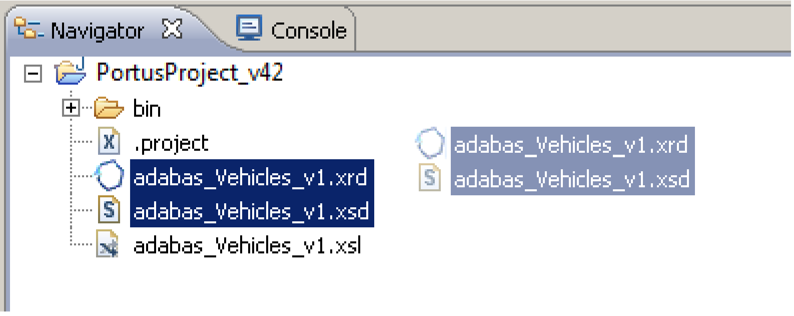
  1 The Adabas "long names" DataViews are available on the Portus Server where Adabas is licensed.

  2 These can be imported using the *Portus View*.

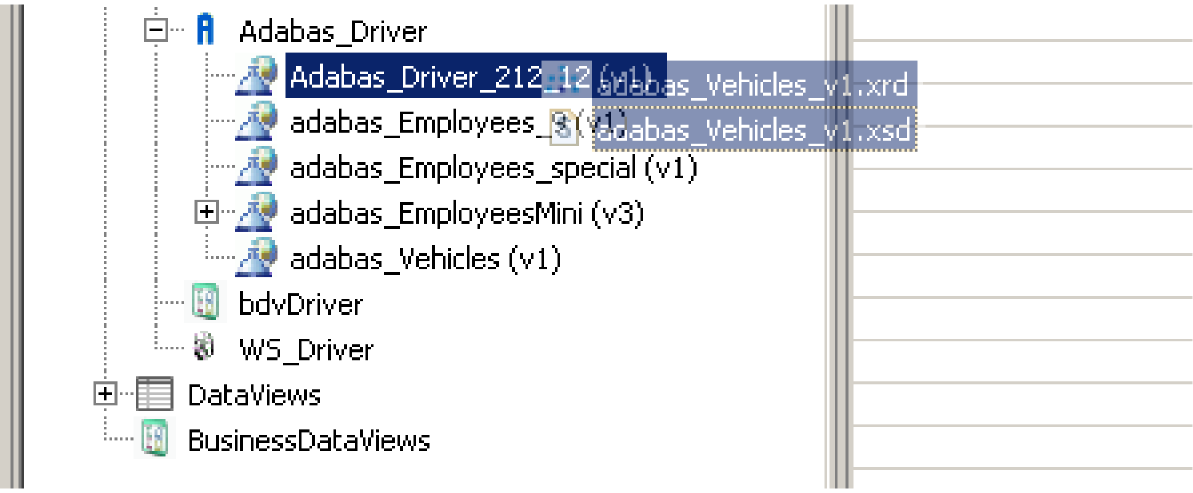
- OR the pre-created (Business)DataView is available on the server already
- The web service you wish to modify has already been created.

If the above assumptions are OK, follow these steps:

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



2 Drop the files onto the service you wish to modify



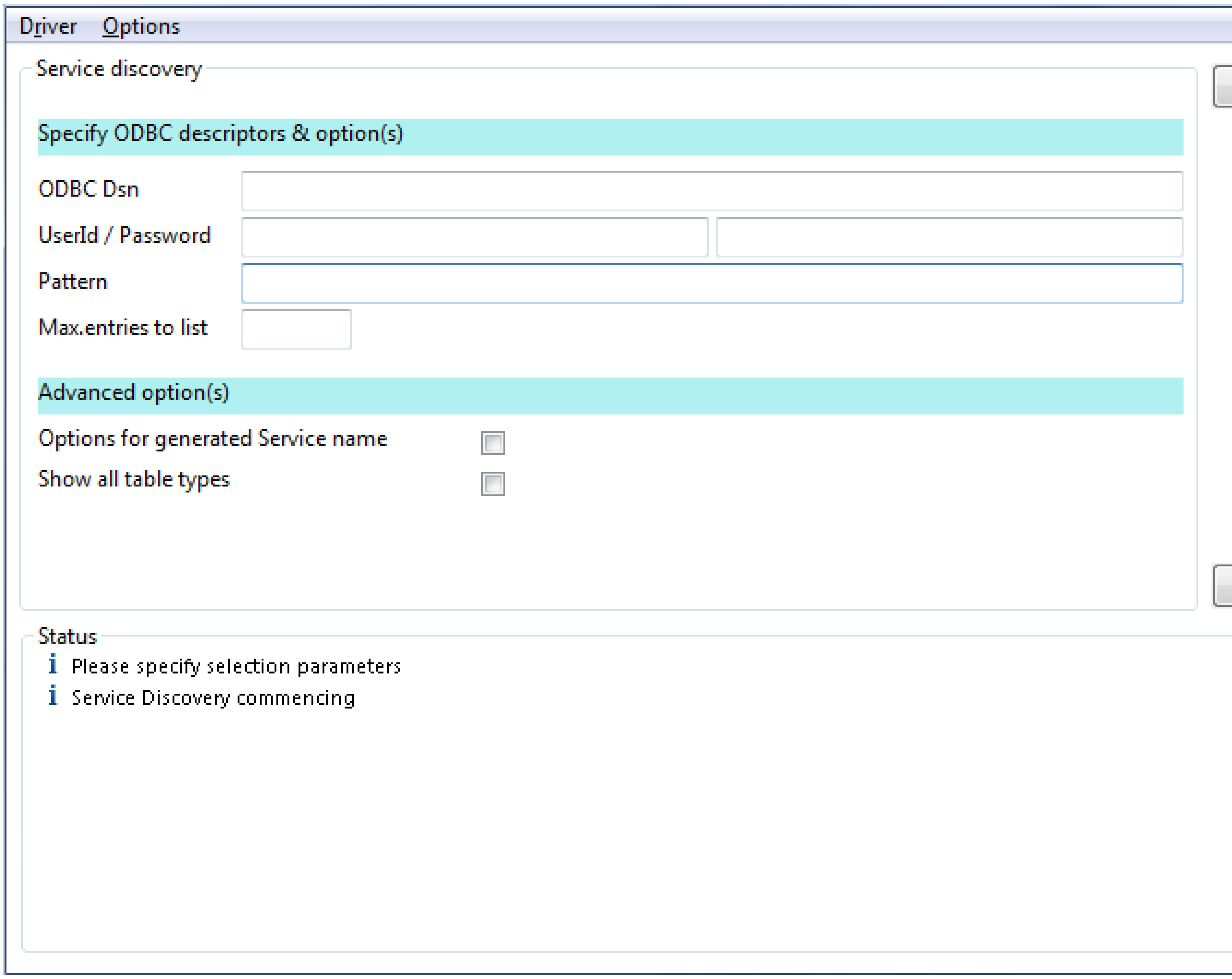
A new (Business)DataView can be assigned by dragging it from the respective Views section and drop it onto the service.

3 Right-click the service and select the *Refresh* function

4 The WSDL should now reflect the DataView. changes

Creating Service(s) from SQL databases

- Select a defined driver pointing to the SQL database system of your choice from the list. This step-by-step guide will use MySQL to demonstrate the process.
- For this example the 'world database' demo has been loaded as per the guide published at the *MySQL website*, and a ODBC Data Source been set up as 'WorldDS', pointing to the 'world' database.
- Right-click the driver and select **Discover Service(s)**
- Specify the ODBC Dsn to be browsed. Optionally enter a 'Pattern' limiting the operation, for example 'c%' will return a list of tables starting with an lowercase 'c'. Another option is to specify 'Max.entries to list' to further limit the number of tables to be listed. If appropriate for your database configuration enter a UserId and Password. Click the 'Discover' button:



Service discovery

Specify ODBC descriptors & option(s)

ODBC Dsn

UserId / Password

Pattern

Max.entries to list

Advanced option(s)

Options for generated Service name □

Show all table types □

Status

Please specify selection parameters

Service Discovery commencing

- zOS and DB2

  In the ODBC Dsn field above - enter the Location name returned from the DB2 command Display DDF

  - in the example below the location name is S1D831

  ```
   COMMAND INPUT ===> /-D831 DIS DDF
  RESPONSE=POC1 DSNL080I -D831 DSNLTDDF DISPLAY DDF REPORT FOLLOWS:
   DSNL081I
  STATUS=STARTD DSNL082I LOCATION LUNAME GENERICLU DSNL083I S1D831
  DEIBMIPA.IPAA77D8 -NONE .......
  ```

  Also ensure that the member CLIINI in the JCL dataset has been modified as necessary. Refer to your DBA on this.
- Select the table(s) to be turned into Services. If appropriate the Service name(s) can be modified by selecting the 'Options for generated Service name' tick box and modifying the entries. Finally click the 'Generate' button
- The Status pane 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.
- 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 Stored Procedures

- Details on how to create/modify/delete a stored procedure driver can be found *here*.

- Select the stored procedure driver from the list This step-by-step guide will use MS SQL Server database AdventureWorks to demonstrate the discovery process.
- Specify the ODBC Dsn to be browsed. Optionally enter a 'Pattern' limiting the operation, for example 'uspGet%' will return a list of stored procedures starting with 'uspGet'. Optionally specify 'Max. entries to list' to further limit the number of stored procedures to be listed. Click the 'Discover' button.

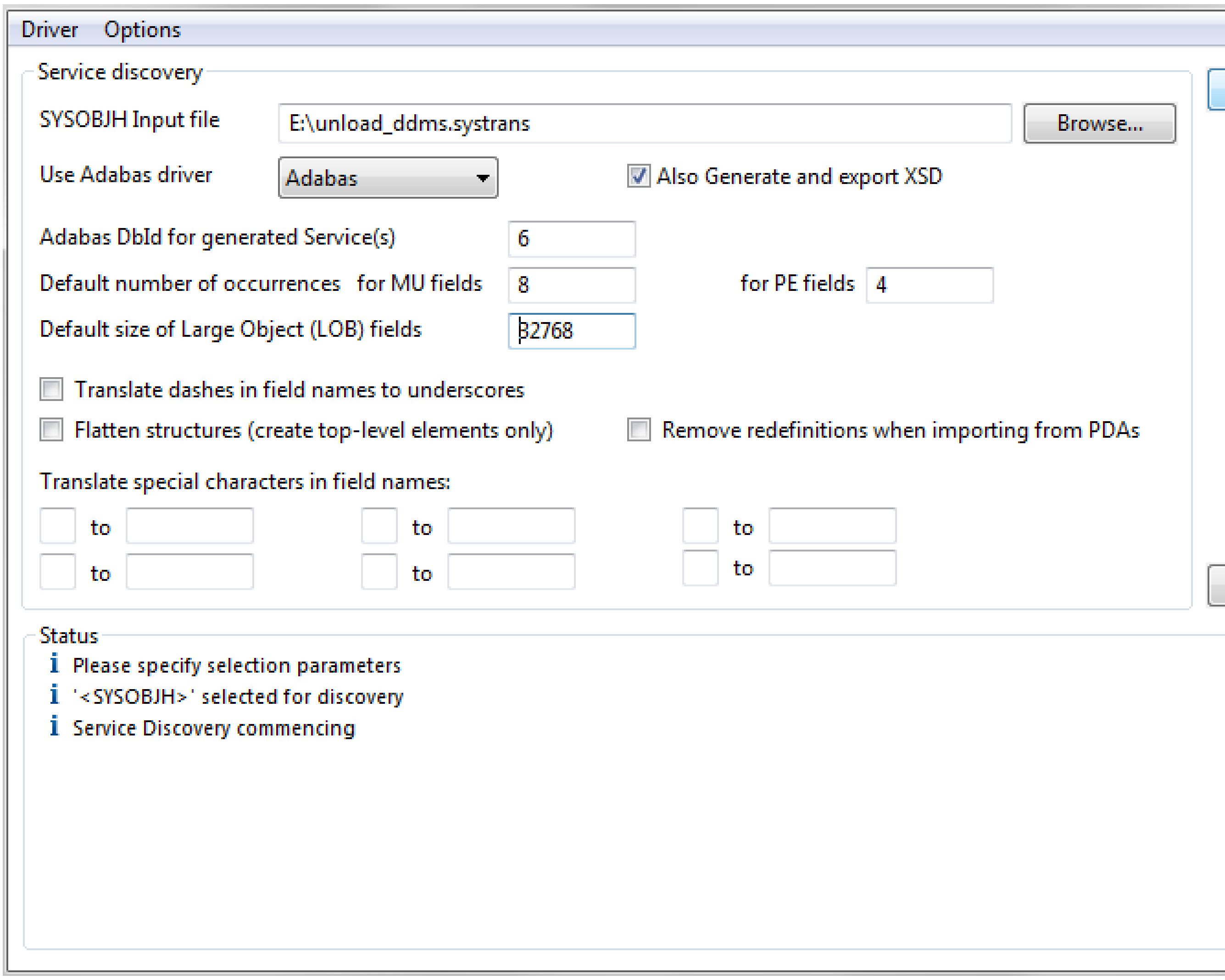
  Note: the generated web service name can also be changed from the default of prefixing with 'storedProcDriver_'

- Select the stored procedure(s) to be turned into Services, click the 'Generate' button
- The Status pane will indicate what has been generated.
- The requested Services and Service definition files (DataView(s) and XSD(s)) have been added to your server configuration
- The Wizard is still active and ready for further discovery, click 'Discover' to start the process again, or 'Exit' to dismiss the wizard.

Creating Service(s) 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 the *Discover Service(s)* context function

  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).



Service discovery

SYSOBJH Input file: E:\unload_ddms.systrans

Use Adabas driver: Adabas

☑ Also Generate and export XSD

Adabas DbId for generated Service(s): 6

Default number of occurrences for MU fields: 8 for PE fields: 4

Default size of Large Object (LOB) fields: 32768

□ Translate dashes in field names to underscores

□ Flatten structures (create top-level elements only)

□ Remove redefinitions when importing from PDAs

Translate special characters in field names:

	to			to			to	
	to			to			to	

Status

i Please specify selection parameters

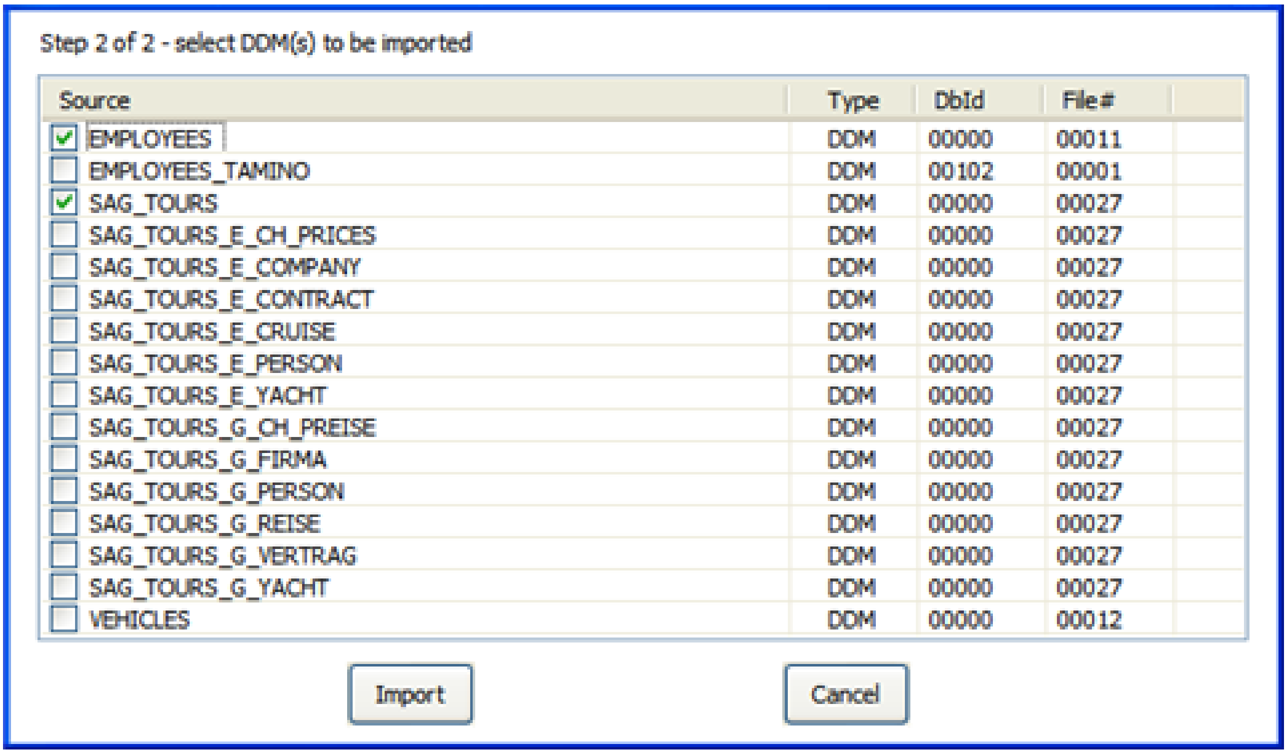
i '<SYSOBJH>' selected for discovery

i Service Discovery commencing

A number of options can be specified here

  - defaults to be used when generating Service(s) and DataView(s) based on Adabas files
    - Database Id used when the DDM is defined with a DbId of 0 (zero)
    - Default number of occurrences for MU(ltiple value) fields and PE(riodic) groups
    - Default size of large object (LOB) fields
  - if you wish dashes ('-') in file and element names to be translated to underscores ('_'). This may be desirable where the target programming language used to access Portus Resources do not accept dashes in field names, an example being PHP.
  - When PDAs include redefinitions you will need to choose the layout that suits you for the DataView to be generated. If you are unsure at the moment, leave the "remove redefinitions" box unchecked, in which case ALL redefinitions will be included in the generated DataView, with a generated fgroup name of "*redefinition_of_<PDA_group_name>". Before the DataView is uploaded to the server, remove all definitions/redefinitions not required. Multiple DataViews (and thus resources) are required when requiring different layouts/set of parameters for the same subprogram.
  - "Flatten structures" when the PDA includes nested groups but you want all fields to be generated at the "top level" without any group/field structure.
  - You may need to translate special characters, like the # (hash) sign if they are used as part of field name(s), because they are not valid in a SOAP context.

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

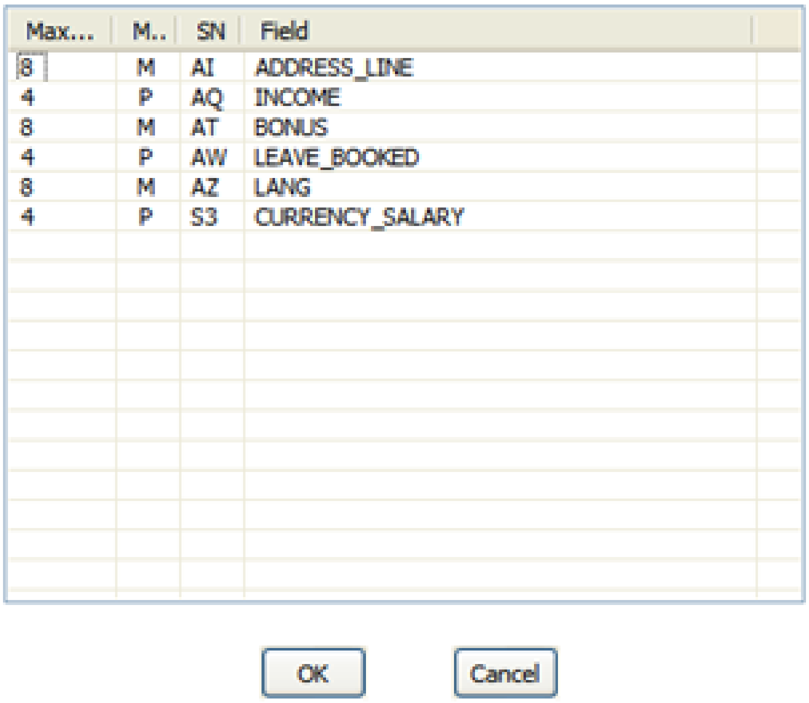


Step 2 of 2 - select DDM(s) to be imported

Source	Type	DbId	File#
☑ EMPLOYEES	DDM	00000	00011
☐ EMPLOYEES_TAMINO	DDM	00102	00001
☑ SAG_TOURS	DDM	00000	00027
☐ SAG_TOURS_E_CH_PRICES	DDM	00000	00027
☐ SAG_TOURS_E_COMPANY	DDM	00000	00027
☐ SAG_TOURS_E_CONTRACT	DDM	00000	00027
☐ SAG_TOURS_E_CRUISE	DDM	00000	00027
☐ SAG_TOURS_E_PERSON	DDM	00000	00027
☐ SAG_TOURS_E_YACHT	DDM	00000	00027
☐ SAG_TOURS_G_CH_PREISE	DDM	00000	00027
☐ SAG_TOURS_G_FIRMA	DDM	00000	00027
☐ SAG_TOURS_G_PERSON	DDM	00000	00027
☐ SAG_TOURS_G_REISE	DDM	00000	00027
☐ SAG_TOURS_G_VERTRAG	DDM	00000	00027
☐ SAG_TOURS_G_YACHT	DDM	00000	00027
☐ VEHICLES	DDM	00000	00012

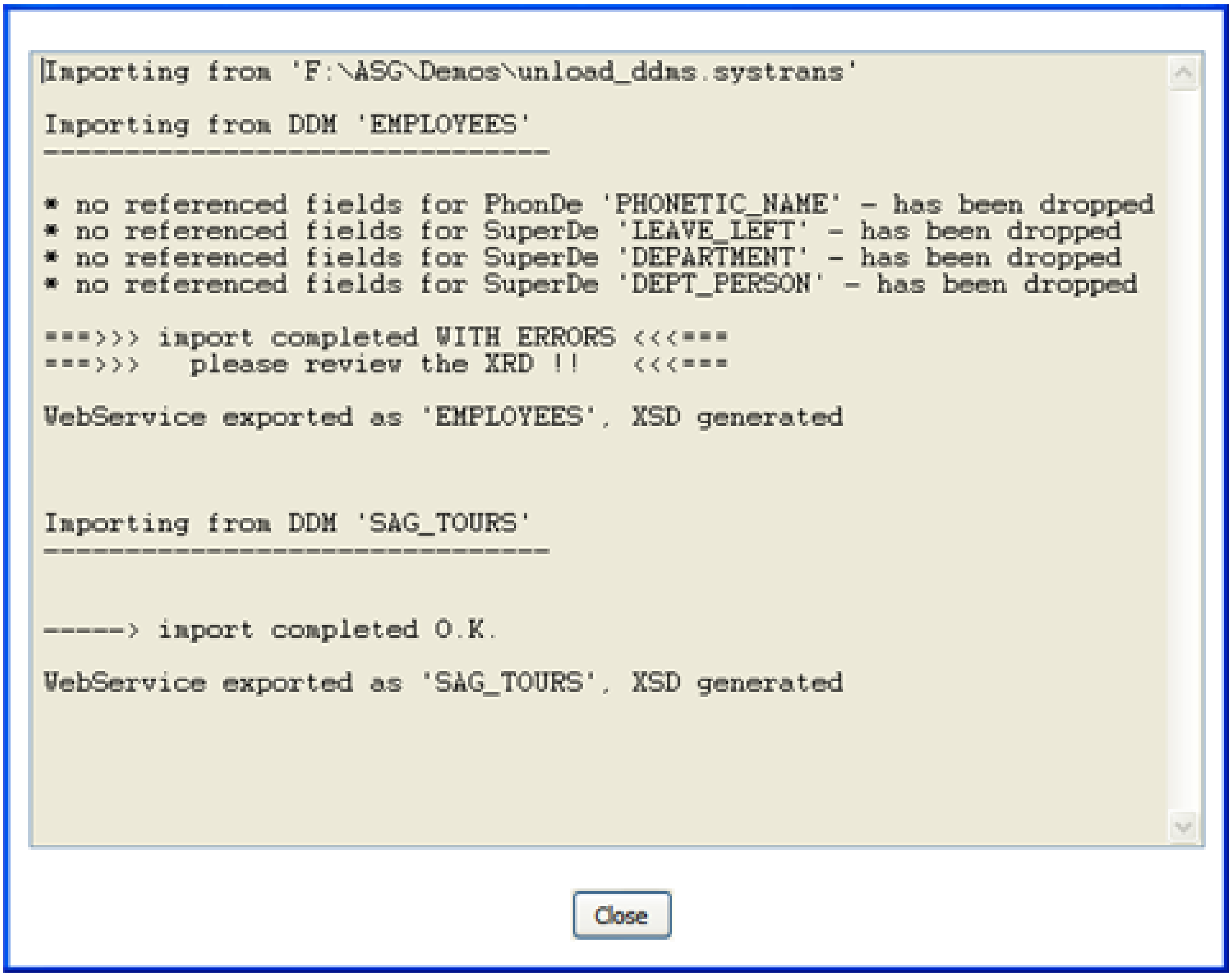
- For each DDM selected, which contains MU and or PE fields, a dialog will request you to specify the maximum number of occurrences expected.

Note: The "MaxOcc" accepted by this dialog is 32767, depending on the ADABAS version the actual limit accepted by the database system may be lower (191).



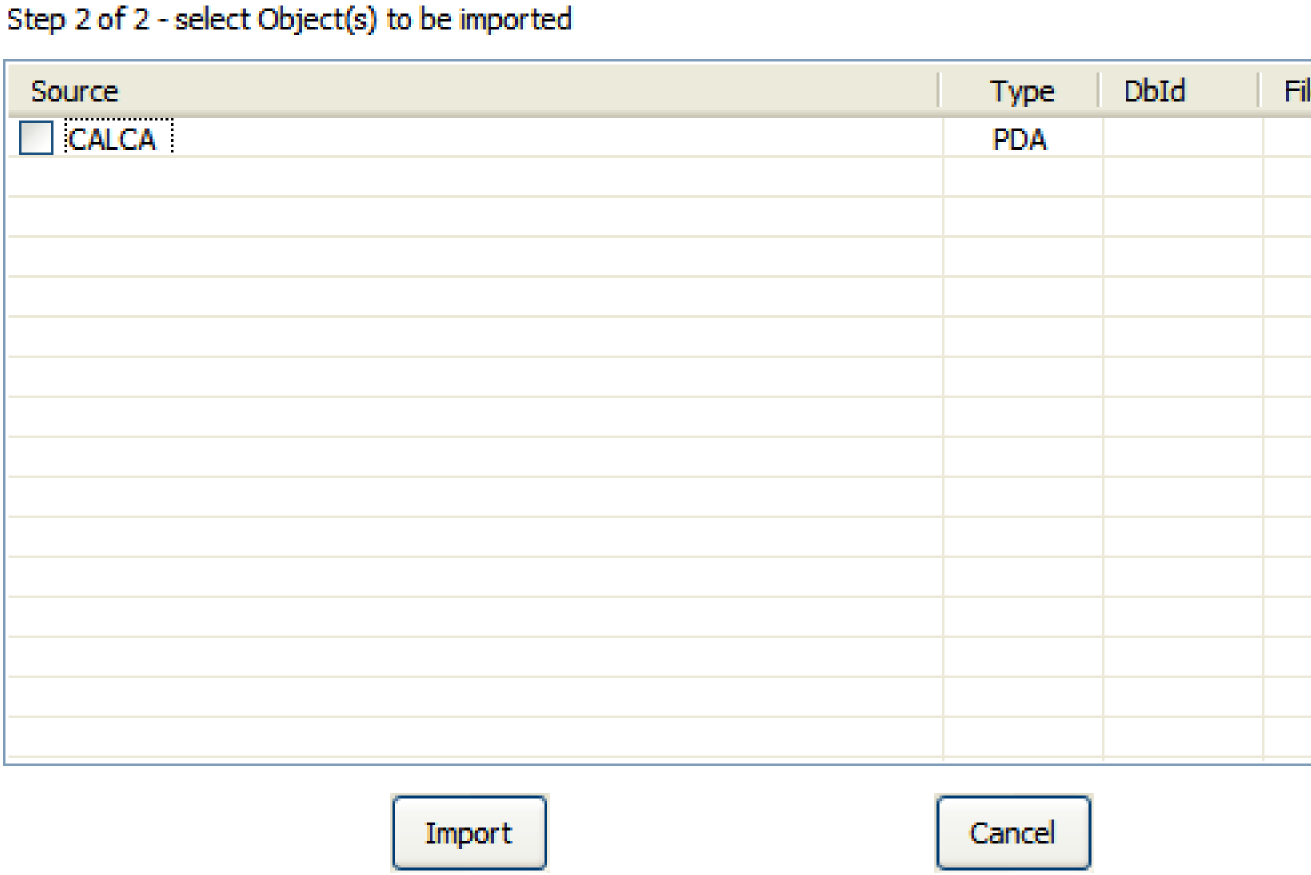
Max...	M..	SN	Field
8	M	AI	ADDRESS_LINE
4	P	AQ	INCOME
8	M	AT	BONUS
4	P	AW	LEAVE_BOOKED
8	M	AZ	LANG
4	P	S3	CURRENCY_SALARY

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



```
Importing from 'F:\ASG\Demos\unload_ddms.systrans'

Importing from DDM 'EMPLOYEES'
--------------------------------

* no referenced fields for PhonDe 'PHONETIC_NAME' - has been dropped
* no referenced fields for SuperDe 'LEAVE_LEFT' - has been dropped
* no referenced fields for SuperDe 'DEPARTMENT' - has been dropped
* no referenced fields for SuperDe 'DEPT_PERSON' - has been dropped

===>>> import completed WITH ERRORS <<<===
===>>>   please review the XRD !!   <<<===

WebService exported as 'EMPLOYEES', XSD generated


Importing from DDM 'SAG_TOURS'
--------------------------------


-----> import completed O.K.

WebService exported as 'SAG_TOURS', XSD generated
```

- In case of a Natural Parameter Data Area (PDA), the type will be shown as "PDA"



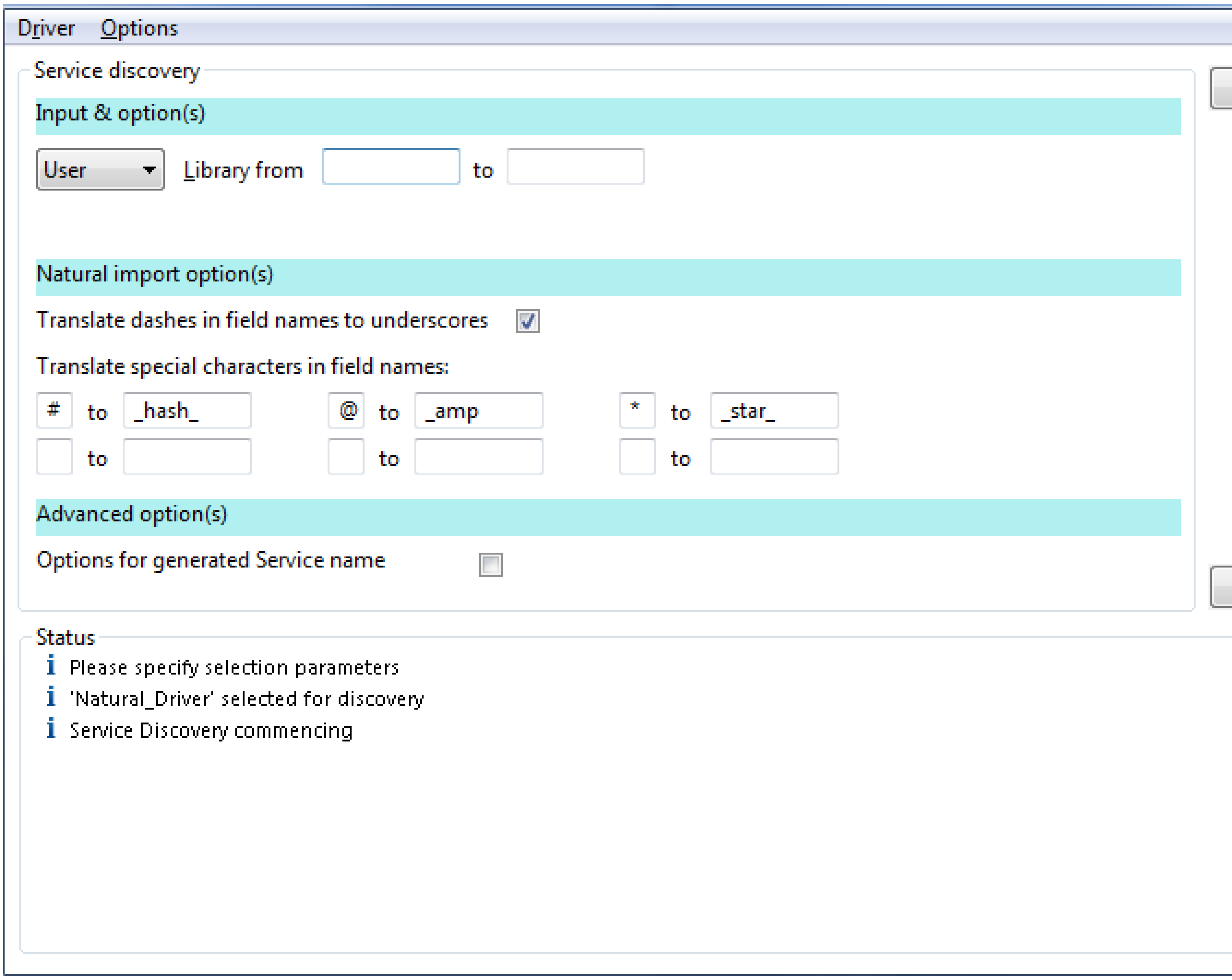
Step 2 of 2 - select Object(s) to be imported

Source	Type	DbId	Fil
□ CALCA	PDA		

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

Creating Service(s) from Natural

- Select a defined driver of type 'Natural' from the list
- Right-click the driver and select *Discover Service(s)*
- Specify the name of the Library to be scanned, select if it is a Natural "User" or a "System" library.



User ▾ Library from ___ to ___

Translate dashes in field names to underscores ☑

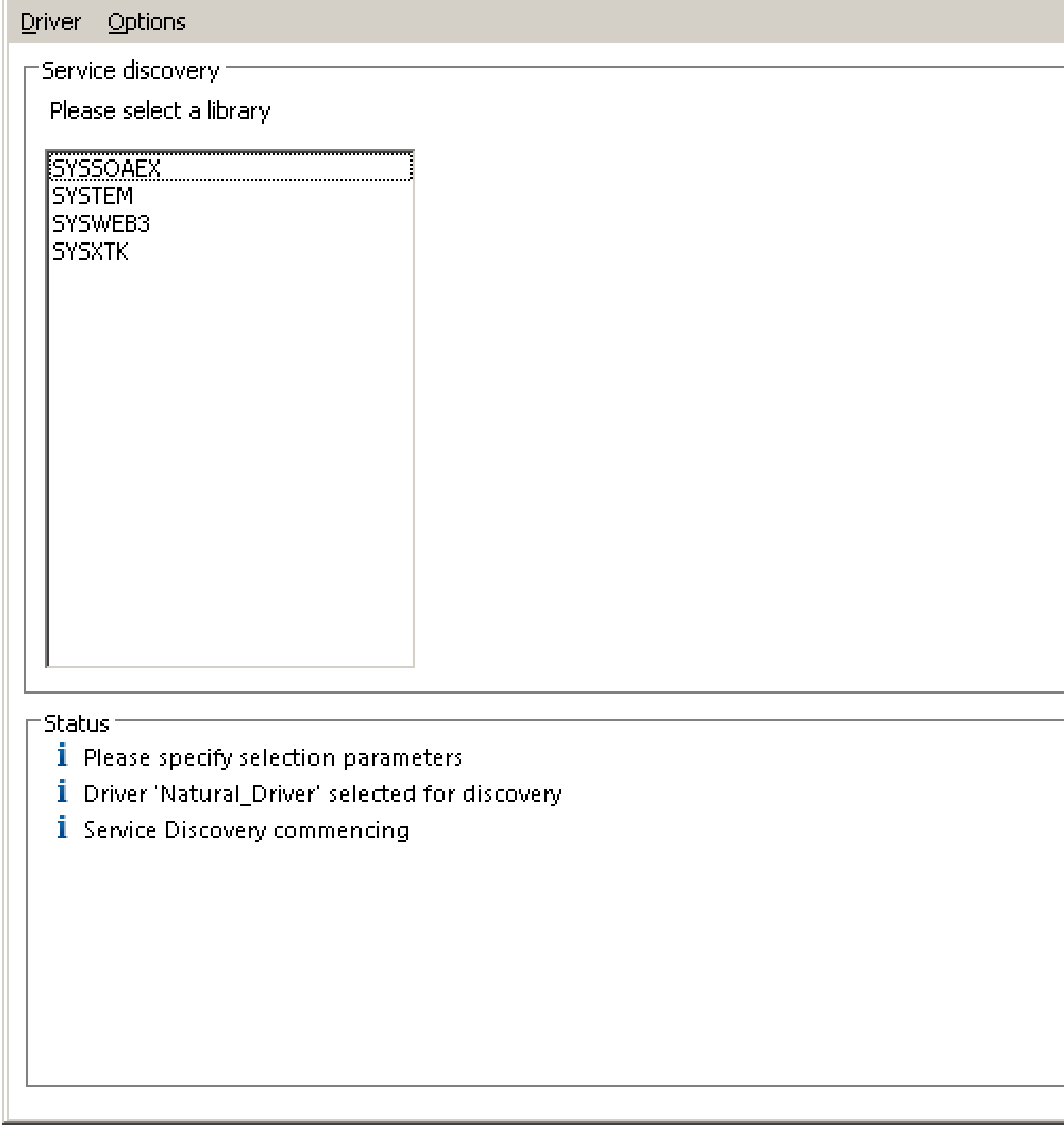
Translate special characters in field names:

#	to	_hash_	@	to	_amp	*	to	_star_
	to			to			to	

Options for generated Service name ☐

ℹ Please specify selection parameters
ℹ 'Natural_Driver' selected for discovery
ℹ Service Discovery commencing

Alternatively, if the library name is unknown, specify a generic library name, or a from-to range and click the **Select...** button. For example, the above input will result in this additional selection dialog to pop up



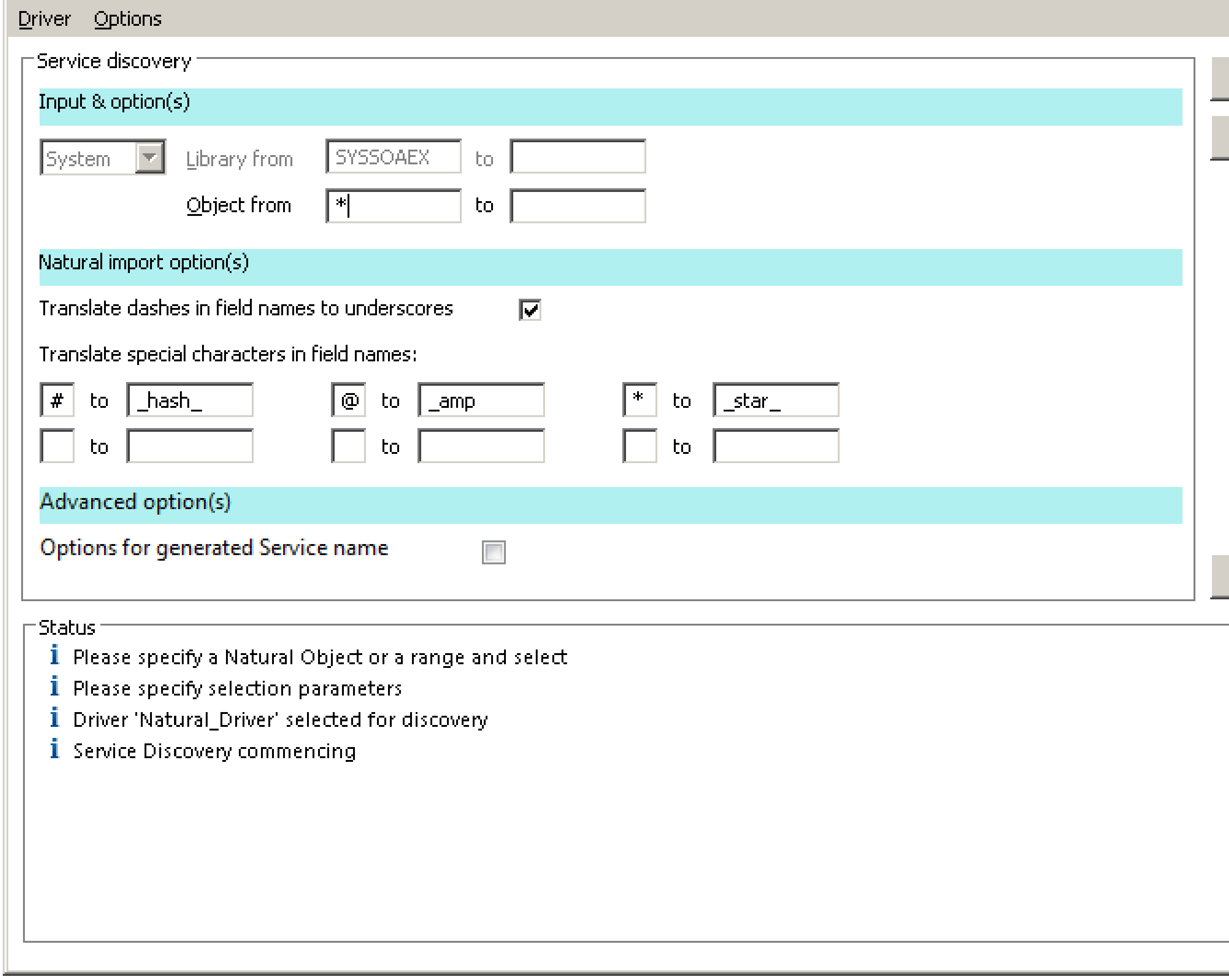
Please select a library

SYSSOAEX
SYSTEM
SYSWEB3
SYSXTK

Please specify selection parameters
Driver 'Natural_Driver' selected for discovery
Service Discovery commencing

Select a library with a double-click, or select a library name and click **Select**, or stop the wizard by clicking the **Exit** button.

- Select the name(s) of the object(s) to be turned into Services, click the **Select...** button. Again, this can be a specific object name, a generic object name, or a range of objects. When a specific object name is entered in the Object "from" field, but the "to" Object name field is left empty, this will result in all objects starting from the specified name being listed



Service discovery

Input & option(s)

System | Library from SYSSOAEX to

Object from * to

Natural import option(s)

Translate dashes in field names to underscores ☑

Translate special characters in field names:

#	to	_hash_	@	to	_amp	*	to	_star_
	to			to			to	

Advanced option(s)

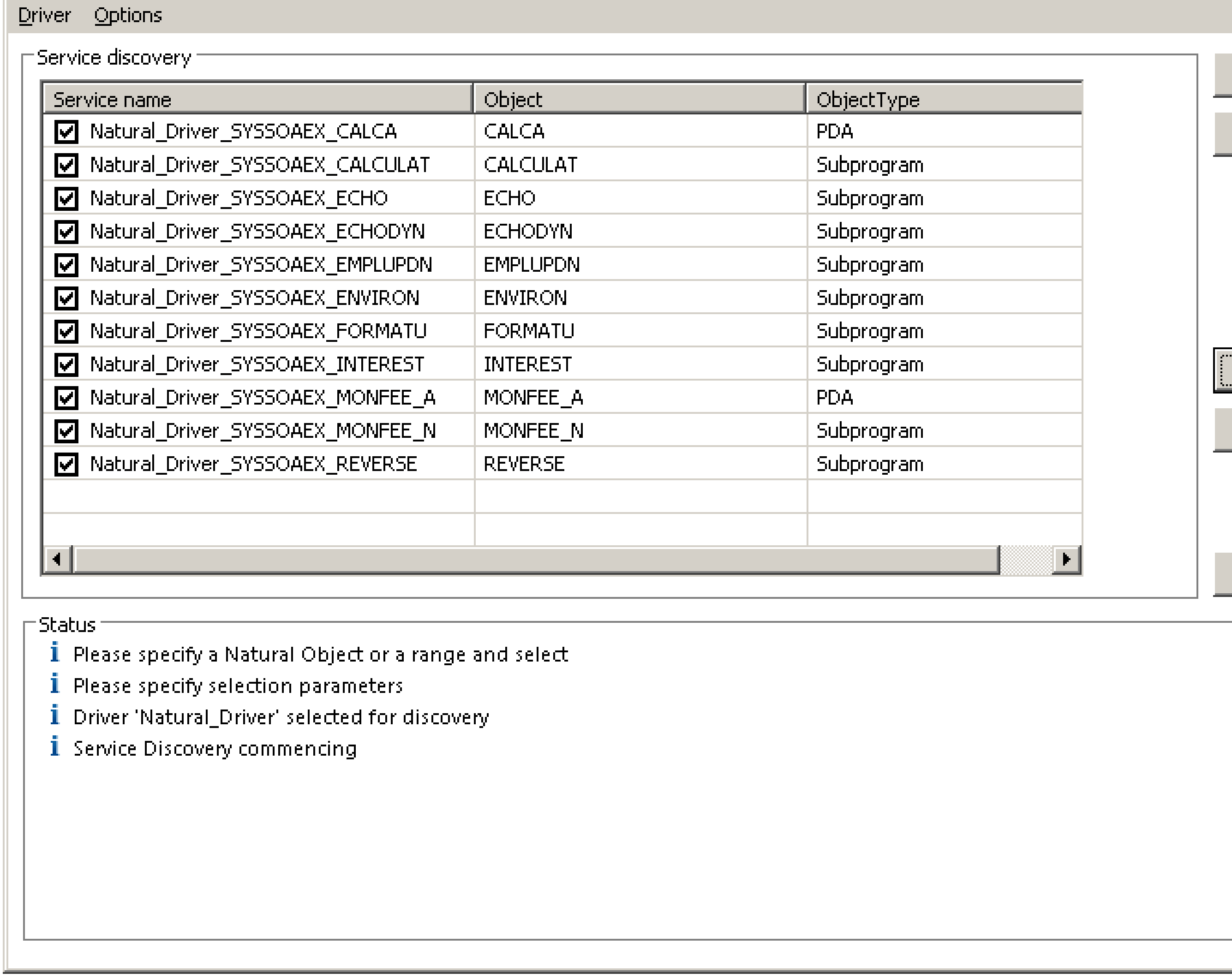
Options for generated Service name □

Status

- Please specify a Natural Object or a range and select
- Please specify selection parameters
- Driver 'Natural_Driver' selected for discovery
- Service Discovery commencing

You may return to Library selection mode by clicking the **Back** button.

- A list of objects matching the selection criteria is displayed



Service name	Object	ObjectType
☑ Natural_Driver_SYSSOAEX_CALCA	CALCA	PDA
☑ Natural_Driver_SYSSOAEX_CALCULAT	CALCULAT	Subprogram
☑ Natural_Driver_SYSSOAEX_ECHO	ECHO	Subprogram
☑ Natural_Driver_SYSSOAEX_ECHODYN	ECHODYN	Subprogram
☑ Natural_Driver_SYSSOAEX_EMPLUPDN	EMPLUPDN	Subprogram
☑ Natural_Driver_SYSSOAEX_ENVIRON	ENVIRON	Subprogram
☑ Natural_Driver_SYSSOAEX_FORMATU	FORMATU	Subprogram
☑ Natural_Driver_SYSSOAEX_INTEREST	INTEREST	Subprogram
☑ Natural_Driver_SYSSOAEX_MONFEE_A	MONFEE_A	PDA
☑ Natural_Driver_SYSSOAEX_MONFEE_N	MONFEE_N	Subprogram
☑ Natural_Driver_SYSSOAEX_REVERSE	REVERSE	Subprogram

```
i Please specify a Natural Object or a range and select
i Please specify selection parameters
i Driver 'Natural_Driver' selected for discovery
i Service Discovery commencing
```

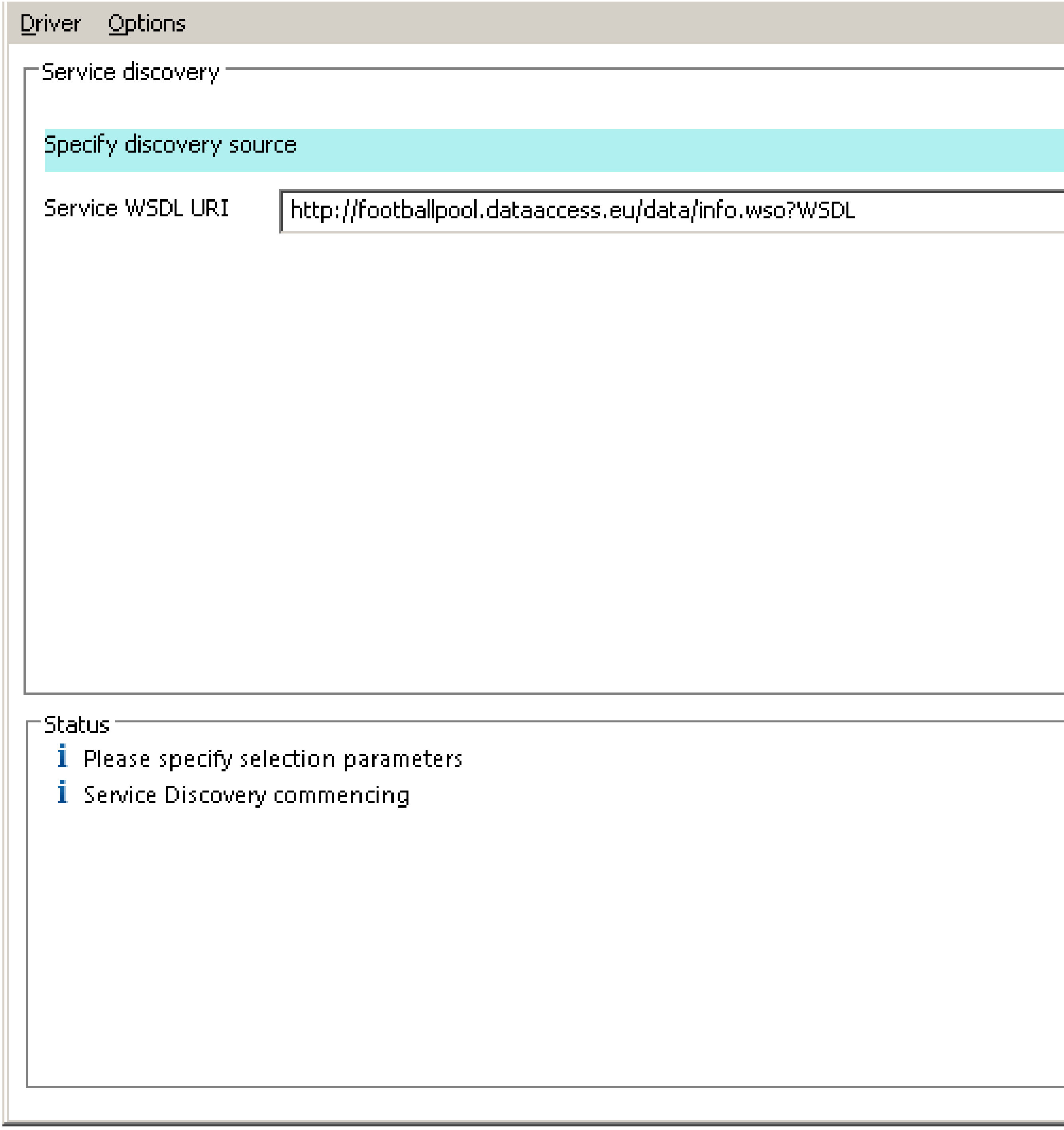
Either select all, or only the one(s) you are specifically interested in, click the **Generate** button.

- The Status pane will report on the progress of each item chosen.

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

Creating Service(s) from 3rd party web services

- Under Services select the Web Services Driver previously added. Right-click and select Discover Service(s).
- Specify the Web Service URI and click the 'Discover' button



Service discovery

Specify discovery source

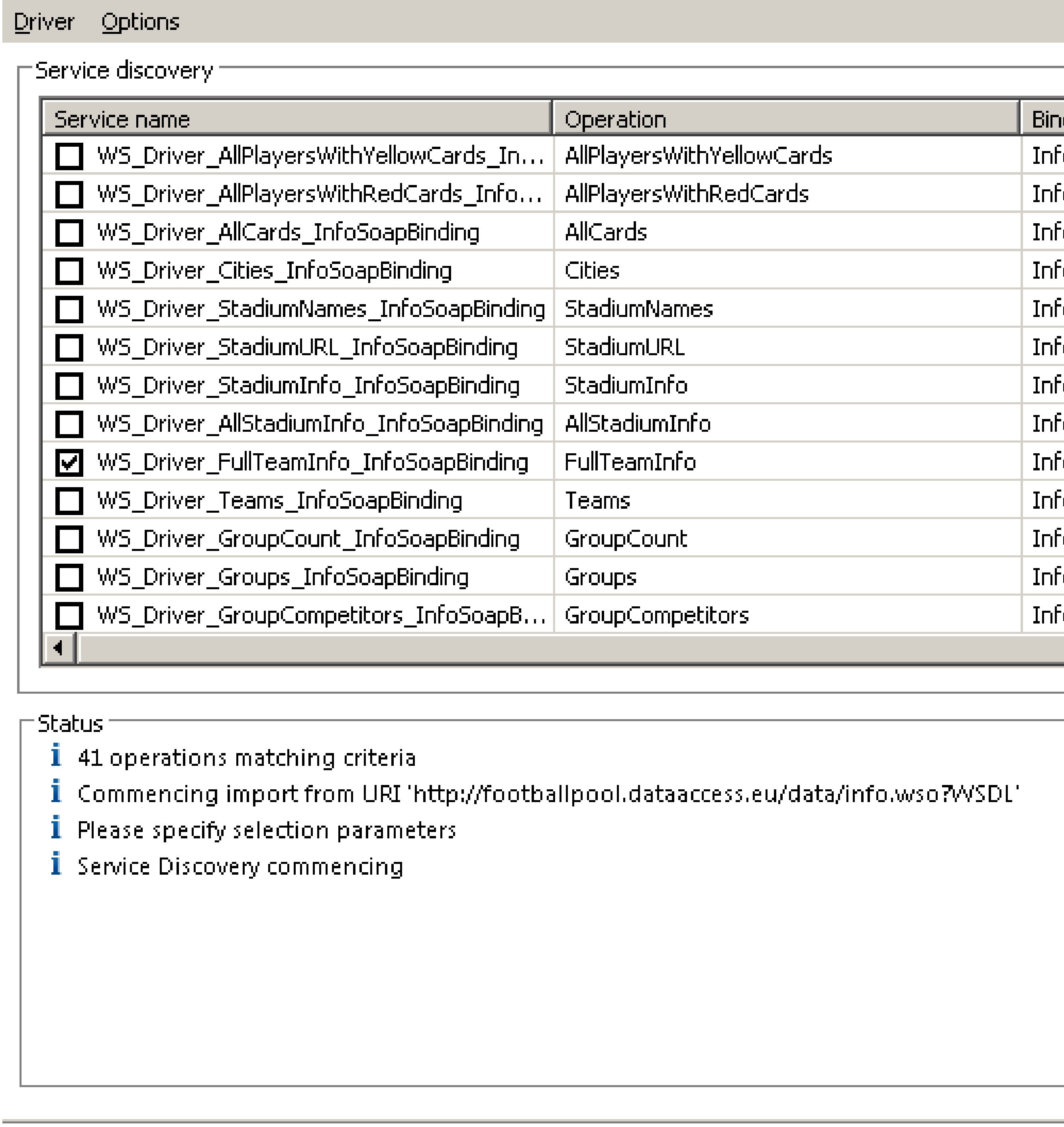
Service WSDL URI: http://footballpool.dataaccess.eu/data/info.wso?WSDL

Status

ℹ Please specify selection parameters

ℹ Service Discovery commencing

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



Service discovery

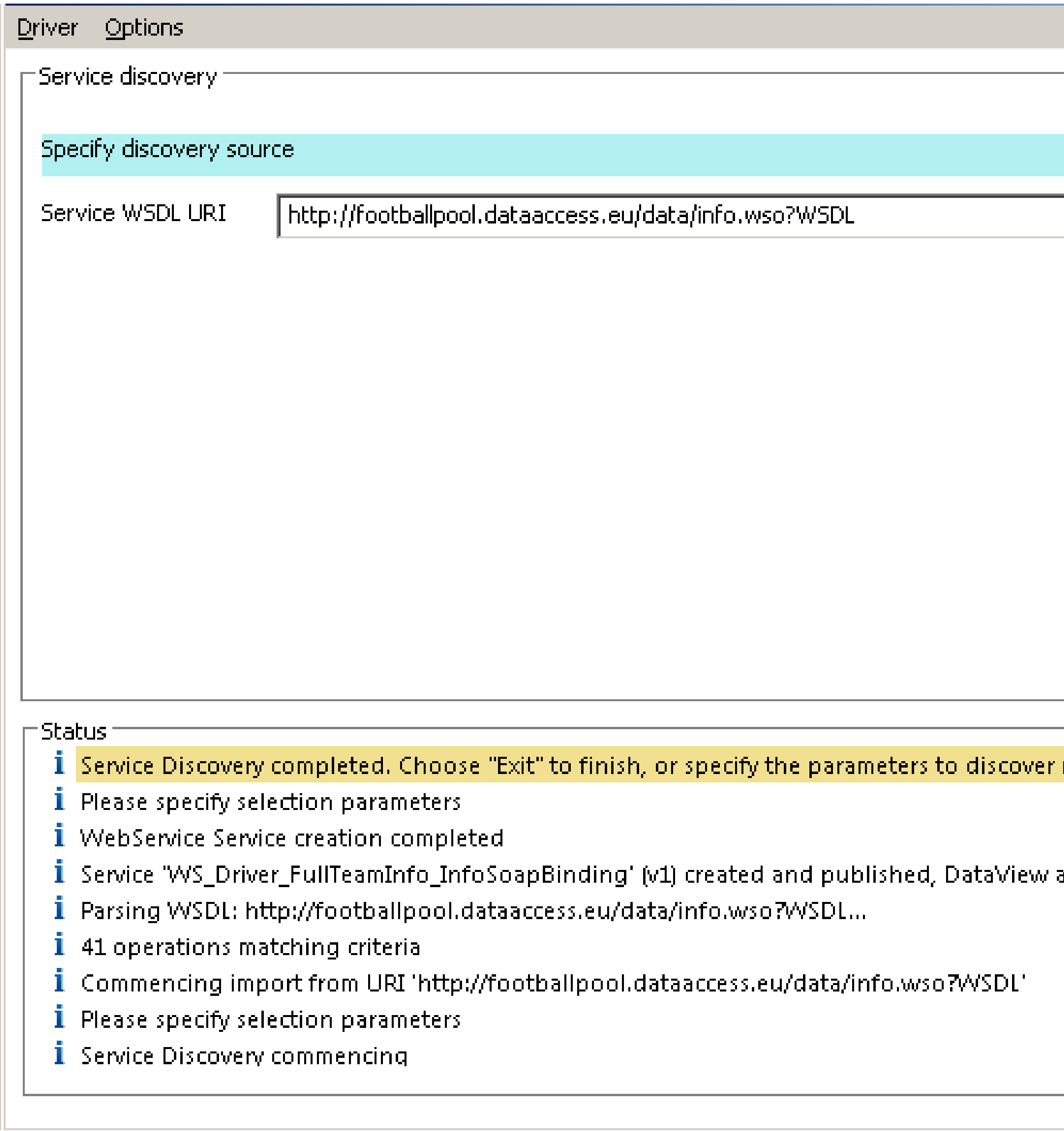
	Service name	Operation	Bin
□	WS_Driver_AllPlayersWithYellowCards_In...	AllPlayersWithYellowCards	Inf
□	WS_Driver_AllPlayersWithRedCards_Info...	AllPlayersWithRedCards	Inf
□	WS_Driver_AllCards_InfoSoapBinding	AllCards	Inf
□	WS_Driver_Cities_InfoSoapBinding	Cities	Inf
□	WS_Driver_StadiumNames_InfoSoapBinding	StadiumNames	Inf
□	WS_Driver_StadiumURL_InfoSoapBinding	StadiumURL	Inf
□	WS_Driver_StadiumInfo_InfoSoapBinding	StadiumInfo	Inf
□	WS_Driver_AllStadiumInfo_InfoSoapBinding	AllStadiumInfo	Inf
☑	WS_Driver_FullTeamInfo_InfoSoapBinding	FullTeamInfo	Inf
□	WS_Driver_Teams_InfoSoapBinding	Teams	Inf
□	WS_Driver_GroupCount_InfoSoapBinding	GroupCount	Inf
□	WS_Driver_Groups_InfoSoapBinding	Groups	Inf
□	WS_Driver_GroupCompetitors_InfoSoapB...	GroupCompetitors	Inf

Status

- i 41 operations matching criteria
- i Commencing import from URI 'http://footballpool.dataaccess.eu/data/info.wso?WSDL'
- i Please specify selection parameters
- i Service Discovery commencing

Use Select all to choose all operations or select on an individual basis. 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.



Service WSDL URI http://footballpool.dataaccess.eu/data/info.wso?WSDL

```
i Service Discovery completed. Choose "Exit" to finish, or specify the parameters to discover
i Please specify selection parameters
i WebService Service creation completed
i Service 'WS_Driver_FullTeamInfo_InfoSoapBinding' (v1) created and published, DataView a
i Parsing WSDL: http://footballpool.dataaccess.eu/data/info.wso?WSDL...
i 41 operations matching criteria
i Commencing import from URI 'http://footballpool.dataaccess.eu/data/info.wso?WSDL'
i Please specify selection parameters
i Service Discovery commencing
```

- 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 the WebService Driver.

Using Portus with ...

This section provides extensive information about how to interact with various data sources using Portus

Adabas	Ultra-high performance, high-availibility transactional database
Natural	Comprehensive high-productivity development environment
Shared Libraries or DLLs	Invoke 3gl programs available as DLLs or shared libraries on any platform

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

Portus Statistics

This section describes the statistics available in Portus

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

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

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

The KPIs gathered for each operation are as follows:

- total_count

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

  This is the accumulated round trip time for the operation.
- high_time

  This is the highest round trip time for the operation.
- low_time

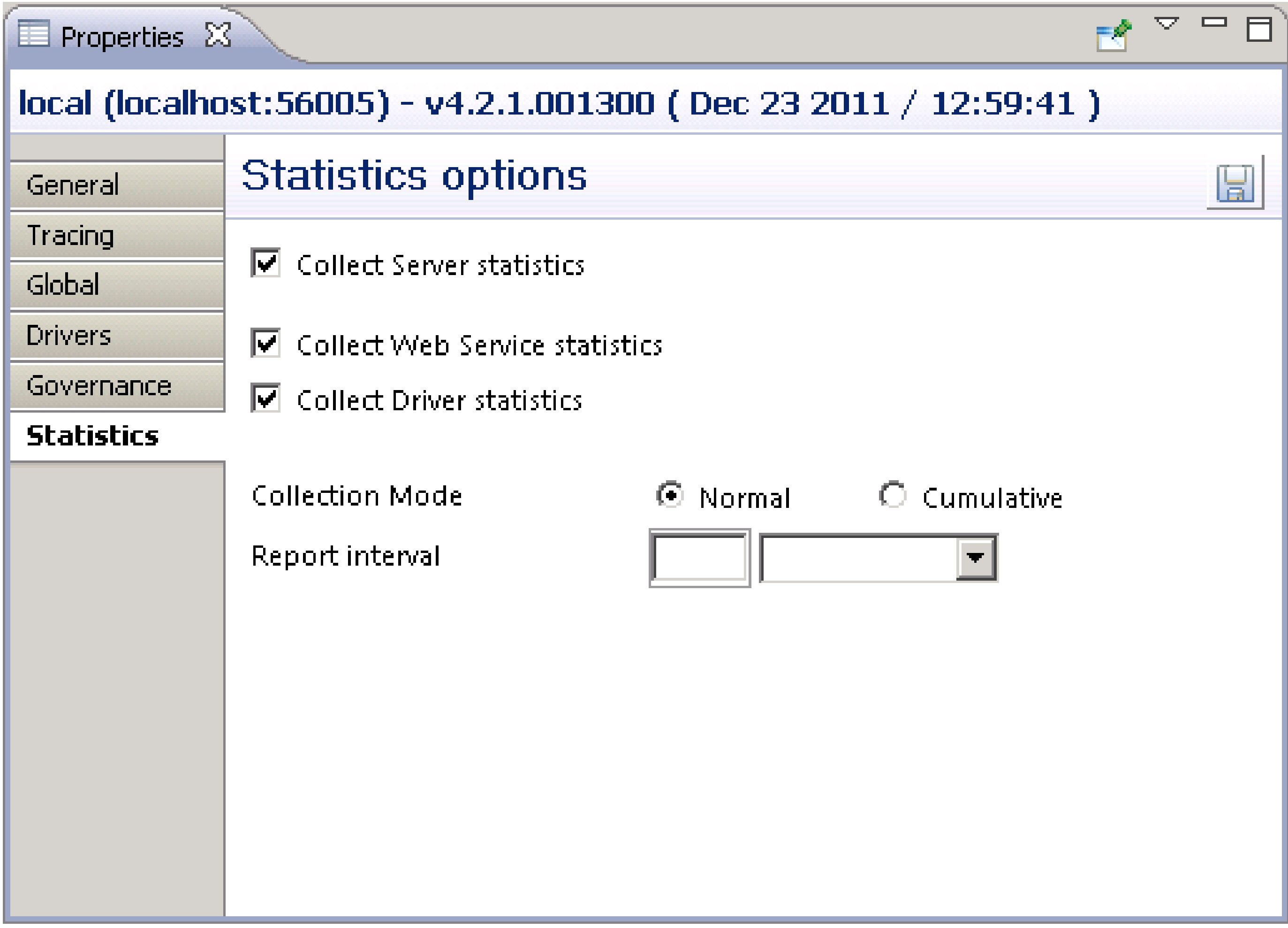
  This is the lowest round trip time for the operation.
- errors_occurred

  This is the number of times the operation has failed.

Server Setup

Setup is carried out using the Portus Control Centre.

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



local (localhost:56005) - v4.2.1.001300 (Dec 23 2011 / 12:59:41)

Statistics options

☑ Collect Server statistics

☑ Collect Web Service statistics

☑ Collect Driver statistics

Collection Mode ● Normal ○ Cumulative

Report interval

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

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

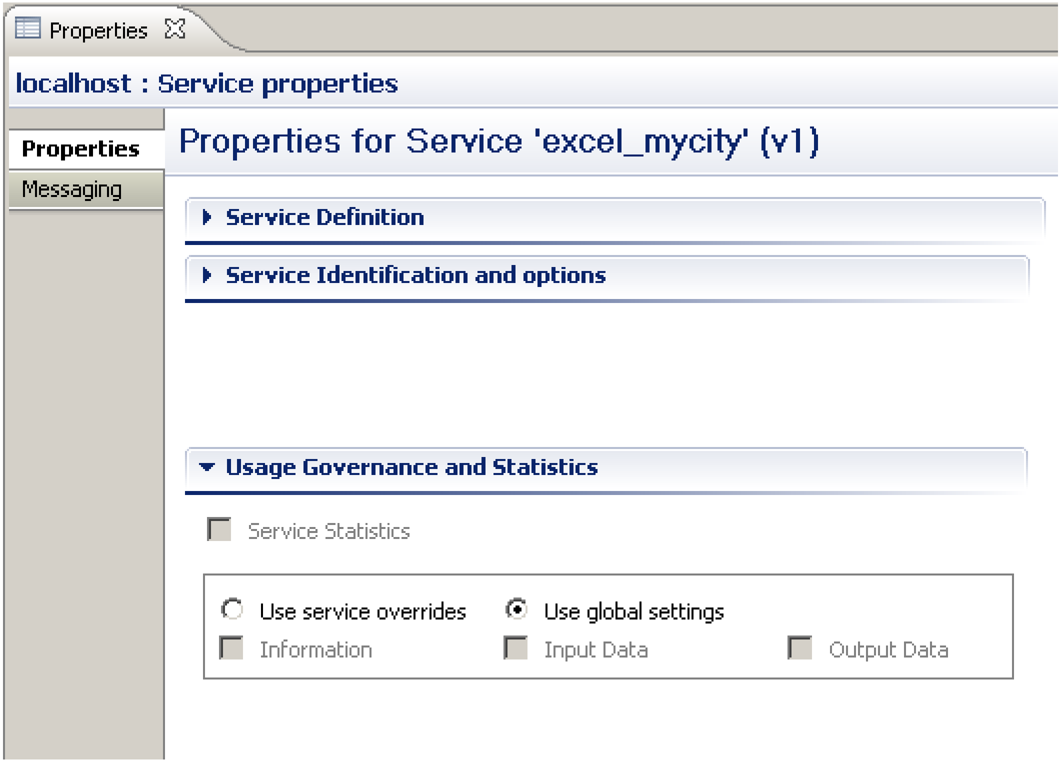
The default behaviour is for a report to be written when the server is stopped. Setting an interval enables reporting at set periods.

Service Setup

Click on the service to be monitored. In the Properties View set Service Statistics on/off.

Note: 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.



Properties

localhost : Service properties

Properties | Messaging

Properties for Service 'excel_mycity' (v1)

▸ Service Definition

▸ Service Identification and options

▾ Usage Governance and Statistics

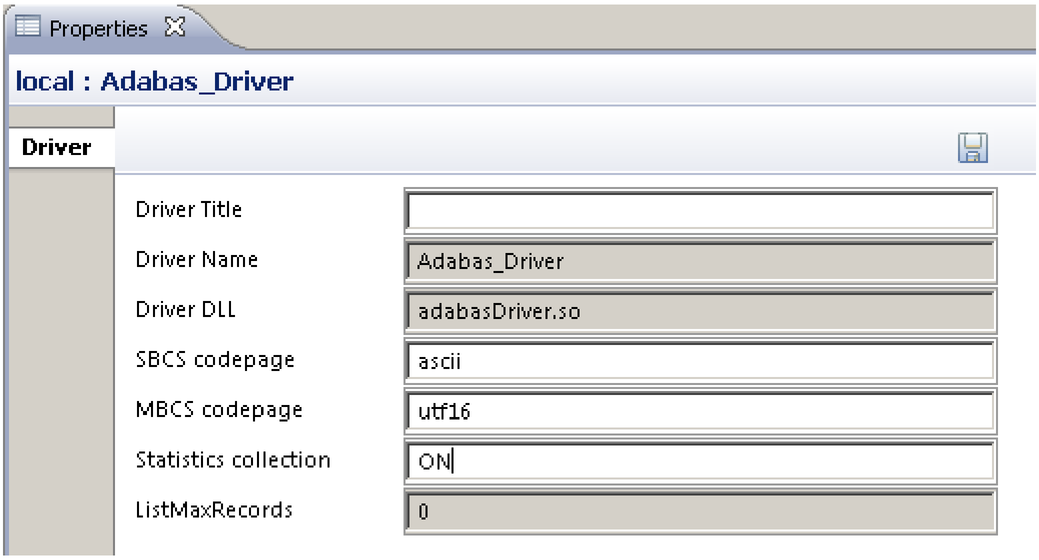
□ Service Statistics

○ Use service overrides ● Use global settings

□ Information □ Input Data □ Output Data

Driver Setup

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



Properties

local : Adabas_Driver

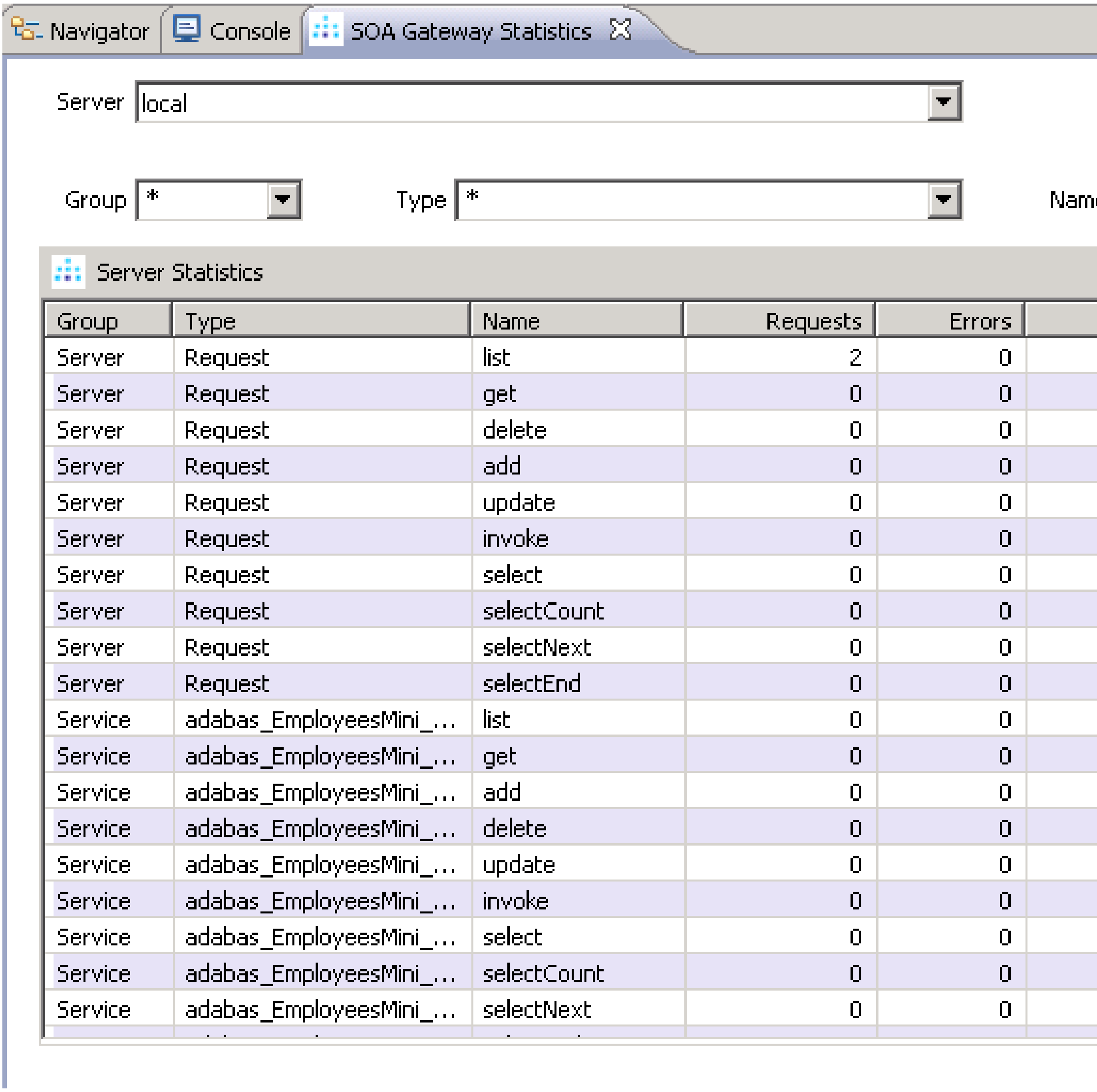
Driver

Driver Title	
Driver Name	Adabas_Driver
Driver DLL	adabasDriver.so
SBCS codepage	ascii
MBCS codepage	utf16
Statistics collection	ON
ListMaxRecords	0

Monitoring

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

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



Server: local

Group: * Type: * Nam

Group	Type	Name	Requests	Errors
Server	Request	list	2	0
Server	Request	get	0	0
Server	Request	delete	0	0
Server	Request	add	0	0
Server	Request	update	0	0
Server	Request	invoke	0	0
Server	Request	select	0	0
Server	Request	selectCount	0	0
Server	Request	selectNext	0	0
Server	Request	selectEnd	0	0
Service	adabas_EmployeesMini_...	list	0	0
Service	adabas_EmployeesMini_...	get	0	0
Service	adabas_EmployeesMini_...	add	0	0
Service	adabas_EmployeesMini_...	delete	0	0
Service	adabas_EmployeesMini_...	update	0	0
Service	adabas_EmployeesMini_...	invoke	0	0
Service	adabas_EmployeesMini_...	select	0	0
Service	adabas_EmployeesMini_...	selectCount	0	0
Service	adabas_EmployeesMini_...	selectNext	0	0

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_Driv

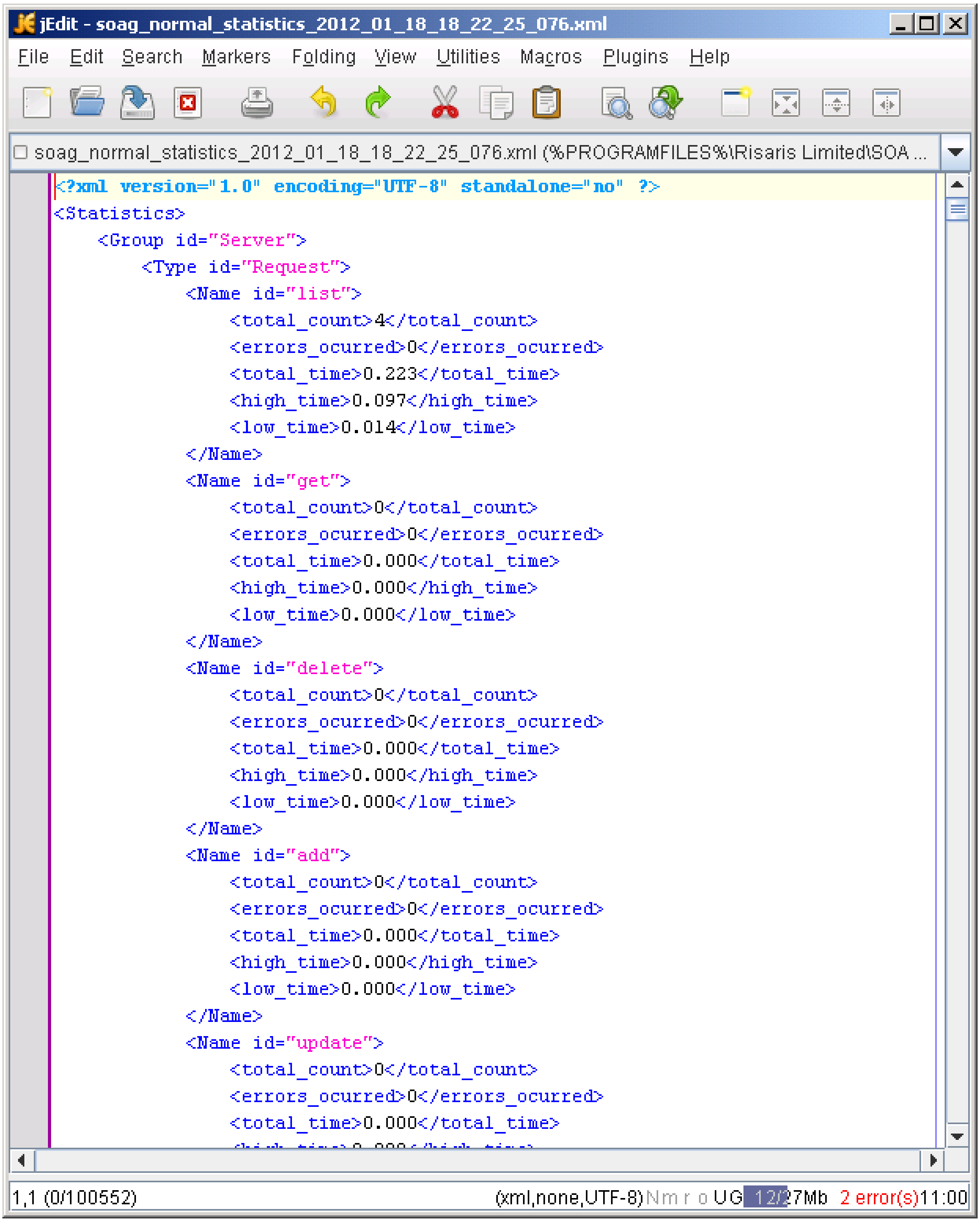
Note that the wildcarding is asterisk only i.e. Adabas* is not acceptable.

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:

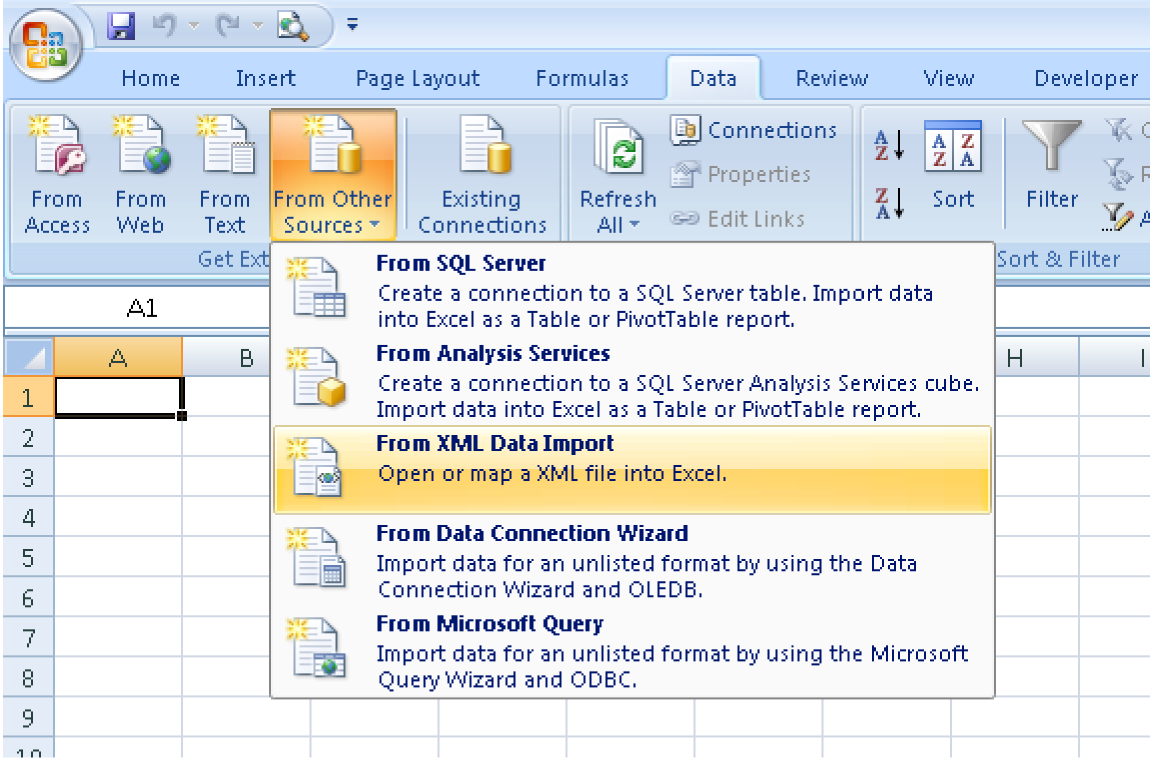

jEdit - soag_normal_statistics_2012_01_18_18_22_25_076.xml
File Edit Search Markers Folding View Utilities Macros Plugins Help
soag_normal_statistics_2012_01_18_18_22_25_076.xml (%PROGRAMFILES%\Risaris Limited\SOA ...
1,1 (0/100552)
(xml,none,UTF-8) Nm r o UG 12/27Mb 2 error(s) 11:00

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

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

XML

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



From SQL Server
Create a connection to a SQL Server table. Import data into Excel as a Table or PivotTable report.

From Analysis Services
Create a connection to a SQL Server Analysis Services cube. Import data into Excel as a Table or PivotTable report.

From XML Data Import
Open or map a XML file into Excel.

From Data Connection Wizard
Import data for an unlisted format by using the Data Connection Wizard and OLEDB.

From Microsoft Query
Import data for an unlisted format by using the Microsoft Query Wizard and ODBC.

resulting in a very readable format:

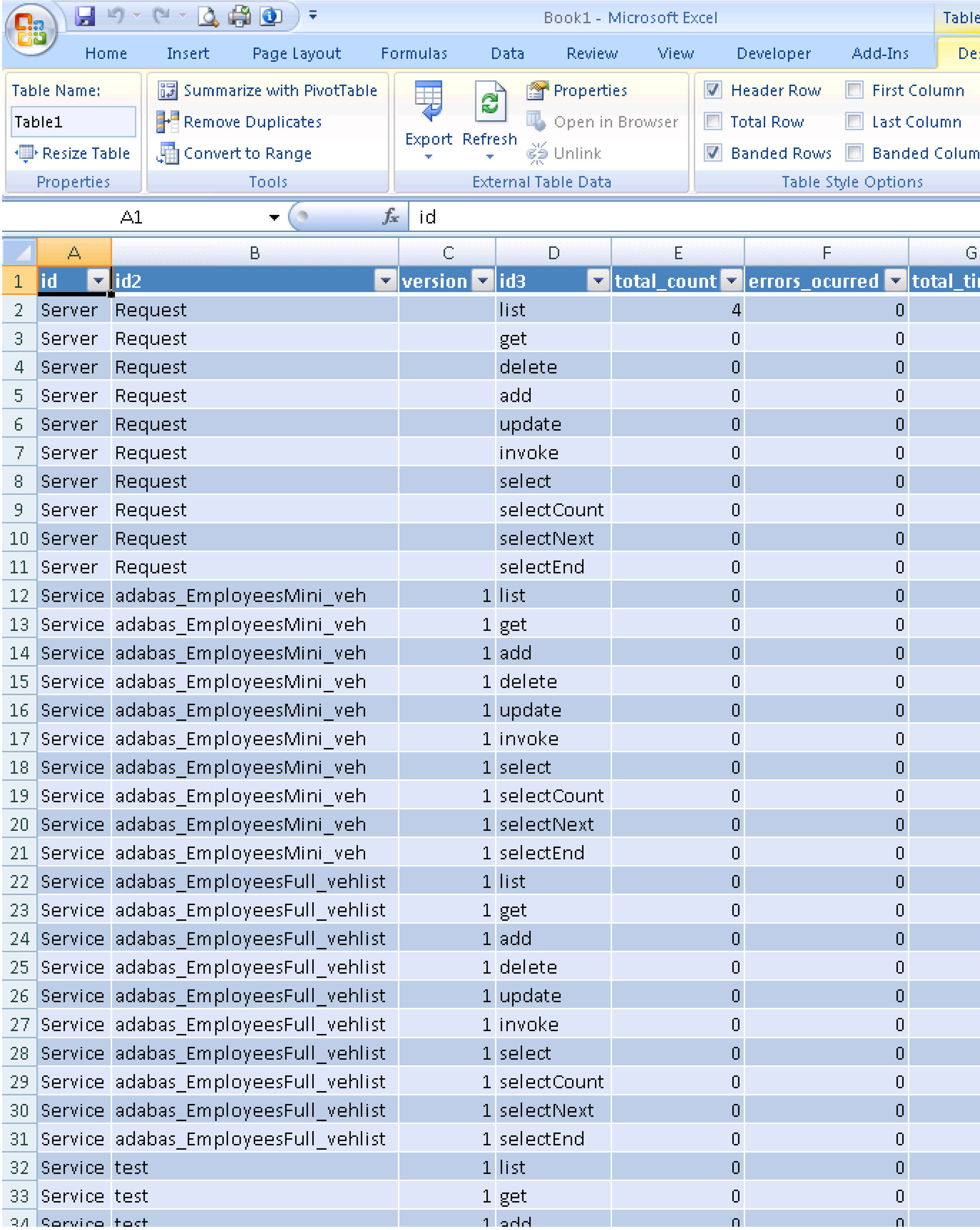


Table Name: Table1

☑ Header Row ☐ First Column
☐ Total Row ☐ Last Column
☑ Banded Rows ☐ Banded Colum

A1 id

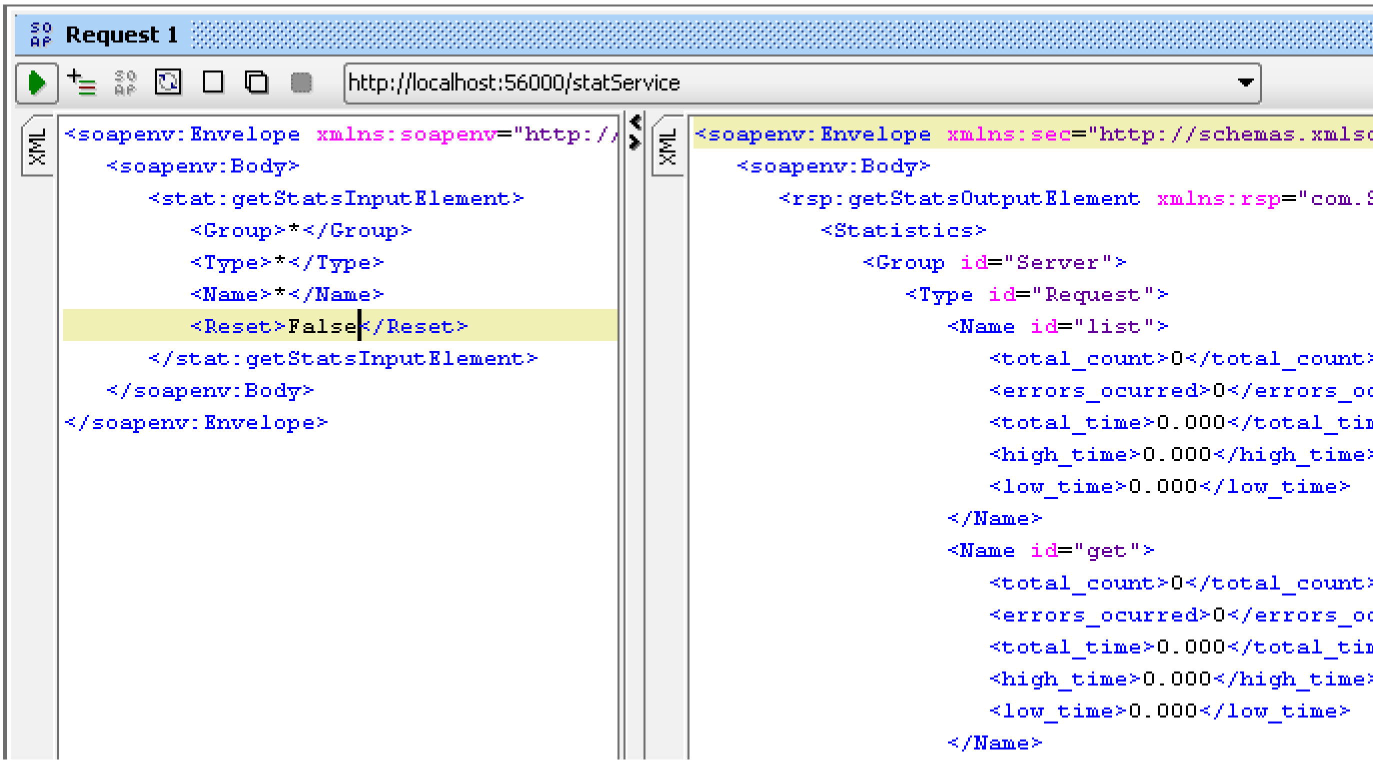
	A	B	C	D	E	F	G
1	id	id2	version	id3	total_count	errors_ocurred	total_ti
2	Server	Request		list	4	0	
3	Server	Request		get	0	0	
4	Server	Request		delete	0	0	
5	Server	Request		add	0	0	
6	Server	Request		update	0	0	
7	Server	Request		invoke	0	0	
8	Server	Request		select	0	0	
9	Server	Request		selectCount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



http://localhost:56000/statService

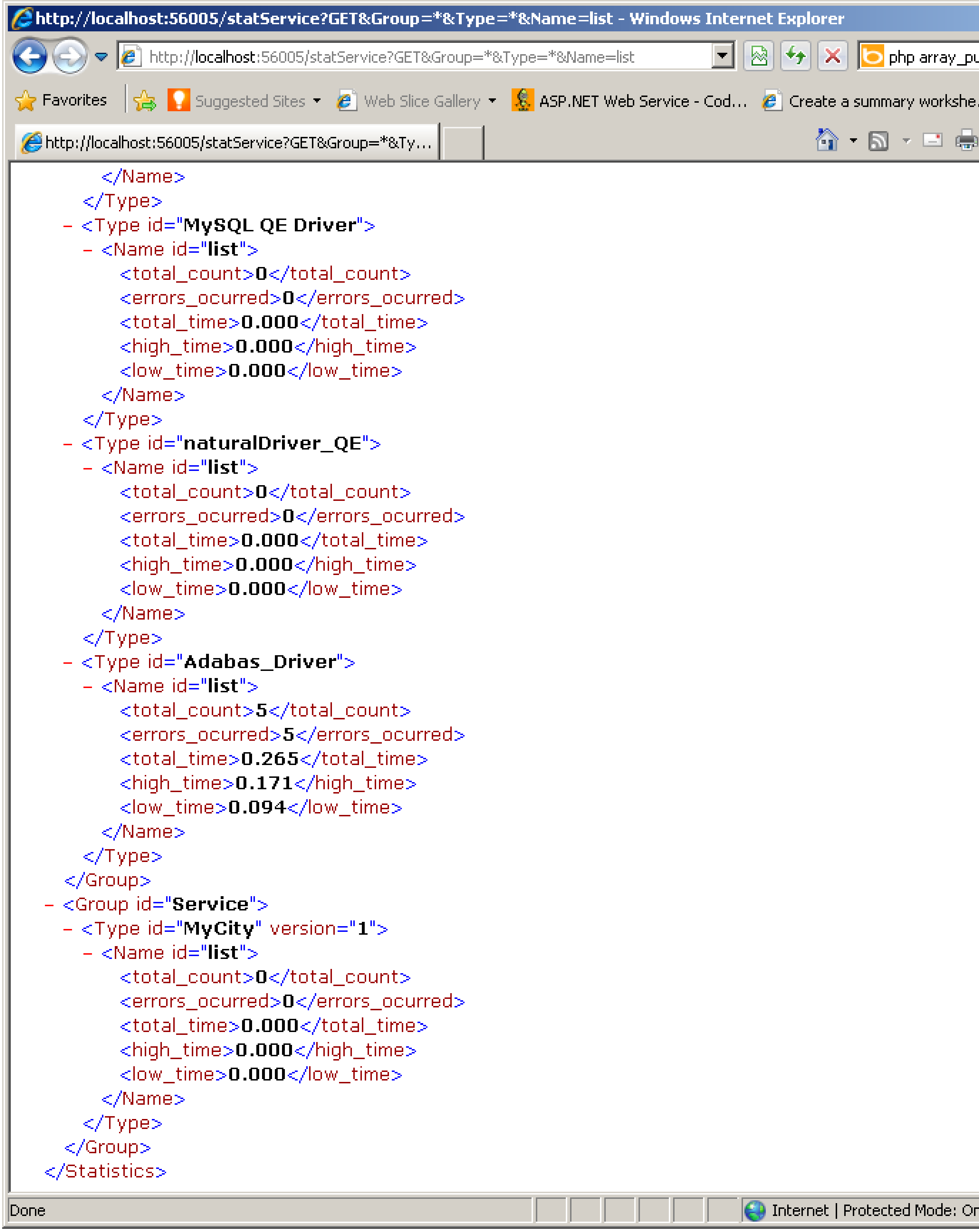
```
<soapenv:Envelope xmlns:soapenv="http://
   <soapenv:Body>
      <stat:getStatsInputElement>
         <Group>*</Group>
         <Type>*</Type>
         <Name>*</Name>
         <Reset>False</Reset>
      </stat:getStatsInputElement>
   </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:sec="http://schemas.xmlsc
   <soapenv:Body>
      <rsp:getStatsOutputElement xmlns:rsp="com.S
         <Statistics>
            <Group id="Server">
               <Type id="Request">
                  <Name id="list">
                     <total_count>0</total_count>
                     <errors_ocurred>0</errors_oc
                     <total_time>0.000</total_tin
                     <high_time>0.000</high_time>
                     <low_time>0.000</low_time>
                  </Name>
                  <Name id="get">
                     <total_count>0</total_count>
                     <errors_ocurred>0</errors_oc
                     <total_time>0.000</total_tin
                     <high_time>0.000</high_time>
                     <low_time>0.000</low_time>
                  </Name>
```

URL

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

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


http://localhost:56005/statService?GET&Group=*&Type=*&Name=list - Windows Internet Explorer
http://localhost:56005/statService?GET&Group=*&Type=*&Name=list
php array_pu
Favorites
Suggested Sites
Web Slice Gallery
ASP.NET Web Service - Cod...
Create a summary workshe
http://localhost:56005/statService?GET&Group=*&Ty...
Done
Internet | Protected Mode: On

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

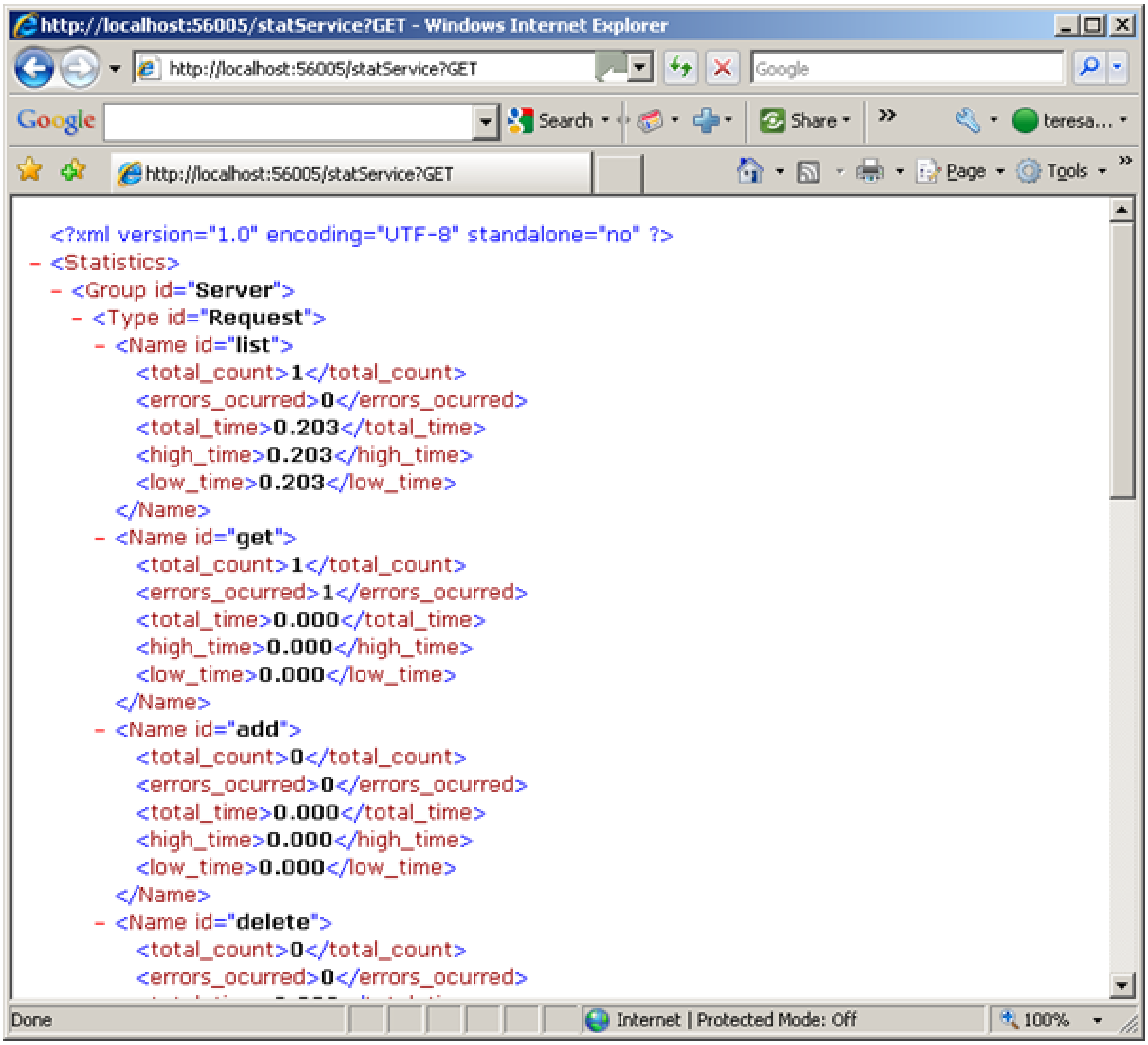
Example

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

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

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

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

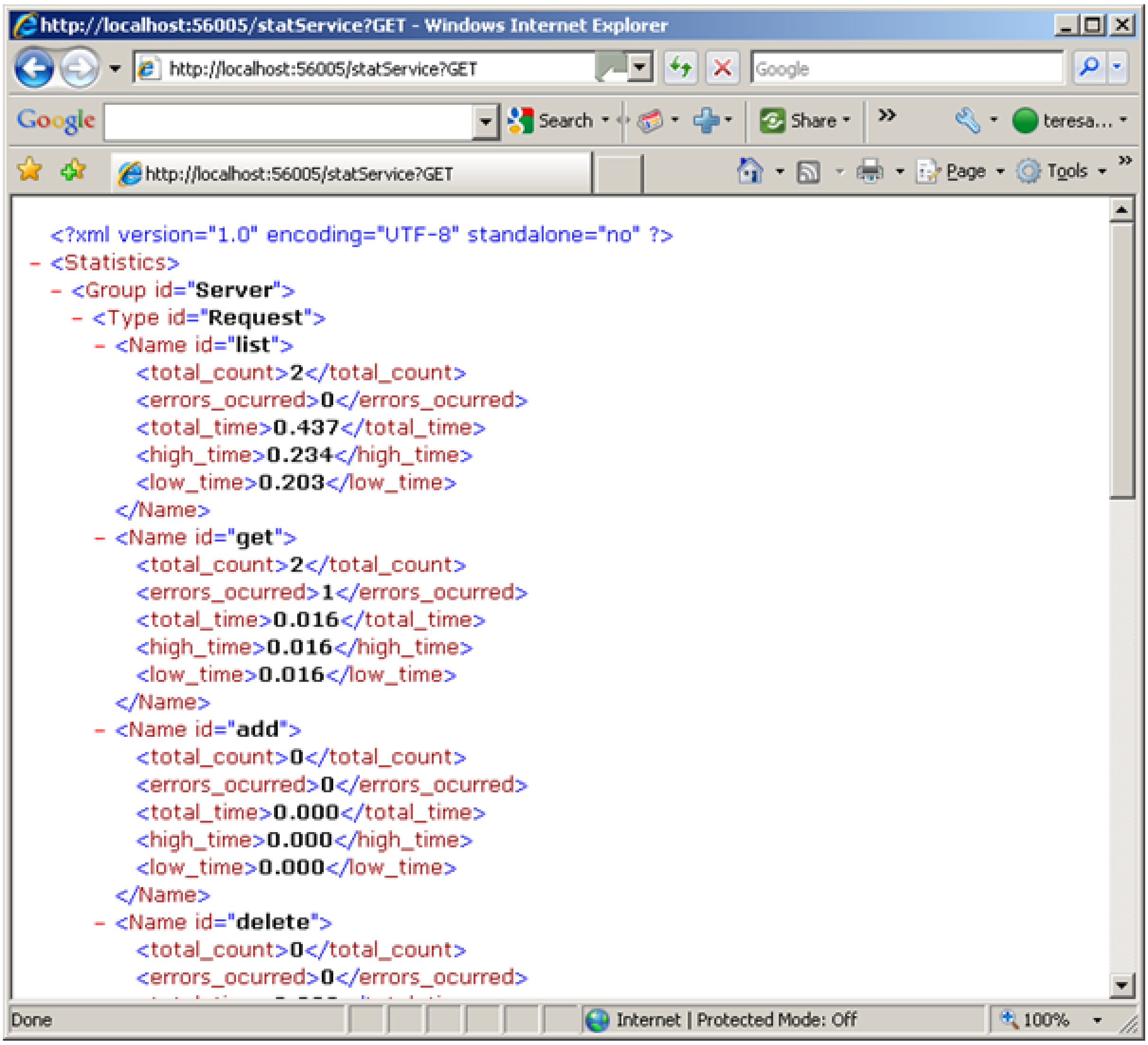

http://localhost:56005/statService?GET - Windows Internet Explorer
http://localhost:56005/statService?GET
Google
Google
Search
Share
teresa...
http://localhost:56005/statService?GET
Page
Tools
Done
Internet | Protected Mode: Off
100%

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

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

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

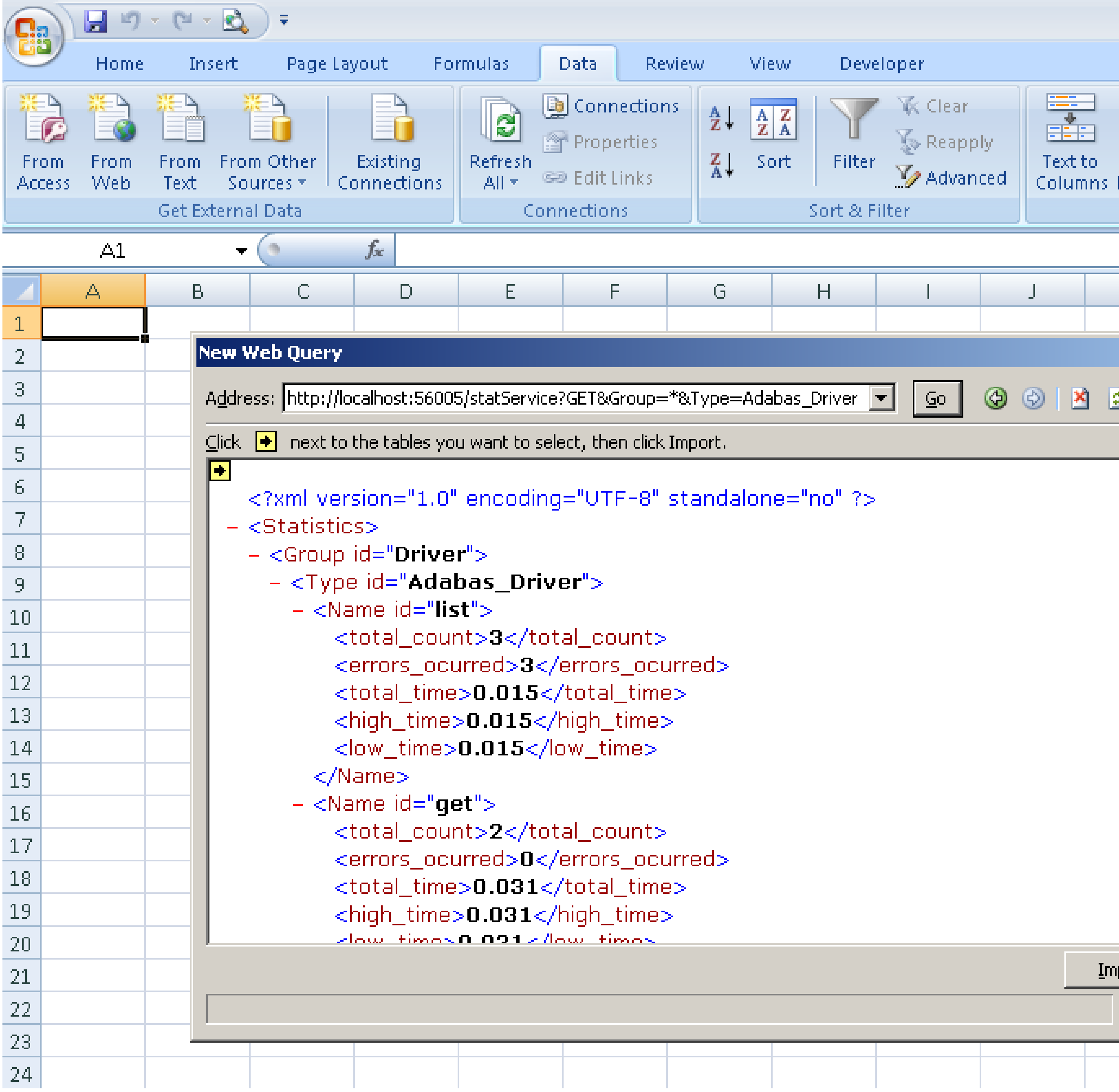
Note the now successful entry for the get operation. Note also that for list the high time is now 0.234 as the query took slightly longer. As there have been only 2 list requests so far the total time equals the sum of low and high times.


http://localhost:56005/statService?GET - Windows Internet Explorer
http://localhost:56005/statService?GET
Google
Search
Share
teresa...
Page
Tools
Done
Internet | Protected Mode: Off
100%

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

Service Import

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



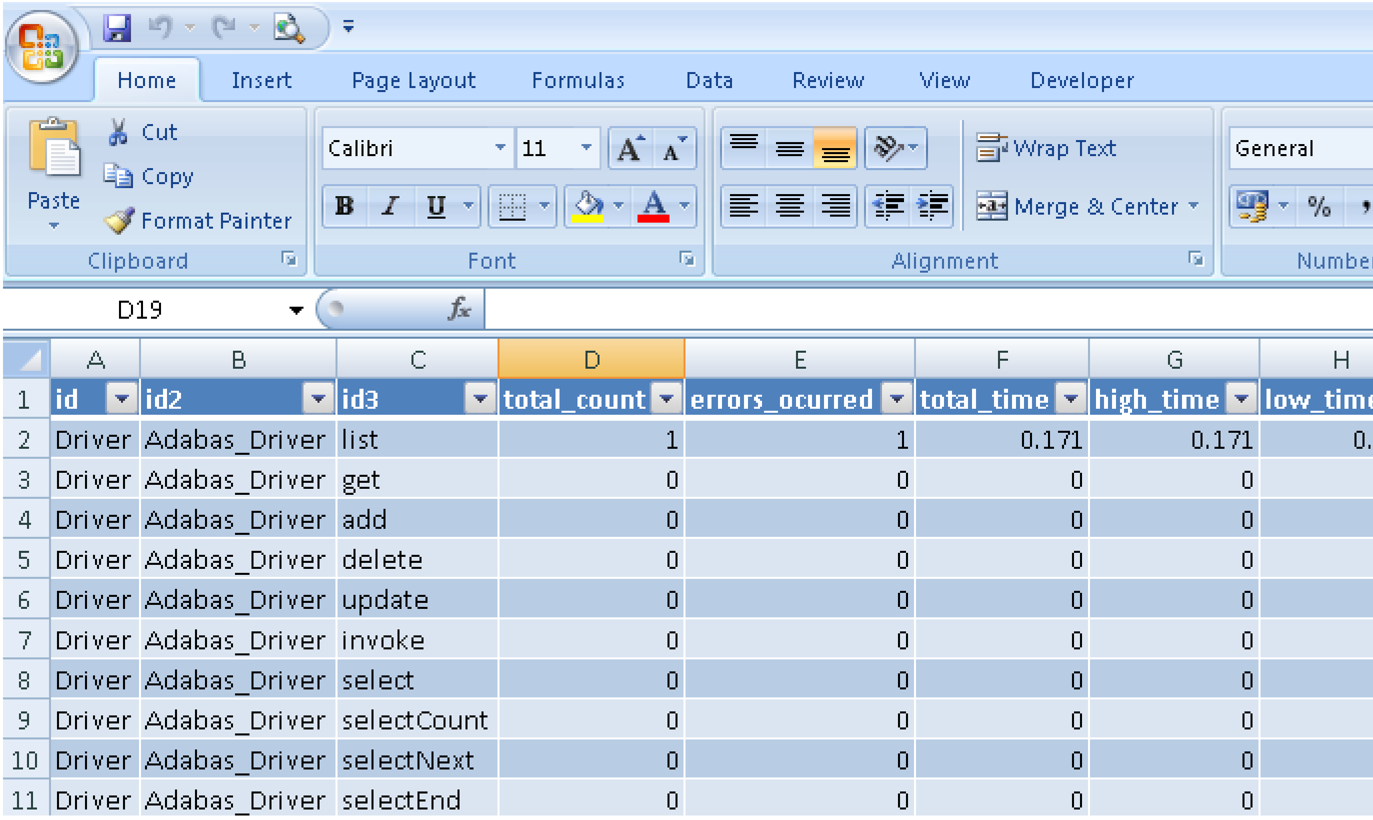
New Web Query

Address: http://localhost:56005/statService?GET&Group=*&Type=Adabas_Driver

Click next to the tables you want to select, then click Import.

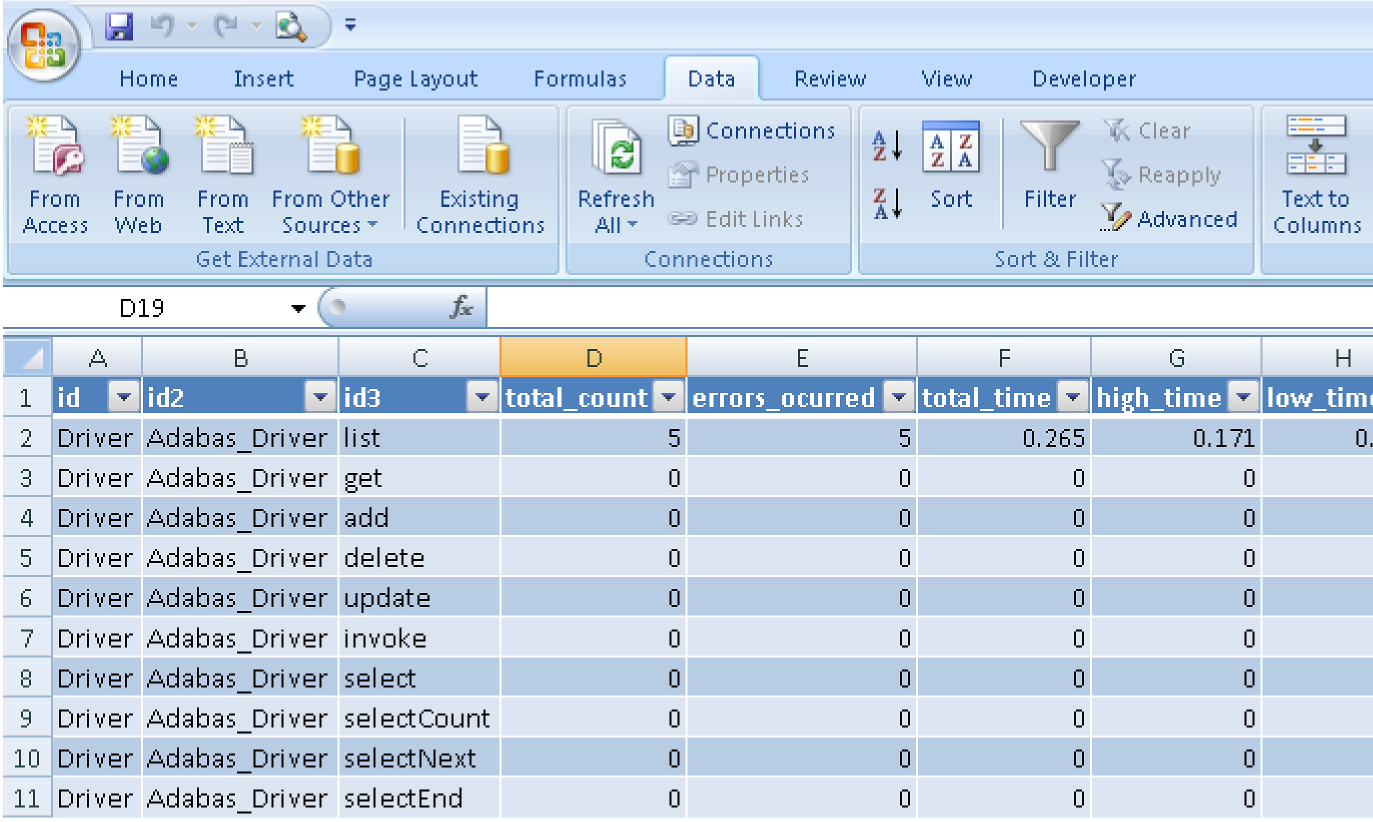
```
  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Statistics>
  - <Group id="Driver">
    - <Type id="Adabas_Driver">
      - <Name id="list">
          <total_count>3</total_count>
          <errors_ocurred>3</errors_ocurred>
          <total_time>0.015</total_time>
          <high_time>0.015</high_time>
          <low_time>0.015</low_time>
        </Name>
      - <Name id="get">
          <total_count>2</total_count>
          <errors_ocurred>0</errors_ocurred>
          <total_time>0.031</total_time>
          <high_time>0.031</high_time>
          <low_time>0.031</low_time>
```

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_tim
2	Driver	Adabas_Driver	list	1	1	0.171	0.171	0.
3	Driver	Adabas_Driver	get	0	0	0	0	
4	Driver	Adabas_Driver	add	0	0	0	0	
5	Driver	Adabas_Driver	delete	0	0	0	0	
6	Driver	Adabas_Driver	update	0	0	0	0	
7	Driver	Adabas_Driver	invoke	0	0	0	0	
8	Driver	Adabas_Driver	select	0	0	0	0	
9	Driver	Adabas_Driver	selectCount	0	0	0	0	
10	Driver	Adabas_Driver	selectNext	0	0	0	0	
11	Driver	Adabas_Driver	selectEnd	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_tim
2	Driver	Adabas_Driver	list	5	5	0.265	0.171	0.
3	Driver	Adabas_Driver	get	0	0	0	0	
4	Driver	Adabas_Driver	add	0	0	0	0	
5	Driver	Adabas_Driver	delete	0	0	0	0	
6	Driver	Adabas_Driver	update	0	0	0	0	
7	Driver	Adabas_Driver	invoke	0	0	0	0	
8	Driver	Adabas_Driver	select	0	0	0	0	
9	Driver	Adabas_Driver	selectCount	0	0	0	0	
10	Driver	Adabas_Driver	selectNext	0	0	0	0	
11	Driver	Adabas_Driver	selectEnd	0	0	0	0	

Governance

There are 2 ways to apply goverance to Portus

Usage Governance

Once a service is deployed and operational, it may be necessary to govern its usage. This section describes this functionality and how it works in Portus

Introduction

When a service has been created and is deployed, there are a number of things that may need to be understood about the usage of service.

- For audit purposes, it may be necessary to know who has used a specific service and when.
- It may be necessary to log the data provided to the service and the data returned by the service to record what the user did
- It may be necessary to understand who is using a service on specific dates or even within a time range.
- When a service has been deprecated (i.e. replaced by a newer version of the service), it is necessary to understand who is still using the service so that they may be upgraded to the newer level of the service and the older version retired.
- For capacity planning purposes, it is necessary to understand how often a service is being called.
- In order to comply with Service Level Agreements (SLAs), the behaviour of the service under normal circumstances must be understood so that a commitment can be made to a SLA and it can be detected when a service is performing outside its normal band.

Governance Data

The following describes the information that is available to enable an organization to properly govern their various services based on their core software assets.

Version

The web service version.

Status

The web service status i.e. Test, Frozen, Deprecated or Historical.

Identity

The user calling the web service.

Time

Time is recorded at various stages of the service invocation e.g. the time the server received the request, the time the input was processed, the time the output was processed and the time the request was returned.

Location

This records the IP address from where the request originated and the IP address of the server.

Operation

This records the HTTP SOAP method and the request operation.

Data

This records the request input data and the request output data.

Service collection

Portus can be configured to collect data or not at the service name level. This represents the logical name of the service which may have different versions of the service active at any one time in a given Portus server instance. The following information may be collected about each service.

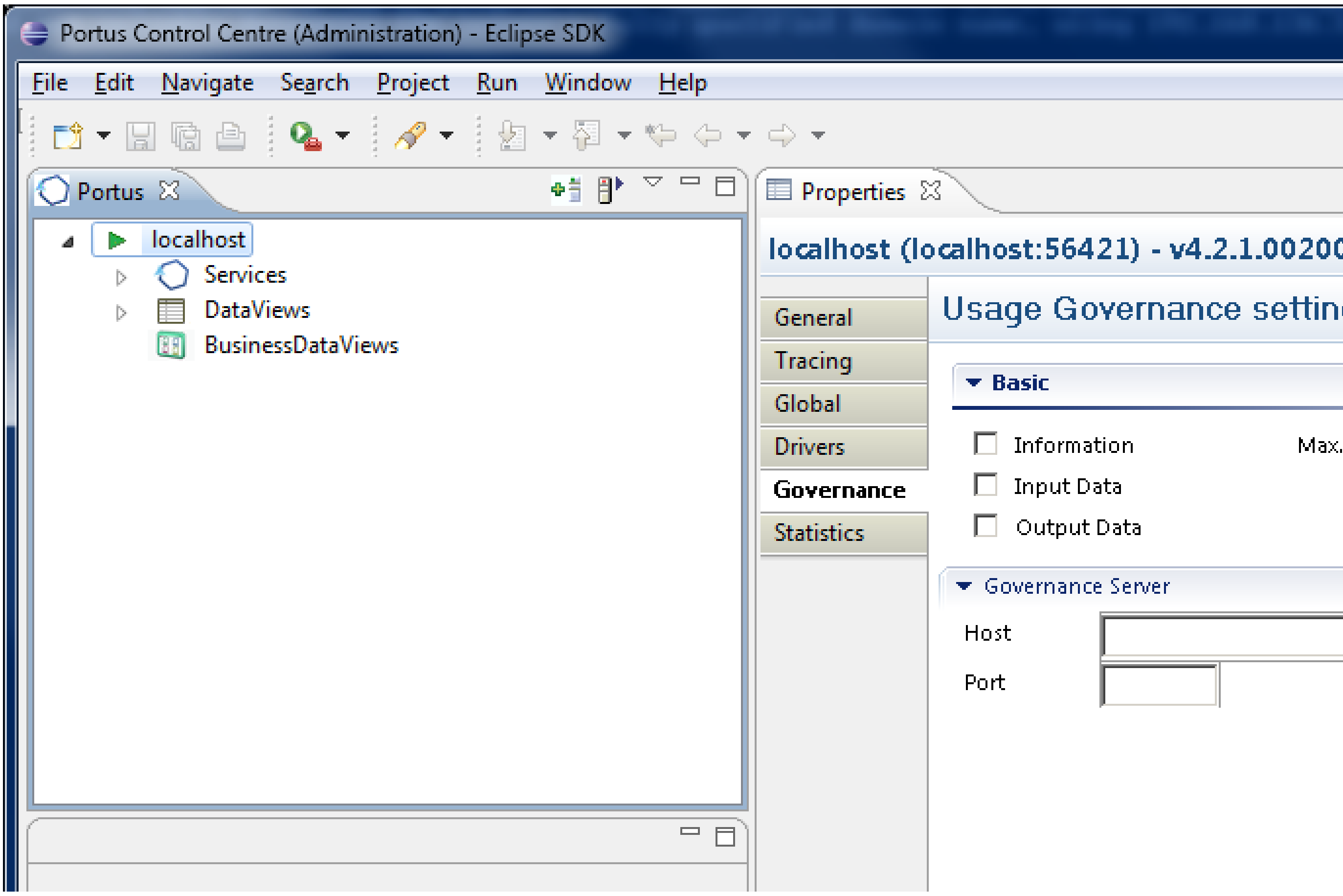
- No governance information about the service at all.
- Detailed access information about the service. This will record the access data (e.g. date and time of access, identity of caller, location of caller etc.) for each request made to the service.
- Request information provided to the service by the caller. This will record the message data provided to the service for each request made to the service.
- Response information returned by the service to the caller. This will record the message data returned by the service to the caller for each request made to the service.

Controlling Collection

Collection will be controlled by higher level defaults for the entire Portus server instance. These may then be overridden at the service level.

Global Collection

This is set on/off via the Portus Control Centre. Selecting the server displays its properties. Click on the Governance tab. Shown below are the items which can be set :


Portus Control Centre (Administration) - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Portus
localhost
Services
DataViews
BusinessDataViews
Properties
localhost (localhost:56421) - v4.2.1.0020
General
Tracing
Global
Drivers
Governance
Statistics
Usage Governance settin

Basic

☐ Information Max.

☐ Input Data

☐ Output Data

Governance Server

Host

Port

Note: To activate/deactivate usage governance, these settings must be saved, the server stopped and restarted

Basic

Here is where Governance collection is set on/off. Selecting Information turns collection on. Selecting Input Data and/or Output Data adds this information to the collection. If writing to a local file then a maximum file size can be set. When this value is reached the file will be written and collection continues with a new file.

Governance Server

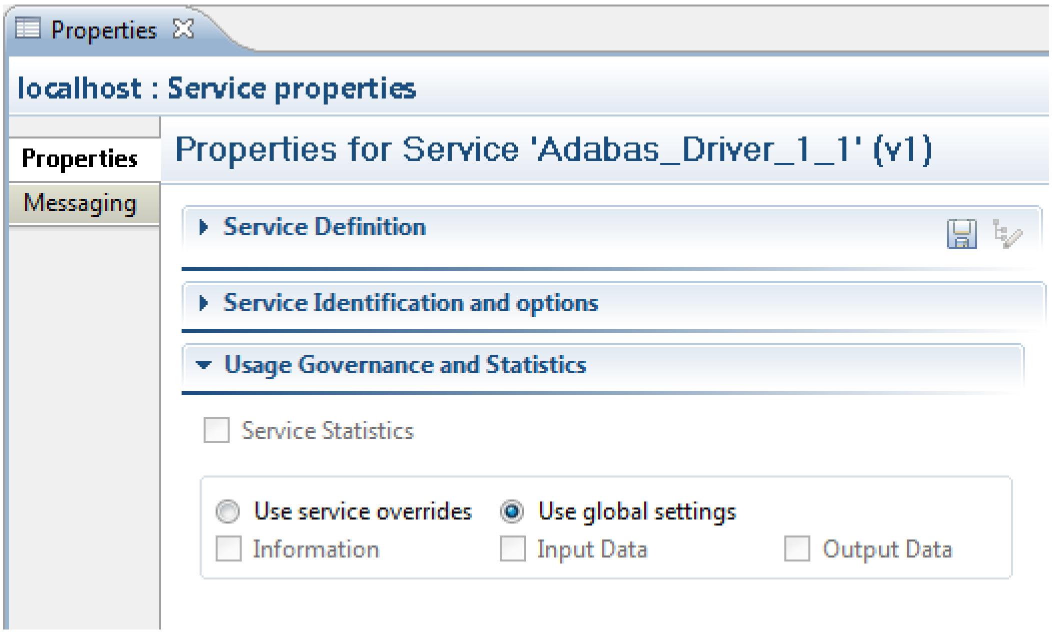
The option to send collection data to another Portus is achieved by setting the Host and Port details. Prior to this a governance web service must be created and the server up and running. See *here* for details on how to do this.

Messaging

A third option is to send data to an existing Messaging system. Currently this can be IBM WebSphere MQ or Software AG EntireX Broker. See bla bla bla for details on Portus Messaging.

Collection at Web Service Level

Select the web service (under Services) in the Control Centre. Here the collection can be set as required for a specific service (Use service overrides). Otherwise the value of the global setting is used (Use global settings).


Properties
Properties
Messaging
Service Definition
Service Identification and options
Usage Governance and Statistics

localhost : Service properties

Properties for Service 'Adabas_Driver_1_1' (v1)

☐ Service Statistics

○ Use service overrides ● Use global settings

☐ Information ☐ Input Data ☐ Output Data

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.

Lifecycle Governance

This section describes how the Portus web services can be governed in terms of their availability, their versions and their status. This concept is referred to as “Lifecycle Governance” – the governance of Portus web services throughout their life time.

The availability (or unavailability) of a web service is something that an organization may wish to set for a period of time. For example, if there is maintenance required on a specific database table, the web service which uses this table could be made unavailable without affecting other services. In this case, clients will receive a specific Portus message to indicating the service is unavailable.

The versioning of web services allows similar services to co-exist, transparent to existing clients, while allowing new clients to take advantage of the latest service modifications. For example, what happens when the type of a parameter needs to change from xs:string to xs:int? This change must be coordinated so that existing clients of the web service can continue unaffected, and ensuring that new clients receive the most up-to-date implementation. Portus can make this possible seamlessly.

The status of a service is a concept where the service is in the process of being migrated from different states as it evolves. For example, a service may be in a “test” or “frozen” state.

For the purposes of this example, we will use the “MySQL City” sample service. See here for more information. The process is the same for any other service defined in Portus.

Concept

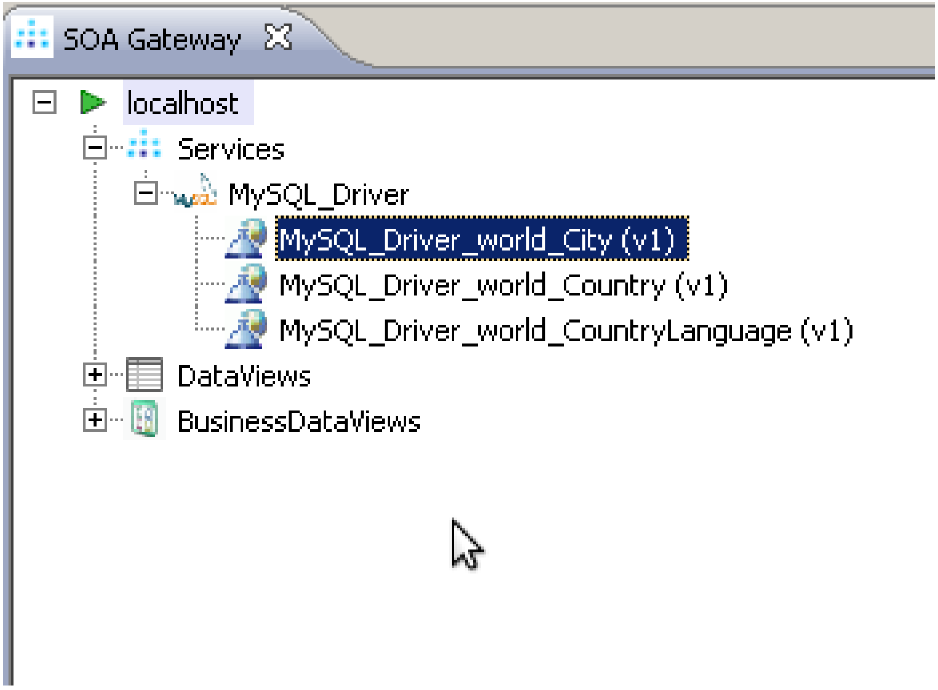
The basic concept of lifecycle governance is that it allows changes to be made to existing services without forcing existing clients to upgrade. Each client that wishes to consume a web service starts with retrieving the service WSDL. Thus, Portus will return different WSDLs based on the different version numbers of the service. The key point is that the endpoint will also have the version number as part of the URL. Thus a client may use “version 1” of a Portus web service. If the administrator decides that the web service needs to be changed, a new WSDL with a new endpoint is made available. New clients can use version 2, while existing clients can continue to use version 1 seamlessly.

From a clients perspective, this process is seamless, there is no requirement for them to have any understanding of web service versions. New clients just start using the WSDL, existing clients work with their services as before.

From an administrators perspective, Portus will control the allocation and incrementation of service version numbers.

Web Service Versioning

Every web service that is created in Portus has an associated version. By default, this version number starts a “1”. The control centre will display the version number of each service after its service name.


SOA Gateway
localhost
Services
MySQL_Driver
MySQL_Driver_world_City (v1)
MySQL_Driver_world_Country (v1)
MySQL_Driver_world_CountryLanguage (v1)
DataViews
BusinessDataViews

When retrieving the WSDL for a service, Portus will always return the most recent version of the WSDL. The most recent version is determined by the incrementally highest version number. For example, version 3 is “more recent” than version 1. It is still possible to retrieve a WSDL other than the most recent, but you must pass an extra “__version=N” argument on the URL. For example:

```
http://host:port/MySQL_Driver_world_City?WSDL&__version=1
```

Note: You will notice that the WSDL endpoint now has a unique version number appended on the end of the URL in the <service> section.

Web Service Status

There are 4 possible states a Portus Web Service can be in.

1 **Test**

In this state, the service may be modified as often as is required to bring the service to a point where it is to be made available to others. The service can be edited and/or modified, and the version number will stay the same.

2 **Frozen**

When the service is to be deployed or made available to others for use, it must be frozen. At this point, the version assigned to the service will represent this version of the service forever. There can be multiple frozen versions of any service, but they must have different version numbers.

3 **Deprecated**

When a service has been marked as deprecated, clients will still be able to call the service, but they will receive a warning message in the response, and the administrator will see a warning message in the system log. In this scenario, it is expected that administrators should recommend a service upgrade to their clients.

4 **Historical**

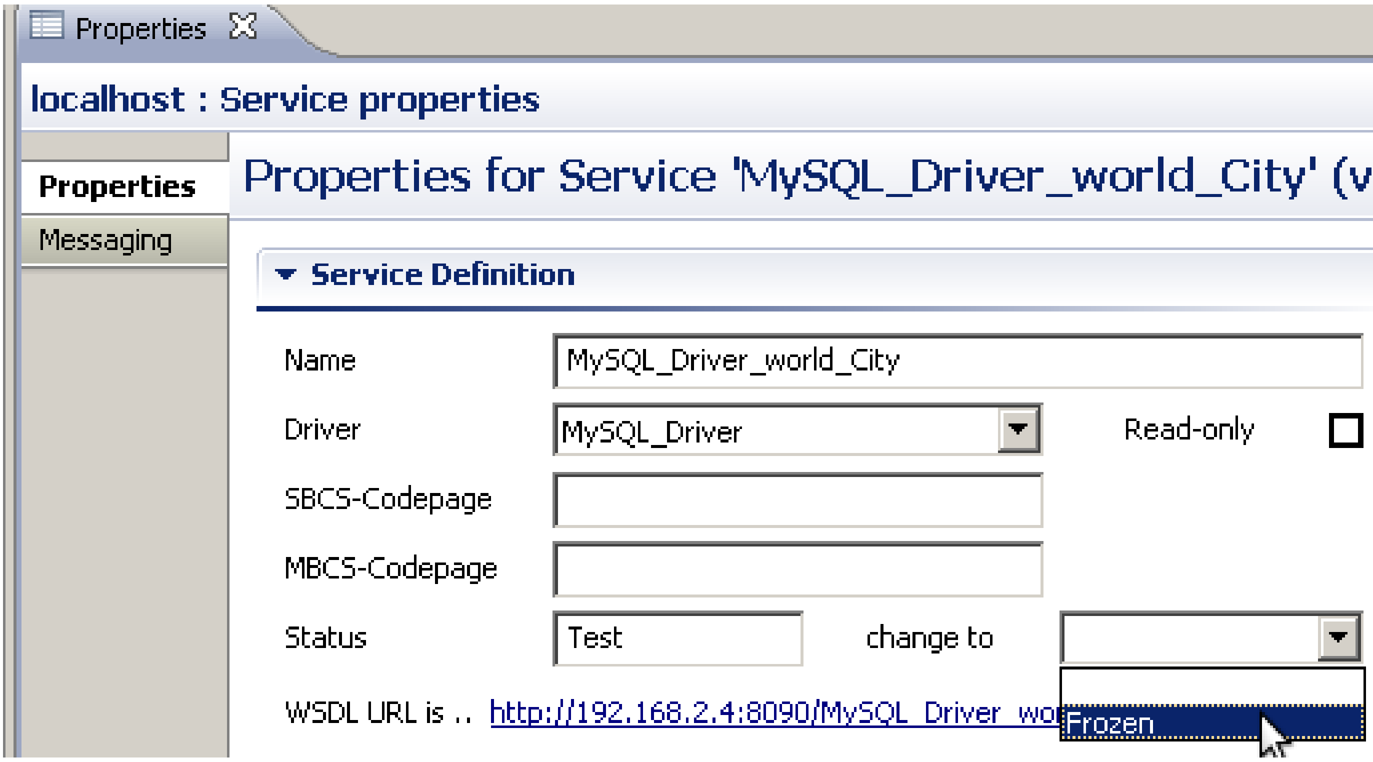
This status will be assigned to a service which is no longer active and has essentially been deleted. It is maintained in the configuration for historical purposes. If this service is invoked, the client will be returned a soap fault and a record logged that an attempt was made to use a historical version of the service.

Practical Example

In this example, we will use the “adabas_EmployeesMini” service that is created when adding a new Adabas driver. The principal is the same for all other Portus web services. Initially, the version number of the web service is “1”. This service can be modified 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.



Properties for Service 'MySQL_Driver_world_City' (v

Service Definition

Name	MySQL_Driver_world_City		
Driver	MySQL_Driver	Read-only	□
SBCS-Codepage			
MBCS-Codepage			
Status	Test	change to	Frozen

WSDL URL is .. http://192.168.2.4:8090/MySQL_Driver_wo

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

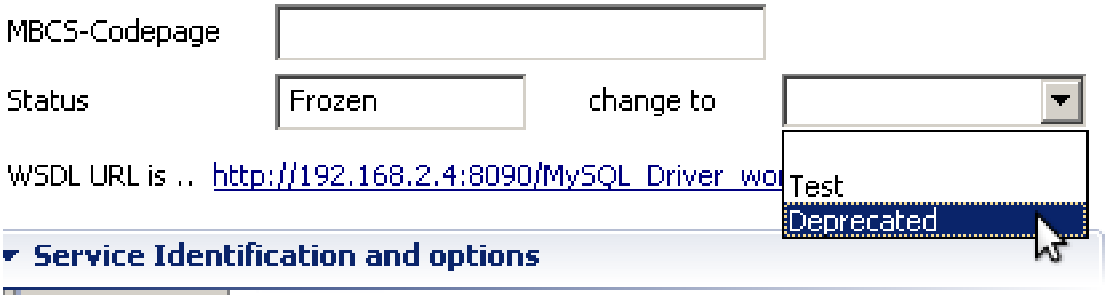
When a client requests the WSDL for this service, by default they will receive the version 2 WSDL. Thus new clients will get the most up to date version of the service. It is assumed that existing clients will only import the WSDL once (when version 1 was the most recent). To access the version 1 WSDL,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

```
http://host:port/MySQL_Driver_world_City?WSDL&__version=1
```

Any client that uses the existing version 1 service will now receive a informational message in the SOAP response Header indicating that a newer version of this service exists. Clients could be configured to recognised this information, and take appropriate action, such as synchronizing with the SOA Gateway to get the latest version of this service.

Deprecating a service

When the administrator wishes to move clients off an old service, they should mark it as deprecated.



MBCS-Codepage

Status: Frozen — change to: Deprecated

WSDL URL is .. http://192.168.2.4:8090/MySQL_Driver_wo

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
```

Portus Security

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

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

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

Advanced Topics

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

- *How to install a server on a remote Windows* on page 143
- *Running multiple instances of Portus* on page 143
- *IPv6 Considerations* on page 144
- *Reducing memory footprint using streaming* on page 145

How to install a server on a remote Windows

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

1 Transfer win32Build.msi to the target machine.

The msi can be located at ../eclipse/features/ com.SOAGateway.feature.install.x86.windows_v.r.m.pppppp.

where ../eclipse is the folder from which your Control Centre is launched and v.r.m.pppppp indicates the version of Portus for Windows.

2 Transfer the Portus license to the target machine.

3 Logon to the target machine and run:

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

Assuming that:

- The msi and your license are in c:\temp\

Important: The license must *NOT* reside on a network drive/share, it *MUST* be copied to a local drive, otherwise the MSI installer will throw an error.

- You want the server to run on port 56000
- Your license is named LIC.xml
- The server prerequisites have already been installed. Check *here* for these.

Running multiple instances of Portus

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

- *Multiple instances of Portus on z/OS* on page 144

- *Multiple instances of Portus on Linux* on page 144

Multiple instances of Portus on z/OS

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

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

Multiple instances of Portus on Linux

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

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

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

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

  ```
  cp -R /home/bre/soaGateway/* /home/bre/PortusNew
  ```
- In the new installation, change the Apache Listen directive, specifying a yet unused port number. This directive can be found in the httpd.conf file in the apache2/conf directory.
- Again, edit the new httpd.conf file and change the Include directive on the last line to pick up the new installation.
- Edit new *adabas_soa_gw.conf* file in the new installation. Modify required directives to pick up the new files and directories in the new installation.
- 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.
  - 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 Portus
- Verify that the Portus start-up messages (found in the Apache logs) refer to the new Portus configuration file.

IPv6 Considerations

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

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

Further information can be found *here*

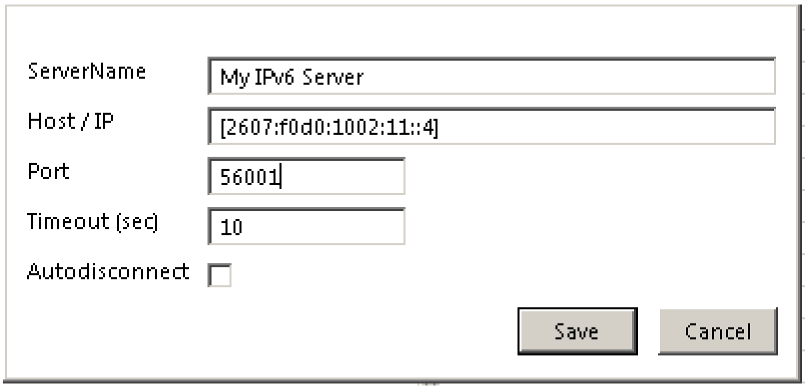
To make Portus listen on the IPv6 interface, provide the IPv6 address of the host machine on the Listen directive in the Apache configuration file. This file is [INSTALL_DIR]/apache2/conf/httpd.conf or [INSTALL_DS].CONF(HT$CONF) depending on your system.

E.g.

- Listen on port 56001 on this machine : Listen [2607:f0d0:1002:11::4]:56001
- Listen on port 56001 for ONLY local IPv6 connections: Listen [::1]:56001

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

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



ServerName: My IPv6 Server

Host / IP: [2607:f0d0:1002:11::4]

Port: 56001

Timeout (sec): 10

Autodisconnect □

Reducing memory footprint using streaming

When Portus handles a request it builds XML structures internally to hold the response payload. This data can become very large, and on systems where memory is limited, unexpected results can occur.

To resolve this, Portus has the ability to stream back data as it is retrieved from the database.

Portus streaming uses HTTP chunked transfer encoding as a transfer mechanism. This is only available in HTTP clients that understand HTTP 1.1 or above.

When data is streamed back to the client, Portus does not have to build large internal 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=*

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:

- Portus may use extra CPU resources when streaming is enabled.

- Not all clients understand HTTP 1.1 or indeed support the chunked transfer encoding.

Chapter

5

Reference

Topics:

- *Tutorials / Samples*
- *SOA Resource Access*
- *Creating a Stylesheet for your Portus Data*
- *Data Masking*
- *Internationalization*
- *Performance Hints*
- *Frequently Asked Questions*
- *Introduction to Eclipse*

This section provides reference information regarding various aspects of the system, frequently asked questions and more.

Tutorials / Samples

This section contains tutorials which will show you how to access your Adabas or MySQL database using Portus. The tutorials are based on a number of programming languages which have built-in Web Service support.

It also contains tutorials detailing how to set up and use the Usage Governace capability offered by Portus.

- *Adabas Tutorials* on page 148
- *MySQL Tutorials* on page 204
- *SOAP over IBM MQ Series Tutorial* on page 217

Other samples:

- *Natural Samples* on page 226

Adabas Tutorials

The Adabas tutorials are based on the well-known Adabas 'Employees' file. The following sample Resource definitions are put in place by the Portus server install process:

- adabas_Employees
- adabas_Employees_9
- adabas_Employees_special
- adabas_EmployeesMini
- adabas_Vehicles

The default Adabas Database ID preset for the demo files is 212, which may not reflect the actual Database ID in your environment, so these will need to be adjusted before trying to run the sample programs. Please familiarize yourself with the *Portus Control Centre functions* required to achieve this before continuing the tutorials trail.

Basic Tutorials

The basic tutorials cover the following areas:

- *Using Java (Axis2) wrapper classes* on page 149
- *Accessing Adabas using the PHP SOAP extension* on page 154
- *Accessing a Portus Service from a Ruby program* on page 187
- *Creating a sample C# application listing "Employees"* on page 163
- *Accessing Adabas through SoapUI* on page 179
- *Accessing Adabas through Microsoft InfoPath* on page 170

Advanced Tutorials

Dealing with LOBs (Large OBjects)

- *Preparing for the LOBs (Large OBjects) samples* on page 188
- *Using LOBs (Large OBjects) with soapUi* on page 189
- *Accessing LOBs from PHP* on page 198
- *Accessing LOBs using a browser* on page 198

Dealing with Transactions

- *Using Transactions with soapUI* on page 200

Samples

This section also includes the following demo files and code samples that you can try out:

- *Tut_02_List.java*
- *First steps (ex01_SoaGatewayFirst.php)*
- *List Employees (ex02_SoaGatewayEmpList.php)*
- *List Employees descending (ex02a_SoaGatewayEmpListDescending.php)*
- *List Employees sorted (ex02a_SoaGatewayEmpListSorted.php)*
- *List by Sub Descriptor (ex06_SoaGatewaySpecial_SubDescriptor.php)*
- *List by Super Descriptor (ex06_SoaGatewaySpecial_SuperDescriptor.php)*
- *Add an Employee (ex03_SoaGatewayEmpAdd.php)*
- *Get an Employee (ex04_SoaGatewayEmpGet.php)*
- *Delete an Employee (ex05_SoaGatewayEmpDel.php)*
- *A simple PHP form for accessing Adabas (ex10_SoaGatewaySimpleForm.php)*
- *All-in-one PHP form accessing the Employees file (ex15_SoaGatewayUpdateForm.php)*
- *empMiniList.rb (Ruby)*
- *ASGDemo.cs (C#)*

Using Java (Axis2) wrapper classes

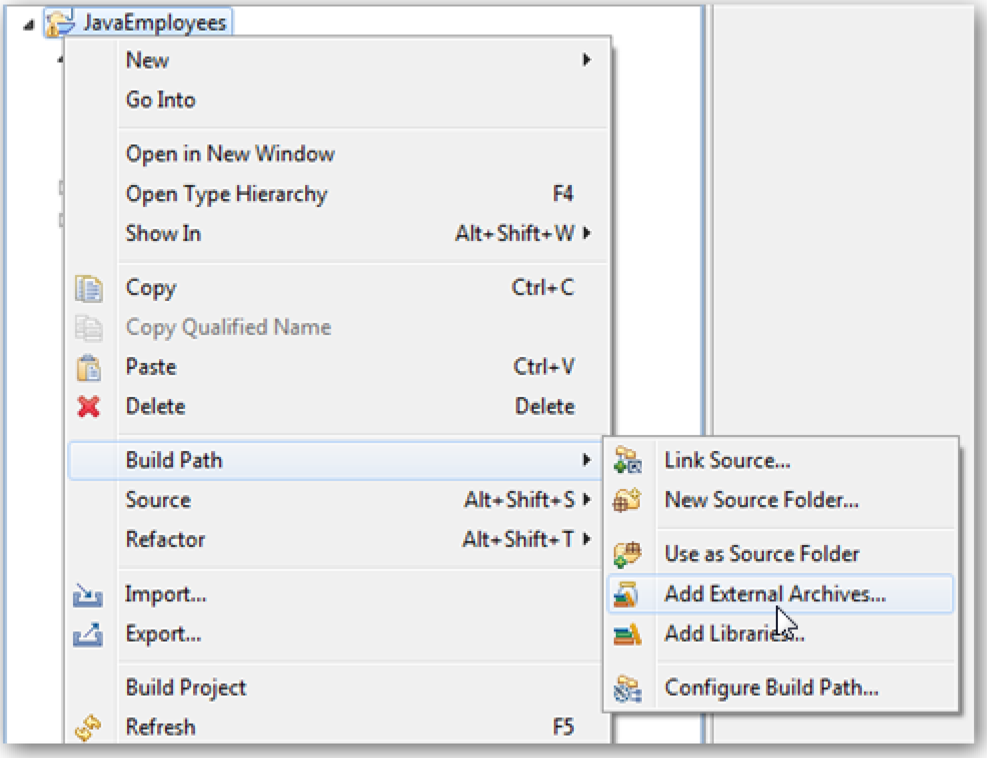
Tutorial: Generate a Java wrapper for the "Employees" file

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

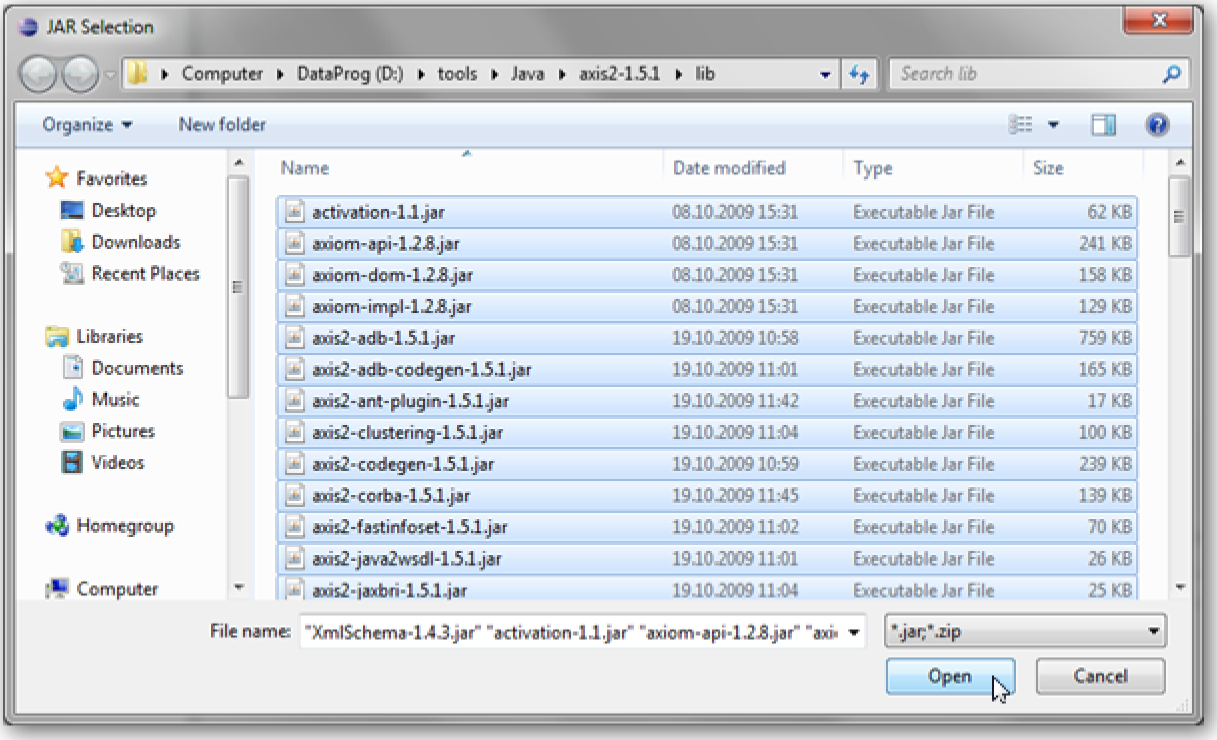
If you do not have it already, download and install the latest Axis2 kit.

These are the steps required to generate the Java wrapper classes for the "adabas_EmployeesMini" DataView supplied with Portus:

1. Create a new Java-project (refer to *Getting started with Eclipse*), name it "JavaEmployees"
2. Right-click the "JavaEmployees" project folder, select "Build Path", then "Add External Archives..:"



Add all .jar files from the axis2 "lib" directory to the project's Build-Path.



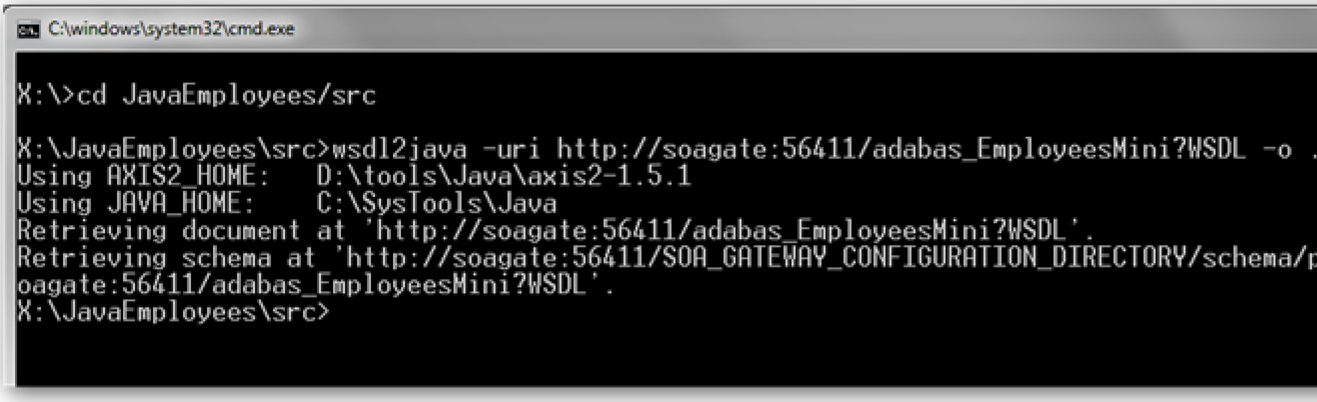
Name	Date modified	Type	Size
activation-1.1.jar	08.10.2009 15:31	Executable Jar File	62 KB
axiom-api-1.2.8.jar	08.10.2009 15:31	Executable Jar File	241 KB
axiom-dom-1.2.8.jar	08.10.2009 15:31	Executable Jar File	158 KB
axiom-impl-1.2.8.jar	08.10.2009 15:31	Executable Jar File	129 KB
axis2-adb-1.5.1.jar	19.10.2009 10:58	Executable Jar File	759 KB
axis2-adb-codegen-1.5.1.jar	19.10.2009 11:01	Executable Jar File	165 KB
axis2-ant-plugin-1.5.1.jar	19.10.2009 11:42	Executable Jar File	17 KB
axis2-clustering-1.5.1.jar	19.10.2009 11:04	Executable Jar File	100 KB
axis2-codegen-1.5.1.jar	19.10.2009 10:59	Executable Jar File	239 KB
axis2-corba-1.5.1.jar	19.10.2009 11:45	Executable Jar File	139 KB
axis2-fastinfoset-1.5.1.jar	19.10.2009 11:02	Executable Jar File	70 KB
axis2-java2wsdl-1.5.1.jar	19.10.2009 11:01	Executable Jar File	26 KB
axis2-jaxbri-1.5.1.jar	19.10.2009 11:04	Executable Jar File	25 KB

File name: "XmlSchema-1.4.3.jar" "activation-1.1.jar" "axiom-api-1.2.8.jar" "axi

.jar;.zip

3 Open a command prompt (aka "DOS box"), change to the "JavaEmployees\src" directory and run the following command

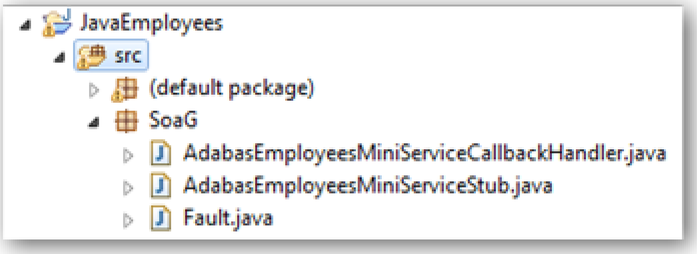
```
wsdl2java -uri http://<yourserver>:<yourport>/
adabas_EmployeesMini?WSDL -o ..\ -p SoaG
```



```
X:\>cd JavaEmployees/src

X:\JavaEmployees\src>wsdl2java -uri http://soagate:56411/adabas_EmployeesMini?WSDL -o .
Using AXIS2_HOME:   D:\tools\Java\axis2-1.5.1
Using JAVA_HOME:    C:\SysTools\Java
Retrieving document at 'http://soagate:56411/adabas_EmployeesMini?WSDL'.
Retrieving schema at 'http://soagate:56411/SOA_GATEWAY_CONFIGURATION_DIRECTORY/schema/p
oagate:56411/adabas_EmployeesMini?WSDL'.
X:\JavaEmployees\src>
```

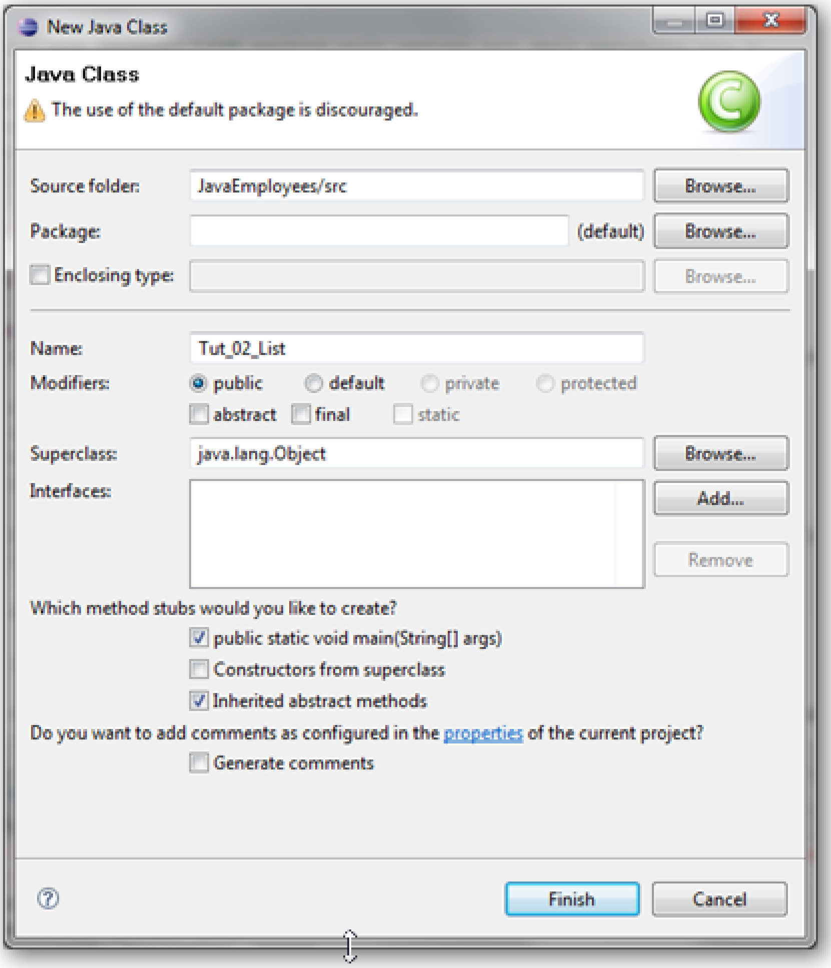
The following items are generated from a Portus WSDL:

- A "Stub" class implementing all types and operations (ports / bindings)
- A CallbackHandler - a stub class (not used in this tutorial) providing hooks for client-side extensions to the generated result- and error handlers.
- A Fault class



```
JavaEmployees
  src
    (default package)
    SoaG
      AdabasEmployeesMiniServiceCallbackHandler.java
      AdabasEmployeesMiniServiceStub.java
      Fault.java
```

4 Add a new Java class named "Tut_02_List" to the project (**File** -> **New** -> **Class**), opt to create a "main" method, click "**Finish**".



New Java Class

Java Class

The use of the default package is discouraged.

Source folder: JavaEmployees/src

Package: (default)

□ Enclosing type:

Name: Tut_02_List

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☑ public static void main(String[] args)

□ Constructors from superclass

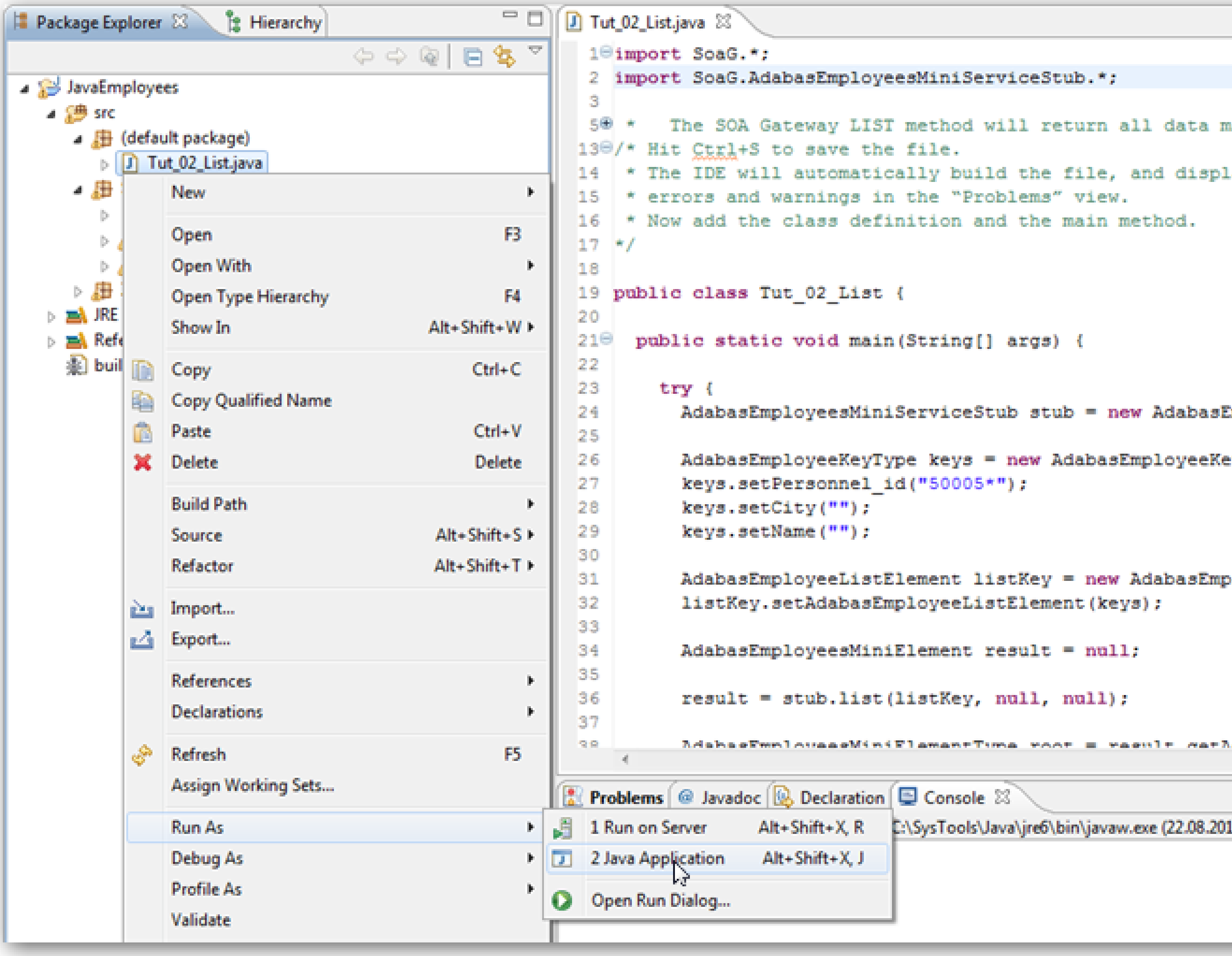
☑ Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

□ Generate comments

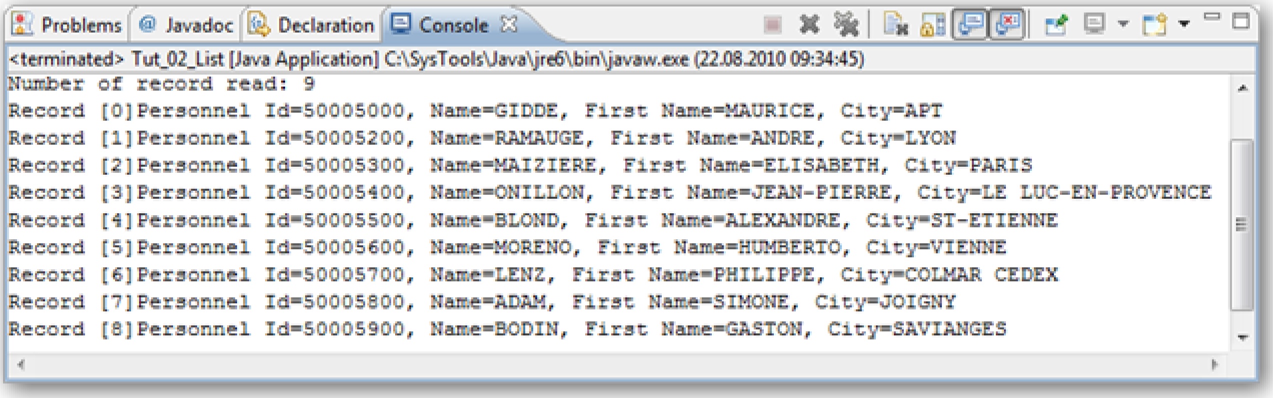
5 Remove the generated code from the newly added class entirely, use (paste) the code from Tut_02_List.java to create your first test program accessing ADABAS data via Portus.

6 Run the program as a "Java Application"



```
import SoaG.*;
import SoaG.AdabasEmployeesMiniServiceStub.*;

 *   The SOA Gateway LIST method will return all data m
/* Hit Ctrl+S to save the file.
 * The IDE will automatically build the file, and displ
 * errors and warnings in the "Problems" view.
 * Now add the class definition and the main method.
 */

public class Tut_02_List {

  public static void main(String[] args) {

    try {
      AdabasEmployeesMiniServiceStub stub = new AdabasE

      AdabasEmployeeKeyType keys = new AdabasEmployeeKe
      keys.setPersonnel_id("50005*");
      keys.setCity("");
      keys.setName("");

      AdabasEmployeeListElement listKey = new AdabasEmp
      listKey.setAdabasEmployeeListElement(keys);

      AdabasEmployeesMiniElement result = null;

      result = stub.list(listKey, null, null);
```

7 The output appears in the "Console" window:



```
<terminated> Tut_02_List [Java Application] C:\SysTools\Java\jre6\bin\javaw.exe (22.08.2010 09:34:45)
Number of record read: 9
Record [0]Personnel Id=50005000, Name=GIDDE, First Name=MAURICE, City=APT
Record [1]Personnel Id=50005200, Name=RAMAUGE, First Name=ANDRE, City=LYON
Record [2]Personnel Id=50005300, Name=MAIZIERE, First Name=ELISABETH, City=PARIS
Record [3]Personnel Id=50005400, Name=ONILLON, First Name=JEAN-PIERRE, City=LE LUC-EN-PROVENCE
Record [4]Personnel Id=50005500, Name=BLOND, First Name=ALEXANDRE, City=ST-ETIENNE
Record [5]Personnel Id=50005600, Name=MORENO, First Name=HUMBERTO, City=VIENNE
Record [6]Personnel Id=50005700, Name=LENZ, First Name=PHILIPPE, City=COLMAR CEDEX
Record [7]Personnel Id=50005800, Name=ADAM, First Name=SIMONE, City=JOIGNY
Record [8]Personnel Id=50005900, Name=BODIN, First Name=GASTON, City=SAVIANGES
```

8 This sample selects all "Employees" records with a personnel-id of 50005nnn, you may want to experiment varying the key data, this is easily done by modifying the properties passed to the generated classes. E.g. try the following to list all records for "Employees" whose names start "SMI", living in cities with names starting "D".

```
keys.setPersonnel_id("");
keys.setName("SMI*");
keys.setCity("D*");
```

The following Java/Axis2 tutorial programs are available:

Tutorial	What it does
Tut_01a_Get.java	GET a single record by Personnel Id
Tut_01b_GetByISN.java	GET a single record by ISN
Tut_02_List.java	LIST some records
Tut_03a_SelectSimple.java	SELECT by a range of Personnel IDs
Tut_03b_SelectSorted.java	SELECT by a range of Personnel IDs and sort by Name
Tut_03c_SelectConversational.java	SELECT by multiple ranges of Personnel IDs, returned in "chunks" of 20 records each
Tut_04a_AddUpdateDelete.java	ADD + UPDATE + DELETE in "autocommit" mode
Tut_04b_AddTransactional.java	ADD in a transactional context

Accessing Adabas using the PHP SOAP extension

This tutorial

- demonstrates how to access Portus from a PHP script
- provides a number of PHP examples

What is PHP ?

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML

The PHP interpreter is available as source code or as pre-compiled binaries for major platforms, including most Linux™ distributions, Windows®, Mac OS X, and iSeries™.

The latest release is PHP 5 and is seeing increasing adoption. PHP 5 introduces improvements to the object model; also, the underlying memory management has been redesigned with multi-threading and performance in mind.

For more information about PHP, or to download the software, please refer to *the PHP homepage*.

New in PHP 5 is a built-in SOAP extension. It is supplied as part of PHP.

For this tutorial to work, you should have PHP 5 up and running in your Web server, see the install.txt document in the PHP distribution library for details.

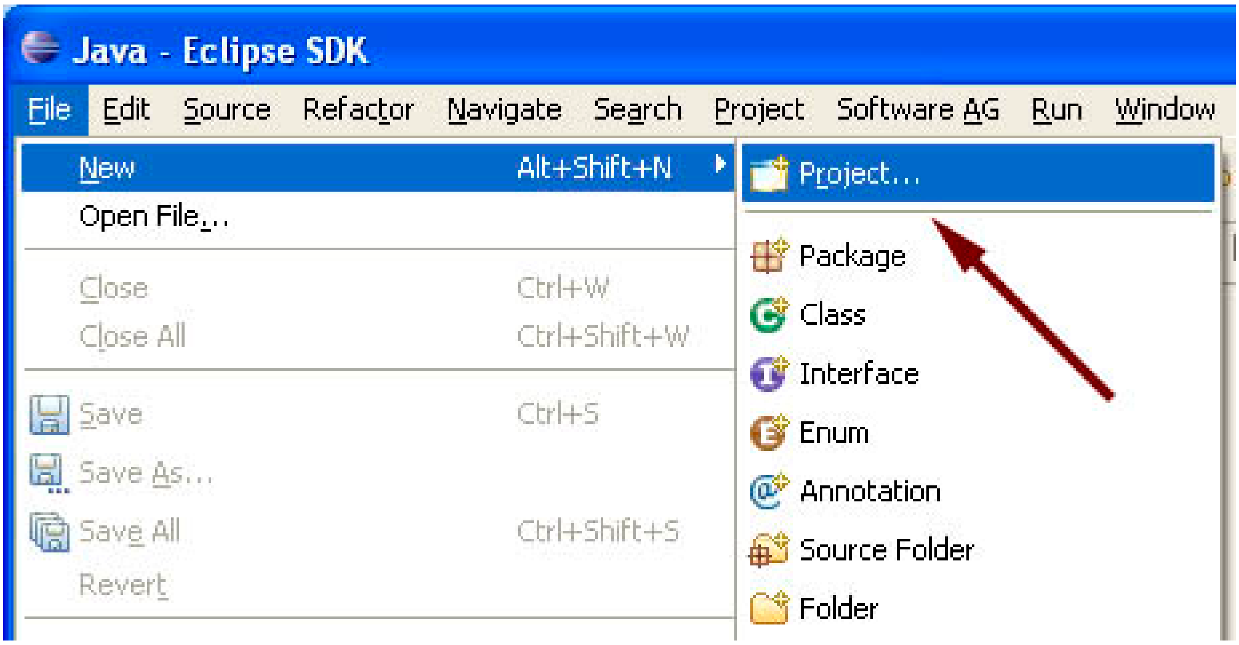
Installing the Eclipse PHP Development Tools

The Eclipse PHP Development Tools (PDT) are not absolutely necessary, but using Eclipse greatly simplifies the development process. This tutorial assumes the PDT to be installed.

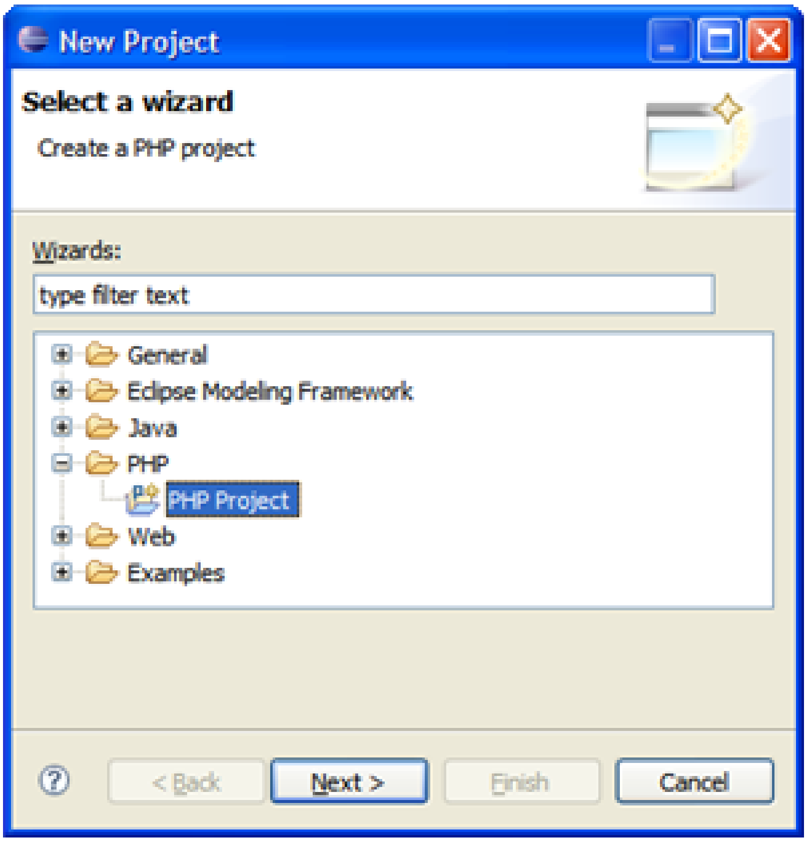
Please refer to *the Eclipse PDT project pages* for installation and configuration instructions.

Accessing Adabas from PHP

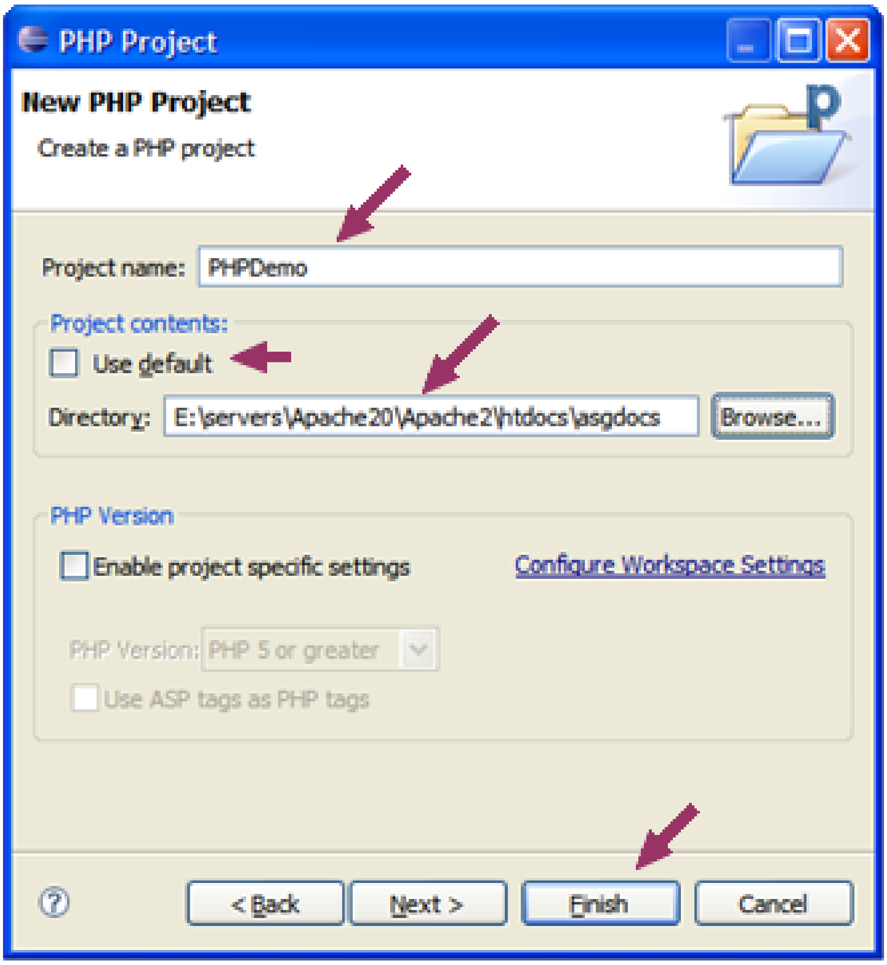
1 First, create a new project within your workspace.



2 Opt to create a PHP Project, click **Next**



Give the project a name and unselect the "Use default" box. Browse to the default location of your html documents, in this case an Apache server document folder called "ASGdocs", click **Finish**



Project name: PHPDemo

Project contents:

□ Use default

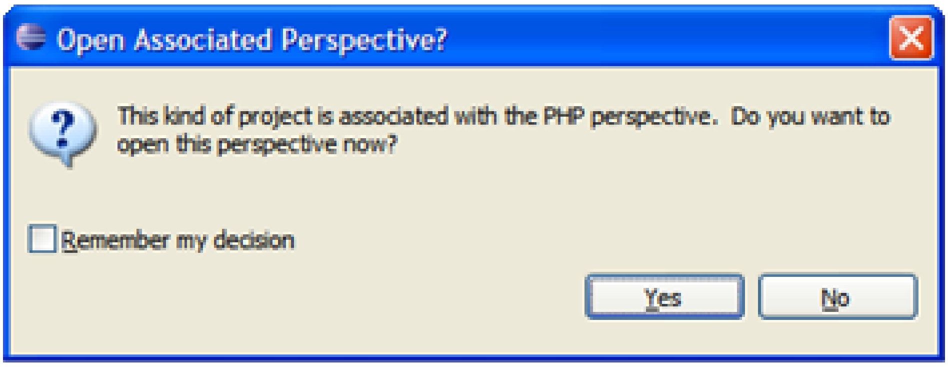
Directory: E:\servers\Apache20\Apache2\htdocs\asgdocs

□ Enable project specific settings

PHP Version: PHP 5 or greater

□ Use ASP tags as PHP tags

If you are asked to switch to the PHP perspective, opt to do so

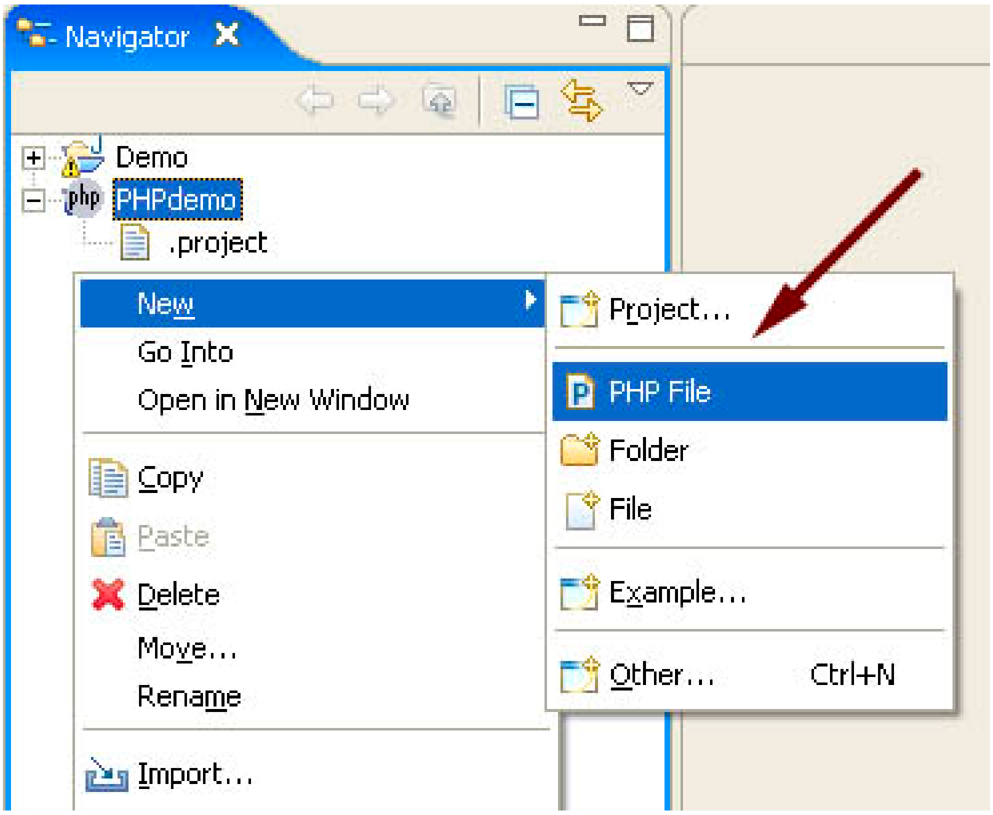


This kind of project is associated with the PHP perspective. Do you want to open this perspective now?

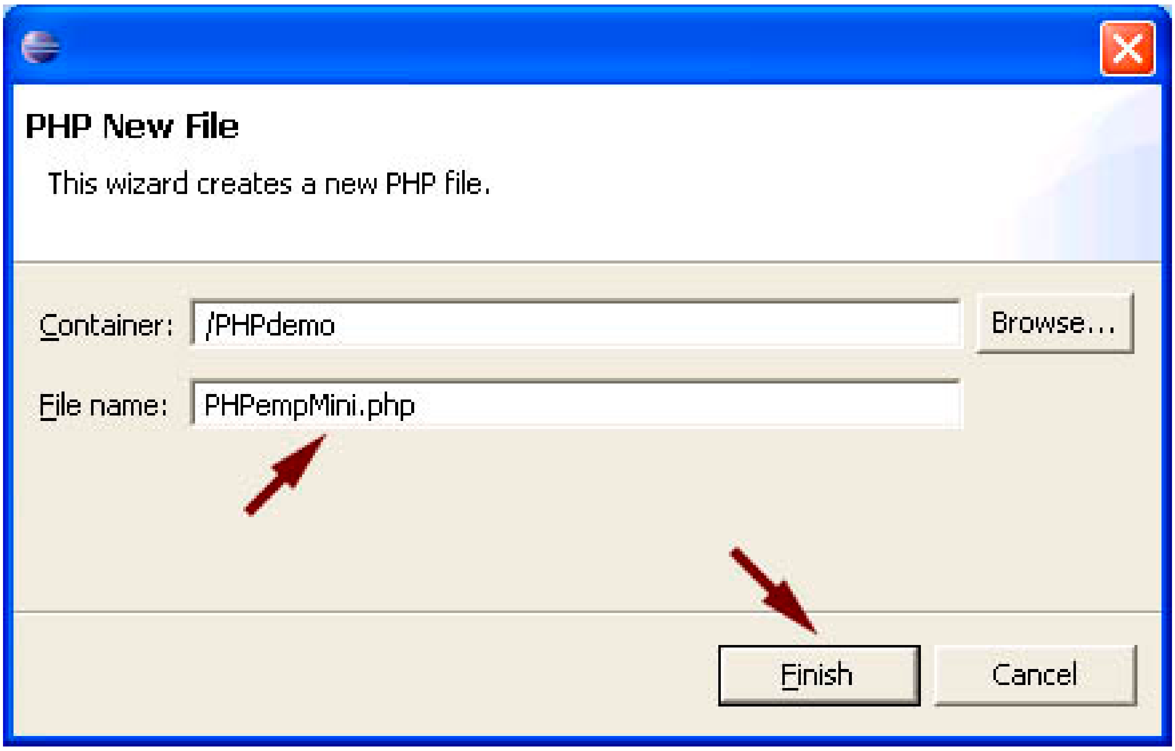
□ Remember my decision

3 Create the PHP SOAP client

Create a PHP script file by right-clicking into the Eclipse Navigator area, select **New -> PHP File**



Specify a File name, click Finish

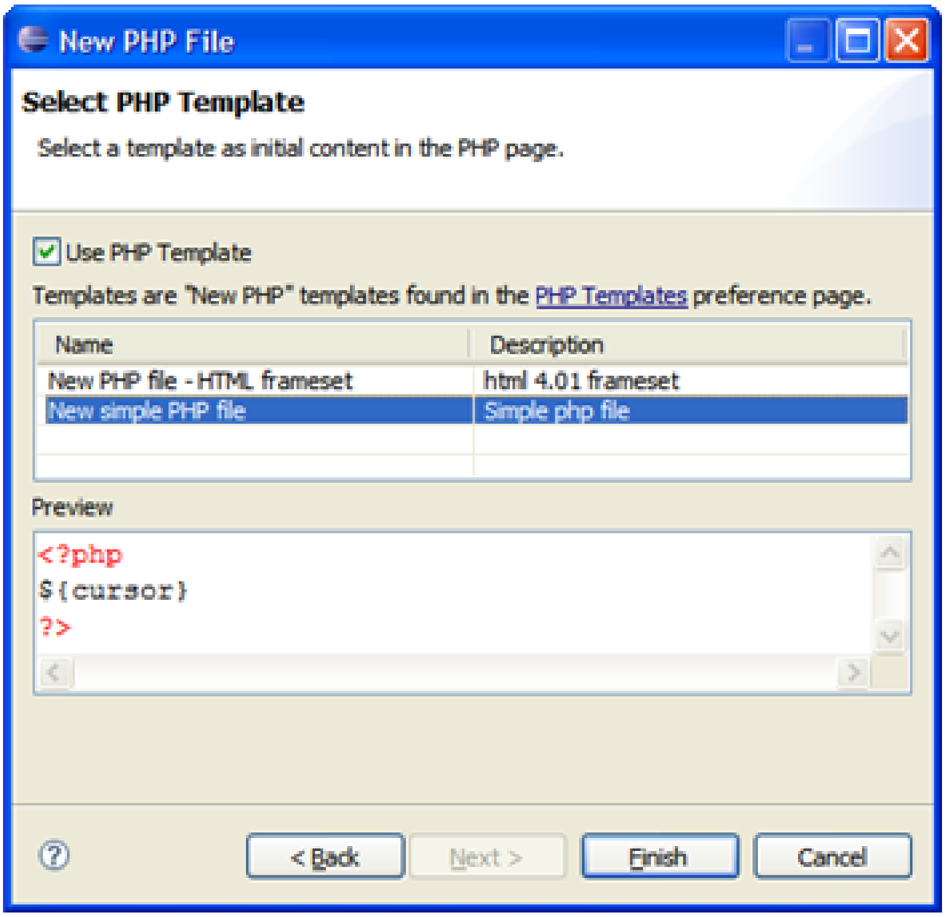


This wizard creates a new PHP file.

Container: /PHPdemo

File name: PHPempMini.php

Opt to create a "Simple PHP File"



Select PHP Template

Select a template as initial content in the PHP page.

☑ Use PHP Template

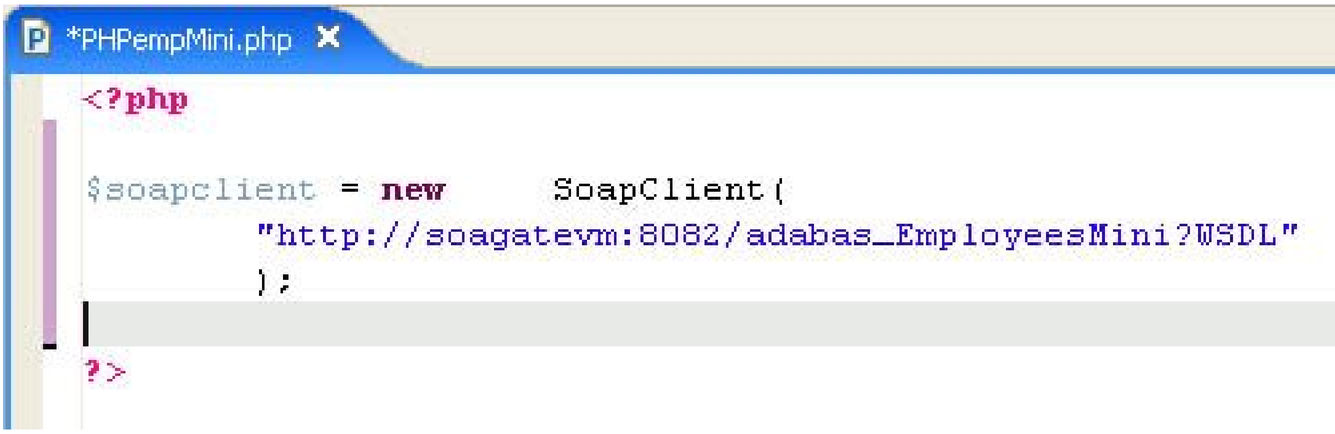
Templates are "New PHP" templates found in the PHP Templates preference page.

Name	Description
New PHP file - HTML frameset	html 4.01 frameset
New simple PHP file	Simple php file

Preview

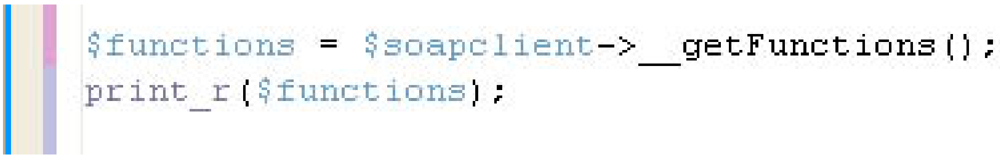
```
<?php
${cursor}
?>
```

The PHP SOAP class to represent the Adabas Service is called SoapClient, the first step will be to instantiate SoapClient, passing the URL of an Portus WSDL as the parameter:



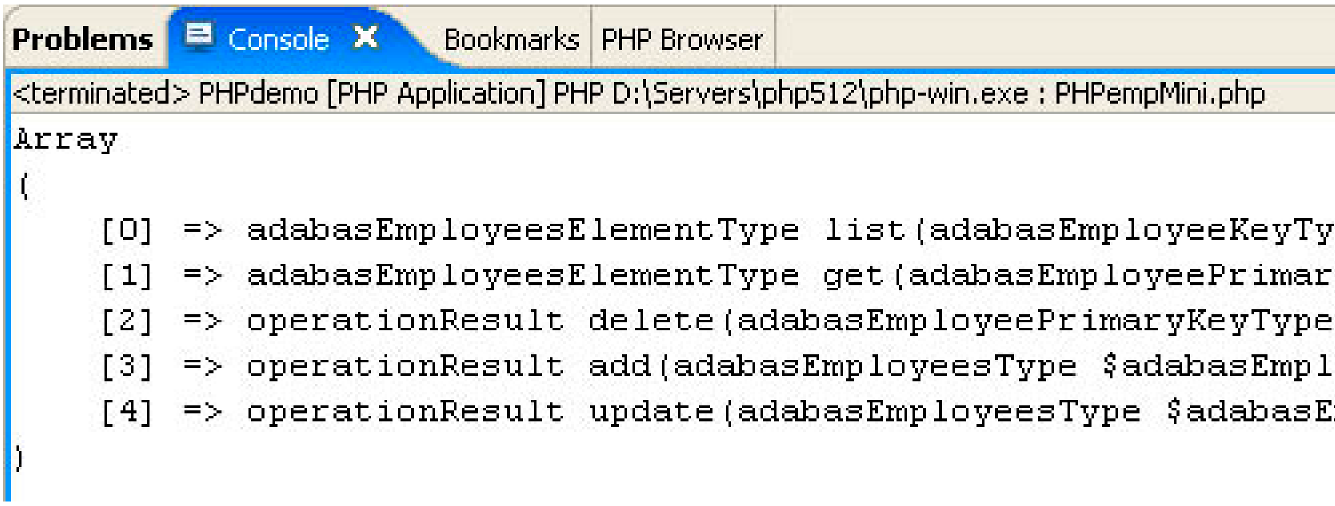
```
<?php

$soapclient = new     SoapClient(
        "http://soagatevm:8082/adabas_EmployeesMini?WSDL"
        );

?>
```

Now that we have instantiated our client we want to see what methods it provides and what parameters are required. Fortunately we can get PHP and the instantiated SoapClient class to do most of the work for us easily:



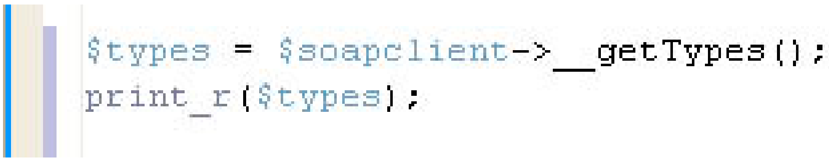
```
$functions = $soapclient->__getFunctions();
print_r($functions);
```

If you run this as a console application (via "Run") the output is much better formatted than running it in the PHP browser. The console window will show the following:



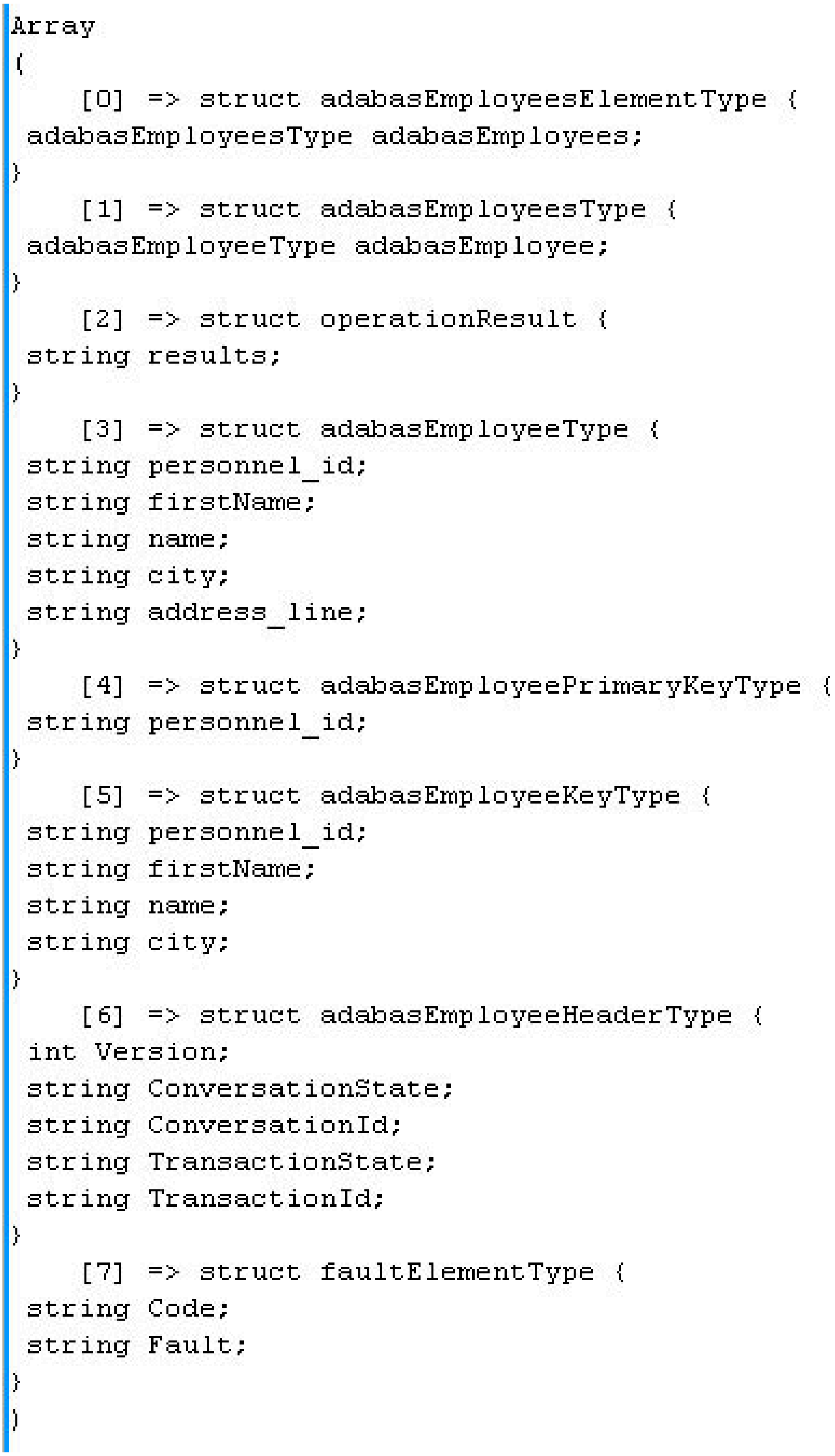
```
<terminated> PHPdemo [PHP Application] PHP D:\Servers\php512\php-win.exe : PHPempMini.php
Array
(
    [0] => adabasEmployeesElementType list(adabasEmployeeKeyTy
    [1] => adabasEmployeesElementType get(adabasEmployeePrimar
    [2] => operationResult delete(adabasEmployeePrimaryKeyType
    [3] => operationResult add(adabasEmployeesType $adabasEmpl
    [4] => operationResult update(adabasEmployeesType $adabasE
)
```

This shows that the service described by the WSDL provides five operations: list, get, delete, add and update; It also lists the required parameters and the responses given. A description of the input and output parameters can be retrieve by calling the __getTypes class:



```
$types = $soapclient->__getTypes();
print_r($types);
```

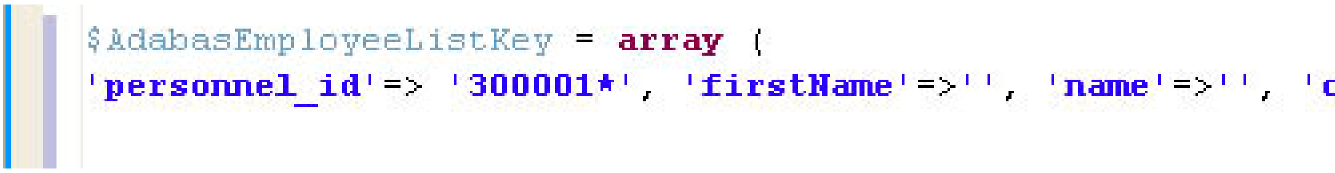
The output will look like this:



```
Array
(
    [0] => struct adabasEmployeesElementType {
 adabasEmployeesType adabasEmployees;
}
    [1] => struct adabasEmployeesType {
 adabasEmployeeType adabasEmployee;
}
    [2] => struct operationResult {
 string results;
}
    [3] => struct adabasEmployeeType {
 string personnel_id;
 string firstName;
 string name;
 string city;
 string address_line;
}
    [4] => struct adabasEmployeePrimaryKeyType {
 string personnel_id;
}
    [5] => struct adabasEmployeeKeyType {
 string personnel_id;
 string firstName;
 string name;
 string city;
}
    [6] => struct adabasEmployeeHeaderType {
 int Version;
 string ConversationState;
 string ConversationId;
 string TransactionState;
 string TransactionId;
}
    [7] => struct faultElementType {
 string Code;
 string Fault;
}
)
```

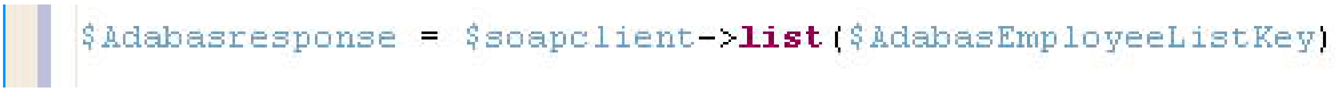
This information is sufficient to construct the first simple call to the "list" operation.

First an array of the required input parameters needs to be constructed:



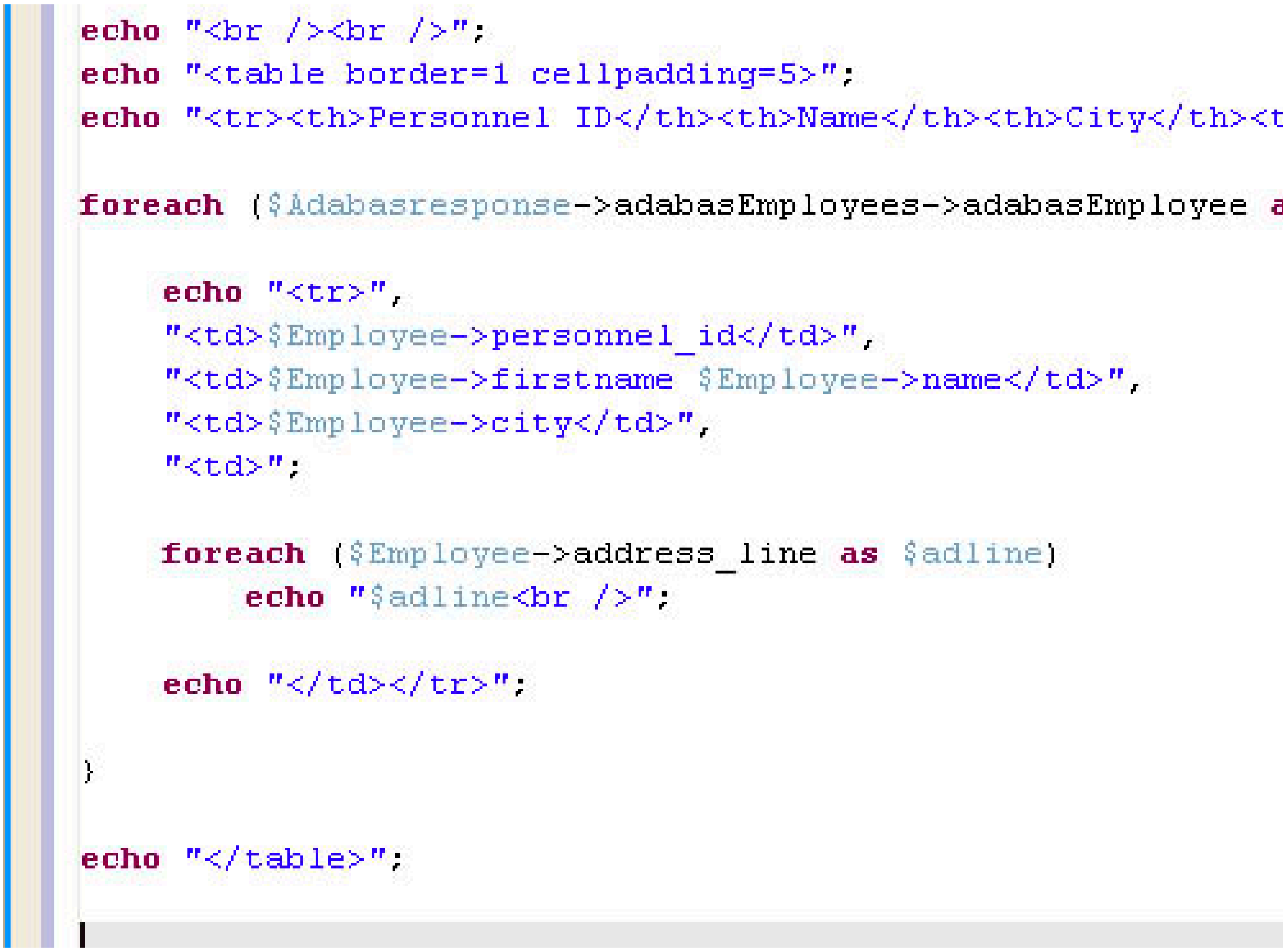
```
$AdabasEmployeeListKey = array (
'personnel_id'=> '300001*', 'firstName'=>'', 'name'=>'', 'c
```

Ready to invoke the "list" operation as a method of the soapclient class:



```
$Adabasresponse = $soapclient->list($AdabasEmployeeListKey)
```

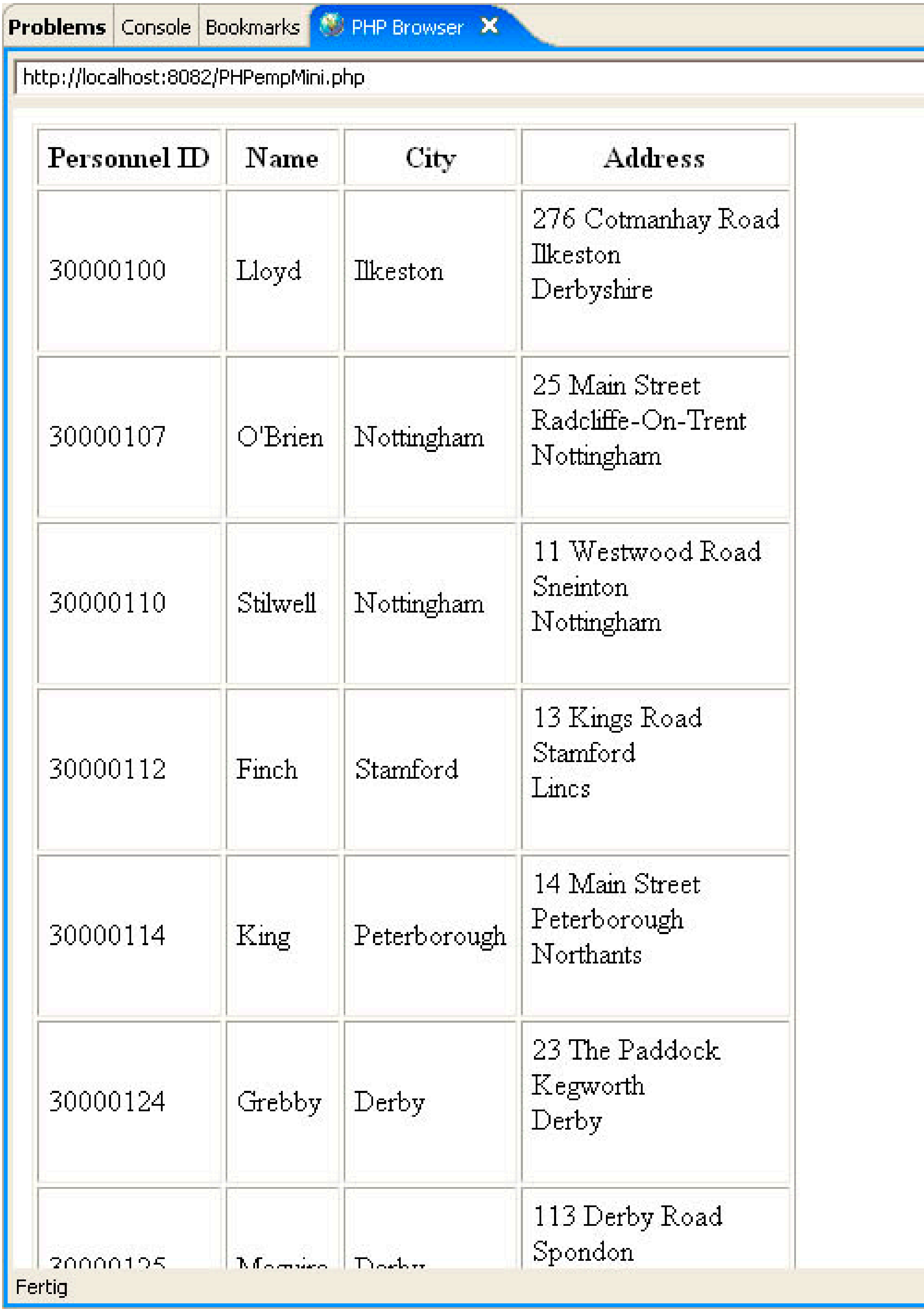
Now it is just a matter of taking the returned object and outputting the required results in a table:



```
echo "<br /><br />";
echo "<table border=1 cellpadding=5>";
echo "<tr><th>Personnel ID</th><th>Name</th><th>City</th><t

foreach ($Adabasresponse->adabasEmployees->adabasEmployee a

    echo "<tr>",
    "<td>$Employee->personnel_id</td>",
    "<td>$Employee->firstname $Employee->name</td>",
    "<td>$Employee->city</td>",
    "<td>";

    foreach ($Employee->address_line as $adline)
        echo "$adline<br />";

    echo "</td></tr>";

}

echo "</table>";
```

Run this in the built in PHP Browser (activated with "**Window**" -> "**Show view**" and select "**PHP Browser**") or an external browser:

```
http://<your_localhost_url>/PHPempMini.php:
```



http://localhost:8082/PHPempMini.php

Personnel ID	Name	City	Address
30000100	Lloyd	Ilkeston	276 Cotmanhay Road Ilkeston Derbyshire
30000107	O'Brien	Nottingham	25 Main Street Radcliffe-On-Trent Nottingham
30000110	Stilwell	Nottingham	11 Westwood Road Sneinton Nottingham
30000112	Finch	Stamford	13 Kings Road Stamford Lincs
30000114	King	Peterborough	14 Main Street Peterborough Northants
30000124	Grebby	Derby	23 The Paddock Kegworth Derby
30000125	Maguire	Derby	113 Derby Road Spondon

PHP Examples

The following PHP examples can be copied from here, moved to your web server's DocumentRoot (for example) and executed:

- *Tut_02_List.java*
- *First steps (ex01_SoaGatewayFirst.php)*
- *List Employees (ex02_SoaGatewayEmpList.php)*
- *List Employees descending (ex02a_SoaGatewayEmpListDescending.php)*
- *List Employees sorted (ex02a_SoaGatewayEmpListSorted.php)*
- *List by Sub Descriptor (ex06_SoaGatewaySpecial_SubDescriptor.php)*
- *List by Super Descriptor (ex06_SoaGatewaySpecial_SuperDescriptor.php)*
- *Add an Employee (ex03_SoaGatewayEmpAdd.php)*
- *Get an Employee (ex04_SoaGatewayEmpGet.php)*
- *Delete an Employee (ex05_SoaGatewayEmpDel.php)*
- *A simple PHP form for accessing Adabas (ex10_SoaGatewaySimpleForm.php)*
- *All-in-one PHP form accessing the Employees file (ex15_SoaGatewayUpdateForm.php)*
- *empMiniList.rb (Ruby)*
- *ASGDemo.cs (C#)*

Creating a sample C# application listing "Employees"

With a service description (WSDL), a proxy class can be created with the .NET Framework SDK **Wsdl.exe** tool.

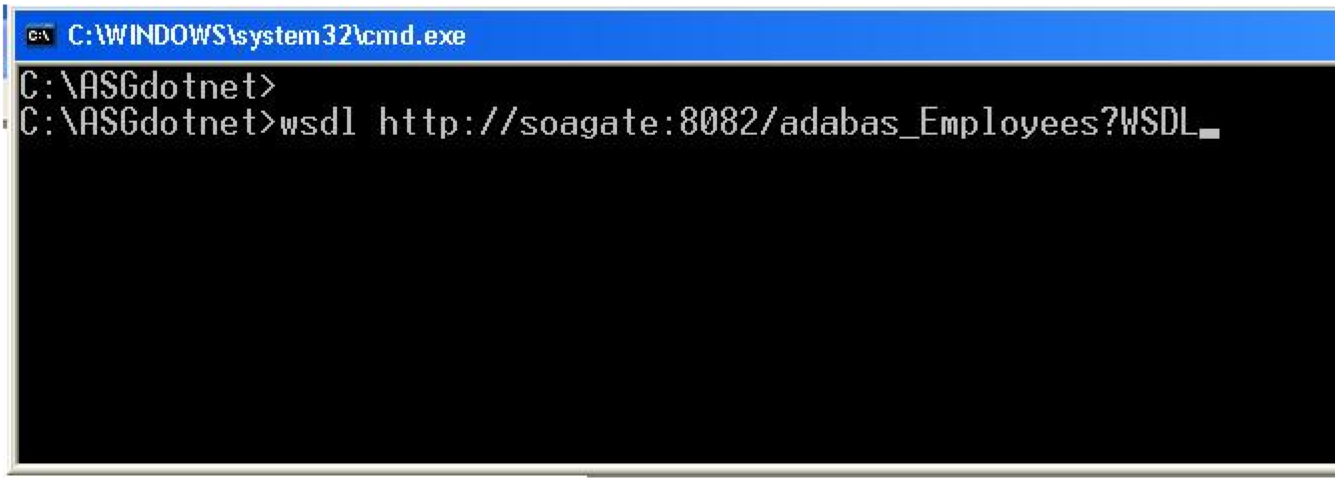
A XML Web service client can then invoke methods of the proxy class, which communicate with Portus over the network by processing the SOAP messages sent to and from Portus server. The proxy class handles the work of mapping parameters to XML elements and then sending the SOAP message over the network.

Wsdl.exe can be used to create proxies for C#, Visual Basic .NET and JScript .NET, for the purpose, we will be generating C#.

Wsdl.exe is a Microsoft .NET tool which is used to create proxies for C#, Visual Basic .NET and JScript .NET. In this tutorial, we will be generating C#.

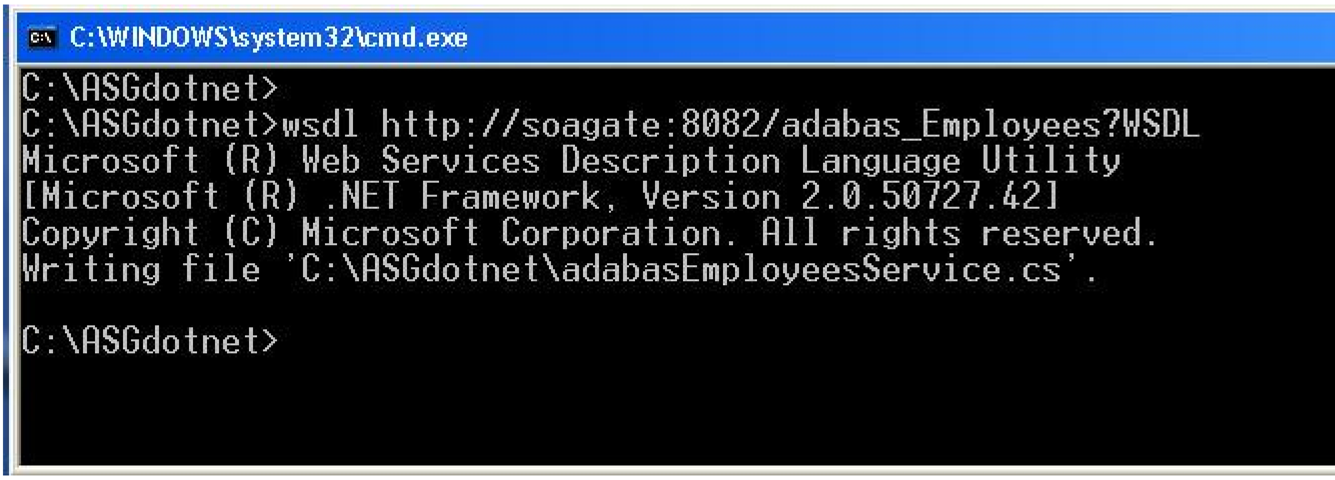
1 From a command prompt, execute Wsdl.exe, specifying the URL / URI of the Portus DataSource to be exposed, append ?WSDL to instruct the Portus server to return the WSDL, not data:

If the Wsdl.exe is not found, open the Visual Studio command prompt via the Start Menu. This location depends on what packages are installed, but often resides under "Microsoft Visual C# " or "Microsoft .NET Framework SDK ".



```
C:\ASGdotnet>
C:\ASGdotnet>wsdl http://soagate:8082/adabas_Employees?WSDL
```

2 A single source file is generated, its name is <rootElementName>Service.cs, in this case the "root element" within the XSD is "adabasEmployees", thus the name of the proxy class source file adabasEmployeesService.cs

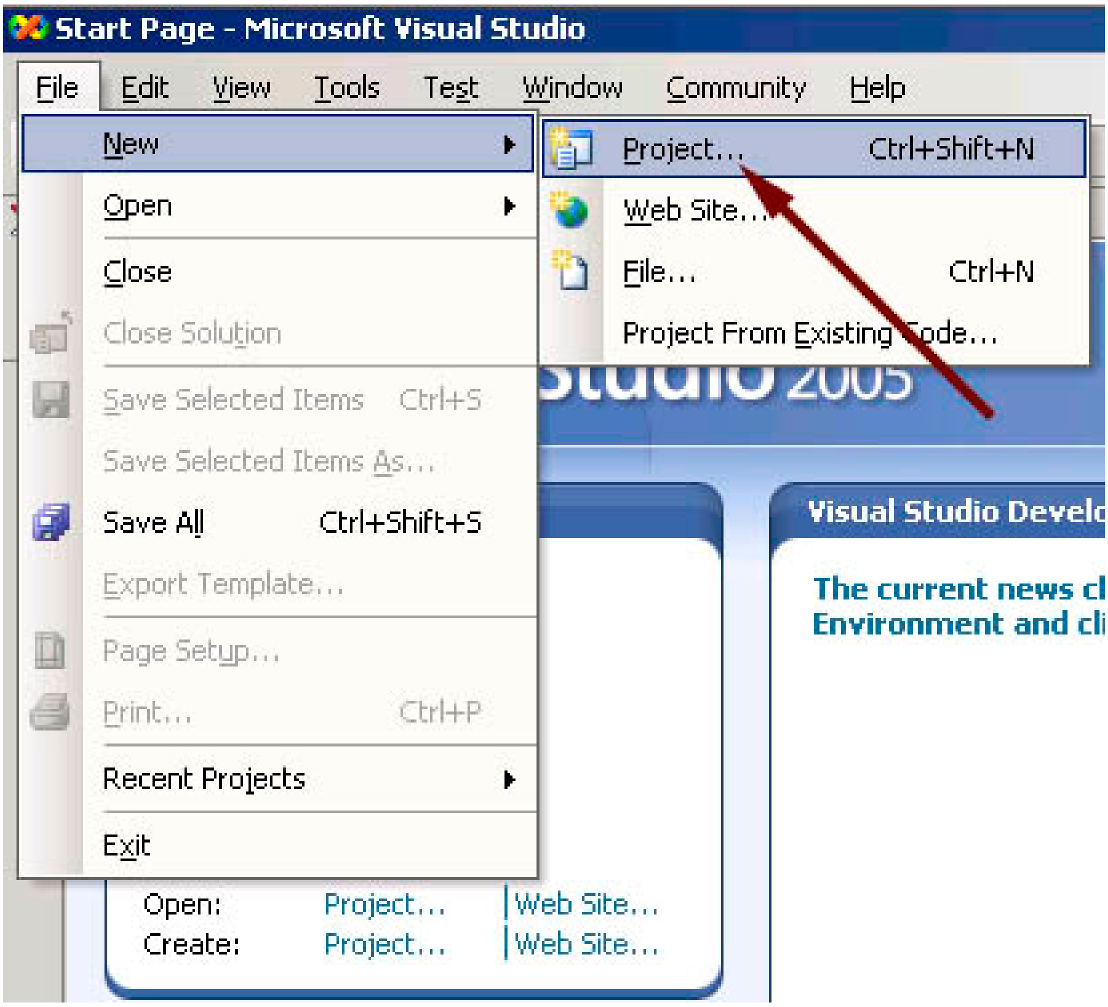


```
C:\ASGdotnet>
C:\ASGdotnet>wsdl http://soagate:8082/adabas_Employees?WSDL
Microsoft (R) Web Services Description Language Utility
[Microsoft (R) .NET Framework, Version 2.0.50727.42]
Copyright (C) Microsoft Corporation. All rights reserved.
Writing file 'C:\ASGdotnet\adabasEmployeesService.cs'.

C:\ASGdotnet>
```

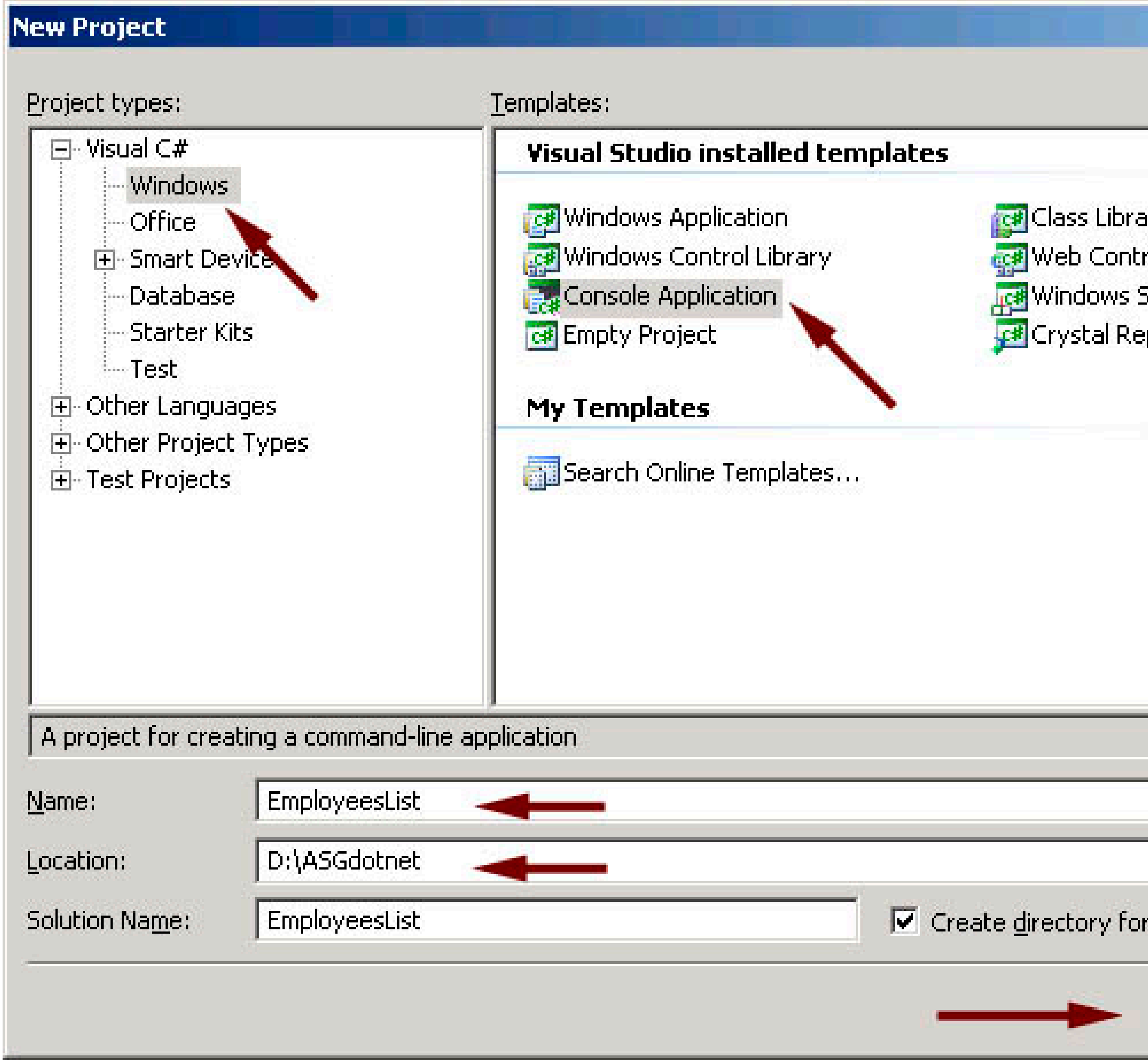
This file contains a proxy class exposing both synchronous and asynchronous methods for each Web Service operation provided by Portus.

For instance, for the list operation, the proxy class has the following methods: list, Beginlist, and Endlist. The list method of the proxy class is used to communicate with Portus synchronously, but the Beginlist and Endlist methods are used to communicate with a Portus server asynchronously. For more information about asynchronous communication with a Web Service please refer to the .NET documentation.

3 Start MS Visual Studio, create a new project with **File -> New - Project** (or the shortcut Ctrl+Shift +N):



Create a C# Console Application, assign a name to it, specify the storage location, click **OK**



A project for creating a command-line application

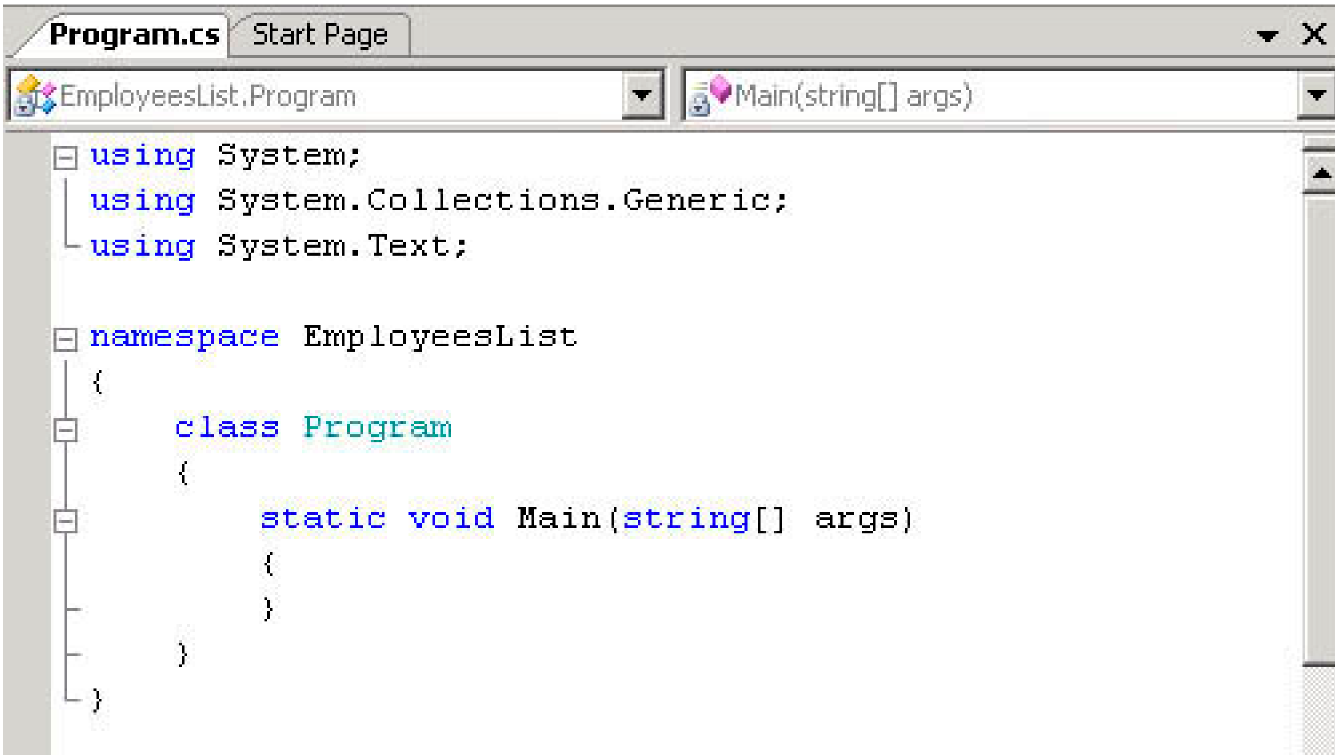
Name: EmployeesList

Location: D:\ASGdotnet

Solution Name: EmployeesList

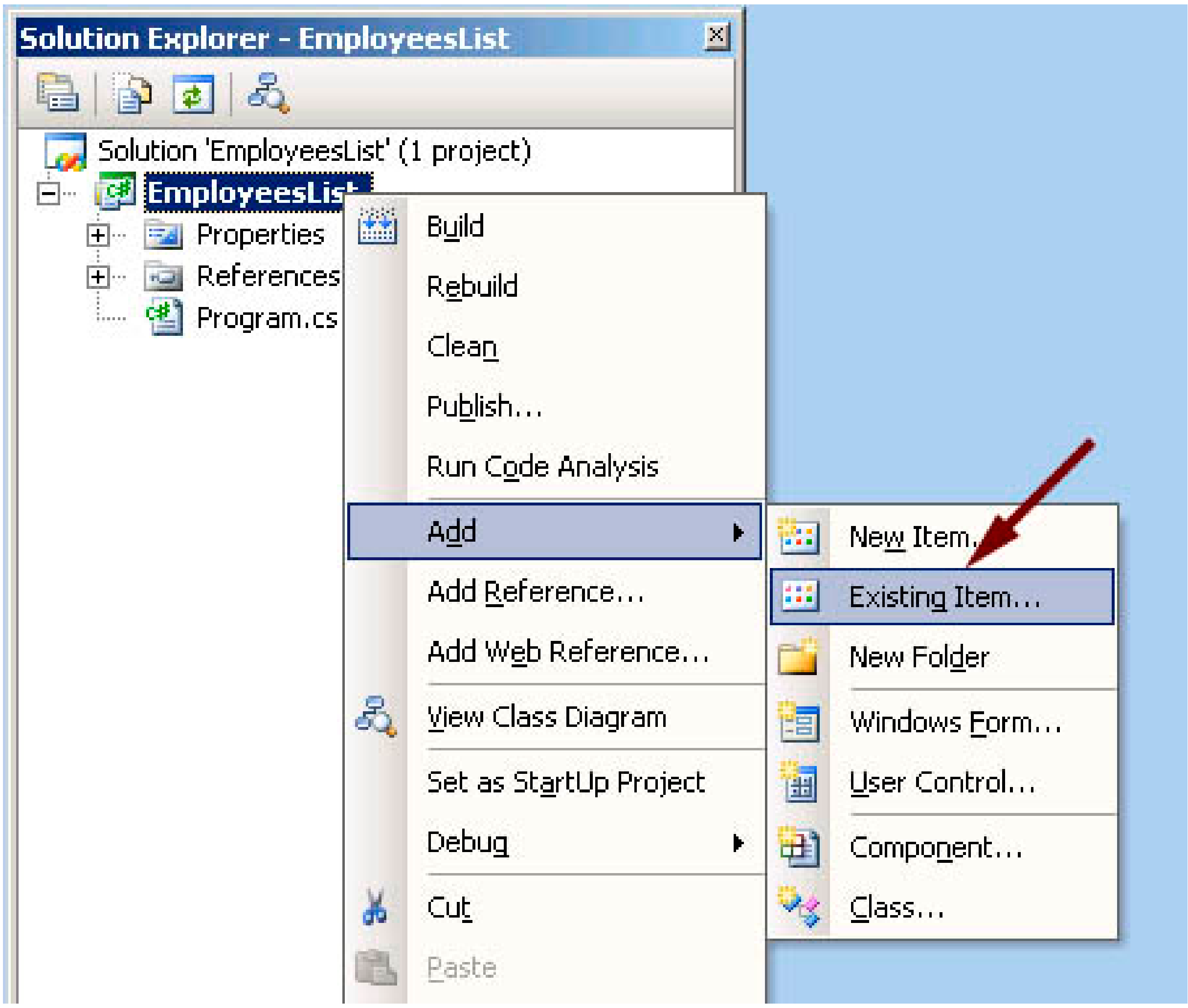
☑ Create directory for

A skeleton class file has been generated into your project workspace, with the required class definition and an empty Main method

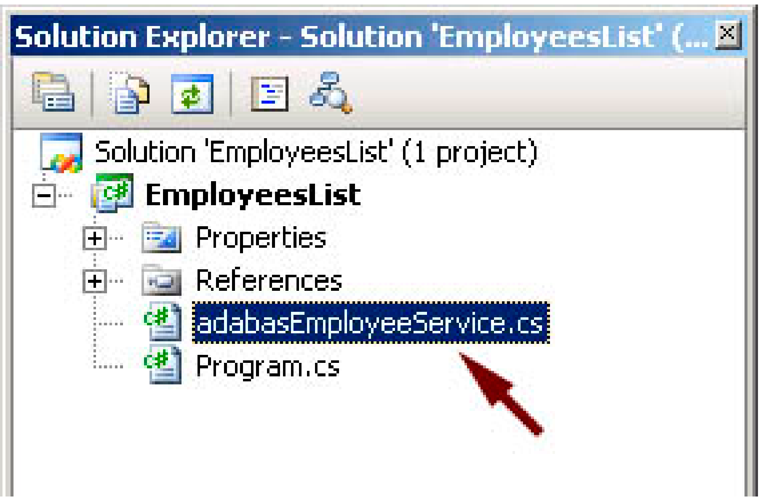


```
using System;
using System.Collections.Generic;
using System.Text;

namespace EmployeesList
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

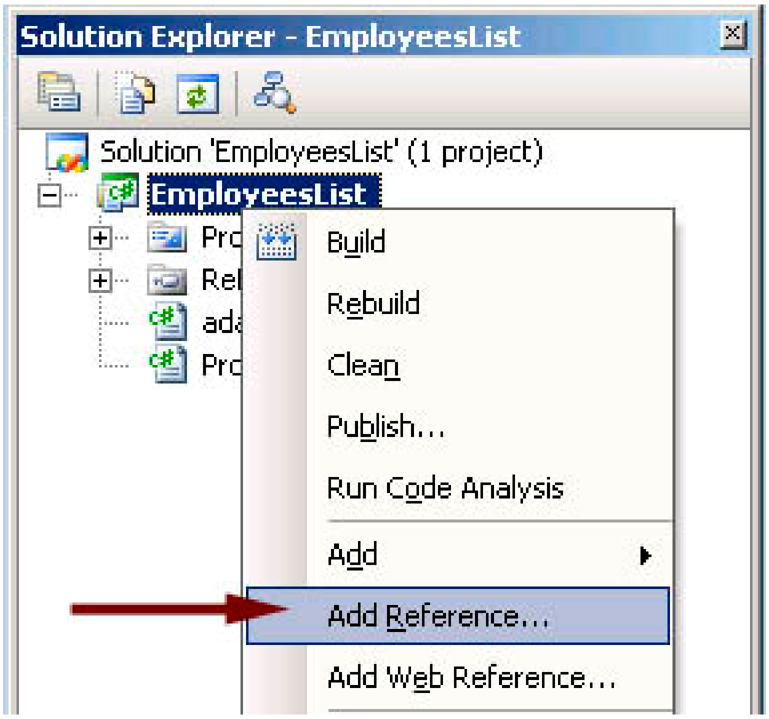
4 First of all, import the generated proxy into the project, right-click on the project name, select **Add -> Existing Item**



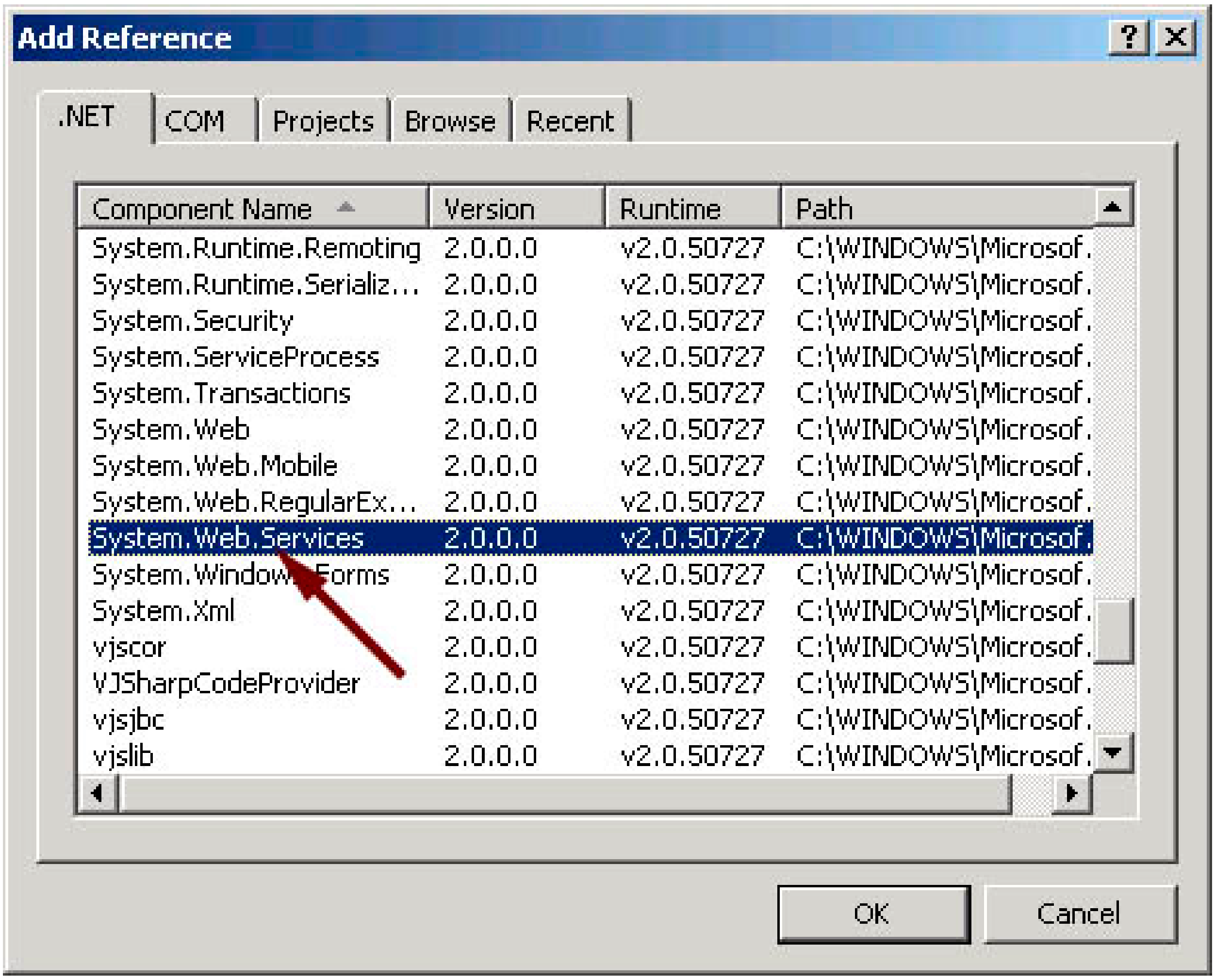
select the AdabasEmployeeService.cs proxy, click **Add**


Solution Explorer - Solution 'EmployeesList' (...
Solution 'EmployeesList' (1 project)
EmployeesList
Properties
References
adabasEmployeeService.cs
Program.cs

You now need to add a reference to the .NET System.Web.Services component implementing the SOAP interface. In the project explorer, right click on the project name, select **Add Reference**

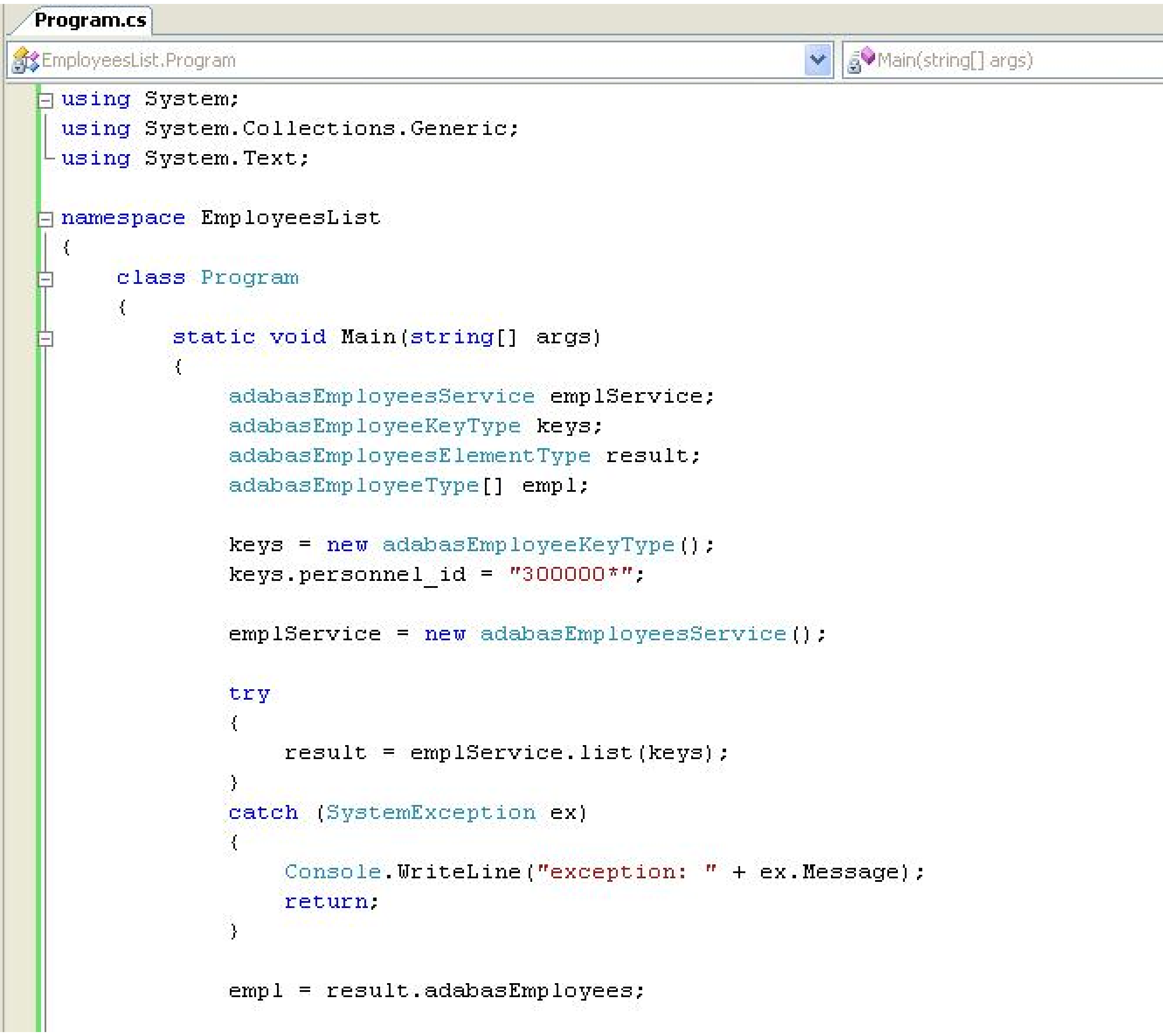


Scroll down to System.Web.Services, click to select it, click to select it, click **OK** to import the reference



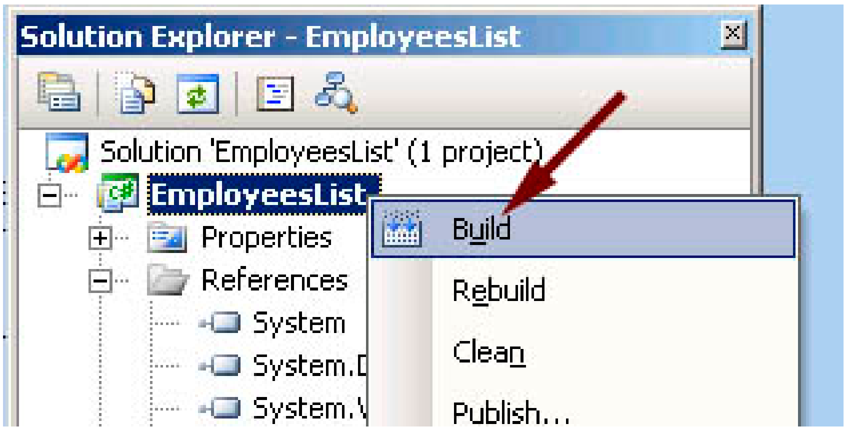
Component Name	Version	Runtime	Path
System.Runtime.Remoting	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.Runtime.Serializ...	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.Security	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.ServiceProcess	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.Transactions	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.Web	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.Web.Mobile	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.Web.RegularEx...	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.Web.Services	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.Windows.Forms	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
System.Xml	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
vjscor	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
VJSharpCodeProvider	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
vjsjbc	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
vjslib	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.

5 Remove the generated code from the newly added class entirely, use (paste) the code from *ASGDemo.cs* to create your first test program accessing Adabas data via Portus.

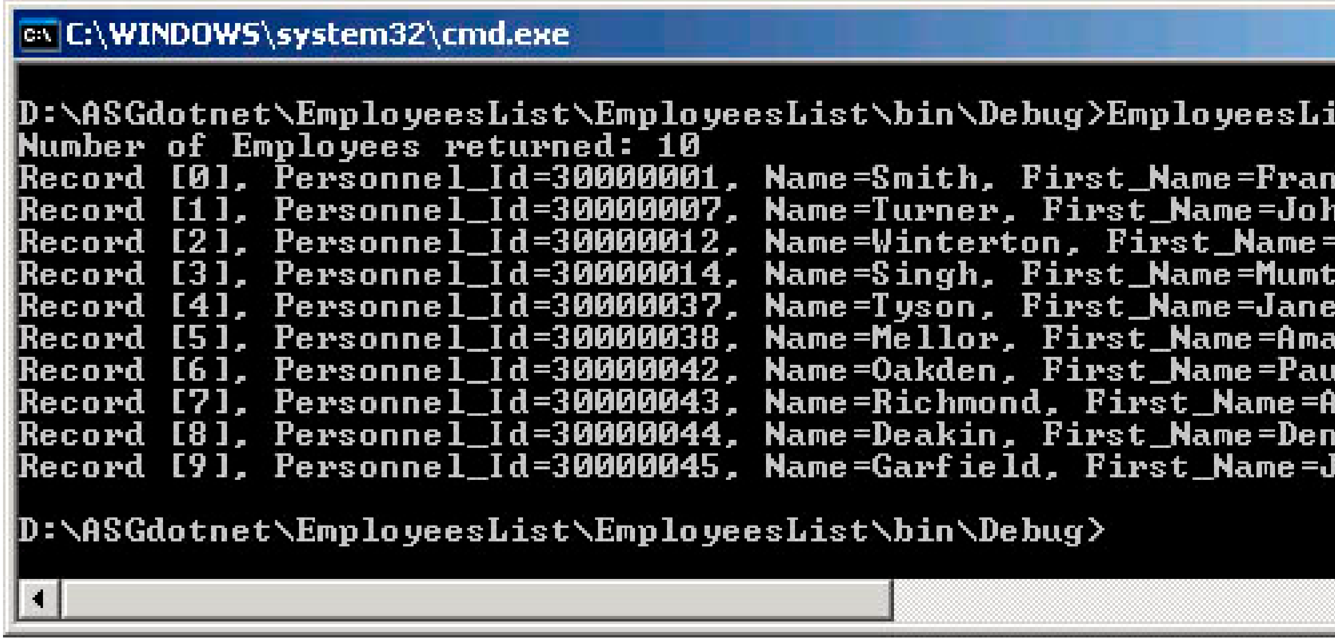


```
using System;
using System.Collections.Generic;
using System.Text;

namespace EmployeesList
{
    class Program
    {
        static void Main(string[] args)
        {
            adabasEmployeesService emplService;
            adabasEmployeeKeyType keys;
            adabasEmployeesElementType result;
            adabasEmployeeType[] empl;

            keys = new adabasEmployeeKeyType();
            keys.personnel_id = "300000*";

            emplService = new adabasEmployeesService();

            try
            {
                result = emplService.list(keys);
            }
            catch (SystemException ex)
            {
                Console.WriteLine("exception: " + ex.Message);
                return;
            }

            empl = result.adabasEmployees;
```

6 Build the application. Right-click on the project name in the project explorer, click **Build**



7 Open a command window, change to the project's build-directory Execute the compiled console application, EmployeesList, the output will look as follows:

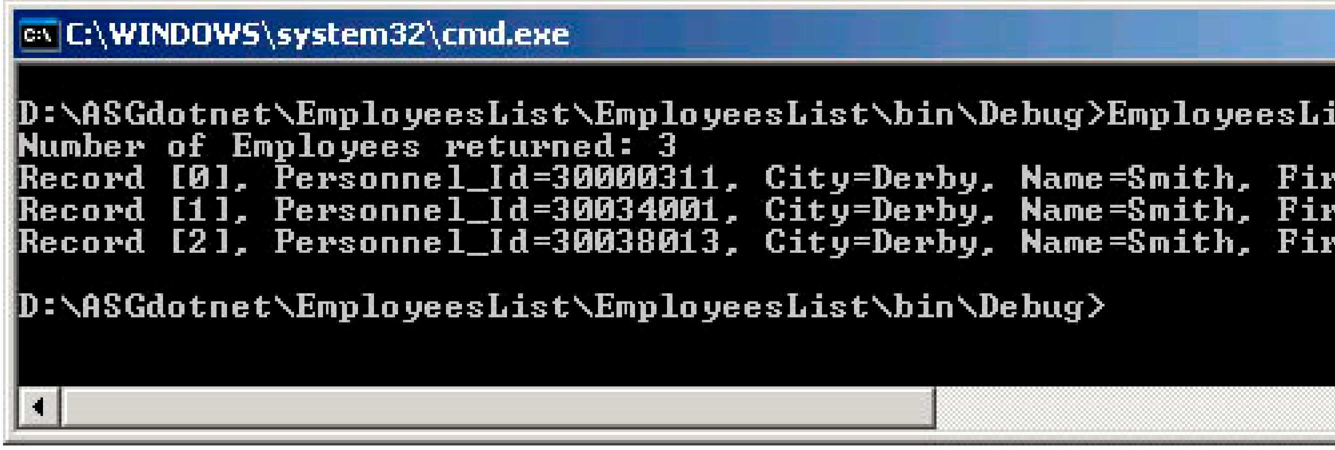


```
D:\ASGdotnet\EmployeesList\EmployeesList\bin\Debug>EmployeesLi
Number of Employees returned: 10
Record [0], Personnel_Id=30000001, Name=Smith, First_Name=Fran
Record [1], Personnel_Id=30000007, Name=Turner, First_Name=Joh
Record [2], Personnel_Id=30000012, Name=Winterton, First_Name=
Record [3], Personnel_Id=30000014, Name=Singh, First_Name=Mumt
Record [4], Personnel_Id=30000037, Name=Tyson, First_Name=Jane
Record [5], Personnel_Id=30000038, Name=Mellor, First_Name=Ama
Record [6], Personnel_Id=30000042, Name=Oakden, First_Name=Pau
Record [7], Personnel_Id=30000043, Name=Richmond, First_Name=A
Record [8], Personnel_Id=30000044, Name=Deakin, First_Name=Den
Record [9], Personnel_Id=30000045, Name=Garfield, First_Name=J

D:\ASGdotnet\EmployeesList\EmployeesList\bin\Debug>
```

8 This sample selects all "Employees" records with a personnel-id of 4000004n, you may want to experiment varying the key data, this is easily done by modifying the properties passed to the generated classes. E.g. try the following to list all records for "Employees" whose names start "SMI", living in cities with names starting "D".

```
keys.name = "SMI*";
keys.city = "D*;
```

The output will look like this:

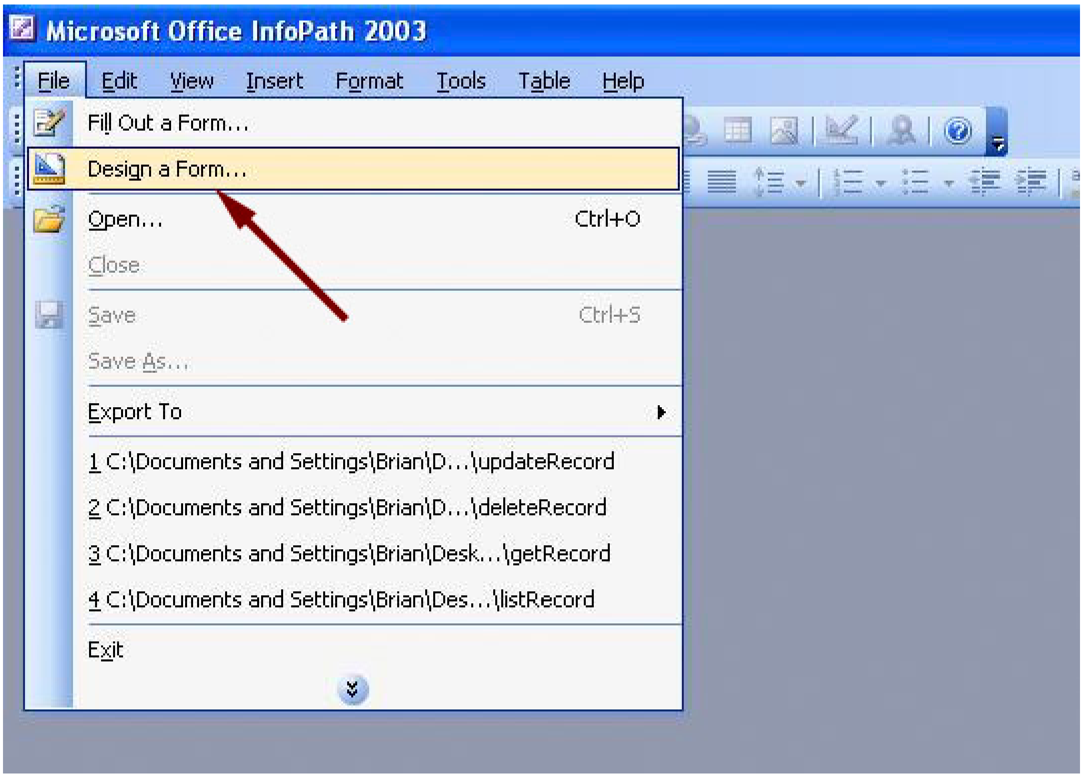


```
D:\ASGdotnet\EmployeesList\EmployeesList\bin\Debug>EmployeesLi
Number of Employees returned: 3
Record [0], Personnel_Id=30000311, City=Derby, Name=Smith, Fir
Record [1], Personnel_Id=30034001, City=Derby, Name=Smith, Fir
Record [2], Personnel_Id=30038013, City=Derby, Name=Smith, Fir

D:\ASGdotnet\EmployeesList\EmployeesList\bin\Debug>
```

Accessing Adabas through Microsoft InfoPath

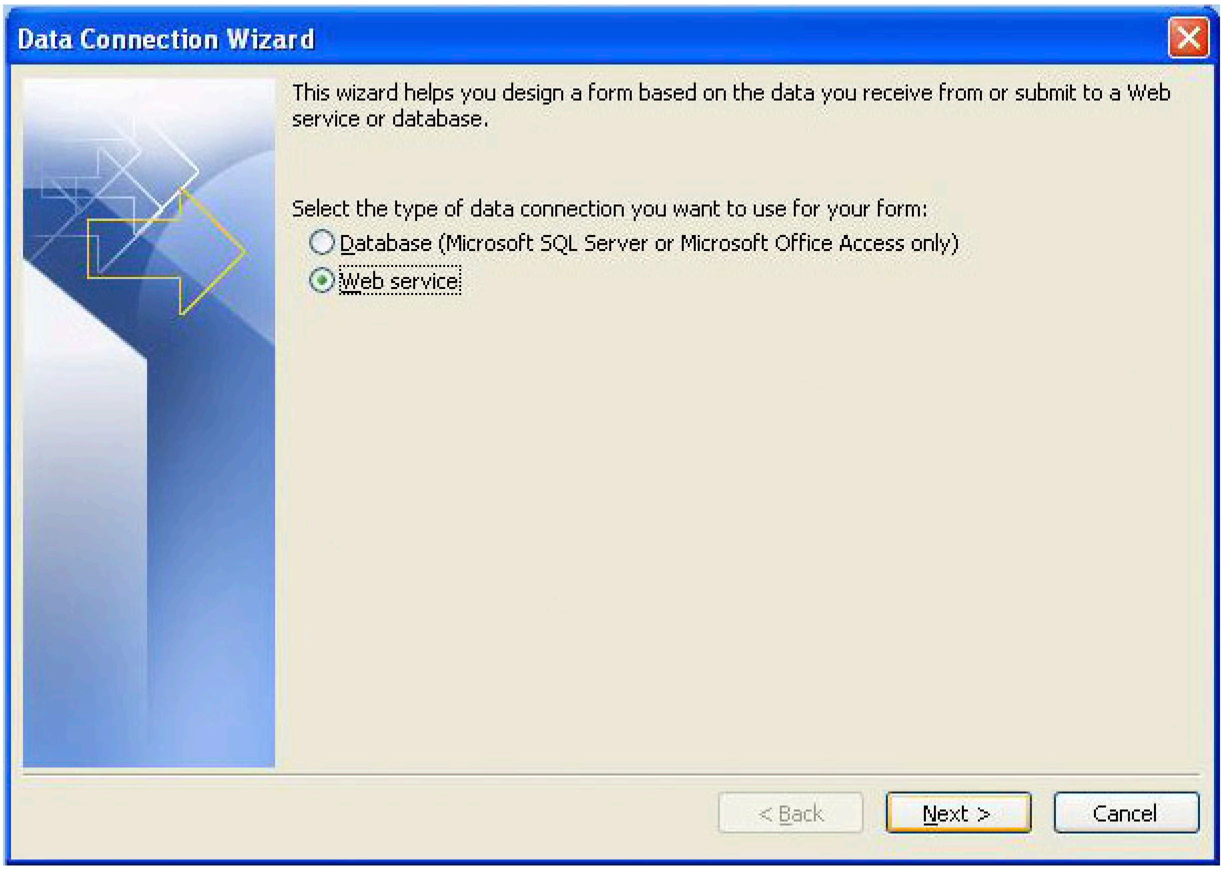
This tutorial demonstrates how to invoke operations on an Adabas DataSource exposed as a "Web service" through Portus from Microsoft InfoPath.

1 From the InfoPath main menu bar, select **File -> Design a Form**.


Microsoft Office InfoPath 2003
File
Edit
View
Insert
Format
Tools
Table
Help
Fill Out a Form...
Design a Form...
Open... Ctrl+O
Close
Save Ctrl+S
Save As...
Export To
1 C:\Documents and Settings\Brian\D...\updateRecord
2 C:\Documents and Settings\Brian\D...\deleteRecord
3 C:\Documents and Settings\Brian\Desk...\getRecord
4 C:\Documents and Settings\Brian\Des...\listRecord
Exit

2 A panel will appear on your right hand side - choose **New** from **Data Connection**

This will start a dialog, first select **Web Service**, click **Next**



Data Connection Wizard

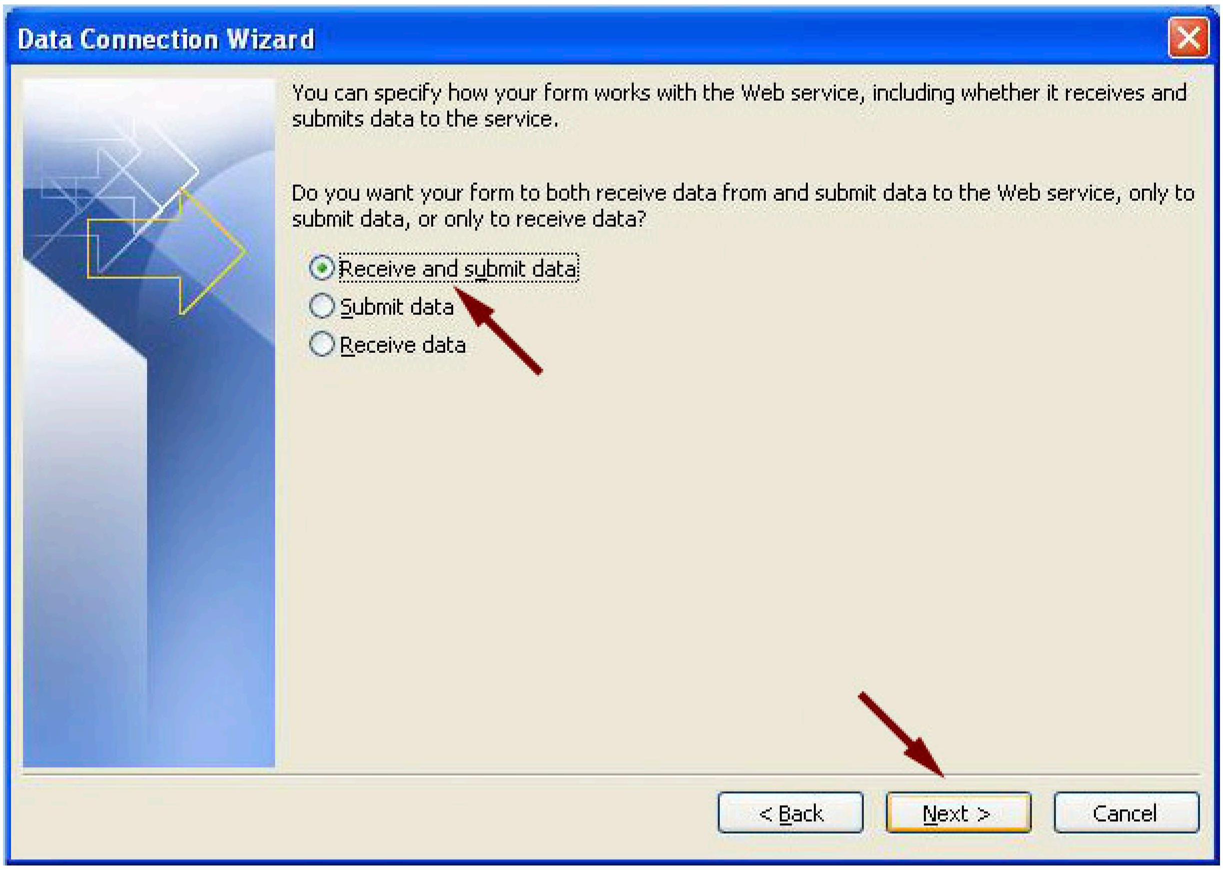
This wizard helps you design a form based on the data you receive from or submit to a Web service or database.

Select the type of data connection you want to use for your form:

○ Database (Microsoft SQL Server or Microsoft Office Access only)

● Web service

Select **Receive and submit data**, click **Next**



Data Connection Wizard

You can specify how your form works with the Web service, including whether it receives and submits data to the service.

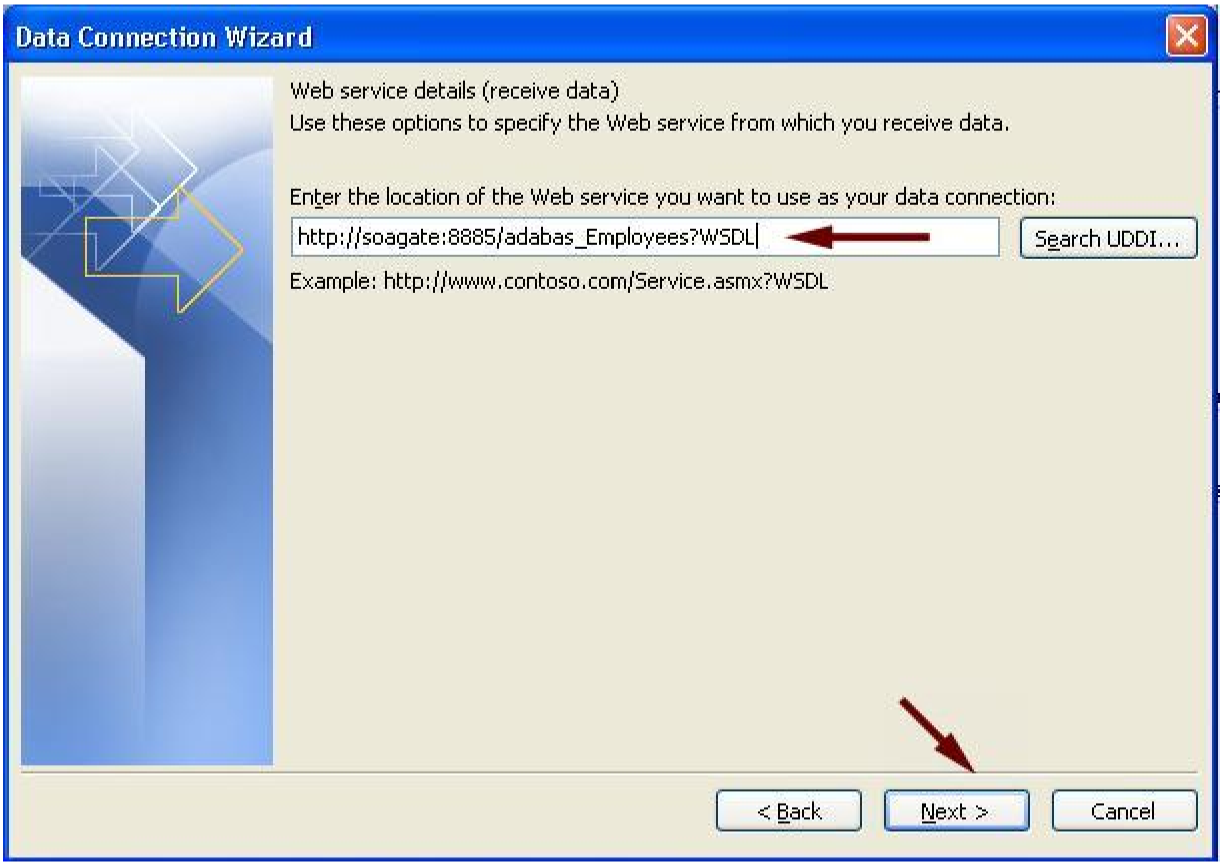
Do you want your form to both receive data from and submit data to the Web service, only to submit data, or only to receive data?

● Receive and submit data
○ Submit data
○ Receive data

Enter the URL, the WSDL for the "Employees" demo file is exposed as:

```
http://soagate:8885/adabas_Employees?WSDL
```

Click **Next**



Data Connection Wizard

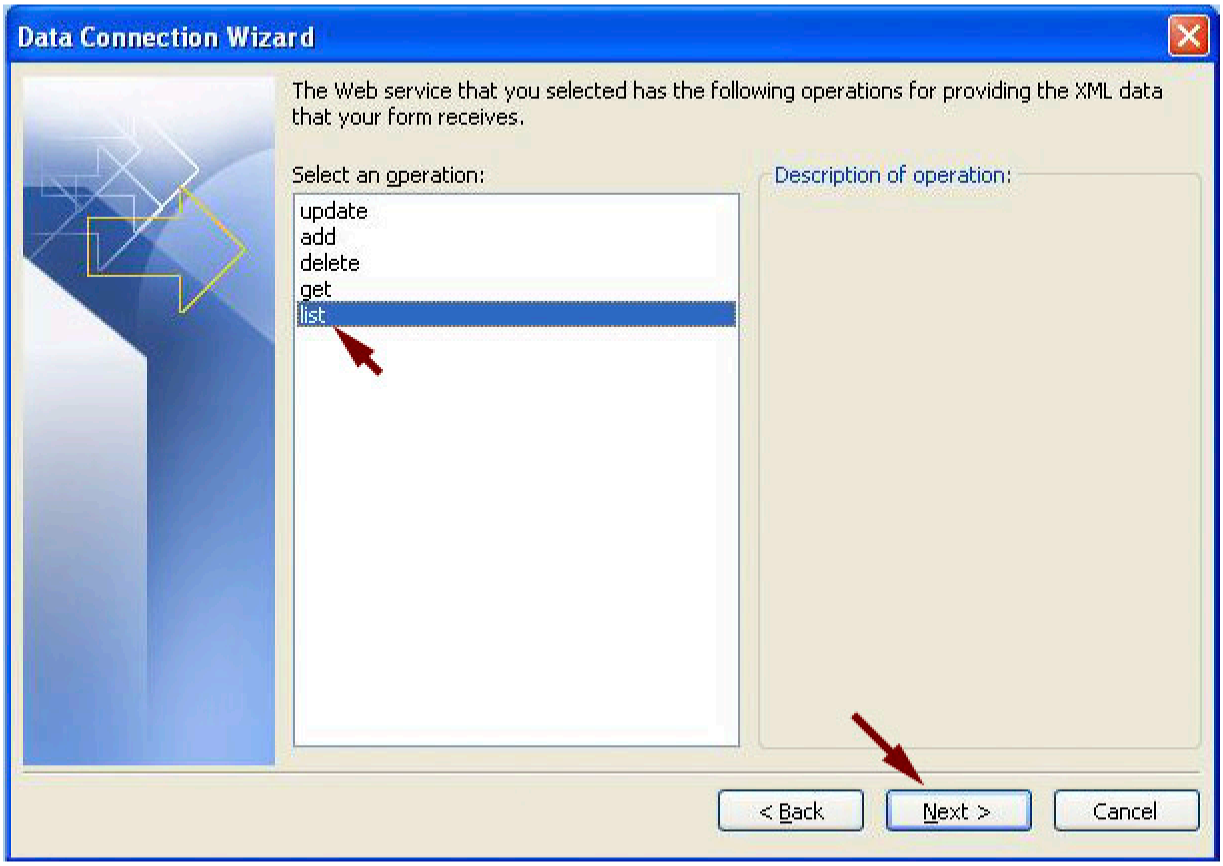
Web service details (receive data)
Use these options to specify the Web service from which you receive data.

Enter the location of the Web service you want to use as your data connection:

http://soagate:8885/adabas_Employees?WSDL

Example: http://www.contoso.com/Service.asmx?WSDL

Select the **list** method, click **Next**



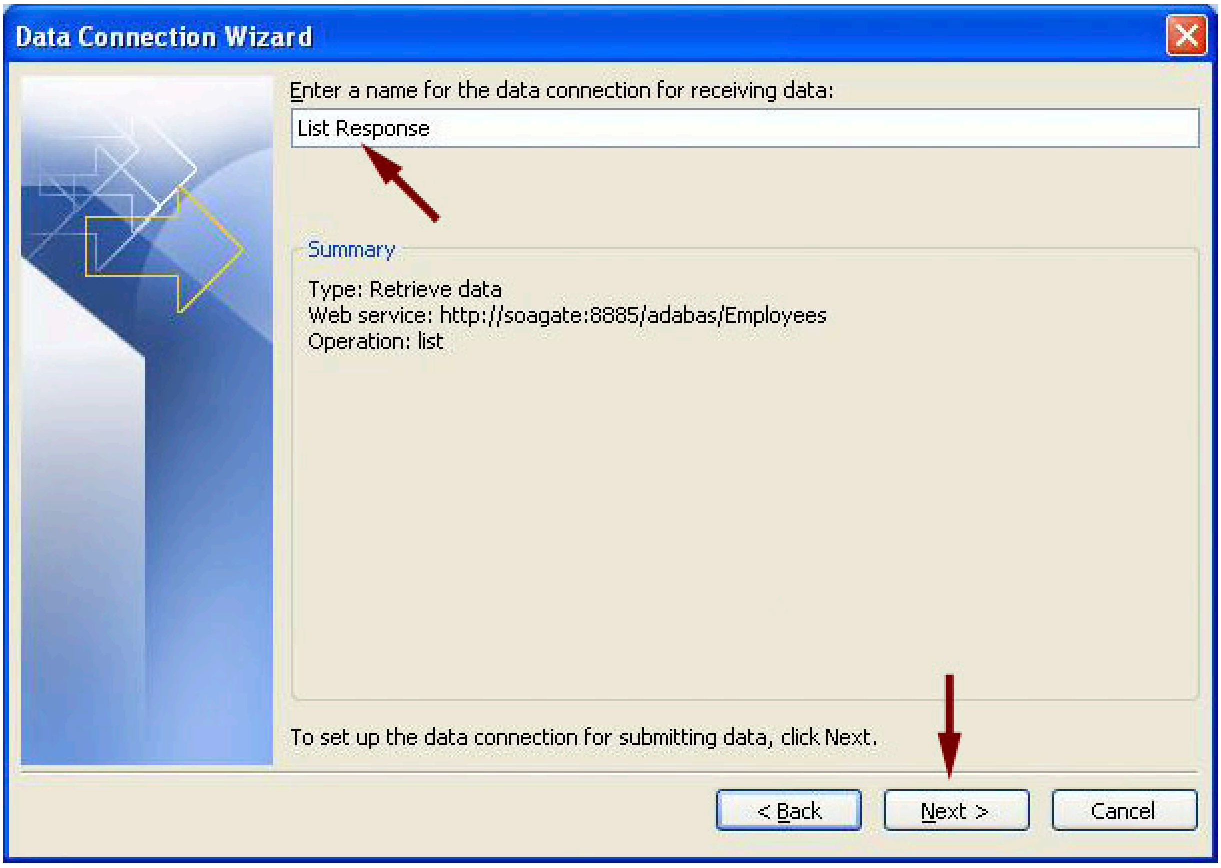
Data Connection Wizard

The Web service that you selected has the following operations for providing the XML data that your form receives.

Select an operation:

update
add
delete
get
list

Assign a name to the response document, click **Next**



Data Connection Wizard

Enter a name for the data connection for receiving data:

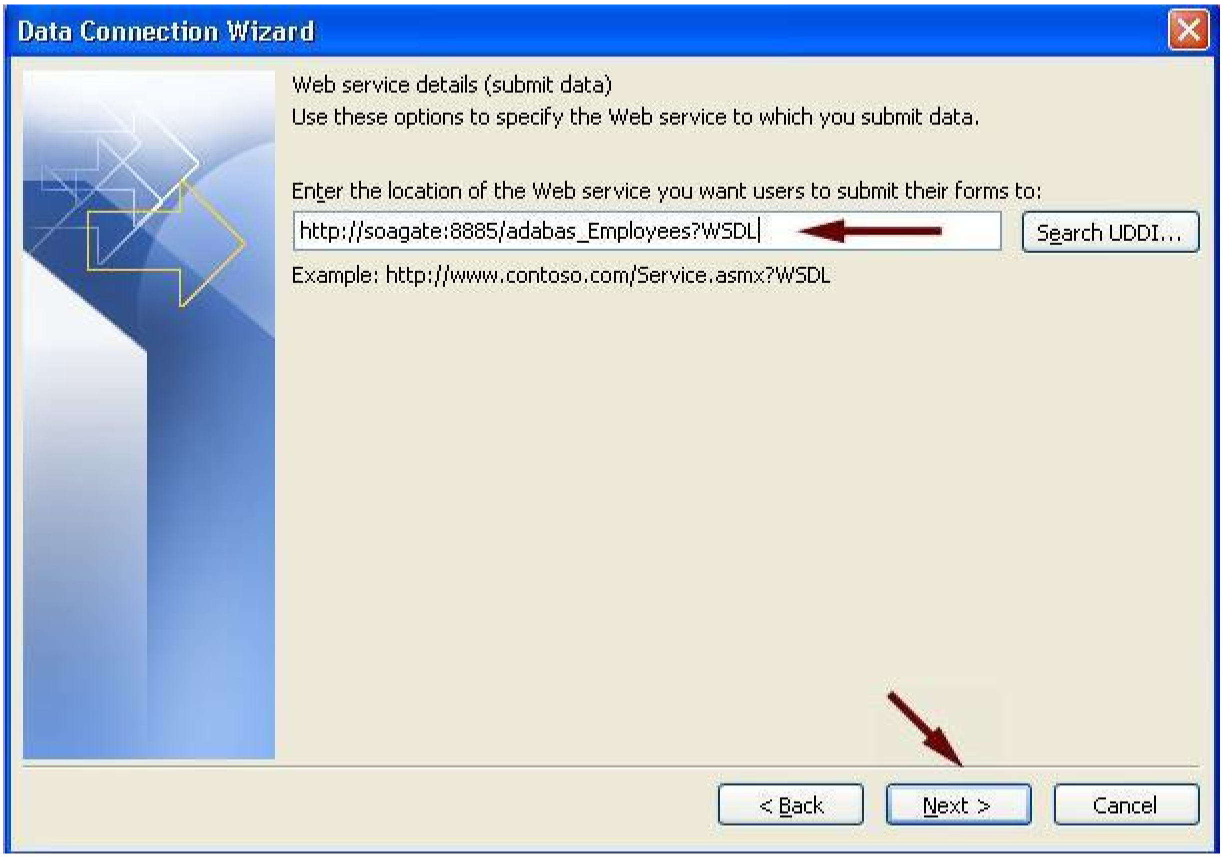
List Response

Summary

Type: Retrieve data
Web service: http://soagate:8885/adabas/Employees
Operation: list

To set up the data connection for submitting data, click Next.

Enter the same URL again, click **Next**



Data Connection Wizard

Web service details (submit data)

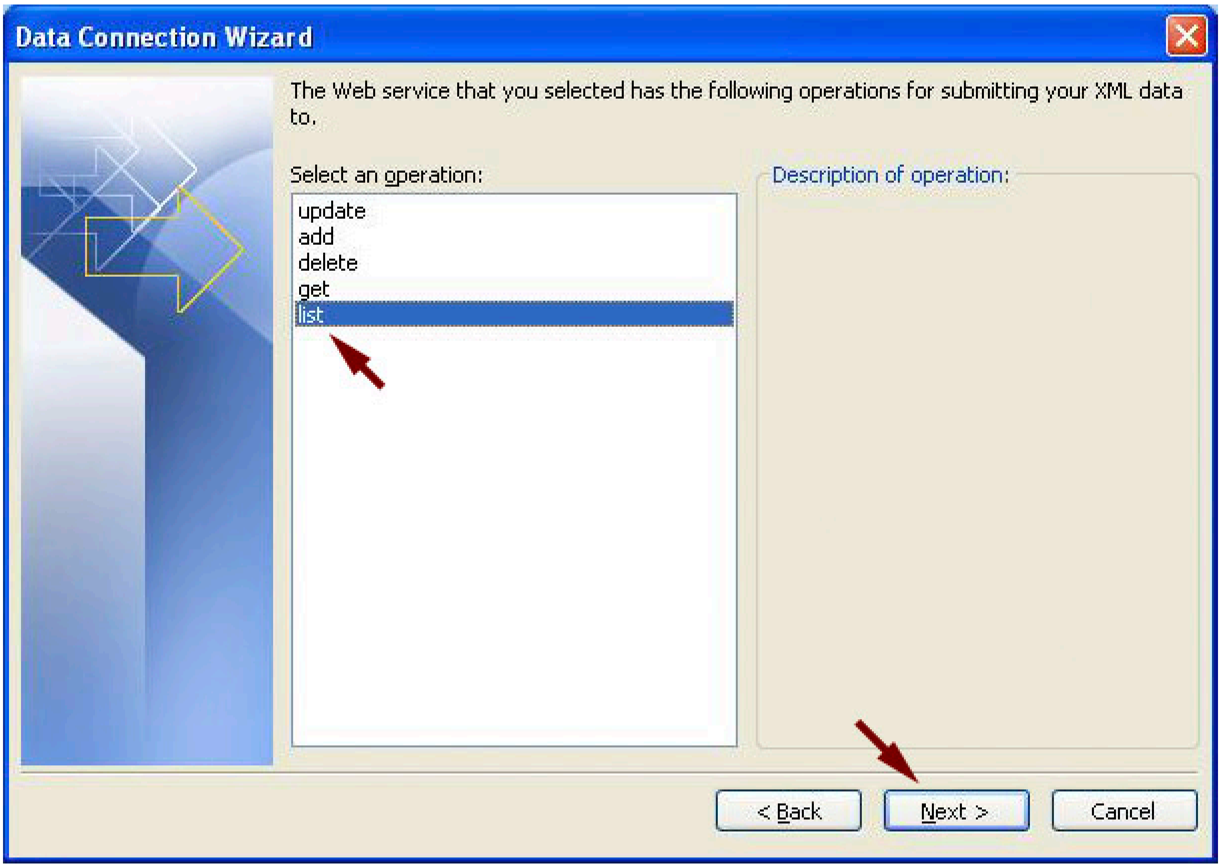
Use these options to specify the Web service to which you submit data.

Enter the location of the Web service you want users to submit their forms to:

http://soagate:8885/adabas_Employees?WSDL

Example: http://www.contoso.com/Service.asmx?WSDL

Choose list and click **Next**



Data Connection Wizard

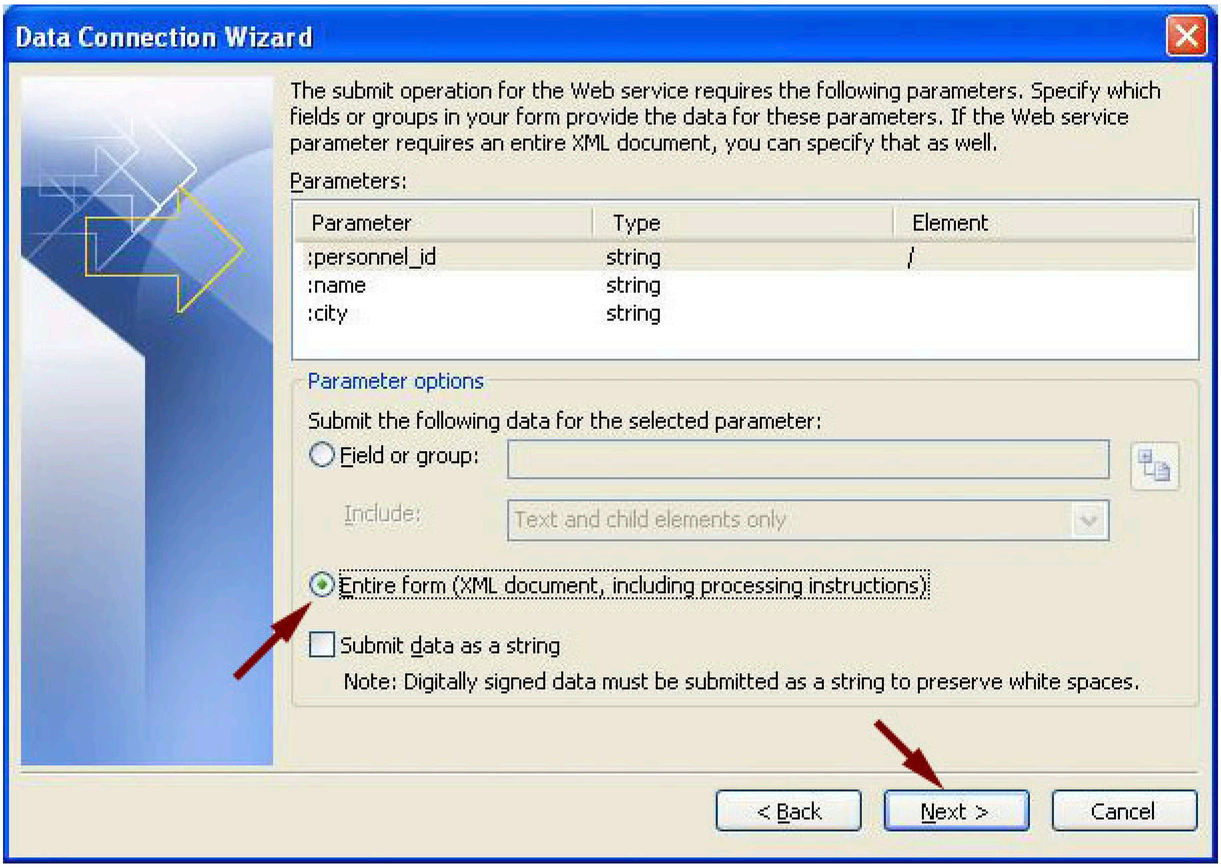
The Web service that you selected has the following operations for submitting your XML data to.

Select an operation:

- update
- add
- delete
- get
- list

Description of operation:

Select "**Entire form (XML document....**", click **Next**



Data Connection Wizard

The submit operation for the Web service requires the following parameters. Specify which fields or groups in your form provide the data for these parameters. If the Web service parameter requires an entire XML document, you can specify that as well.

Parameters:

Parameter	Type	Element
:personnel_id	string	/
:name	string	
:city	string	

Parameter options

Submit the following data for the selected parameter:

○ Field or group:

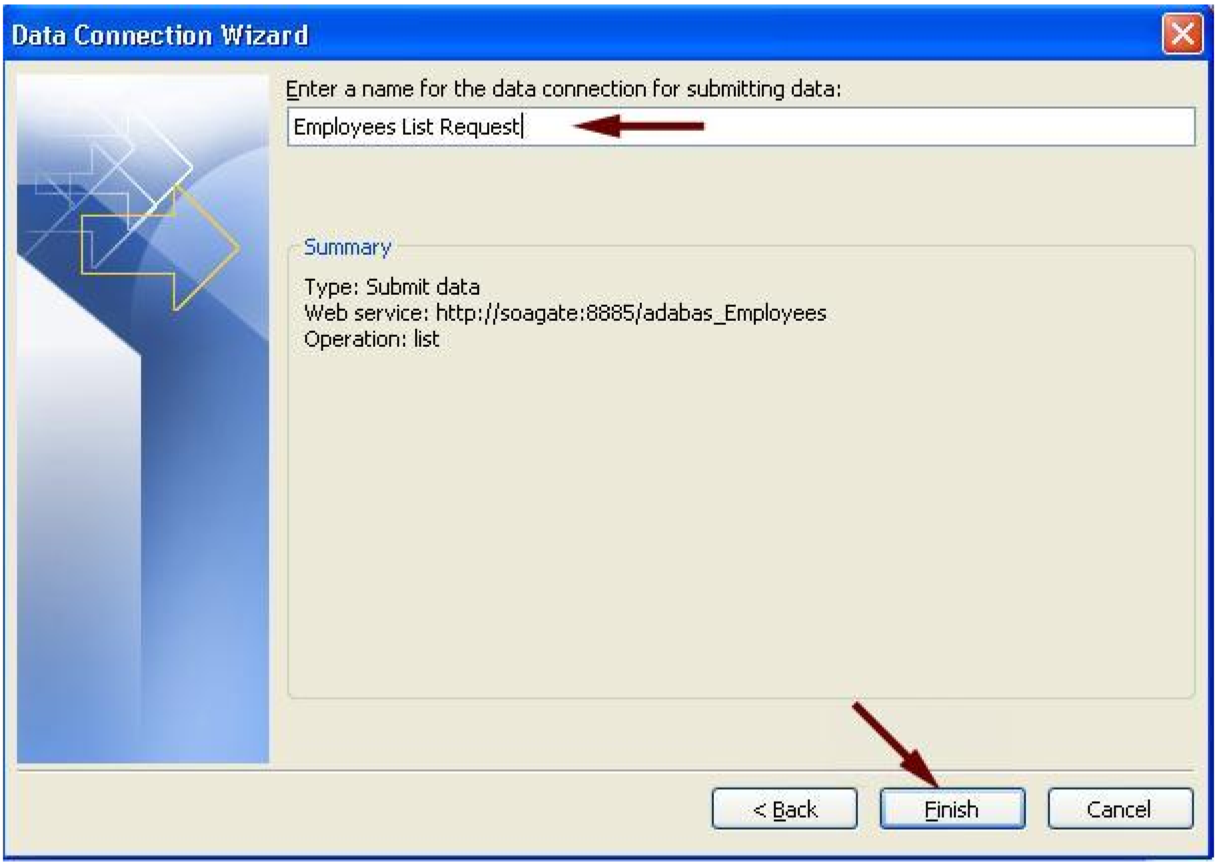
Include: Text and child elements only

● Entire form (XML document, including processing instructions)

□ Submit data as a string

Note: Digitally signed data must be submitted as a string to preserve white spaces.

Assign a name to the Send Connection, click **Finish**



Data Connection Wizard

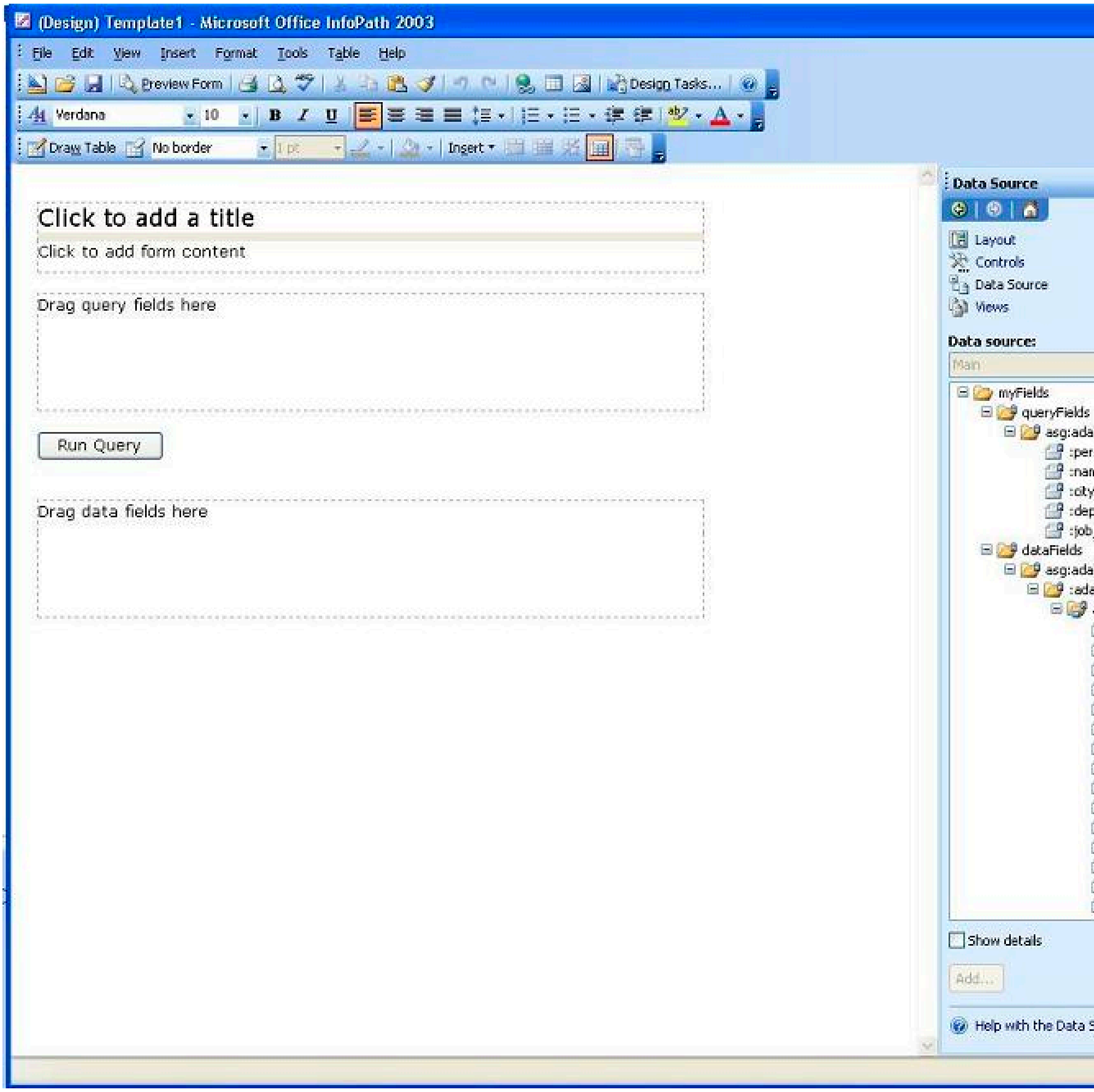
Enter a name for the data connection for submitting data:

Employees List Request

Summary

Type: Submit data
Web service: http://soagate:8885/adabas_Employees
Operation: list

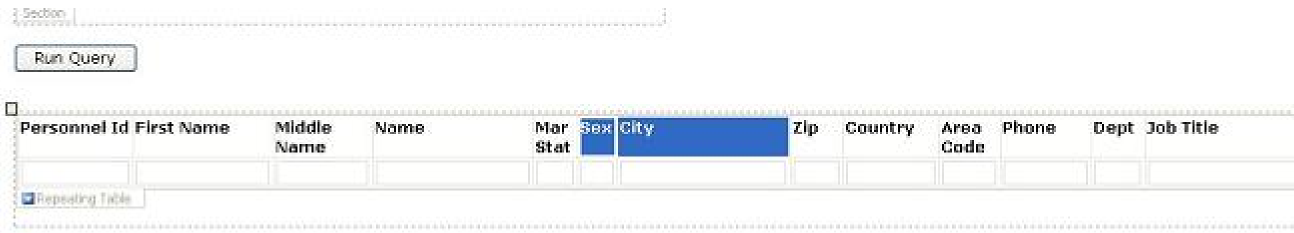
3 You will now be presented with a form



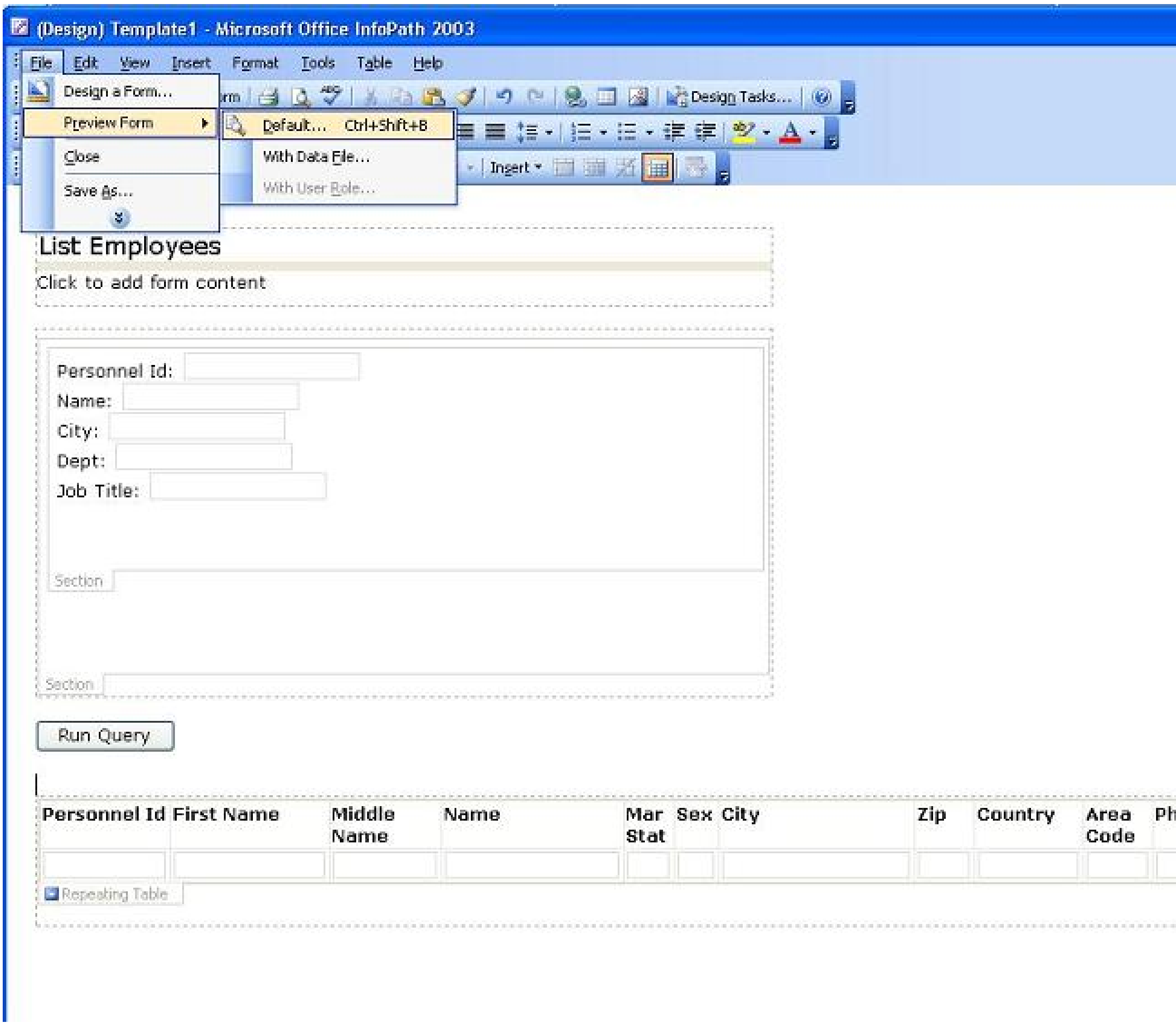
Do the following:

- Enter a title "List Employees" for example
- Expand the **queryFields**, drag them to the area "Drag query fields here"
- Expand the **dataFields**, right-click on the **adabasEmployee** element. Now drag the adabasEmployee to the area "Drag data fields here", insert it as a **Repeating table**

  At this point, it might make sense to resize the table and the fields in the "repeating table"



- Once you are happy with how your form looks like, select **File -> Preview form -> Default**



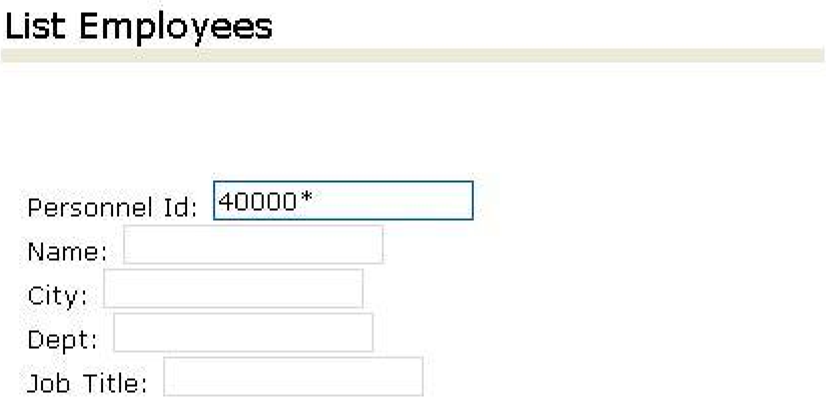
List Employees

Click to add form content

Personnel Id:	
Name:	
City:	
Dept:	
Job Title:	

Personnel Id	First Name	Middle Name	Name	Mar Stat	Sex	City	Zip	Country	Area Code	Ph

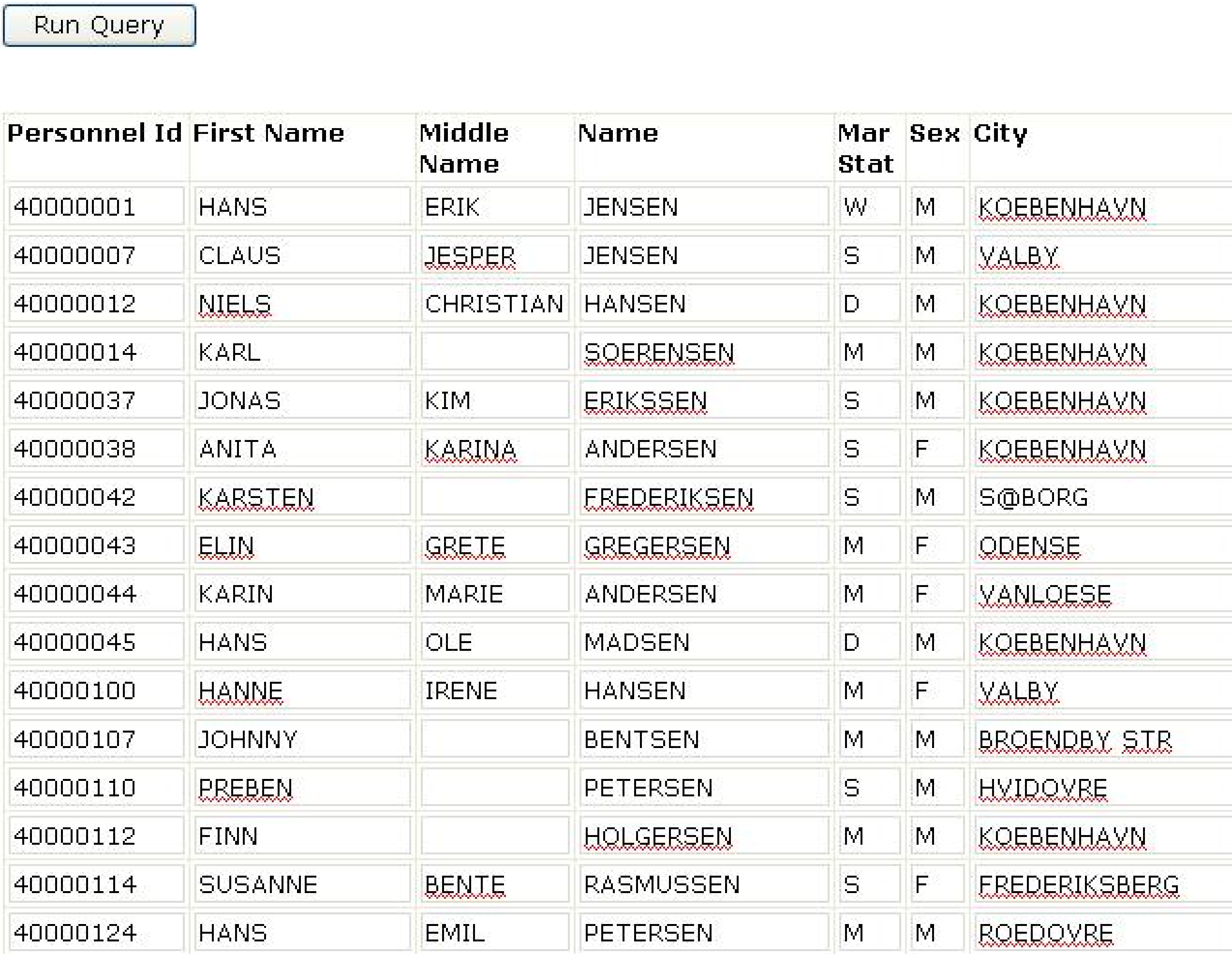
- You will be presented with a form, enter 4000004* in the Personnel Id field and send the request to the server



List Employees

Personnel Id:	40000*
Name:	
City:	
Dept:	
Job Title:	

- Your table will now be populated with the data based on your request:



Personnel Id	First Name	Middle Name	Name	Mar Stat	Sex	City
40000001	HANS	ERIK	JENSEN	W	M	KOEBENHAVN
40000007	CLAUS	JESPER	JENSEN	S	M	VALBY
40000012	NIELS	CHRISTIAN	HANSEN	D	M	KOEBENHAVN
40000014	KARL		SOERENSEN	M	M	KOEBENHAVN
40000037	JONAS	KIM	ERIKSSEN	S	M	KOEBENHAVN
40000038	ANITA	KARINA	ANDERSEN	S	F	KOEBENHAVN
40000042	KARSTEN		FREDERIKSEN	S	M	S@BORG
40000043	ELIN	GRETE	GREGERSEN	M	F	ODENSE
40000044	KARIN	MARIE	ANDERSEN	M	F	VANLOESE
40000045	HANS	OLE	MADSEN	D	M	KOEBENHAVN
40000100	HANNE	IRENE	HANSEN	M	F	VALBY
40000107	JOHNNY		BENTSEN	M	M	BROENDBY STR
40000110	PREBEN		PETERSEN	S	M	HVIDOVRE
40000112	FINN		HOLGERSEN	M	M	KOEBENHAVN
40000114	SUSANNE	BENTE	RASMUSSEN	S	F	FREDERIKSBERG
40000124	HANS	EMIL	PETERSEN	M	M	ROEDOVRE

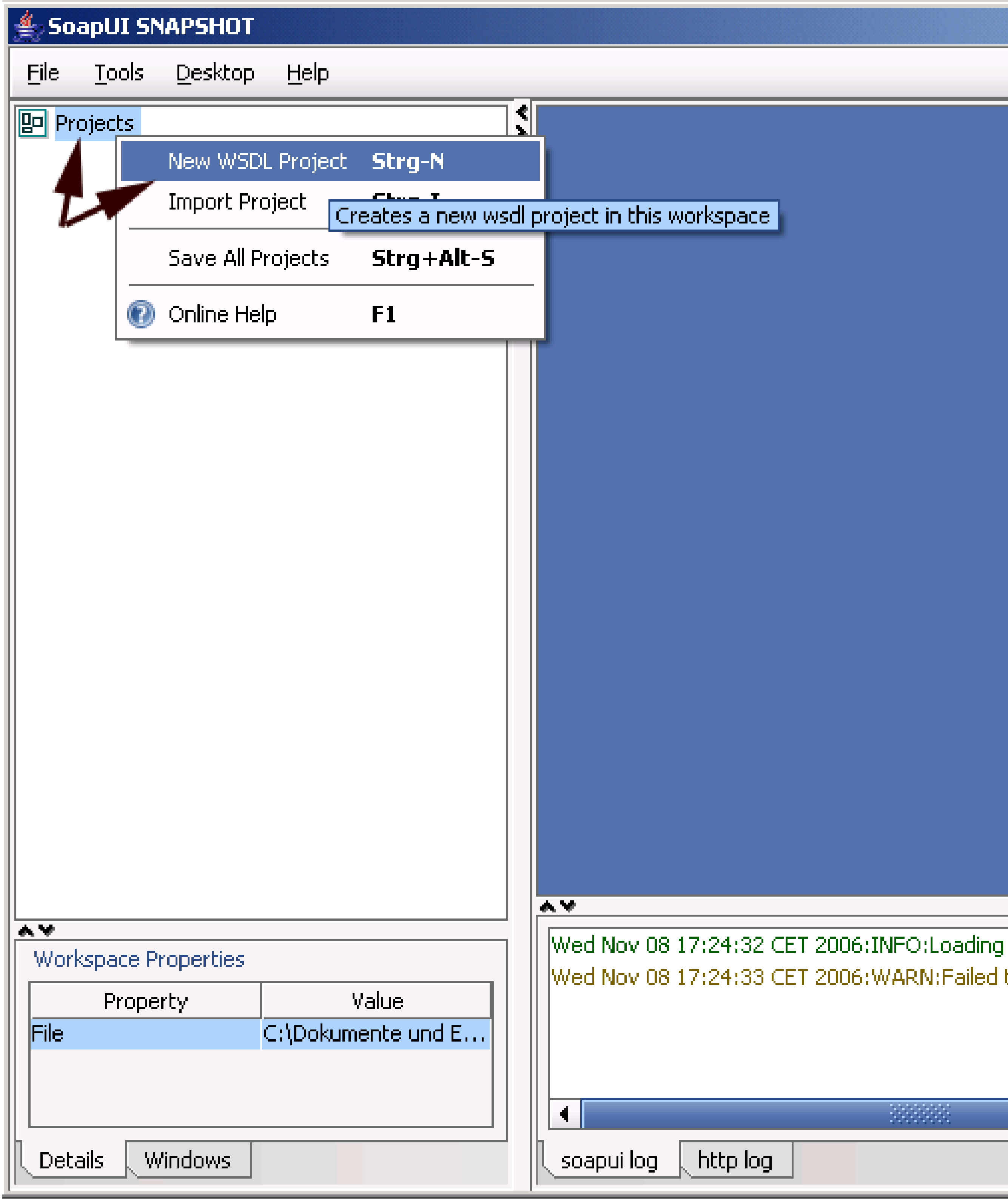
Accessing Adabas through SoapUI

This simple scenario demonstrates how to invoke operations on an Adabas DataSource exposed as a "Web service" through Portus from SoapUI.

soapUI is a (freeware) desktop application for inspecting, invoking, developing and functional / load / compliance testing of web services over HTTP and can be downloaded *here*.

Additionally, soapUI can be integrated into the Eclipse framework, read *here* for more information.

1 When starting soapUI for the very first time an "empty" workspace is generated, right-click on the top-level **Projects** item, select **New WSDL Project** to create your first soapUI project.

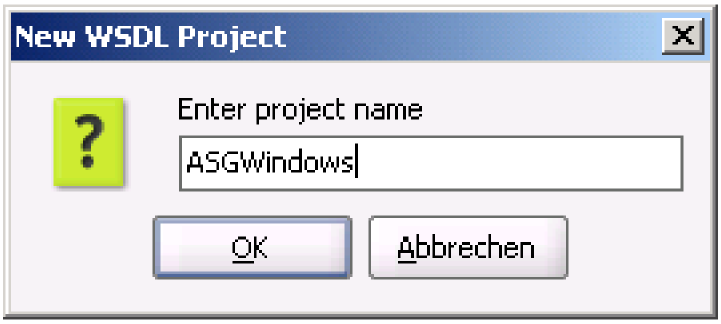


Creates a new wsdl project in this workspace

Property	Value
File	C:\Dokumente und E...

```
Wed Nov 08 17:24:32 CET 2006:INFO:Loading
Wed Nov 08 17:24:33 CET 2006:WARN:Failed l
```

You will be prompted for a project name, enter one and click OK:

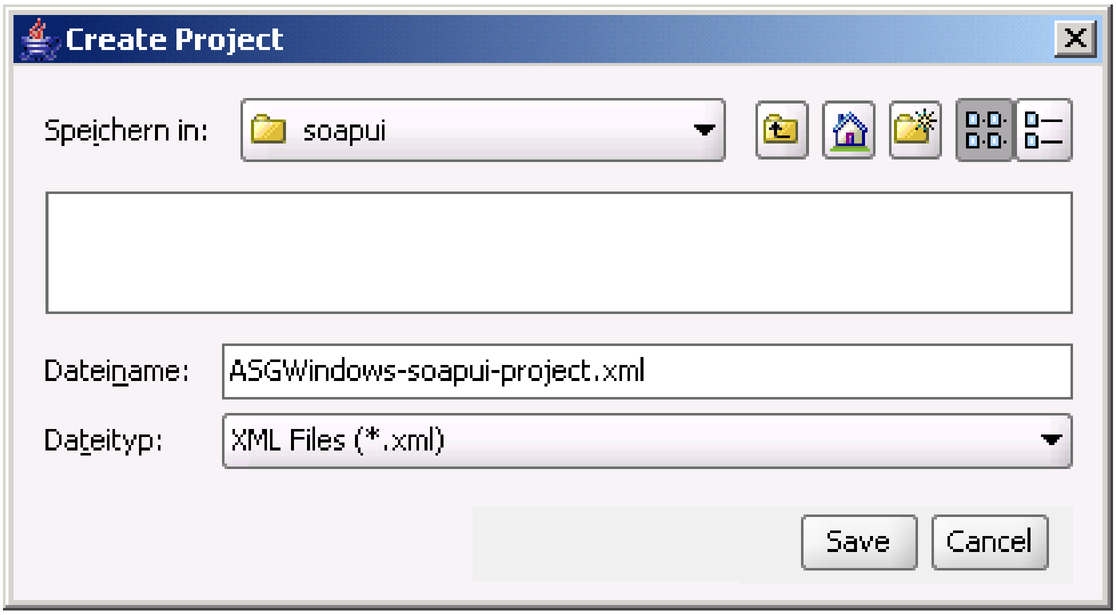


New WSDL Project

Enter project name

ASGWindows

Specify a target location for the project files, click **Save**:



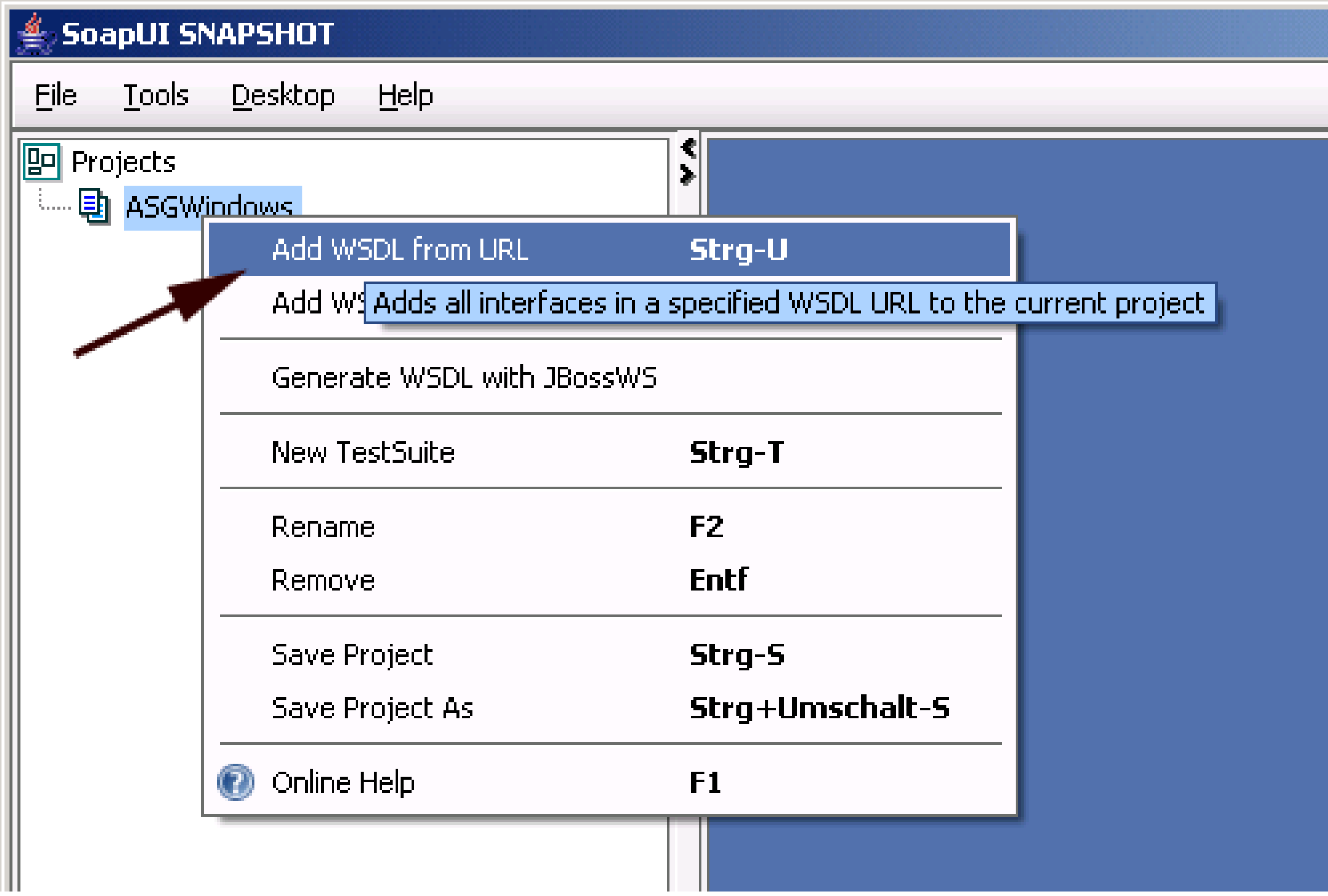
Create Project

Speichern in: soapui

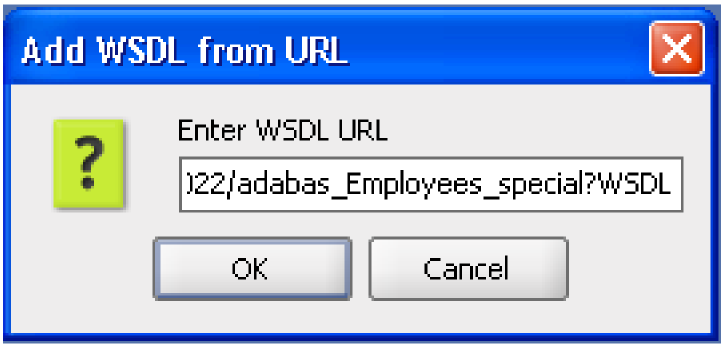
Dateiname: ASGWindows-soapui-project.xml

Dateityp: XML Files (*.xml)

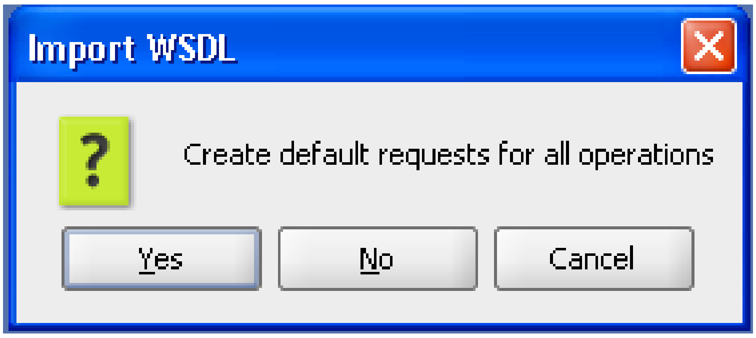
2 Now import an Portus "web service" into the newly created project, right-click the project name, then select **Add WSDL from URL**



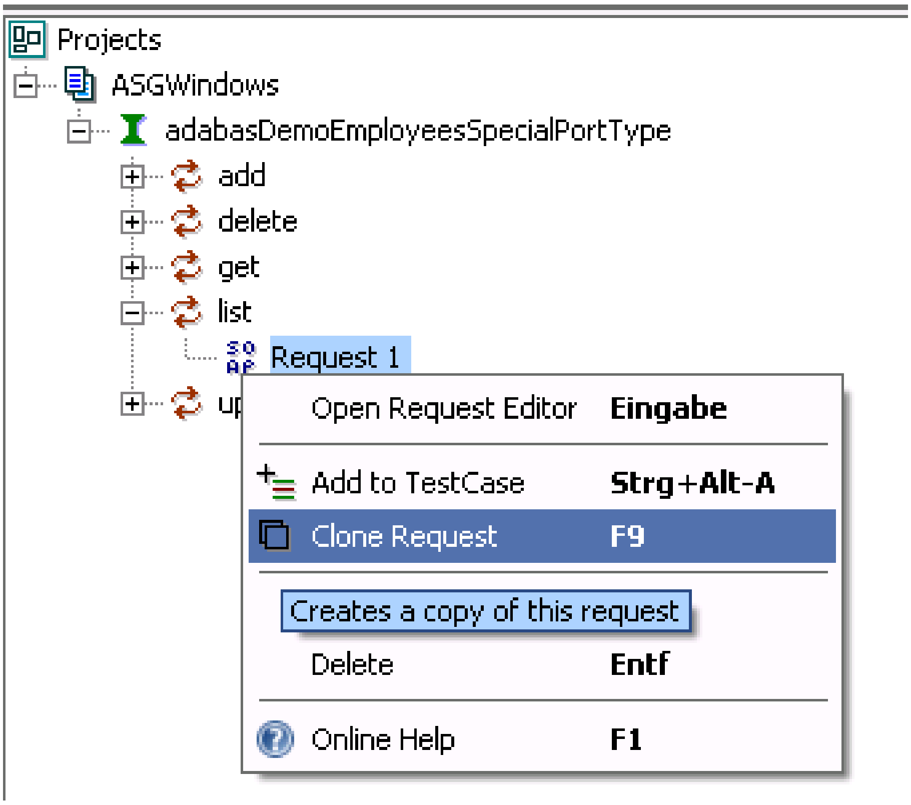
Enter the URI of the adabas_Employees_special resource, as in http://<yourserverhost>:<yourserverport>/adabas_Employees_special?WSDL and click OK to import the webservice definitions from the resources / WSDL



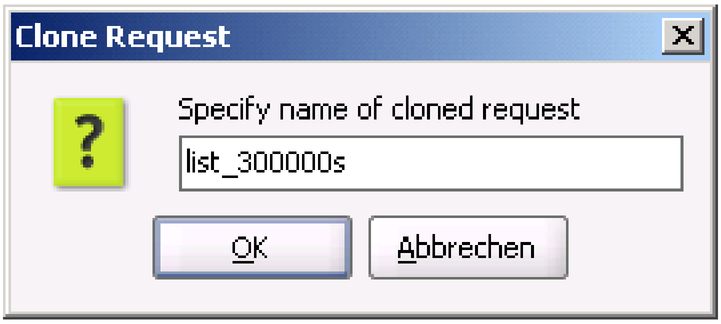
You will be asked if default requests for all operations are to be created, click **Yes**.



3 Default requests have been generated for all Portus operations for a webservice, undfold the "list" request by clicking the **plus** sign left of it, right-click the created **Request 1**, select **Clone Request**, this allows for unlimited duplication of the original request, which may be desirable when testing various options or "canning" requests.


Projects
ASGWindows
adabasDemoEmployeesSpecialPortType
add
delete
get
list
Request 1
up
Open Request Editor Eingabe
Add to TestCase Strg+Alt-A
Clone Request F9
Creates a copy of this request
Delete Entf
Online Help F1

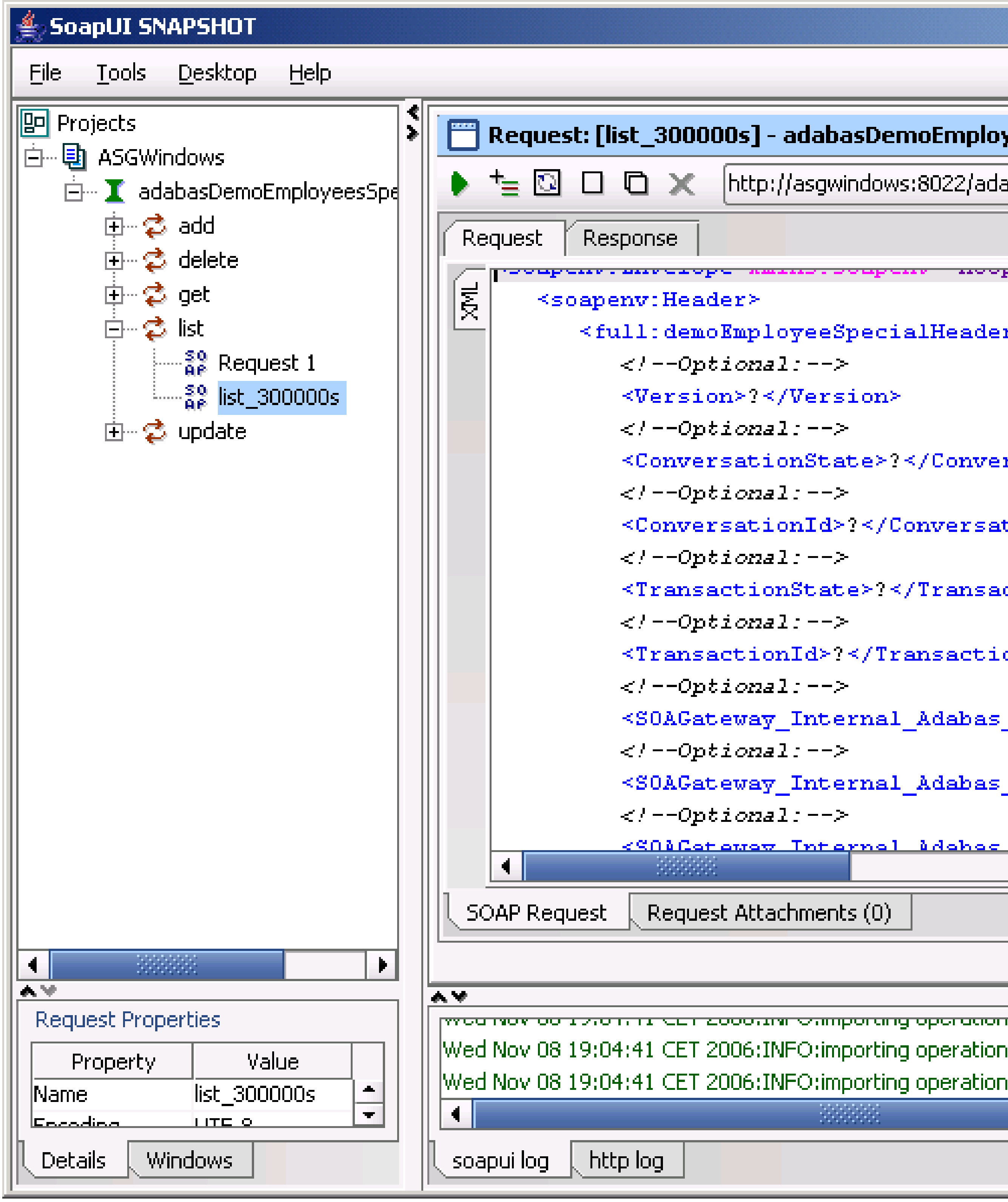
Assign a name to the cloned request, click **OK**



Specify name of cloned request

list_300000s

4 soapUI now opens the request document.



```
<soapenv:Envelope xmlns:soapenv="http
   <soapenv:Header>
      <full:demoEmployeeSpecialHeader
         <!--Optional:-->
         <Version>?</Version>
         <!--Optional:-->
         <ConversationState>?</Conver
         <!--Optional:-->
         <ConversationId>?</Conversat
         <!--Optional:-->
         <TransactionState>?</Transac
         <!--Optional:-->
         <TransactionId>?</Transactio
         <!--Optional:-->
         <SOAGateway_Internal_Adabas_
         <!--Optional:-->
         <SOAGateway_Internal_Adabas_
         <!--Optional:-->
         <SOAGateway_Internal_Adabas_
```

Property	Value
Name	list_300000s
Encoding	UTF-8

```
Wed Nov 08 19:04:41 CET 2006:INFO:importing operation
Wed Nov 08 19:04:41 CET 2006:INFO:importing operation
Wed Nov 08 19:04:41 CET 2006:INFO:importing operation
```

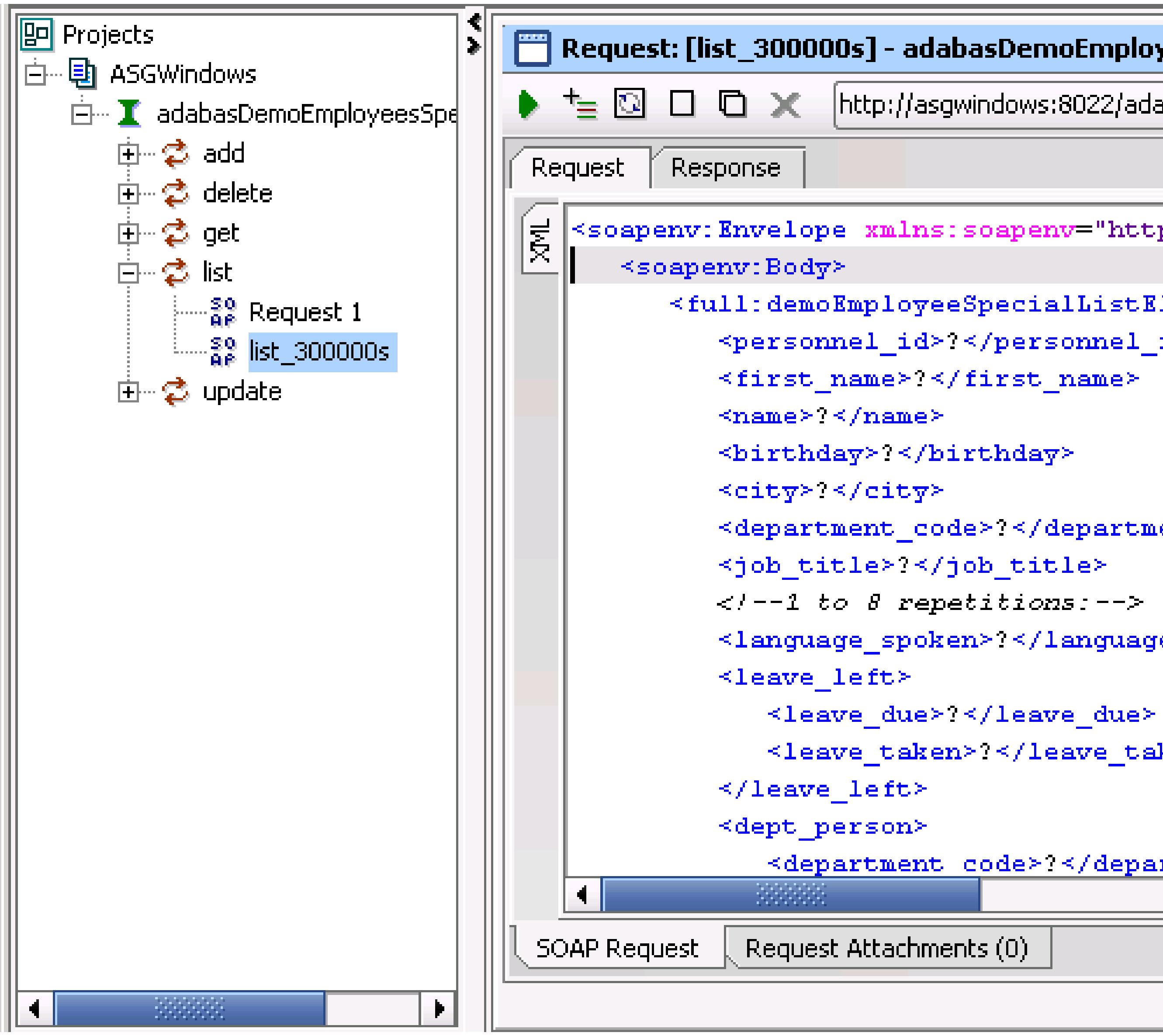
Remove the soapenv:Header section, everything from, and including,

```
<soapenv:Header>
```

to

```
</soapenv:Header>
```

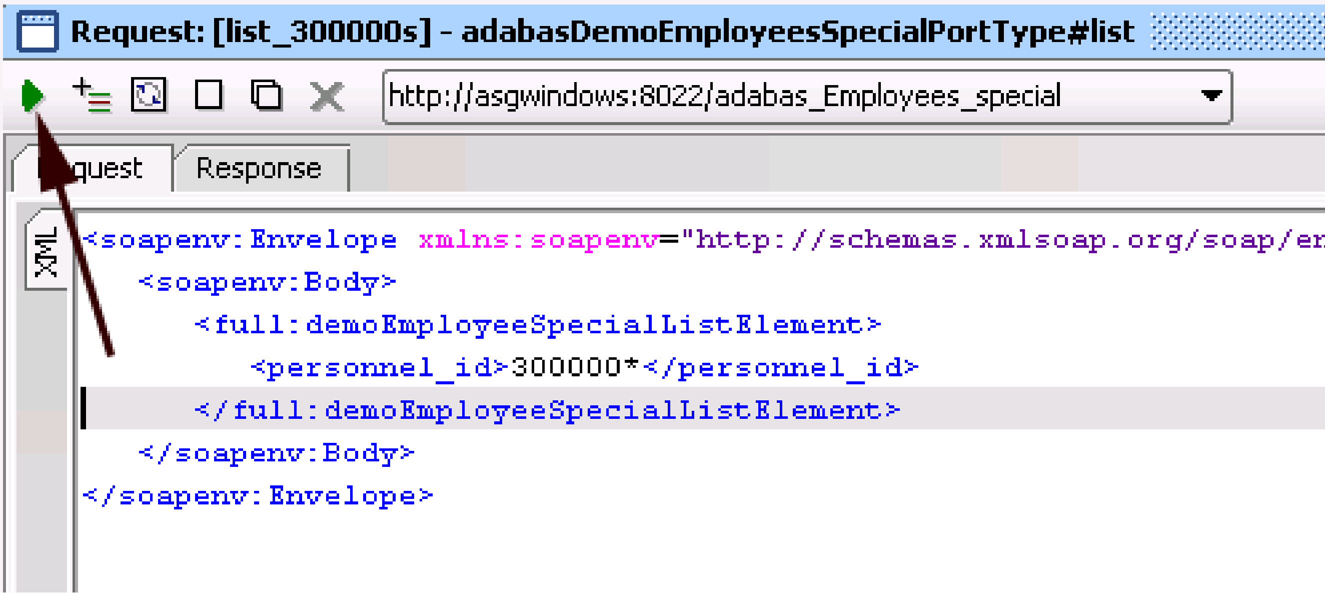
This leaves you with the soap Body section, which holds all key information:



http://asgwindows:8022/ada

```
<soapenv:Envelope xmlns:soapenv="http
   <soapenv:Body>
      <full:demoEmployeeSpecialListE
         <personnel_id>?</personnel_
         <first_name>?</first_name>
         <name>?</name>
         <birthday>?</birthday>
         <city>?</city>
         <department_code>?</departme
         <job_title>?</job_title>
         <!--1 to 8 repetitions:-->
         <language_spoken>?</language
         <leave_left>
            <leave_due>?</leave_due>
            <leave_tak
         </leave_left>
         <dept_person>
            <department_code>?</depar
```

Remove all key elements but the personnel_id, enter 300000* as the key value, the request should look like this now, then click the **green arrow** to send the request to your Portus server:

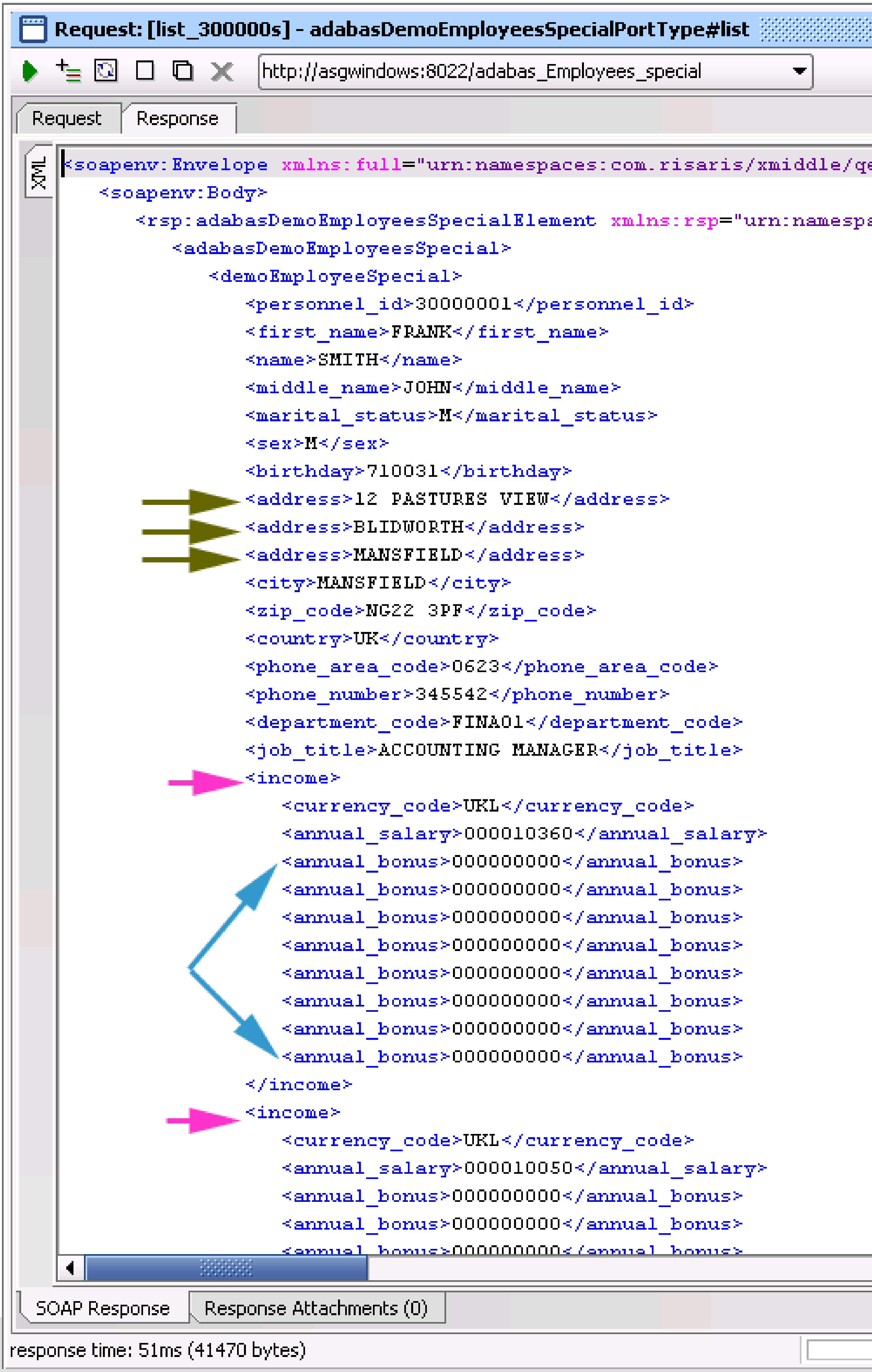

Request: [list_300000s] - adabasDemoEmployeesSpecialPortType#list
quest
Response
XML

http://asgwindows:8022/adabas_Employees_special

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/en
   <soapenv:Body>
      <full:demoEmployeeSpecialListElement>
         <personnel_id>300000*</personnel_id>
      </full:demoEmployeeSpecialListElement>
   </soapenv:Body>
</soapenv:Envelope>
```

5 The response will look like this - formatted XML

This response shows "simple" as well as repeating fields like MUs (multiple value fields, here: olive arrows), PEs (periodic groups, pink arrows) and MUs within PEs (blue arrows):


Request: [list_300000s] - adabasDemoEmployeesSpecialPortType#list
Request
Response
XML
SOAP Response
Response Attachments (0)

http://asgwindows:8022/adabas_Employees_special

```
<soapenv:Envelope xmlns:full="urn:namespaces:com.risaris/xmiddle/qe
   <soapenv:Body>
      <rsp:adabasDemoEmployeesSpecialElement xmlns:rsp="urn:namespa
         <adabasDemoEmployeesSpecial>
            <demoEmployeeSpecial>
               <personnel_id>30000001</personnel_id>
               <first_name>FRANK</first_name>
               <name>SMITH</name>
               <middle_name>JOHN</middle_name>
               <marital_status>M</marital_status>
               <sex>M</sex>
               <birthday>710031</birthday>
               <address>12 PASTURES VIEW</address>
               <address>BLIDWORTH</address>
               <address>MANSFIELD</address>
               <city>MANSFIELD</city>
               <zip_code>NG22 3PF</zip_code>
               <country>UK</country>
               <phone_area_code>0623</phone_area_code>
               <phone_number>345542</phone_number>
               <department_code>FINA01</department_code>
               <job_title>ACCOUNTING MANAGER</job_title>
               <income>
                  <currency_code>UKL</currency_code>
                  <annual_salary>000010360</annual_salary>
                  <annual_bonus>000000000</annual_bonus>
                  <annual_bonus>000000000</annual_bonus>
                  <annual_bonus>000000000</annual_bonus>
                  <annual_bonus>000000000</annual_bonus>
                  <annual_bonus>000000000</annual_bonus>
                  <annual_bonus>000000000</annual_bonus>
                  <annual_bonus>000000000</annual_bonus>
                  <annual_bonus>000000000</annual_bonus>
               </income>
               <income>
                  <currency_code>UKL</currency_code>
                  <annual_salary>000010050</annual_salary>
                  <annual_bonus>000000000</annual_bonus>
                  <annual_bonus>000000000</annual_bonus>
                  <annual_bonus>000000000</annual_bonus>
```

response time: 51ms (41470 bytes)

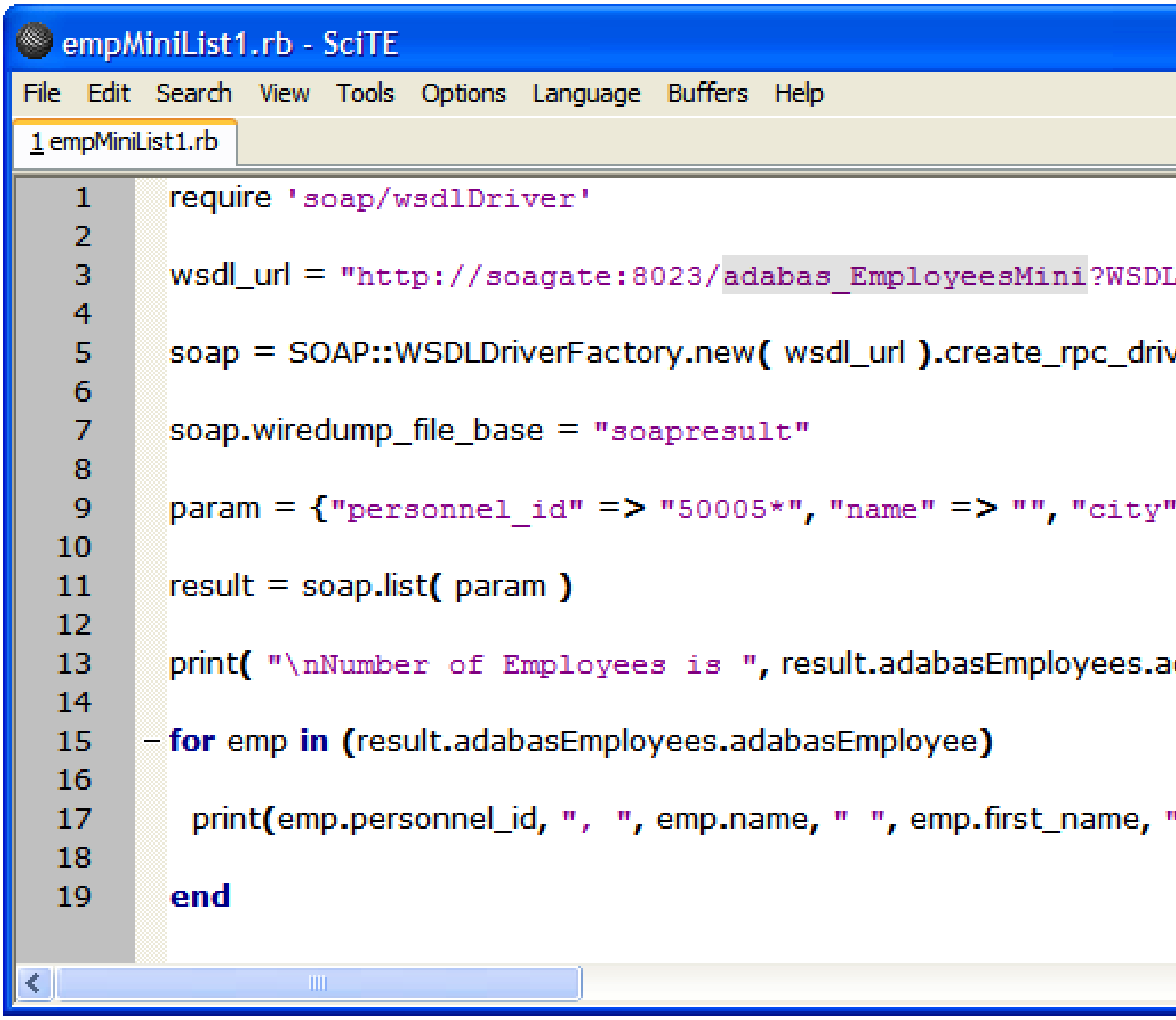
Accessing a Portus Service from a Ruby program

Ruby is an Open Source object oriented language with a very simple, yet powerful SOAP interface.

If you do not have Ruby installed yet it can be *dowbloaded from here*.

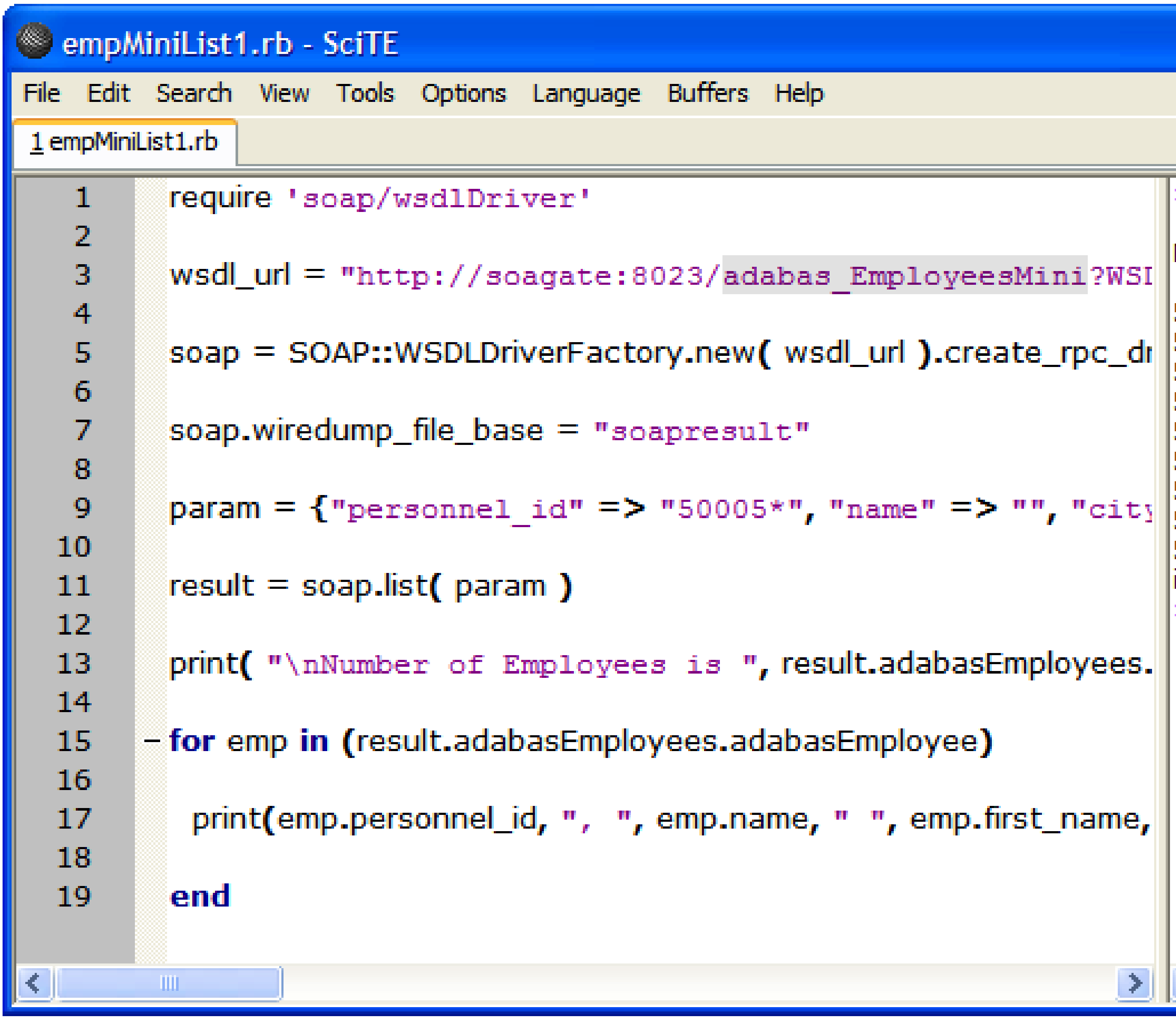
Ruby comes with a very powerful editor, SciTE, which not only allows editing, but also compiling & execution of programs.

This is an example of a complete Ruby program listing all Employees with personnel-IDs starting with 50005*, using the 'adabas_EmployeesMini' Resource that comes with the sample Portus configuration.



```
require 'soap/wsdlDriver'

wsdl_url = "http://soagate:8023/adabas_EmployeesMini?WSDL

soap = SOAP::WSDLDriverFactory.new( wsdl_url ).create_rpc_driv

soap.wiredump_file_base = "soapresult"

param = {"personnel_id" => "50005*", "name" => "", "city"

result = soap.list( param )

print( "\nNumber of Employees is ", result.adabasEmployees.a

for emp in (result.adabasEmployees.adabasEmployee)

  print(emp.personnel_id, "  ", emp.name, "  ", emp.first_name, "

end
```

When running this program, by either pressing the F5 key, or selecting Tools->Go from the SciTE menu, an output window will be attached to the editing window, and show the result of the Ruby query against Portus



```
require 'soap/wsdlDriver'

wsdl_url = "http://soagate:8023/adabas_EmployeesMini?WSI

soap = SOAP::WSDLDriverFactory.new( wsdl_url ).create_rpc_dr

soap.wiredump_file_base = "soapresult"

param = {"personnel_id" => "50005*", "name" => "", "city

result = soap.list( param )

print( "\nNumber of Employees is ", result.adabasEmployees.

for emp in (result.adabasEmployees.adabasEmployee)

  print(emp.personnel_id, ",  ", emp.name, "  ", emp.first_name,

end
```

Preparing for the LOBs (Large OBjects) samples

While the basic samples use the well-known 'Emplyoees' file delivered with Adabas, the LOBs demo programs require loading of a small Adabas file containing LOB fields.

Note: LOB access requires a minimum Adabas version of v6 on OpenSystems platforms, Adabas v8 on mainframe platforms.

Loading the LOB file into an OpenSystems database

These are the steps to load the demo LOB file into an OpenSystems (Windows, *IX) Adabas database:

- Save *FILE90.FDT* on the target system.
- Save *FILE90.FDU* on the target system, adjust the dbid, file and lobfile parameters according to your needs.
- Set the environment variable FDUFDT, for example on a Linux system:

```
export FDUFDT=FILE90.FDT
```

- Run the command:

```
adafdu <FILE90.FDU
```

The demo 'base' and 'LOB' files are now loaded, which can be verified with the adarep utility:

```
adarep db=<yourdbid>,cont
```

The LOB demo file is now ready to be used.

Loading the LOB file into a Mainframe database

The steps to create the LOB demo file in a mainframe Adabas database are as follows:

- Use *this FDT* as input to ADACMP, specifying dummy input (DDEBAND). Sample JCL can be found on the ADAvrm.JOBS library distributed with Adabas.
- Run the ADALOD utility, *these parameters* may serve as template input to the utility. Adjust the file number and size specifications based on your requirements.

Note: the Adabas space / buffer parameters may need to be increased for LOB access, please consult the Adabas documentation for a description of required changes to your Adabas nucleus parameters.

Using LOBs (Large OBjects) with soapUi

This tutorial shows how to add, delete and get BLOBs using Portus and Adabas.

Portus uses the *MTOM specification* to send/receive the XML and binary data to/from the required web service. This involves attaching the required binary file(s) to the SOAP message, and then transforming this into a MIME message to send across the wire.

For the purposes of this tutorial, we will use *soapUi* to send and receive messages to our web service.

Other web service clients may also be used, such as PHP with the *WSO2 Web Services Framework* extension

- You have at least soapUi v1.7 installed.
- You are using a version of Adabas which support LOBs
- You have set up a LOB file at FNR 90.

1 Start the Control Center and add a new resource.

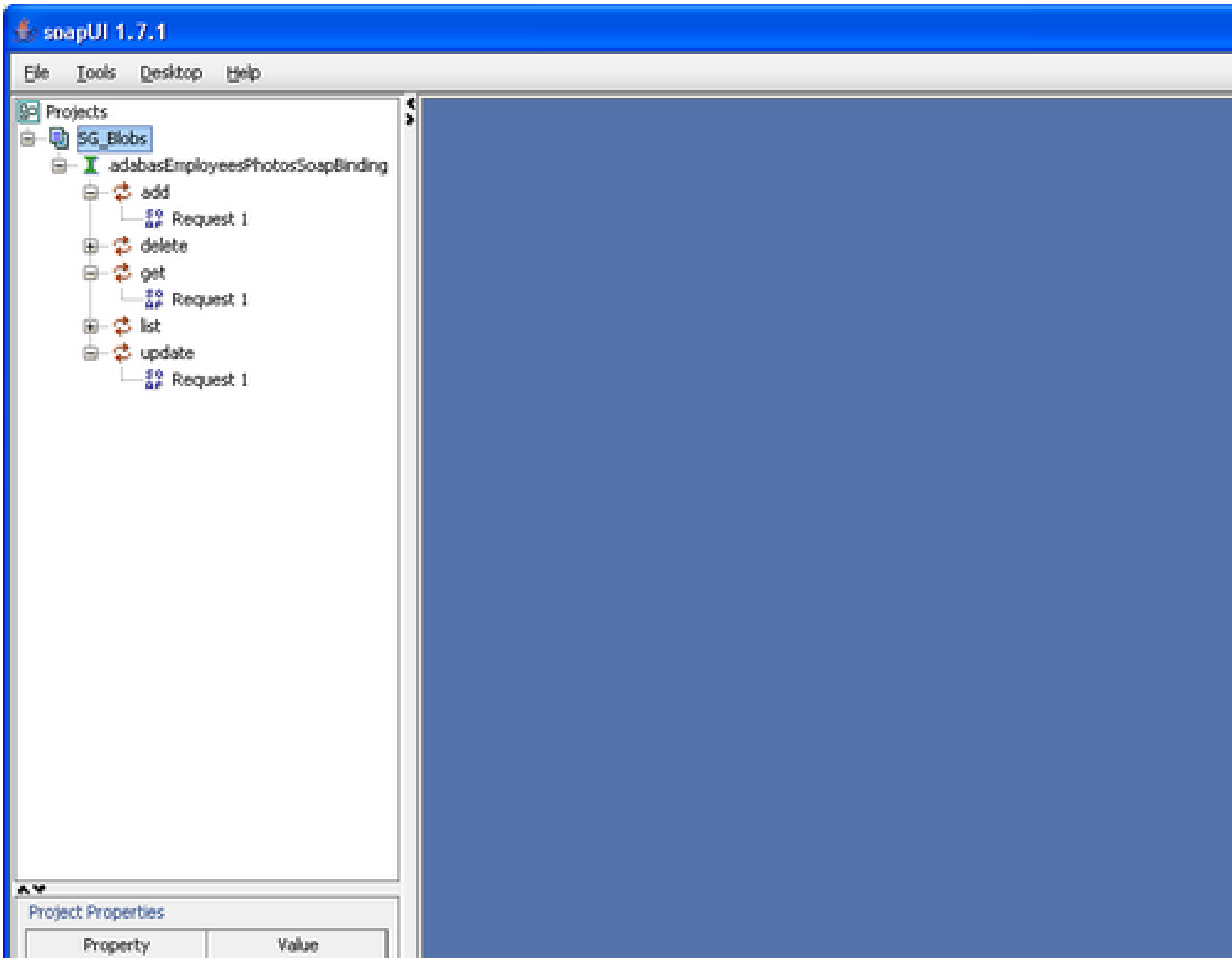
Enter the following information

- Name: adabas_blobs
- Assign the'adabas_photoblobs' DataView, which is one of the sample definitions delivered with Portus, to the service as described *here*
- DatabaseId: 212 (or the DBID relevant on your system)
- FileNumber: 90

Publish your changes to the server.

2 Start soapUi, and add the WSDL for the adabas_blobs service. If you are unfamiliar with soapUi, please run through *this tutorial* first.

3 You should now have a screen similar to this

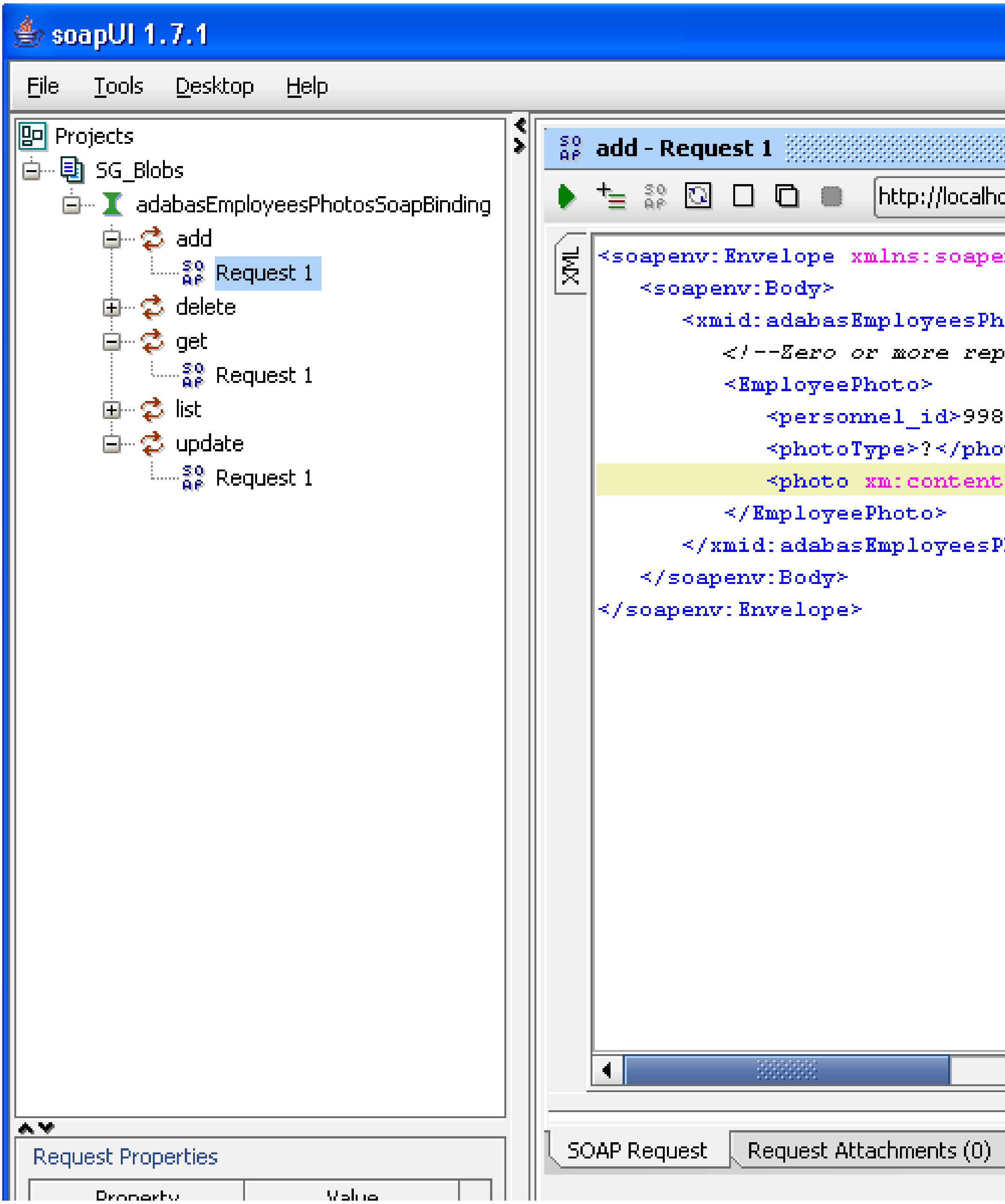


Under **add** double-click **Request 1**

4 Remove the <soapenv:Header> element and all child elements.

Add the personnel_id of the record you would like the BLOB to be added

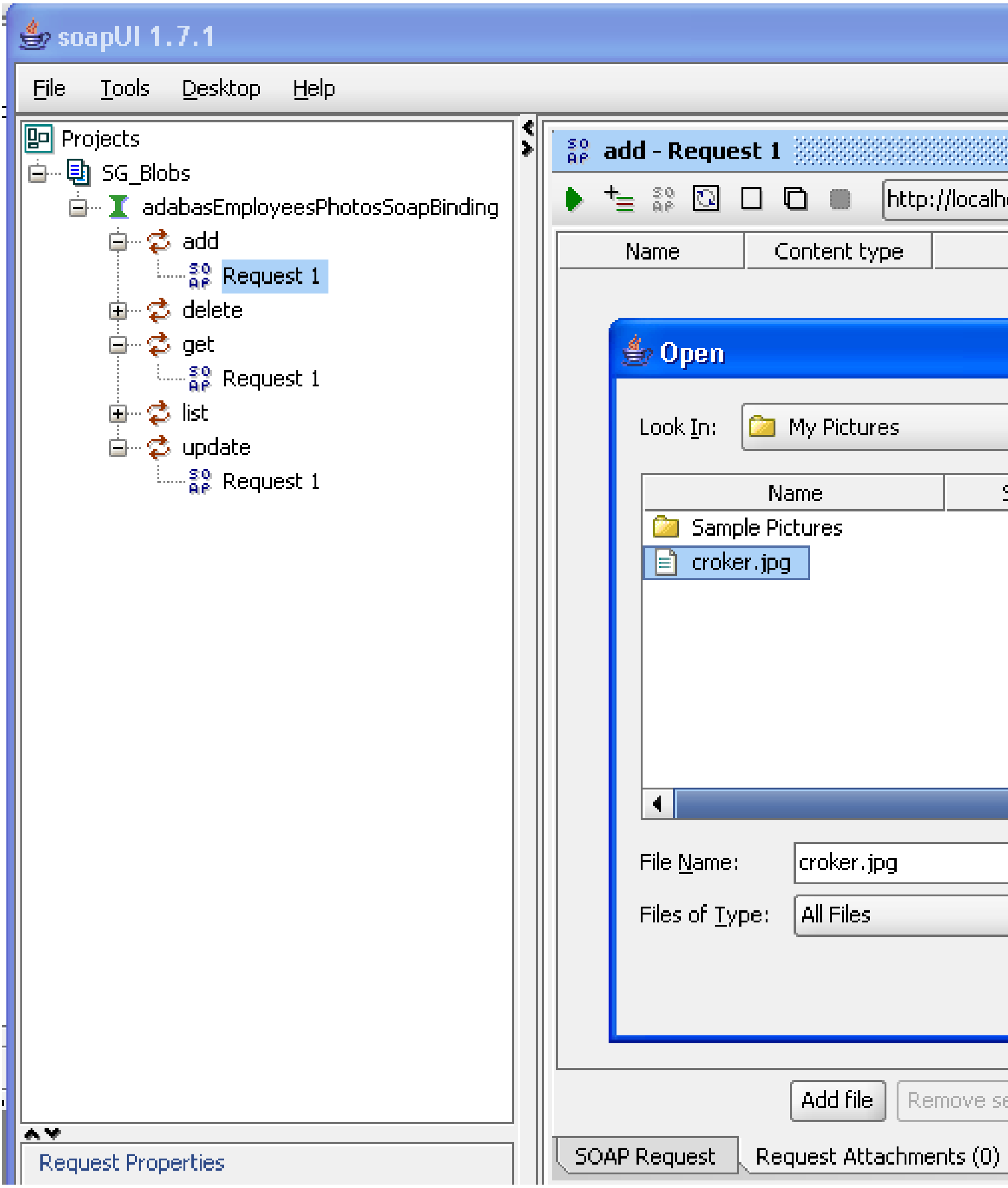
Change the cid:XYZ reference to be something familiar, for example **cid:myPhoto**



```
<soapenv:Envelope xmlns:soape
   <soapenv:Body>
      <xmid:adabasEmployeesPh
         <!--Zero or more rep
         <EmployeePhoto>
            <personnel_id>998
            <photoType>?</pho
            <photo xm:content
         </EmployeePhoto>
      </xmid:adabasEmployeesP
   </soapenv:Body>
</soapenv:Envelope>
```

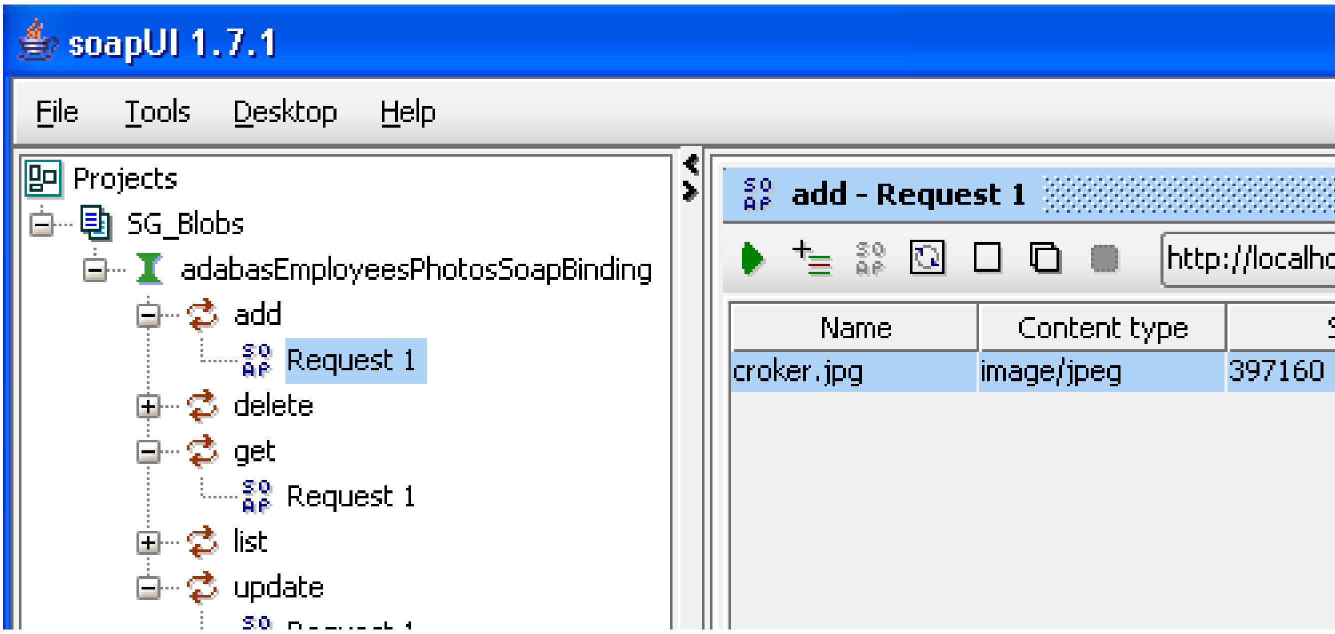
5 Click the **Request Attachments** tab, and click **Add File**

Select the required LOB file and click **Open**



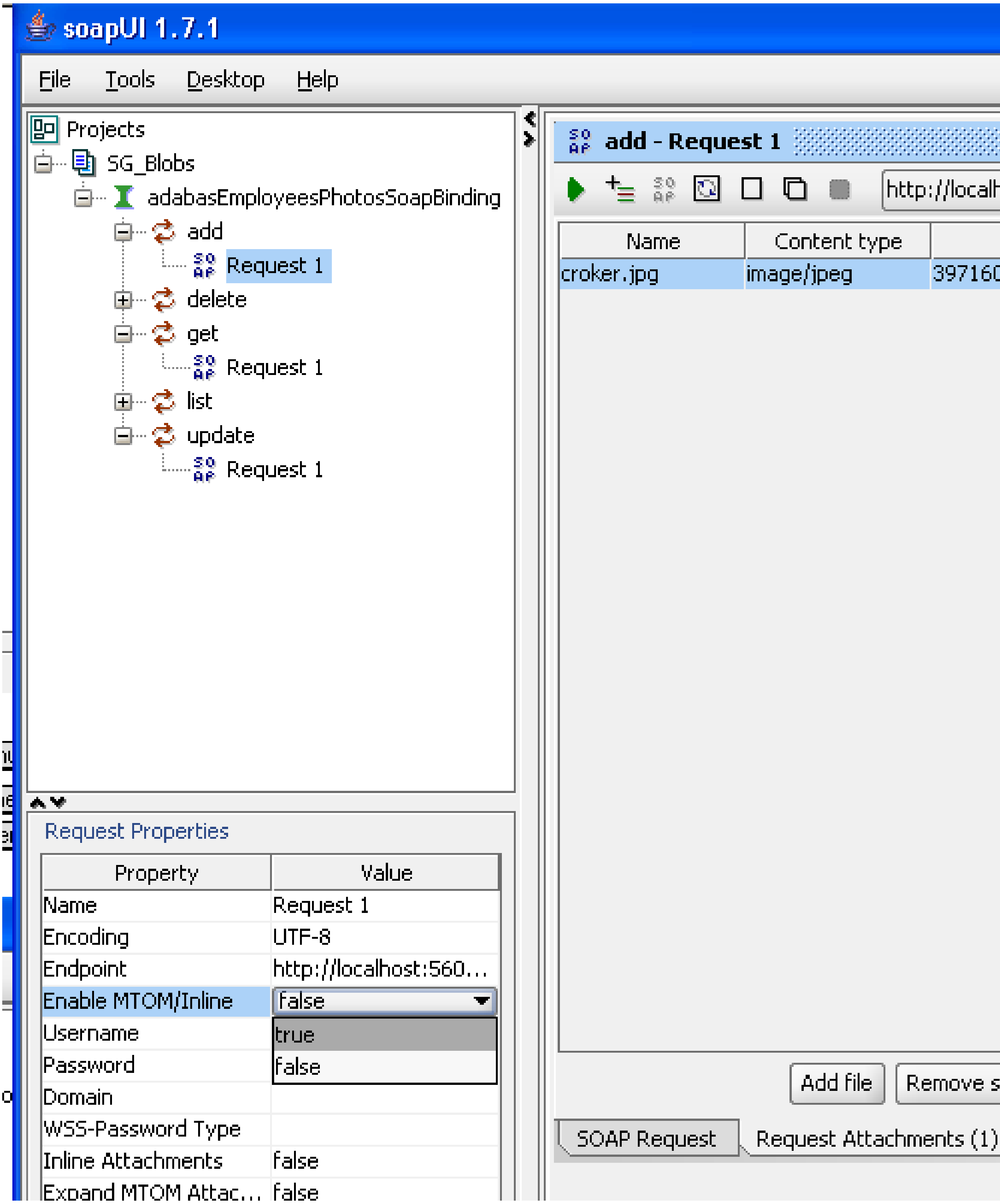
When asked to **Cache Attachment** select **No**

6 Click the **Part** column, and select the CID reference you changed earlier, for example **myPhoto**



Name	Content type	S
croker.jpg	image/jpeg	397160

7 In the **Request Properties** change Enable **MTOM/Inline** to true

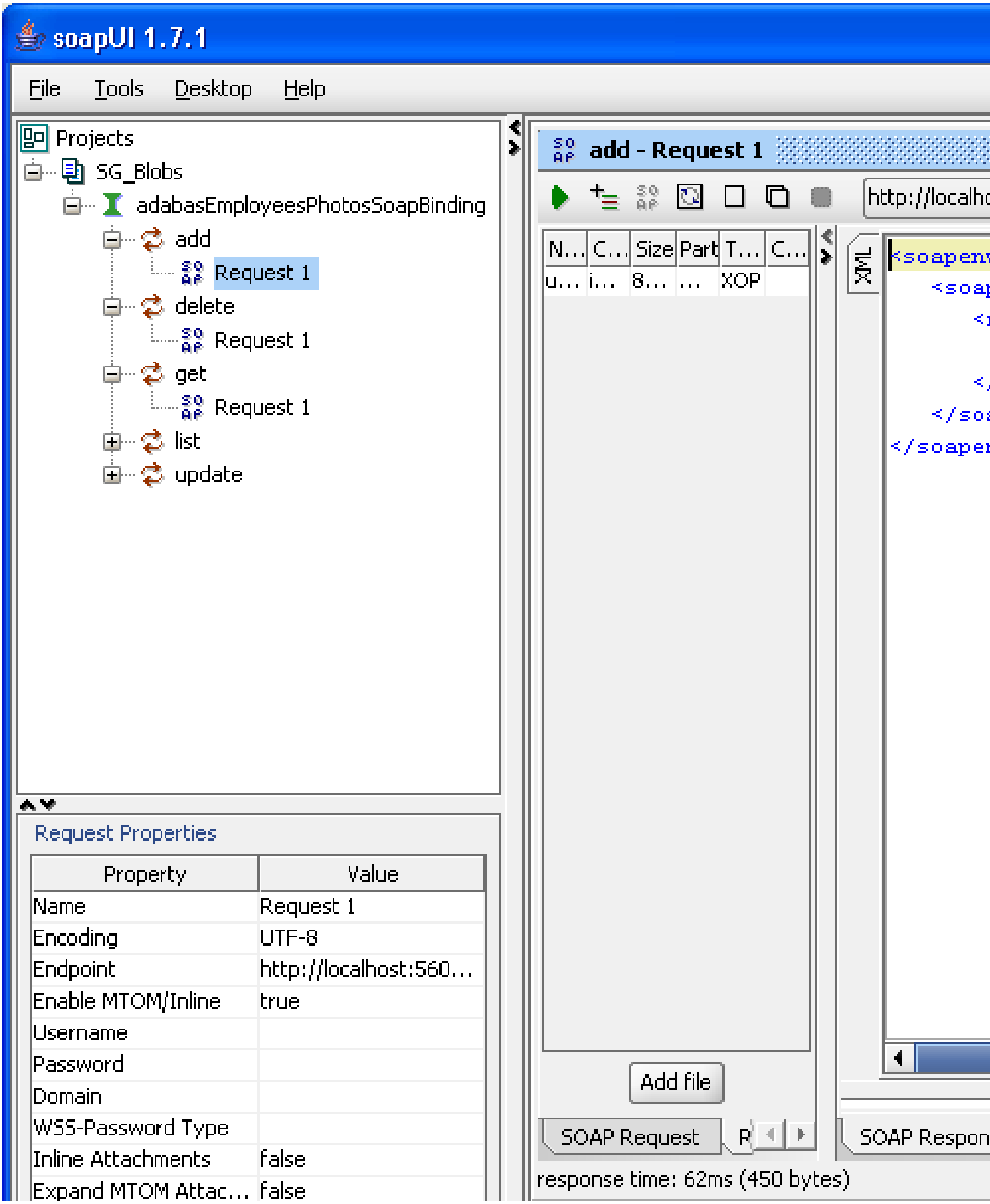


Name	Content type	
croker.jpg	image/jpeg	39716C

Request Properties

Property	Value
Name	Request 1
Encoding	UTF-8
Endpoint	http://localhost:560...
Enable MTOM/Inline	false true false
Username	
Password	
Domain	
WSS-Password Type	
Inline Attachments	false
Expand MTOM Attac...	false

8 Click the green arrow to send the request. The server's response will appear in the right pane



```
<soapenv
  <soap
    <
    </
  </soa
</soapen
```

Request Properties

Property	Value
Name	Request 1
Encoding	UTF-8
Endpoint	http://localhost:560...
Enable MTOM/Inline	true
Username	
Password	
Domain	
WSS-Password Type	
Inline Attachments	false
Expand MTOM Attac...	false

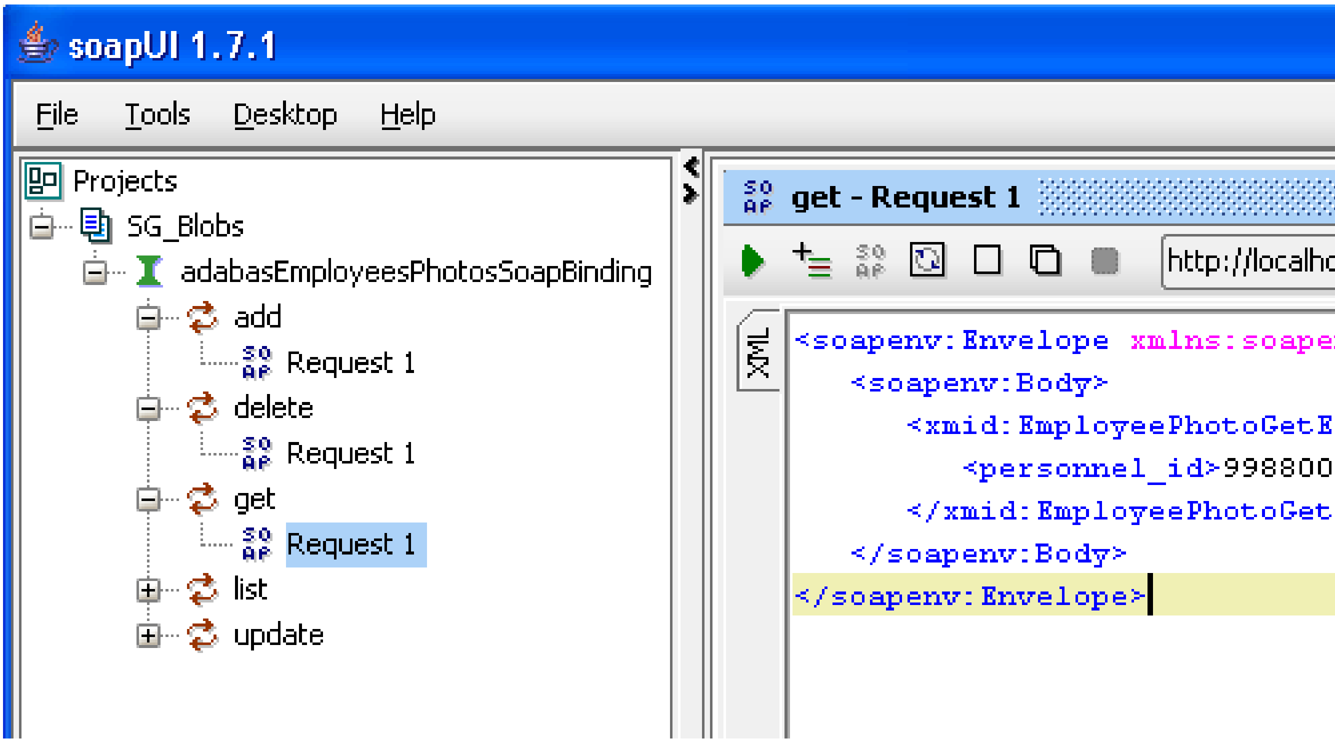
response time: 62ms (450 bytes)

9 Now that the record, including the LOB, has been added to your Adabas file, you will want to retrieve it.

In Portus LOBs can be retrieved using the **get** request

Under **get**, double-click **Request 1**. Remove the <soapenv:Header> elements as before.

10 Add the personnel_id of the record that you wish to retrieve, for example 99880000.

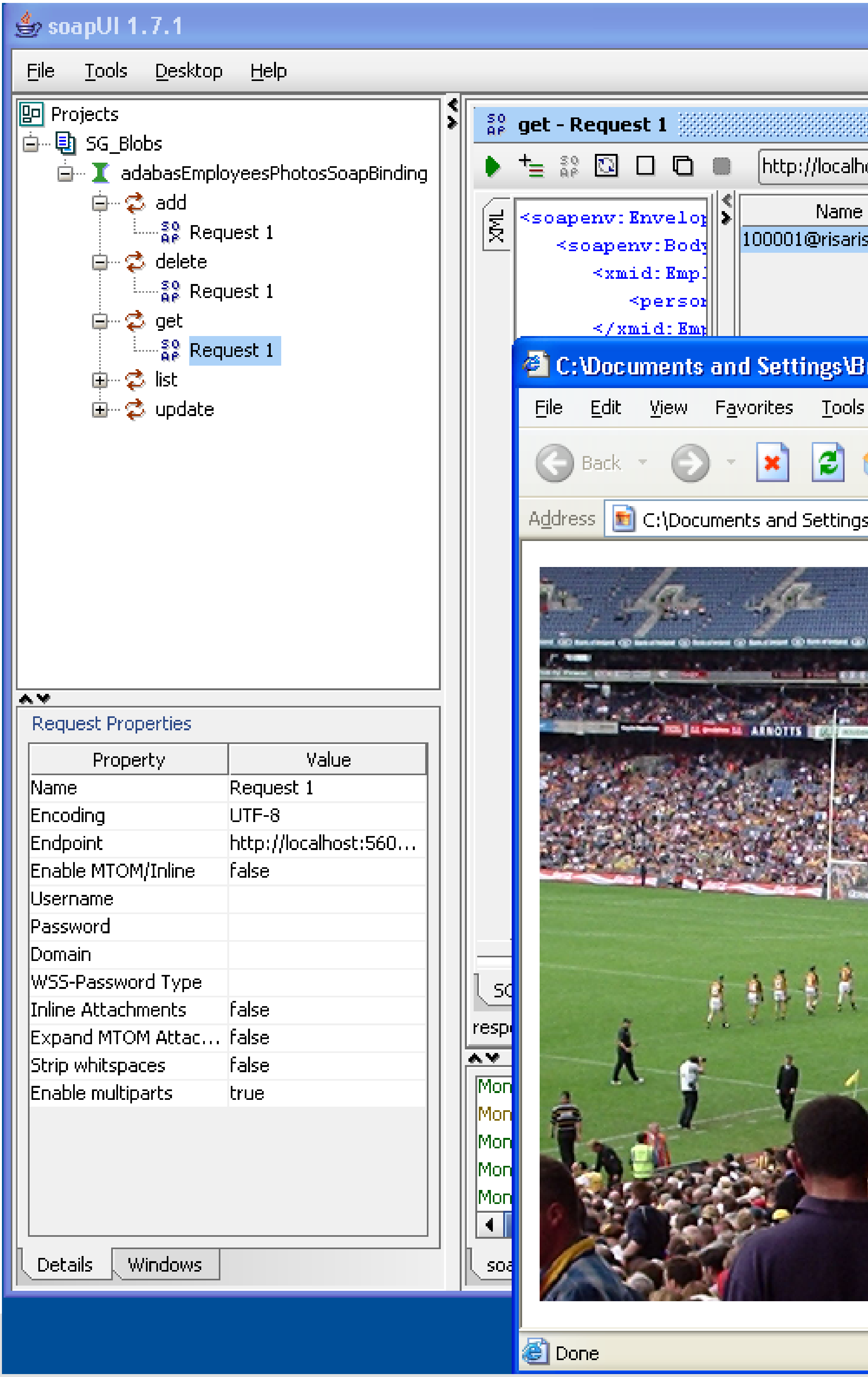


```
<soapenv:Envelope xmlns:soape
   <soapenv:Body>
      <xmid:EmployeePhotoGetE
         <personnel_id>998800
      </xmid:EmployeePhotoGet
   </soapenv:Body>
</soapenv:Envelope>
```

11 Click the green arrow, and this record will be retrieved from Adabas.

Select the **Response Attachments** tab, and double-click the attachment.

It should open in a web-browser, or you can save the attachment to disk by selecting **Export selected**



```
<soapenv:Envelop
  <soapenv:Body
    <xmid:Empl
      <person
    </xmid:Emp
```

Request Properties

Property	Value
Name	Request 1
Encoding	UTF-8
Endpoint	http://localhost:560...
Enable MTOM/Inline	false
Username	
Password	
Domain	
WSS-Password Type	
Inline Attachments	false
Expand MTOM Attac...	false
Strip whitspaces	false
Enable multiparts	true

12 You may delete the record, including the LOB, by selecting the delete operation and entering the corresponding personnel_id.

Accessing LOBs from PHP

LOBs sent as MTOM attachments on a SOAP response from Portus can be handled by a PHP program.

This, however, requires the presence of the *WSO2 Web Services Framework*

Provided here are the following sample PHP programs dealing with LOBs access:

- *Get a record, save the LOB to a file (wsf_lobsGet.php)* on page 286

Accessing LOBs using a browser

You can also use Portus to retrieve LOBs from Adabas into a web browser. This can be any browser of your choice, for example IE running on Windows, retrieving LOBs from Adabas running on z/OS.

In the tutorial below, we will use the popular *Firefox* browser and the Portus running on z/OS.

This tutorial assumes you've already got some LOBs stored in your database. See the previous tutorial to find out how to do this.

1 Start the Control Center and add a new resource.

Enter the following information

- Name: adabas_blobs
- Assign the'adabas_photoblobs' DataView, which is one of the sample definitions delivered with Portus, to the service as described *here*
- DatabaseId: 212 (or the DBID relevant on your system)
- FileNumber: 90

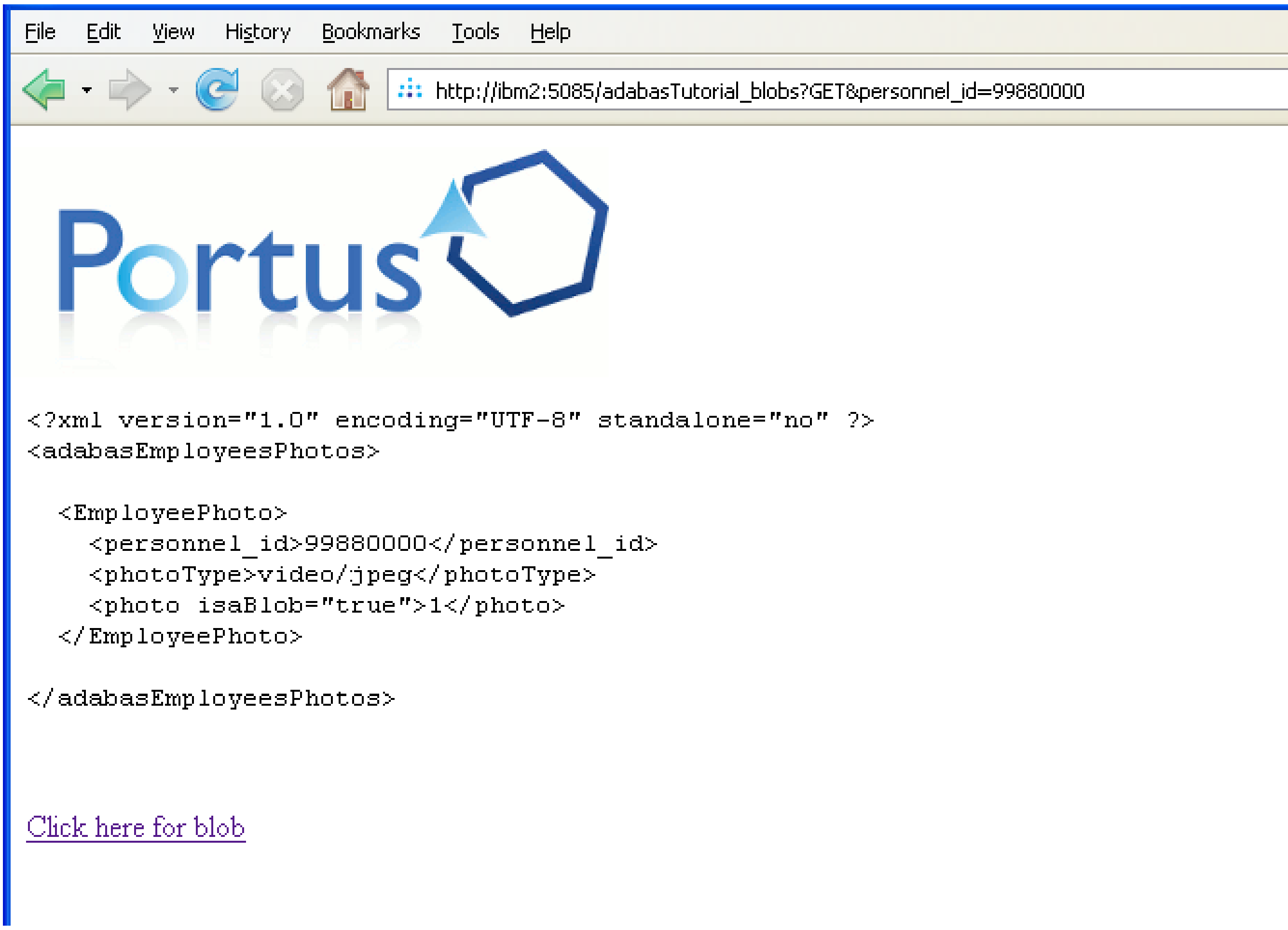
Publish your changes to the server.

2 Start your browser, and enter the following URL

```
http://soagate:56000/adabasTutorial_blobs?
GET&personnel_id=99880000
```

Replacing "soagate:56000" with the hostname and port where Portus is running.

3 If there is a blob associated with this record, the following is returned


File
Edit
View
History
Bookmarks
Tools
Help
http://ibm2:5085/adabasTutorial_blobs?GET&personnel_id=99880000
Portus

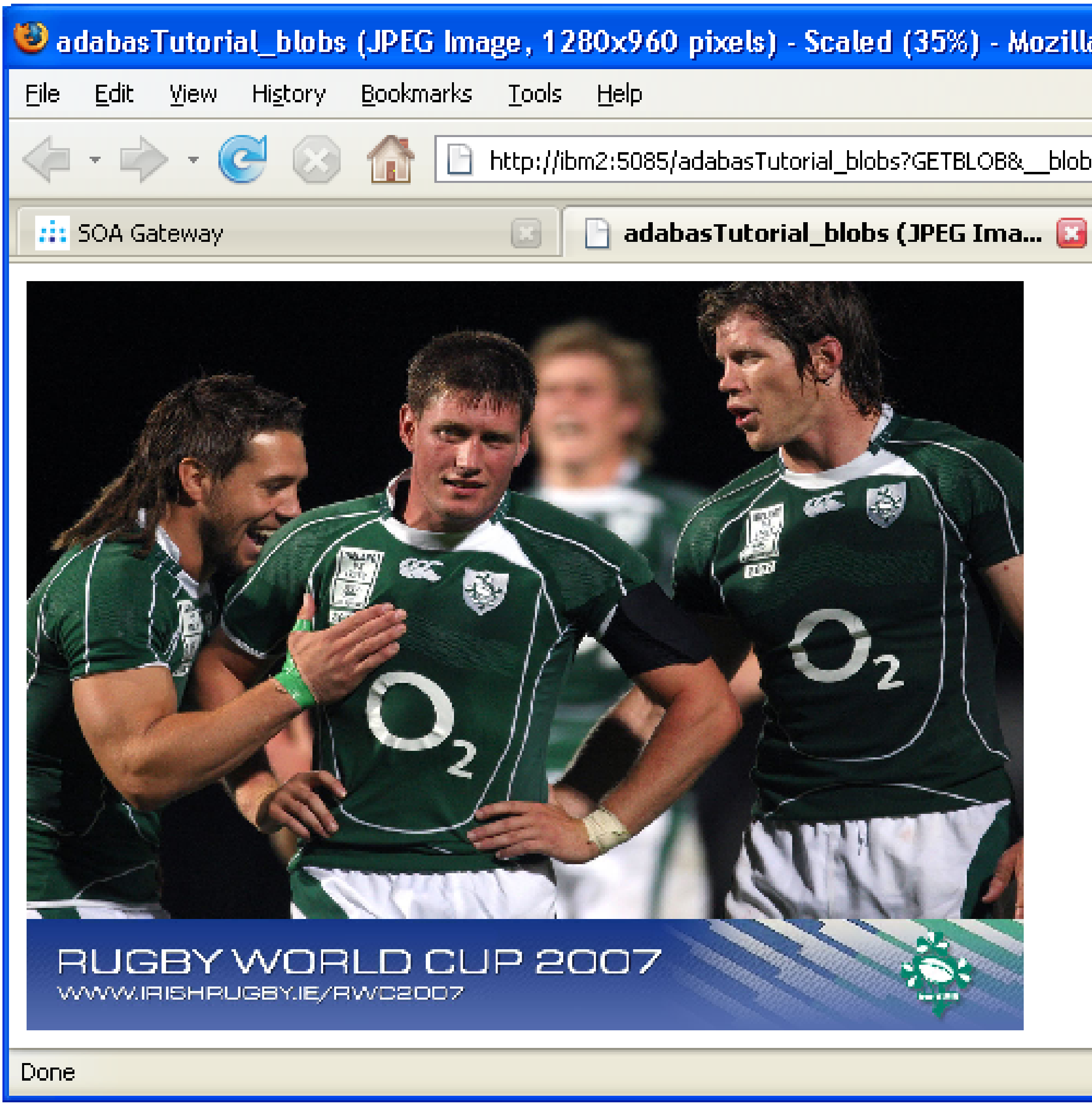
```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<adabasEmployeesPhotos>

  <EmployeePhoto>
    <personnel_id>99880000</personnel_id>
    <photoType>video/jpeg</photoType>
    <photo isaBlob="true">1</photo>
  </EmployeePhoto>

</adabasEmployeesPhotos>
```

Click here for blob

4 Click on the link "Click here for blob" to display the LOB file in the browser.

Note: The LOB will open automatically in your browser, if it of a MIME type that your browser understands. Commonly used types are PDF, WMV, JPEG, etc. Refer to your browser documentation if you need more information.

5 Here you can see a JPEG file opened up in a new tab in firefox after the LOB link was clicked.


adabasTutorial_blobs (JPEG Image, 1280x960 pixels) - Scaled (35%) - Mozill
File Edit View History Bookmarks Tools Help
http://ibm2:5085/adabasTutorial_blobs?GETBLOB&__blob
SOA Gateway
adabasTutorial_blobs (JPEG Ima...
RUGBY WORLD CUP 2007
WWW.IRISHRUGBY.IE/RWC2007
Done

Using Transactions with soapUI

The following tutorial demonstrates how Portus can be used with Adabas transactions. It is assumed you are already familiar with the following

- soapUi (see *here*)
- Creating Portus web services from Adabas (see *here*)
- Familar with SOAP Header usage concepts (see *here*)

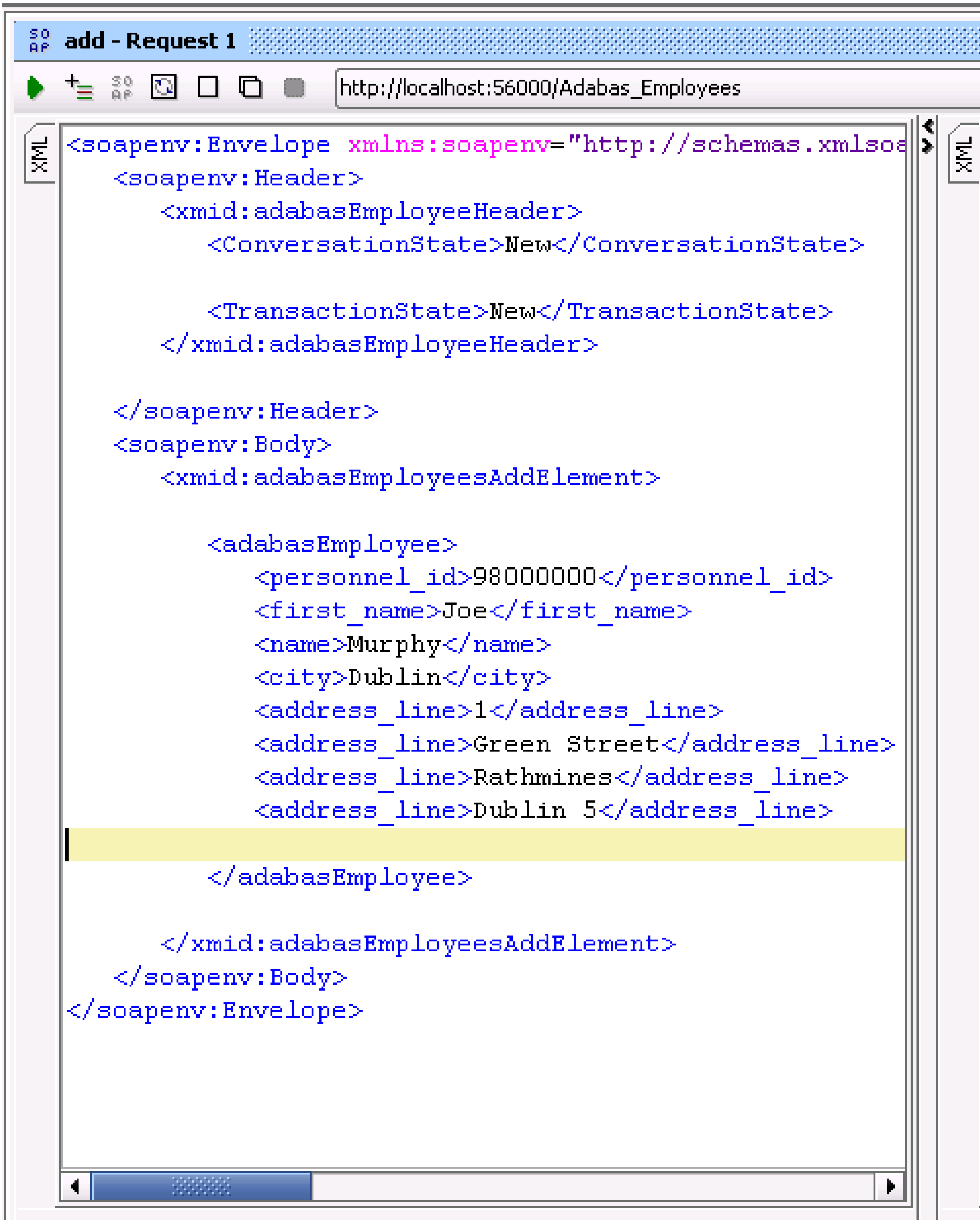
For this tutorial, we will use the Employees demo file (usually file number 11) that comes with Adabas, and the adabas_employees_mini_view data view. It is assumed you have already created a web service for your Adabas file.

Important: By default, Portus will time-out and kill existing Conversations after a period of time. This is can be configured using the Control Centre. with a maximum value of 3600 (10 minutes). See *here* for more information.

1 Import the WSDL into soapUi.

2 Choose the request for the add operation, and in the XML set the ConversationState to "New" and the TransactionState to "New". Also remove the other Header values.

Send the request to Portus. e.g.:

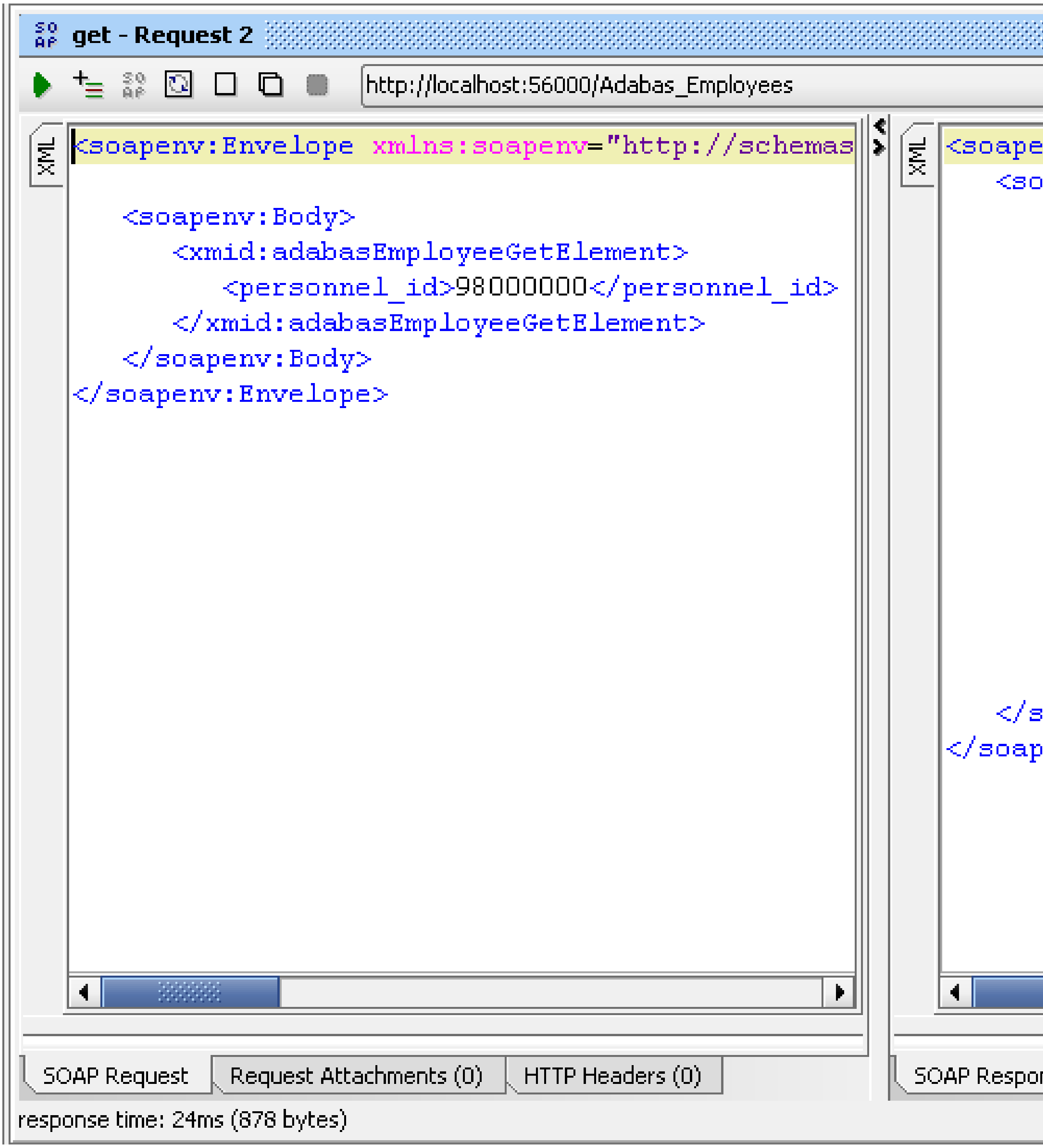


```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoa
   <soapenv:Header>
      <xmid:adabasEmployeeHeader>
         <ConversationState>New</ConversationState>

         <TransactionState>New</TransactionState>
      </xmid:adabasEmployeeHeader>

   </soapenv:Header>
   <soapenv:Body>
      <xmid:adabasEmployeesAddElement>

         <adabasEmployee>
            <personnel_id>98000000</personnel_id>
            <first_name>Joe</first_name>
            <name>Murphy</name>
            <city>Dublin</city>
            <address_line>1</address_line>
            <address_line>Green Street</address_line>
            <address_line>Rathmines</address_line>
            <address_line>Dublin 5</address_line>

         </adabasEmployee>

      </xmid:adabasEmployeesAddElement>
   </soapenv:Body>
</soapenv:Envelope>
```

Portus has now created a new Conversation and a new Transaction for this request. The IDs for each of these are returned.

3 Verify that the record has been added successfully, (but not yet committed).

Choose a get request, remove the soapenv:Header element, and click the green arrow to get the record, e.g.



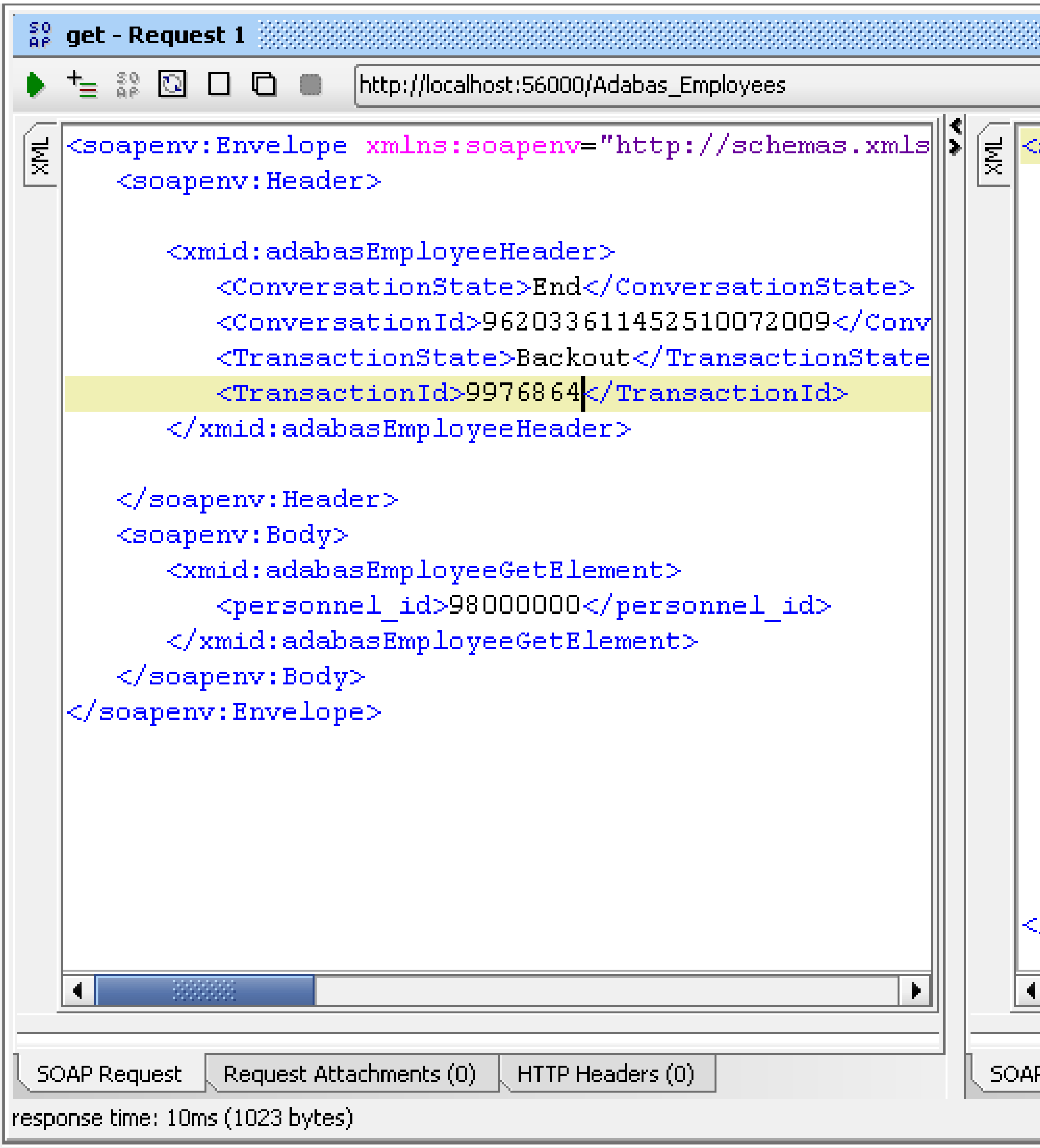
```
<soapenv:Envelope xmlns:soapenv="http://schemas

   <soapenv:Body>
      <xmid:adabasEmployeeGetElement>
         <personnel_id>98000000</personnel_id>
      </xmid:adabasEmployeeGetElement>
   </soapenv:Body>
</soapenv:Envelope>
```

The record has been added, but not yet committed. Because Adabas' isolation level is "Read Uncommitted" (also known as "dirty read"), a request from a non-conversational request will still return the added, but-yet-uncommitted, record.

4 Now backout the transaction.

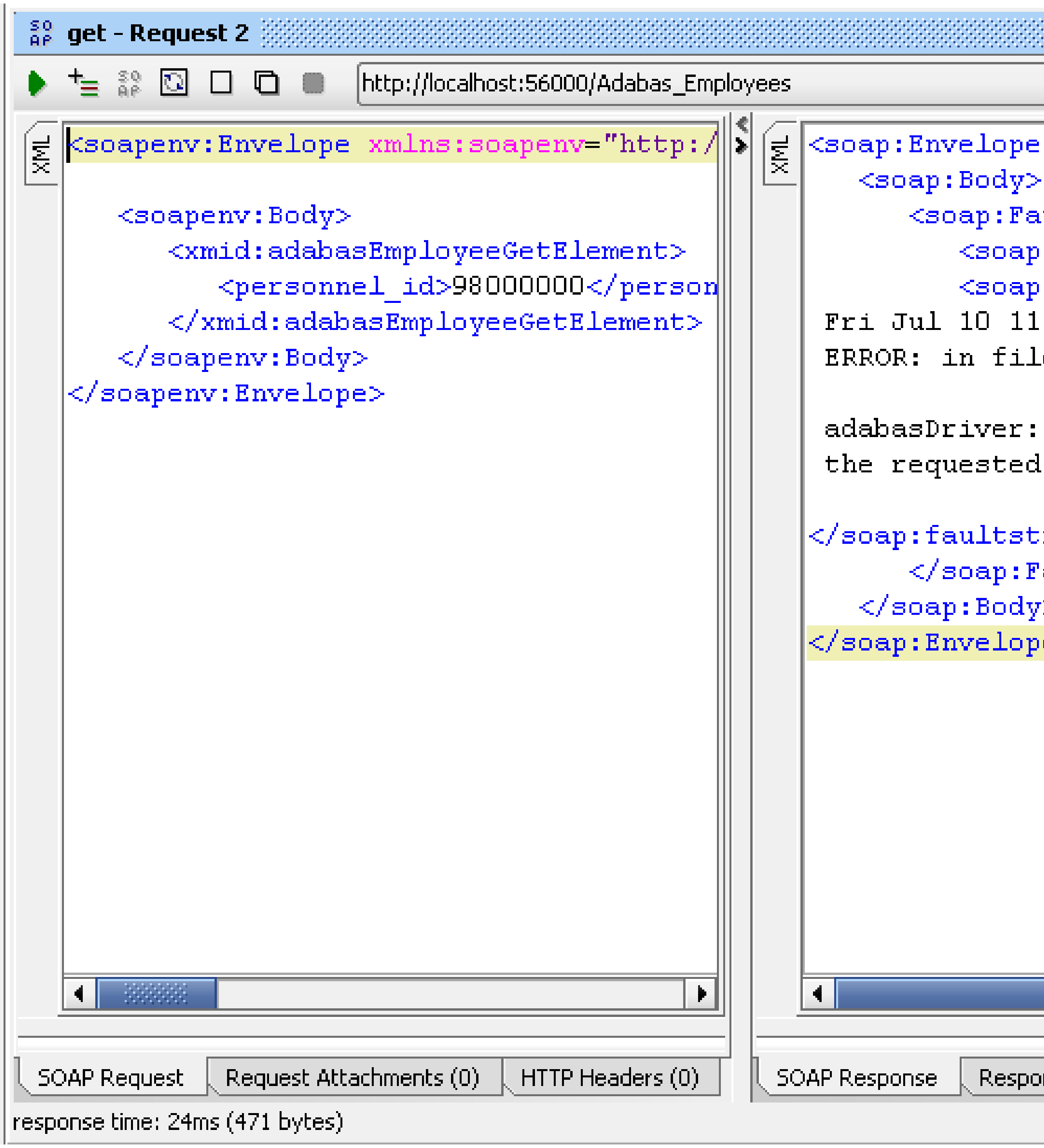
Choose a get request, and enter the following:

- *ConversationState* to "End"
- *ConversationID* to the value returned in the add response
- *TransactionState* to "Backout"
- *TransactionID* to the value returned in the add response



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmls
   <soapenv:Header>

      <xmid:adabasEmployeeHeader>
         <ConversationState>End</ConversationState>
         <ConversationId>962033611452510072009</Conv
         <TransactionState>Backout</TransactionState
         <TransactionId>9976864</TransactionId>
      </xmid:adabasEmployeeHeader>

   </soapenv:Header>
   <soapenv:Body>
      <xmid:adabasEmployeeGetElement>
         <personnel_id>98000000</personnel_id>
      </xmid:adabasEmployeeGetElement>
   </soapenv:Body>
</soapenv:Envelope>
```

5 Now, if you re-run the request from Step 3, the item does not exist as the previous add has been backed out.



```
<soapenv:Envelope xmlns:soapenv="http:/

   <soapenv:Body>
      <xmid:adabasEmployeeGetElement>
         <personnel_id>98000000</person
      </xmid:adabasEmployeeGetElement>
   </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope
   <soap:Body>
      <soap:Fa
         <soap
         <soap
 Fri Jul 10 11
 ERROR: in fil

 adabasDriver:
 the requested

</soap:faultst
      </soap:F
   </soap:Body
</soap:Envelop
```

response time: 24ms (471 bytes)

MySQL Tutorials

The MySQL tutorials are based on the World Database. This is a publically available database which is used for MySQL testing and demonstation.

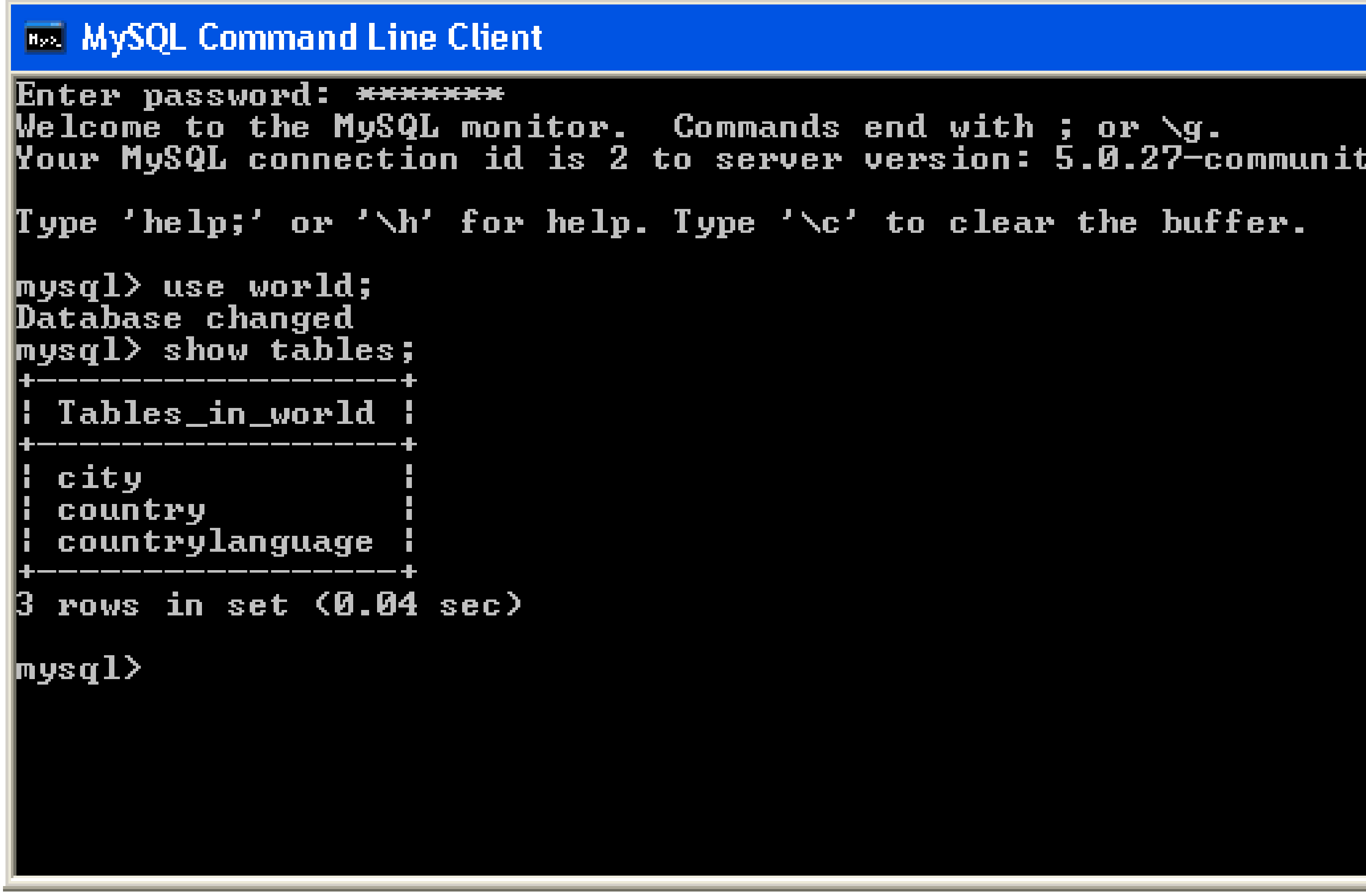
- *Introduction* on page 204
- *Using C# to access MySQL* on page 207

Introduction

You can find out more about the World Database *here*.

Before starting this tutorial, it is recommended you see the "Getting started with MySQL" section *here*.

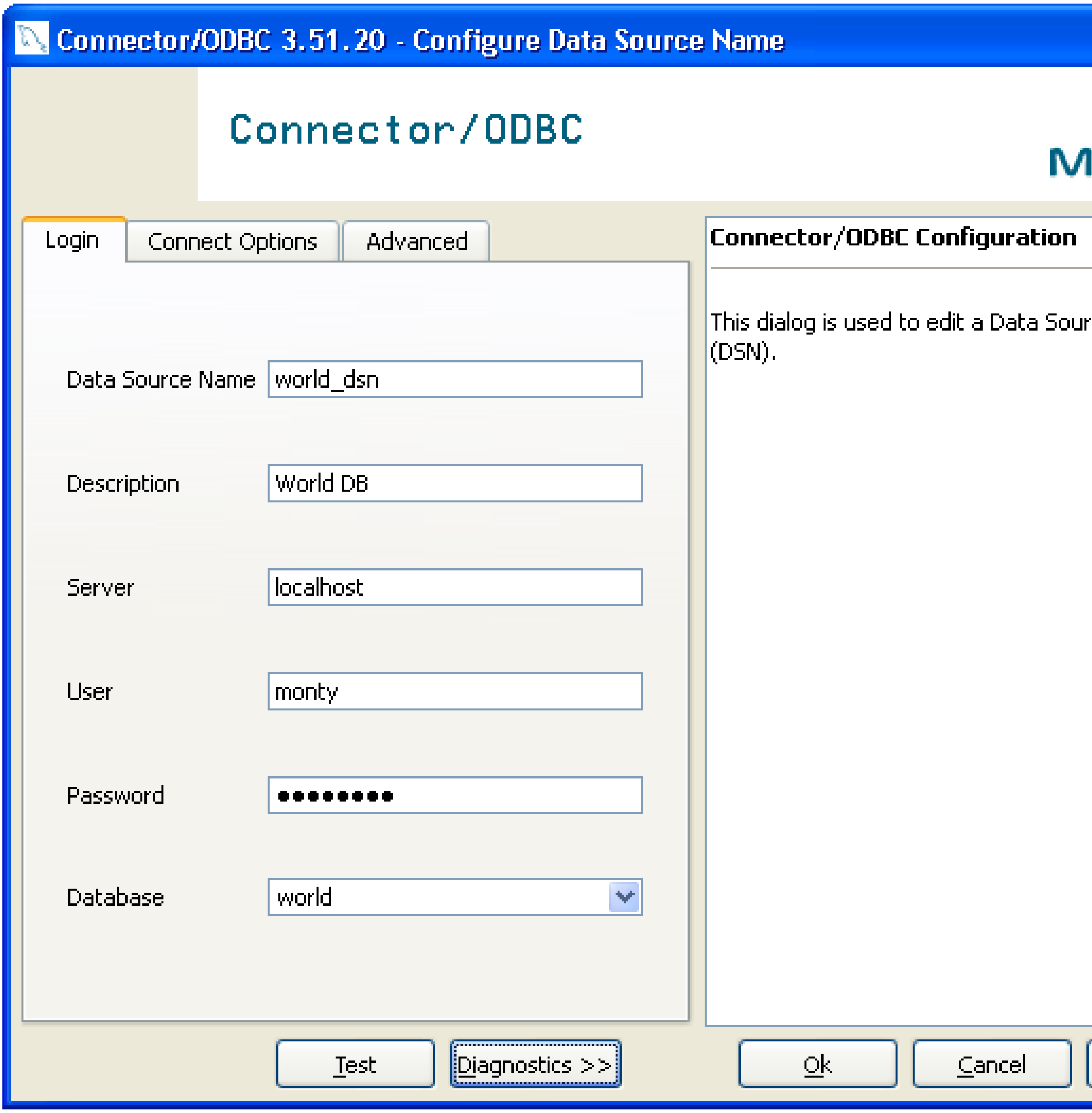
It is assumed that the world database has been set up and and populated as follows



```
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.27-communit

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use world;
Database changed
mysql> show tables;
+-----------------+
| Tables_in_world |
+-----------------+
| city            |
| country         |
| countrylanguage |
+-----------------+
3 rows in set (0.04 sec)

mysql>
```

It is also assumed that an ODBC System DSN called "world_dsn" has been set up as follows

We use the user *monty* to connect to our datasource, but in your instance the user might be different. In many cases, the *root* user ID is used.


Connector/ODBC 3.51.20 - Configure Data Source Name
Connector/ODBC
M
Login
Connect Options
Advanced
Connector/ODBC Configuration
Test
Diagnostics >>
Ok
Cancel

This dialog is used to edit a Data Sour
(DSN).

Data Source Name: world_dsn
Description: World DB
Server: localhost
User: monty
Password: ●●●●●●●●●
Database: world

Or on Linux, assuming you are using the unixODBC driver, the following information in your odbc.ini

```
[world_dsn]
Description      = The world database in mysql
Driver           = DriverMysql
Trace            = off
TraceFile        = stderr
Server           = localhost
Port             = 3306
Database         = world
UserName         =
Password         =
```

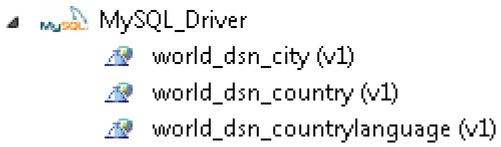
And this information in your odbcinst.ini

```
[DriverMysql]
Description      = ODBC for MySQL
```

```
Driver          = /usr/lib/unixODBC/libmyodbc3.so
Setup           = /usr/lib/unixODBC/libodbcmyS.so
UsageCount      = 1
```

For more information on setting up an ODBC System DSN, see *this section*

Finally, it assumed that the Web Service Discovery Wizard has been used to create 3 web services based on the 3 tables in the World database.



MySQL_Driver
world_dsn_city (v1)
world_dsn_country (v1)
world_dsn_countrylanguage (v1)

Using C# to access MySQL

This tutorial shows how to access the "City" table in MySQL from the C# environment. It assumes a C# environment is available, and some basic knowledge of the C# language.

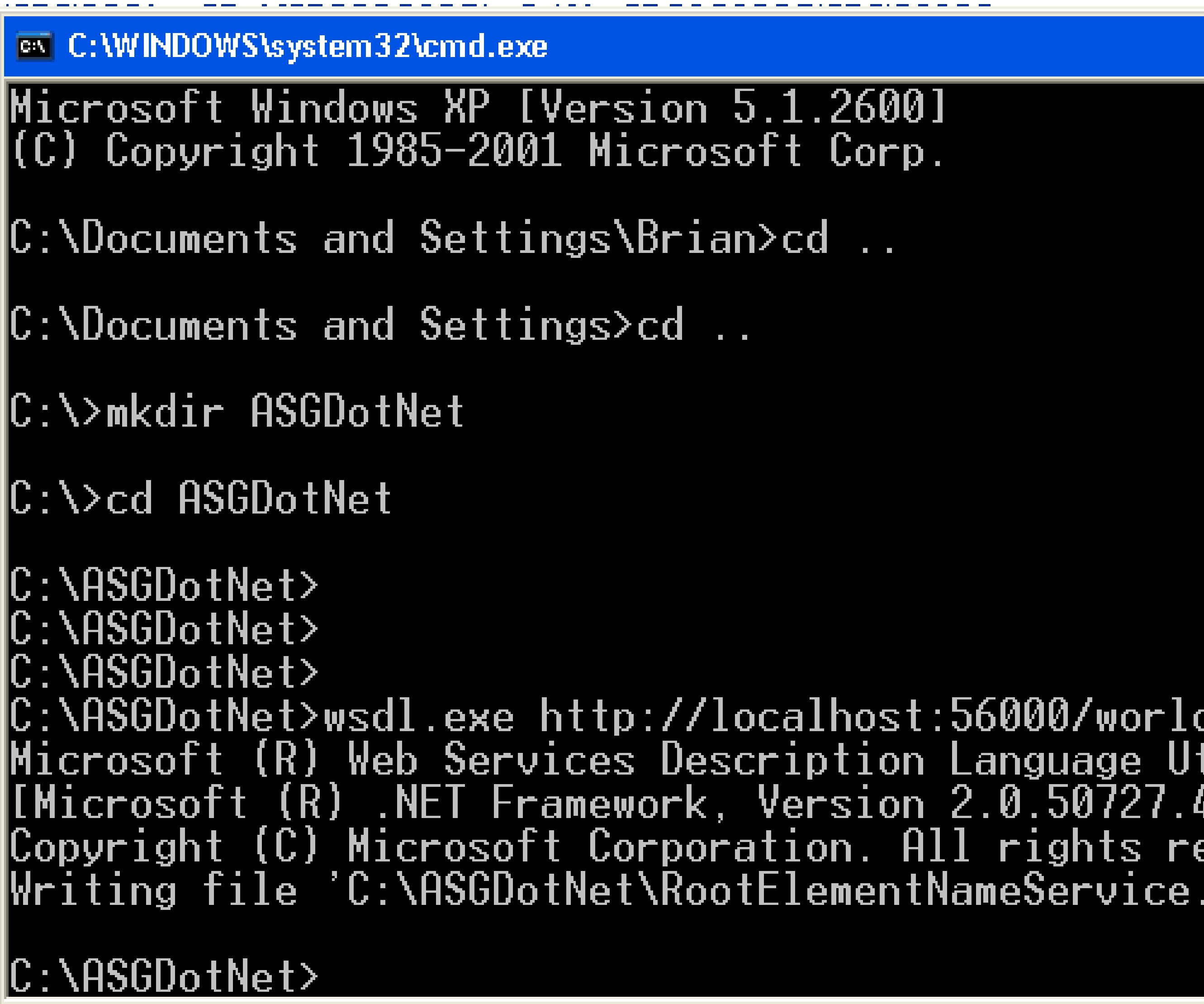
With a service description (WSDL), a proxy class can be created with the .NET Framework SDK **Wsdl.exe** tool.

A XML Web service client can then invoke methods of the proxy class, which communicate with Portus over the network by processing the SOAP messages sent to and from Portus server. The proxy class handles the work of mapping parameters to XML elements and then sending the SOAP message over the network.

Wsdl.exe can be used to create proxies for C#, Visual Basic .NET and JScript .NET, for the purpose, we will be generating C#.

These are the steps required to generate the C# wrapper class using Wsdl.exe and create / run a program listing records from the City table using the generated proxy class

1 From a command prompt, execute Wsdl.exe, specifying the URL / URI of the Portus Web Service to be exposed.



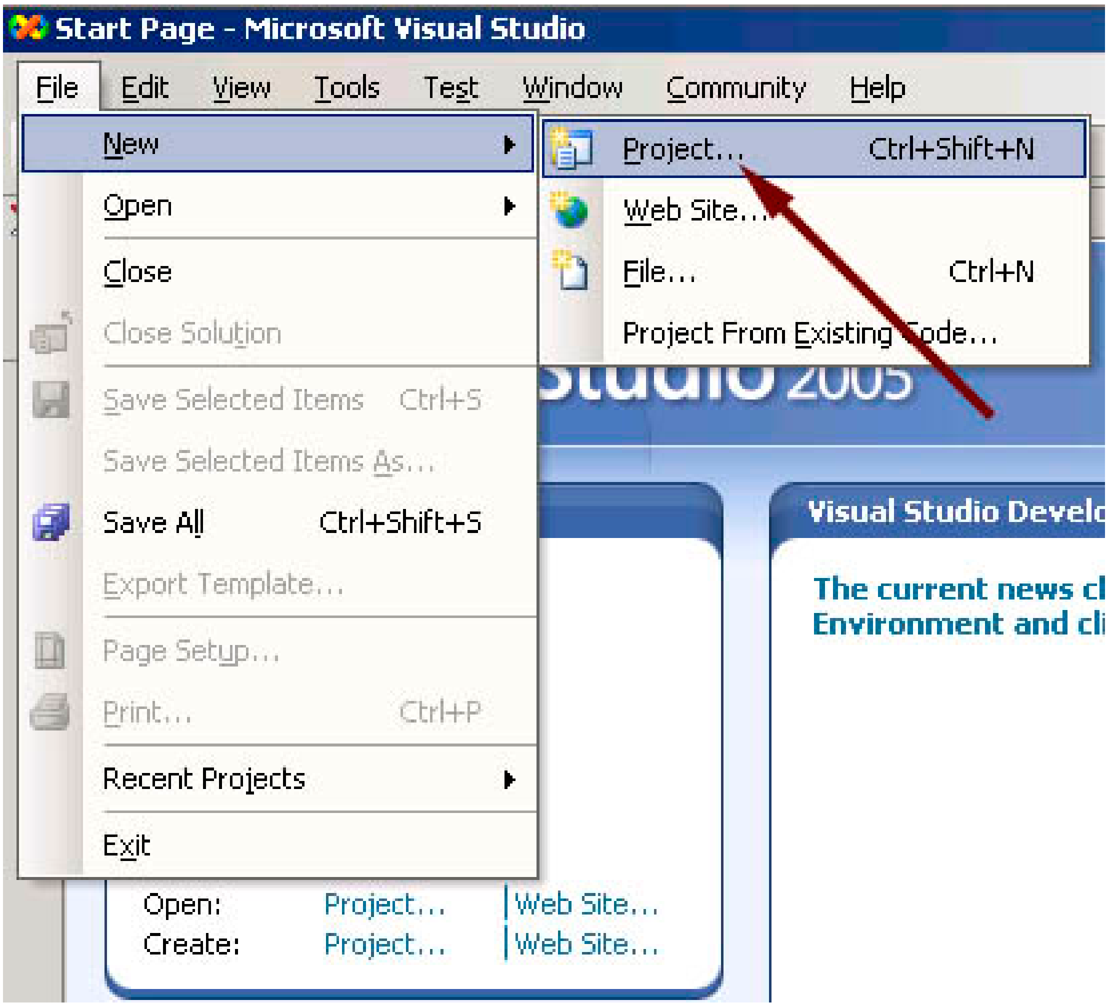
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Brian>cd ..

C:\Documents and Settings>cd ..

C:\>mkdir ASGDotNet

C:\>cd ASGDotNet

C:\ASGDotNet>
C:\ASGDotNet>
C:\ASGDotNet>
C:\ASGDotNet>wsdl.exe http://localhost:56000/worl
Microsoft (R) Web Services Description Language Ut
[Microsoft (R) .NET Framework, Version 2.0.50727.4
Copyright (C) Microsoft Corporation. All rights re
Writing file 'C:\ASGDotNet\RootElementNameService.

C:\ASGDotNet>
```

A single source file is generated, its name is <rootElementName>Service.cs, in this case the "root element" within the XRD is "RootElementName", thus the name of the proxy class source file RootElementNameService.cs

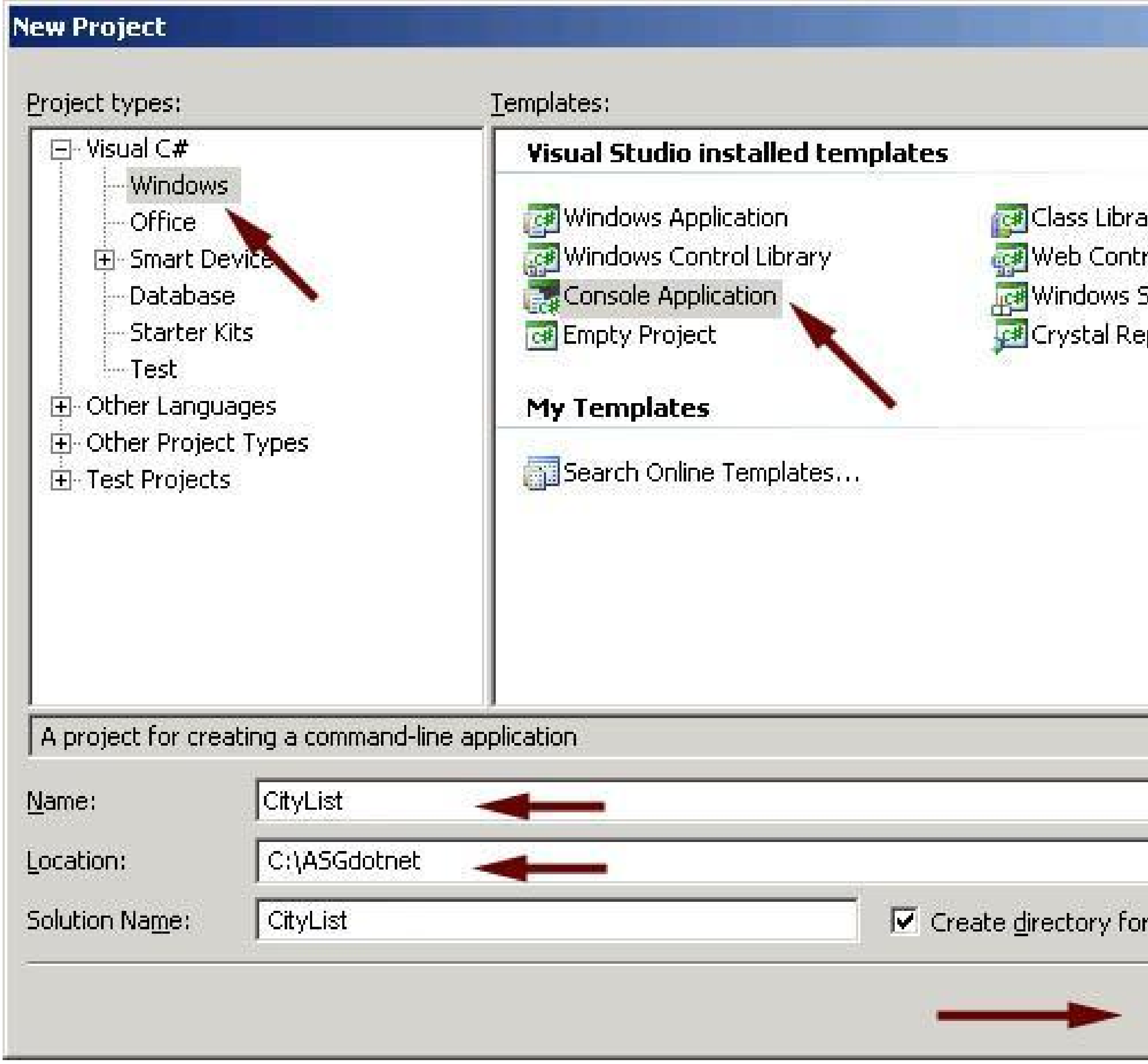
This file contains a proxy class exposing both synchronous and asynchronous methods for each Web Service operation provided by Portus.

For instance, for the list operation, the proxy class has the following methods: list, Beginlist, and Endlist. The list method of the proxy class is used to communicate with Portus synchronously, but the Beginlist and Endlist methods are used to communicate with a Portus server asynchronously. For more information about asynchronous communication with a Web Service please refer to the .NET documentation.

2 Start MS Visual Studio, create a new project with File -> New - Project (or the shortcut Ctrl+Shift +N):



Create a C# Console Application, assign a name to it, specify the storage location, click **OK**



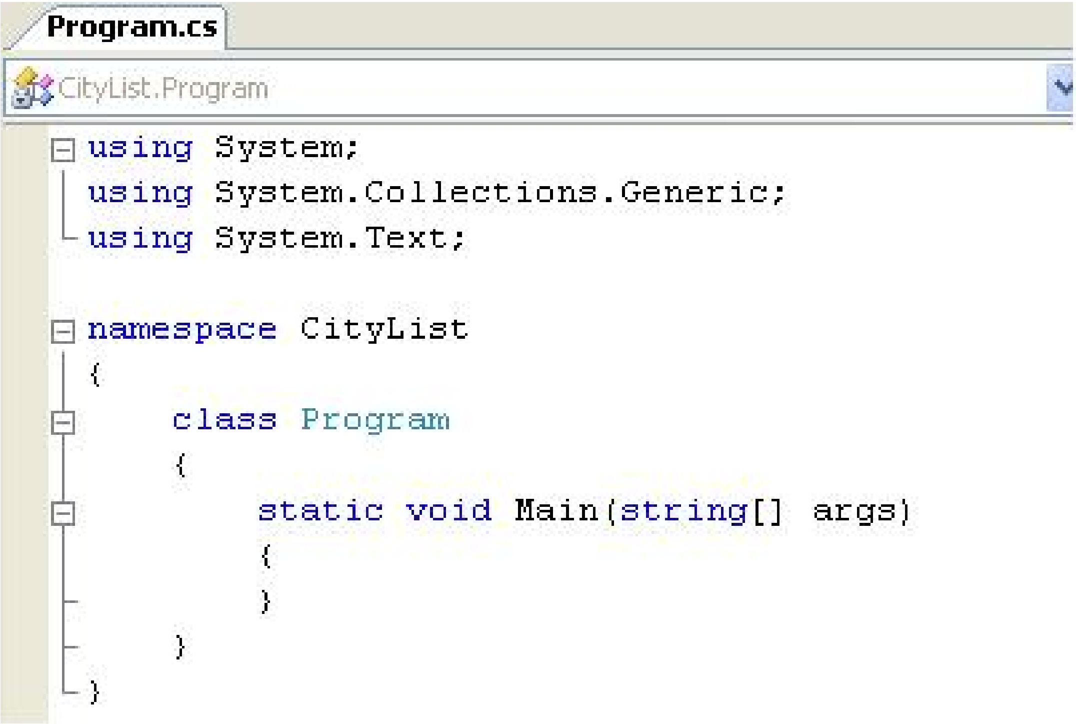
A project for creating a command-line application

Name: CityList

Location: C:\ASGdotnet

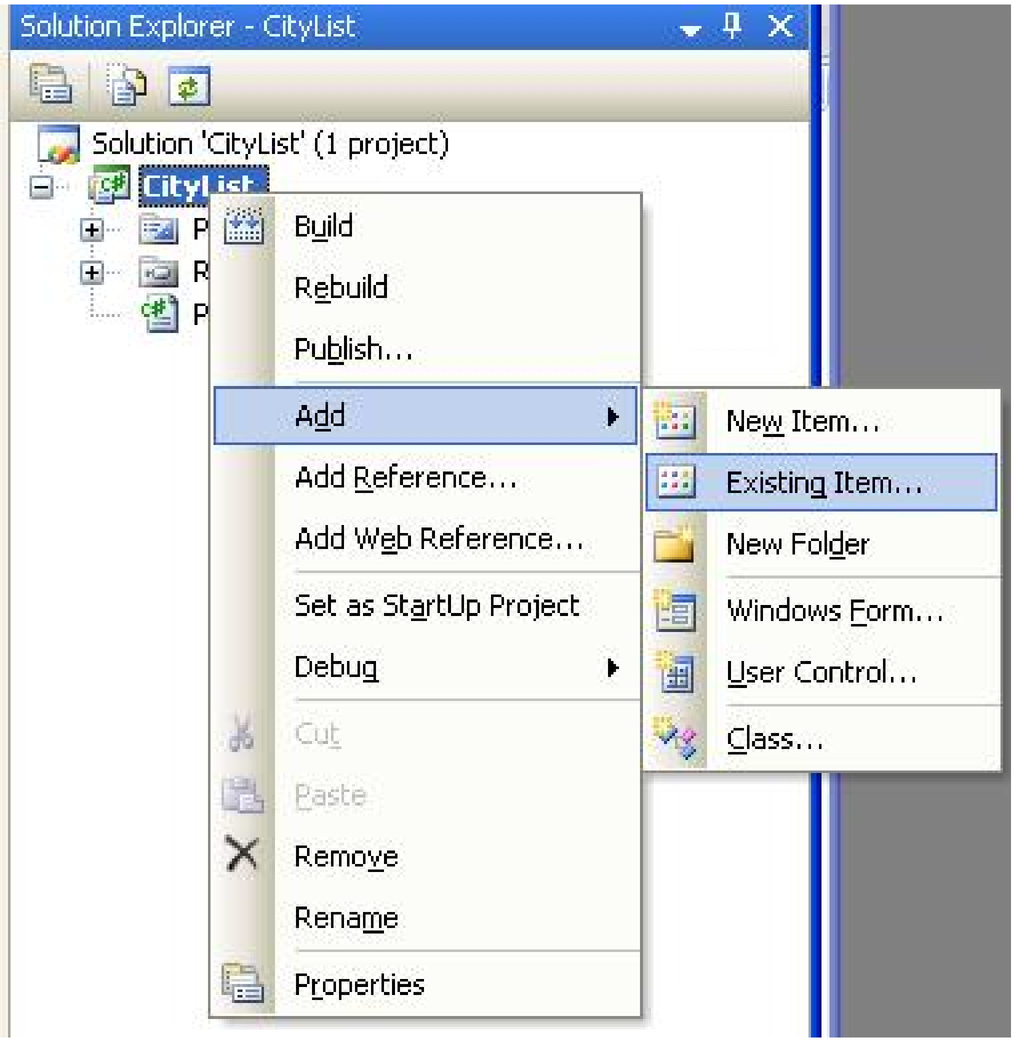
Solution Name: CityList ☑ Create directory for

A skeleton class file has been generated into your project workspace, with the required class definition and an empty Main method

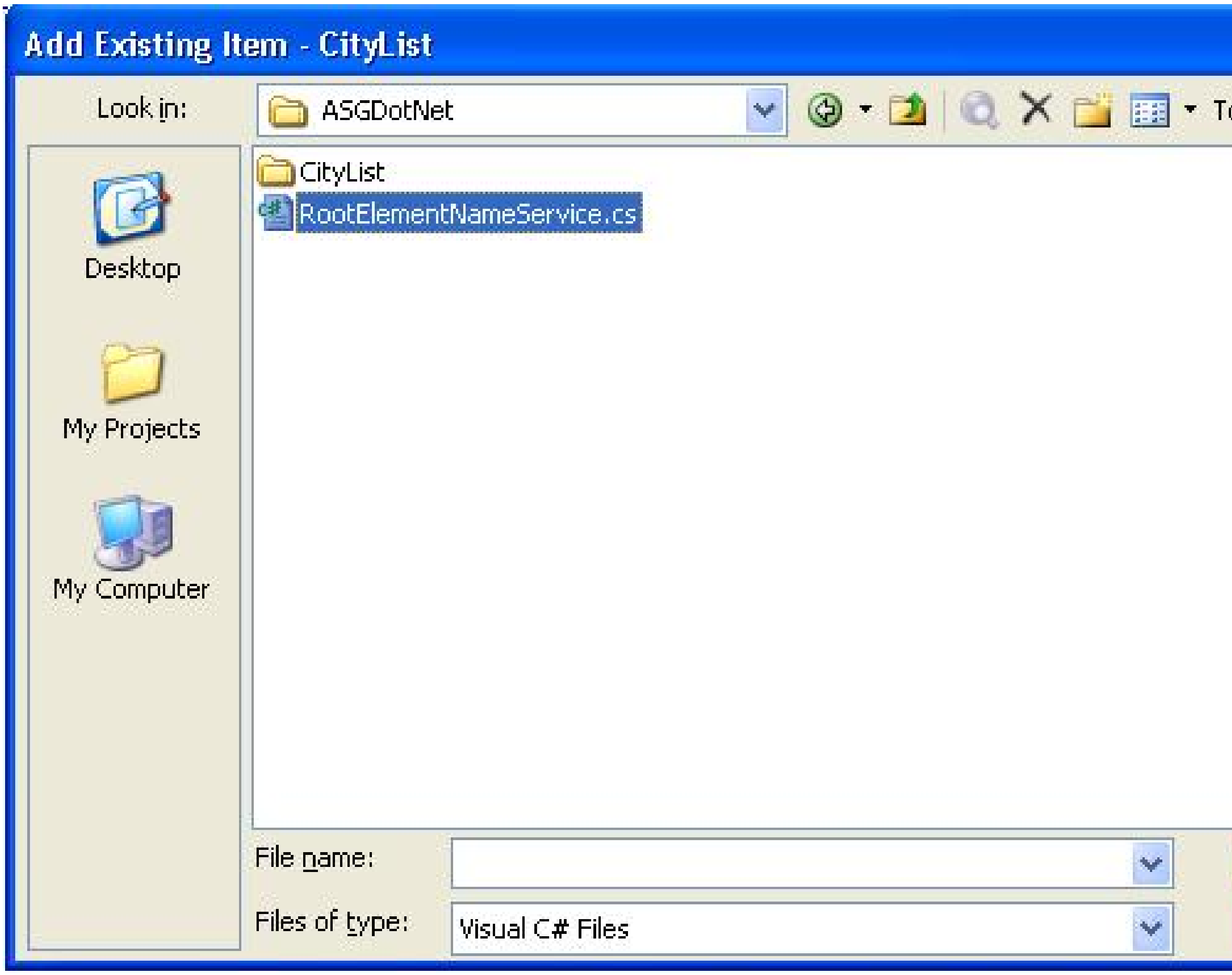


```
using System;
using System.Collections.Generic;
using System.Text;

namespace CityList
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

3 First of all, import the generated proxy into the project, right-click on the project name, select **Add Existing Item**

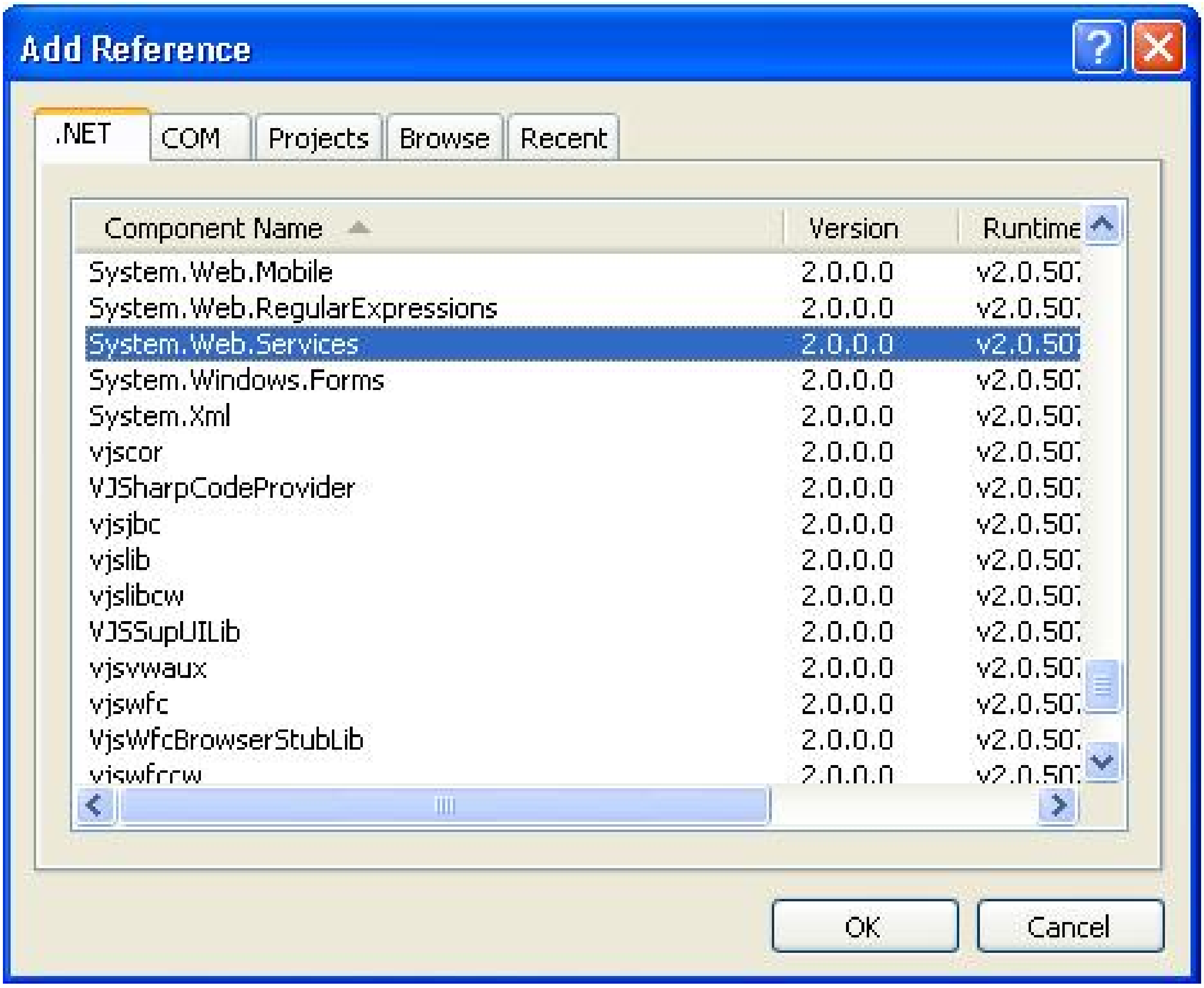


select the RootElementNameService.cs proxy file you created earlier, and click **Add**



The proxy has been added to the project

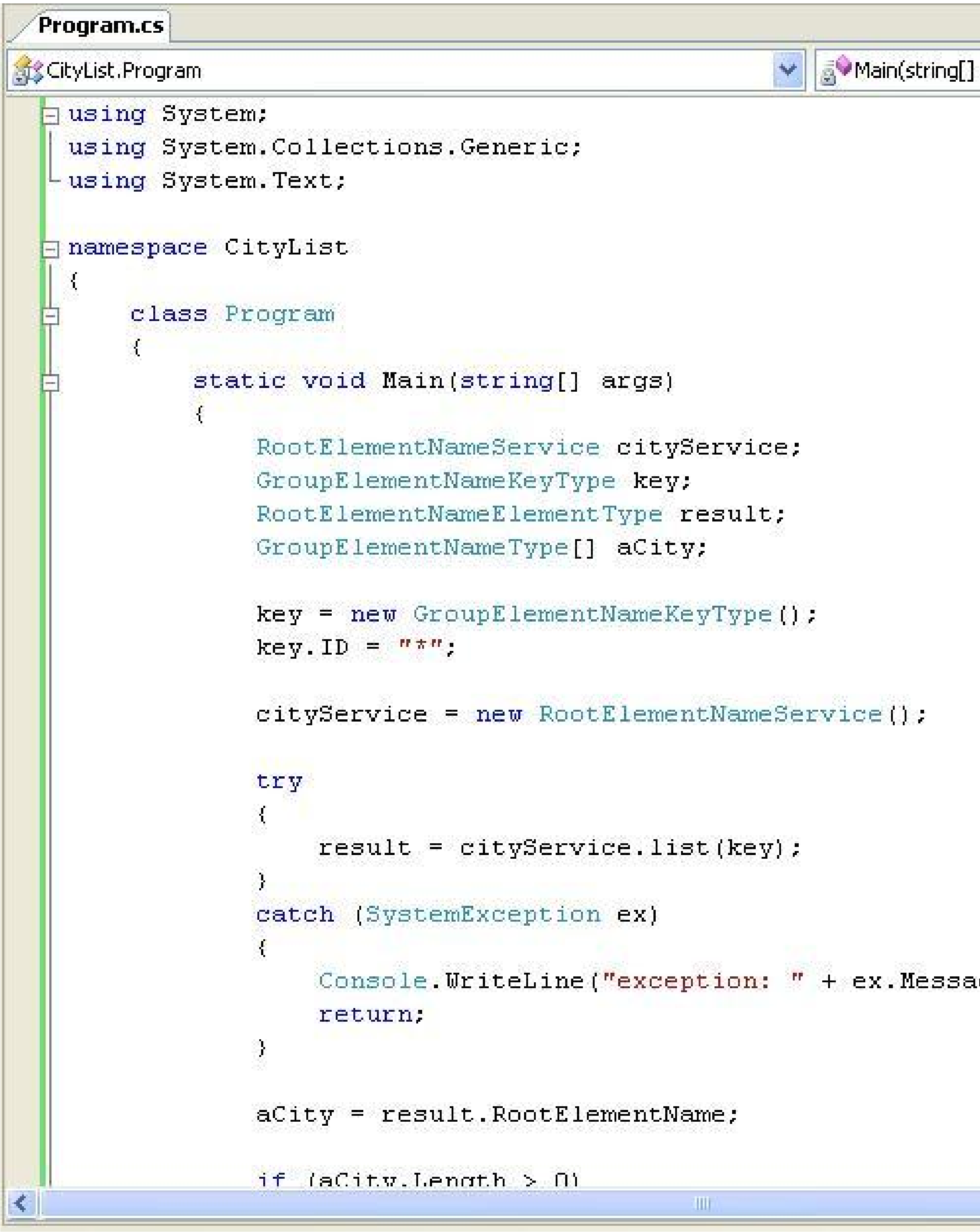
You now need to add a reference to the .NET System.Web.Services component implementing the SOAP interface. In the project explorer, right click on the project name, select **Add Reference**



Component Name	Version	Runtime
System.Web.Mobile	2.0.0.0	v2.0.50
System.Web.RegularExpressions	2.0.0.0	v2.0.50
System.Web.Services	2.0.0.0	v2.0.50
System.Windows.Forms	2.0.0.0	v2.0.50
System.Xml	2.0.0.0	v2.0.50
vjscor	2.0.0.0	v2.0.50
VJSharpCodeProvider	2.0.0.0	v2.0.50
vjsjbc	2.0.0.0	v2.0.50
vjslib	2.0.0.0	v2.0.50
vjslibcw	2.0.0.0	v2.0.50
VJSSupUILib	2.0.0.0	v2.0.50
vjsvwaux	2.0.0.0	v2.0.50
vjswfc	2.0.0.0	v2.0.50
VjsWfcBrowserStubLib	2.0.0.0	v2.0.50
vjswfccw	2.0.0.0	v2.0.50

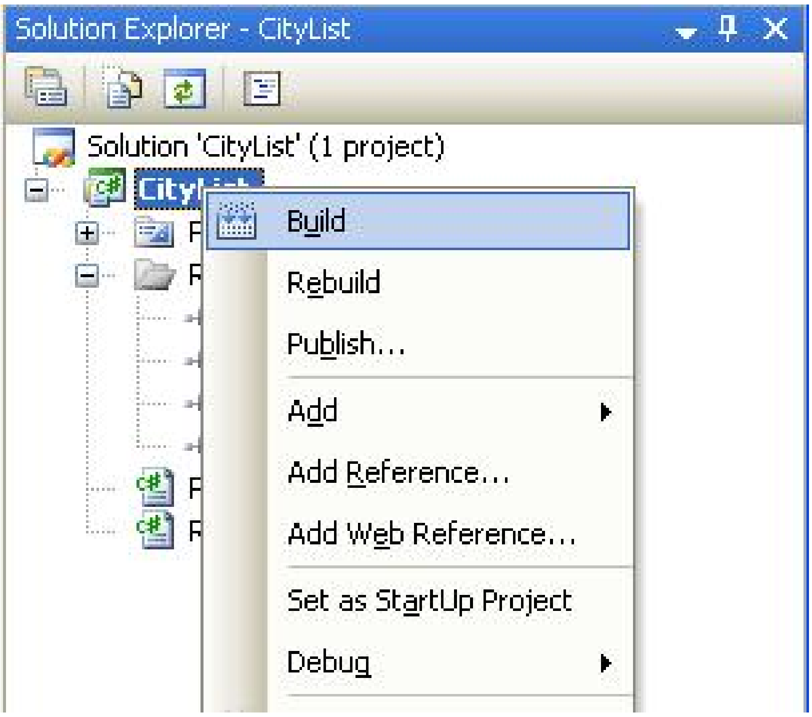
Select **System.Web.Services** and click **OK**

4 Remove the generated code from the newly added class entirely, use (paste) the code from *MySQL_City.cs* to create your first C# program accessing MySQL Web Service via Portus.

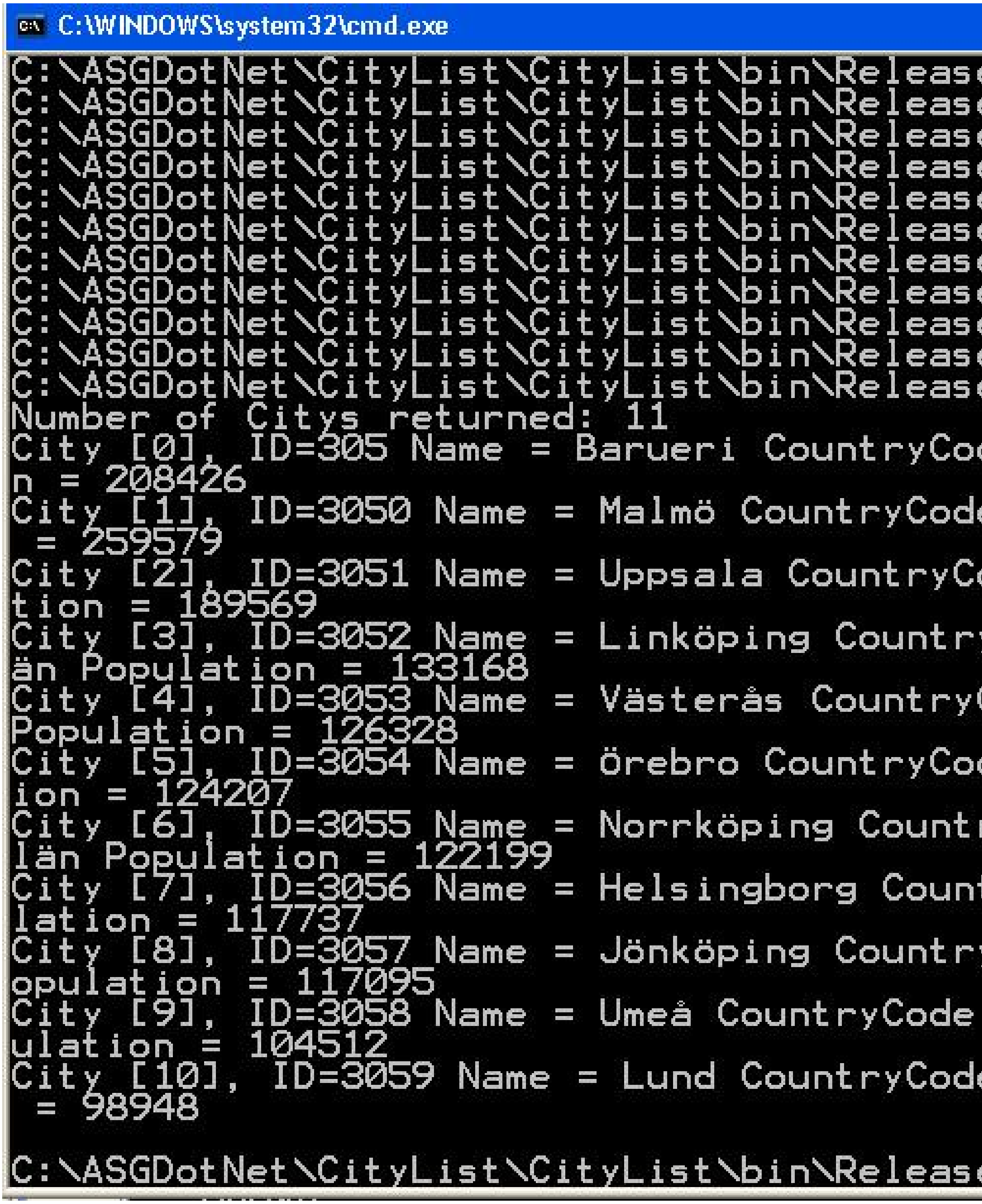


```
using System;
using System.Collections.Generic;
using System.Text;

namespace CityList
{
    class Program
    {
        static void Main(string[] args)
        {
            RootElementNameService cityService;
            GroupElementNameKeyType key;
            RootElementNameElementType result;
            GroupElementNameType[] aCity;

            key = new GroupElementNameKeyType();
            key.ID = "*";

            cityService = new RootElementNameService();

            try
            {
                result = cityService.list(key);
            }
            catch (SystemException ex)
            {
                Console.WriteLine("exception: " + ex.Messa
                return;
            }

            aCity = result.RootElementName;

            if (aCity.Length > 0)
```

5 Build the application. Right-click on the project name in the project explorer, click **Build**



6 Open a command window, change to the project's build-directory Execute the compiled console application, CityList.exe, the output will look as follows:



```
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
C:\ASGDotNet\CityList\CityList\bin\Releas
Number of Citys returned: 11
City [0], ID=305 Name = Barueri CountryCo
n = 208426
City [1], ID=3050 Name = Malmö CountryCod
 = 259579
City [2], ID=3051 Name = Uppsala CountryC
tion = 189569
City [3], ID=3052 Name = Linköping Countr
än Population = 133168
City [4], ID=3053 Name = Västerås Country
Population = 126328
City [5], ID=3054 Name = Örebro CountryCo
ion = 124207
City [6], ID=3055 Name = Norrköping Count
län Population = 122199
City [7], ID=3056 Name = Helsingborg Coun
lation = 117737
City [8], ID=3057 Name = Jönköping Countr
opulation = 117095
City [9], ID=3058 Name = Umeå CountryCode
ulation = 104512
City [10], ID=3059 Name = Lund CountryCod
 = 98948

C:\ASGDotNet\CityList\CityList\bin\Releas
```

7 This sample selects all Cities records with a ID of 305n, you may want to experiment varying the key data, this is easily done by modifying the properties passed to the generated classes. E.g. try the following to list all records for Cities whose ID start "400".

```
key.ID = "400*";
```

Or to get a list of all rows in this table:

```
key.ID = "*";
```

SOAP over IBM MQ Series Tutorial

Tutorial: Send SOAP requests to MQ

Portus and IBM MQ Series can be used together to enable sending SOAP requests to an MQ queue and to receive a response from a queue.

These request are asynchronous so it means that multiple requests may be sent from the client even though Portus may not be running.

These requests are held on the queue until Portus reads and processes them. Likewise the responses will reside on a queue until such time that a client may request them.

Here is a top level view on the steps involved.

- For a particular web service specify the MQ Manager, a queue to store requests and a queue to store responses.
- The resultant WSDL for the web service can be read by a client which will enable it to generate the appropriate SOAP request to send to the input queue.
- The resultant WSDL for the web service can be read by a client which will enable it to generate the appropriate SOAP request to read from an output queue.

Details

This tutorial will make use of Java wrapper/stub classes to access a Portus web service.

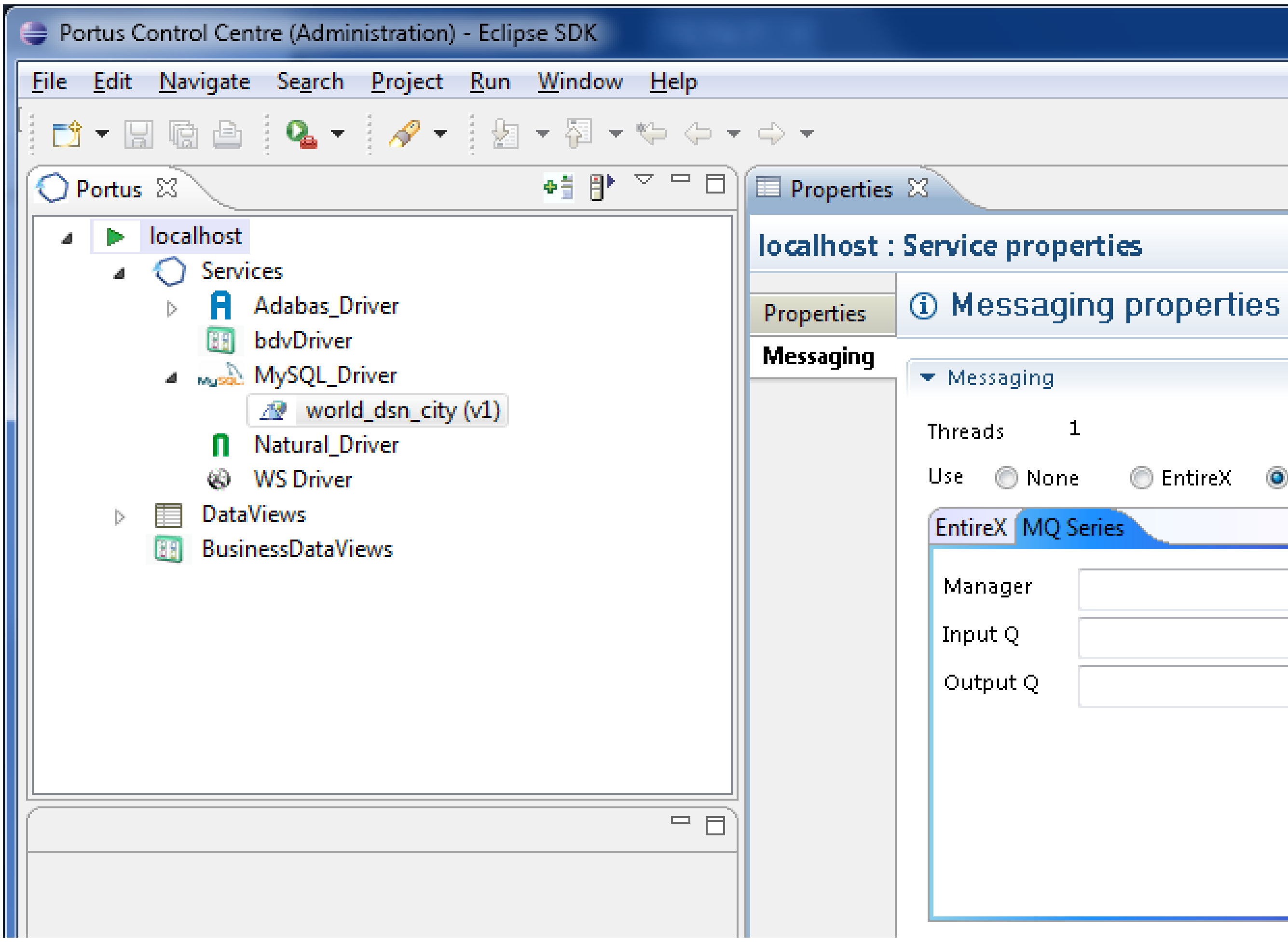
Java wrapper/stub classes are generated using the *Apache Axis2* feature *WSDL2Java*.

If you do not have it already, download and install the latest Axis2 kit.

The web service which we will be using is that generated by discovering the city table in the World database supplied by a MySQL installation. See *MySQL Tutorials* for details on how this is achieved.

Ensure that the MQ Manager to be used has been started and that an input and output queue created. Also ensure that a listener has been started.

1 Select the world_dsn_city web service and open the Messaging tab. Select the MQ radio button and tab. Fill in the details appropriate to your MQ installation.



localhost : Service properties

Messaging properties

Threads 1

Use ○ None ○ EntireX ●

Manager

Input Q

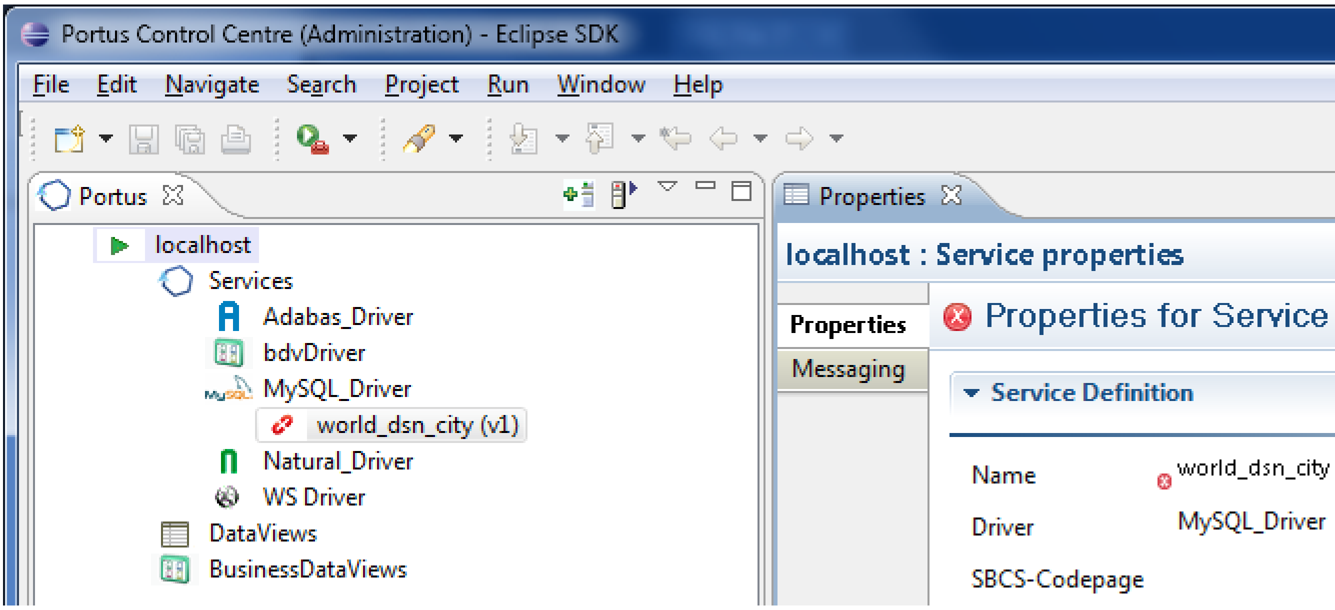
Output Q

2 As per the hint, switch back to Service properties and select the **Save** button when complete.

3 Portus will now make contact with the appropriate messaging system. If an error occurs it will be highlighted as follows.

Check the error_log to find the cause. The default location of the Apache error_log is [SERVER_INST]/Apache22/logs/error_log replacing [SERVER_INST] with the location in which you have installed Portus. Please correct and try again.

If successful then carry on to next step.



localhost : Service properties

Properties for Service

Name world_dsn_city

Driver MySQL_Driver

SBCS-Codepage

4 Switch to the Java perspective (Window -> Open Perspective -> Other... -> Java (default). For more information on this aspect of Eclipse see *Getting started with Eclipse.*

5 Create a new Java-project naming it "SOAPMQTutorial".

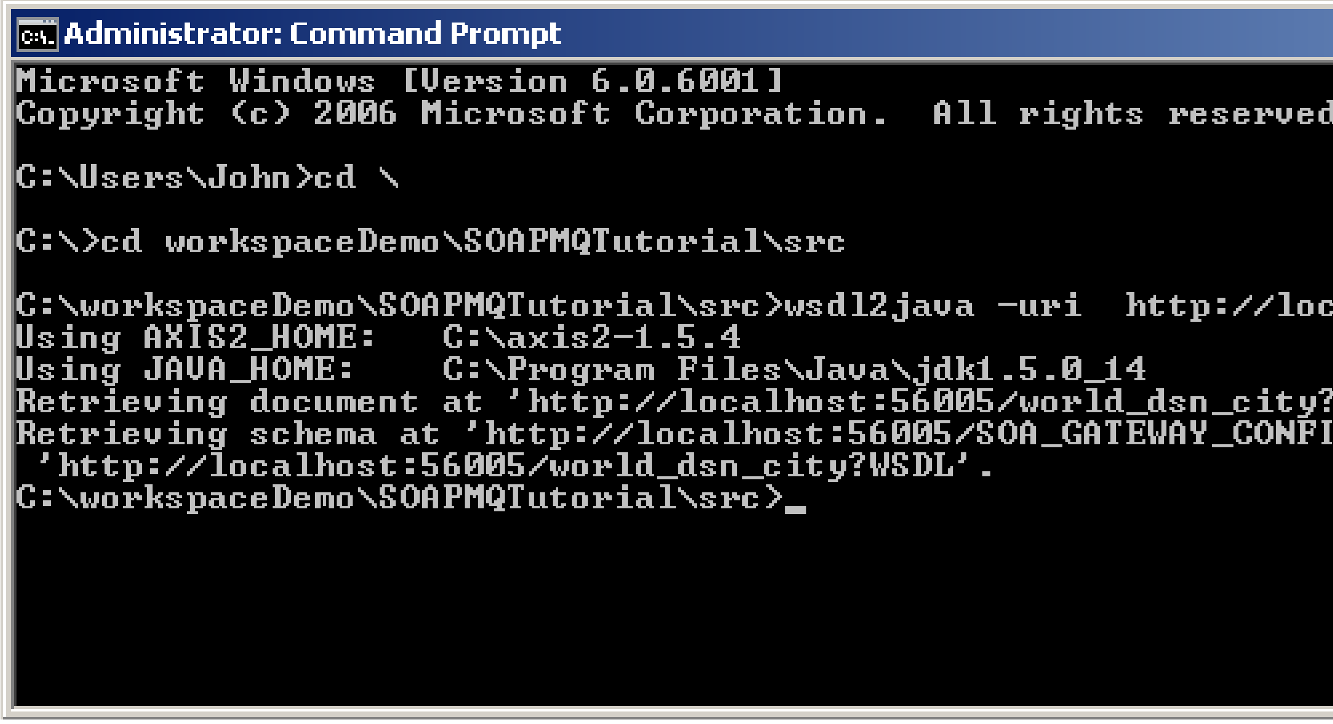
6 Add all .jar files from the axis2 "lib" directory to the project's Build-Path.

7 Right-click the "SOAPMQTutorial" project folder, select "Build Path", then "Add External Archives..."

8 Add all the .jar files from your MQ lib directory e.g. ..\IBM\WebSphere MQ\java\lib

9 Open a command prompt (aka "DOS box"), change to the "SOAPMQTutorial\src" directory and run the following command:

```
wsdl2java -uri http://<yourserver>:<yourport>/world_dsn_city?WSDL
 -o ..\ -p SoaGW
```



```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved

C:\Users\John>cd \

C:\>cd workspaceDemo\SOAPMQTutorial\src

C:\workspaceDemo\SOAPMQTutorial\src>wsdl2java -uri  http://loc
Using AXIS2_HOME:   C:\axis2-1.5.4
Using JAVA_HOME:    C:\Program Files\Java\jdk1.5.0_14
Retrieving document at 'http://localhost:56005/world_dsn_city?
Retrieving schema at 'http://localhost:56005/SOA_GATEWAY_CONFI
 'http://localhost:56005/world_dsn_city?WSDL'.
C:\workspaceDemo\SOAPMQTutorial\src>_
```

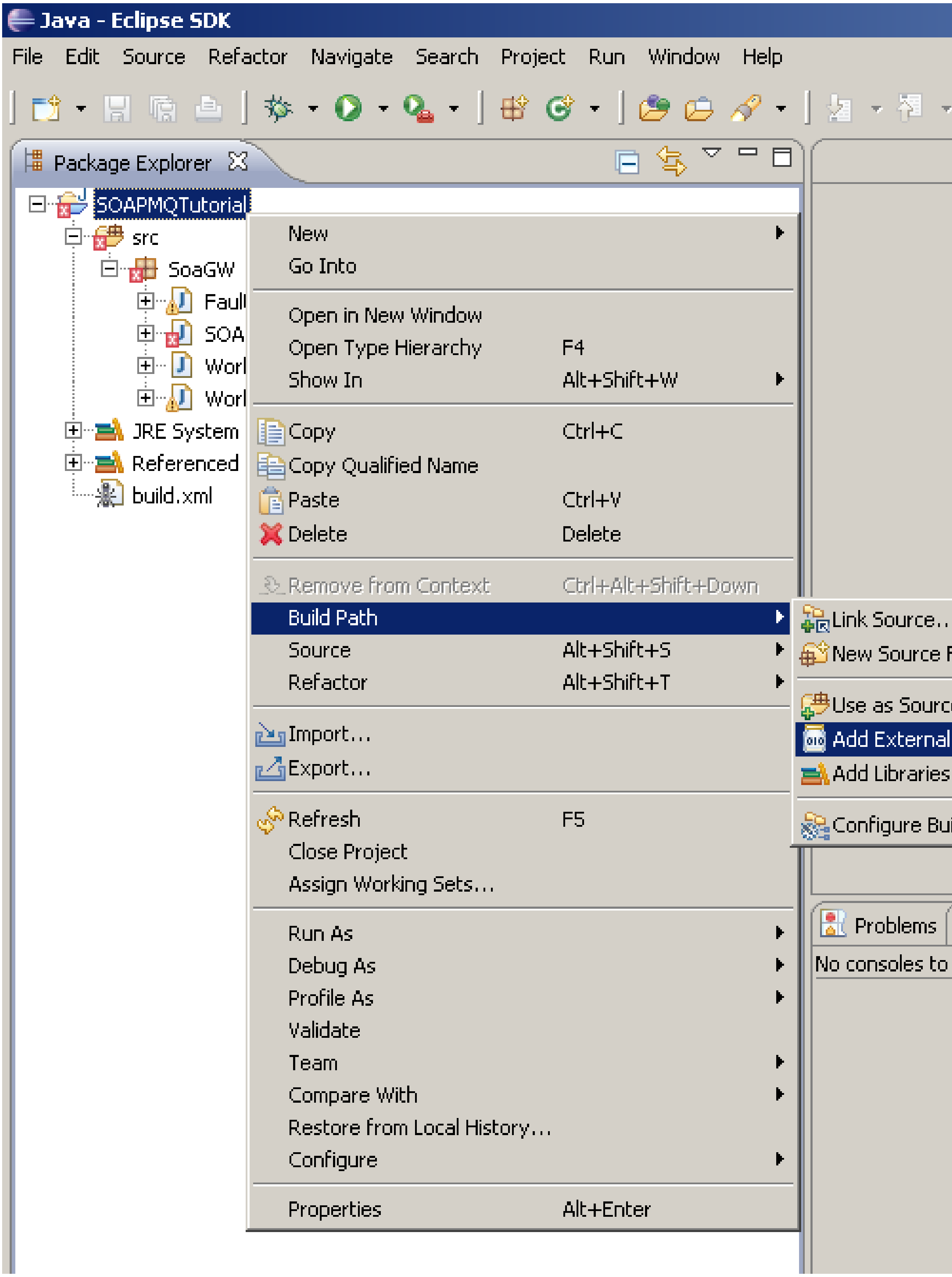
10 The following items are generated from the Portus WSDL:

- A "Stub" class implementing all types and operations (ports / bindings).
- A CallbackHandler - a stub class (not used in this tutorial) providing hooks for client-side extensions to the generated result- and error handlers.
- A Fault class.

11 Get file *SOAPEntireXTut.java* and, when prompted, save in the ..\SOAPMQTutorial\src\SoaGW directory. Right-click on SOAPMQTutorial and select Refresh(F5). SOAPMQTut.java should appear in the explorer window.

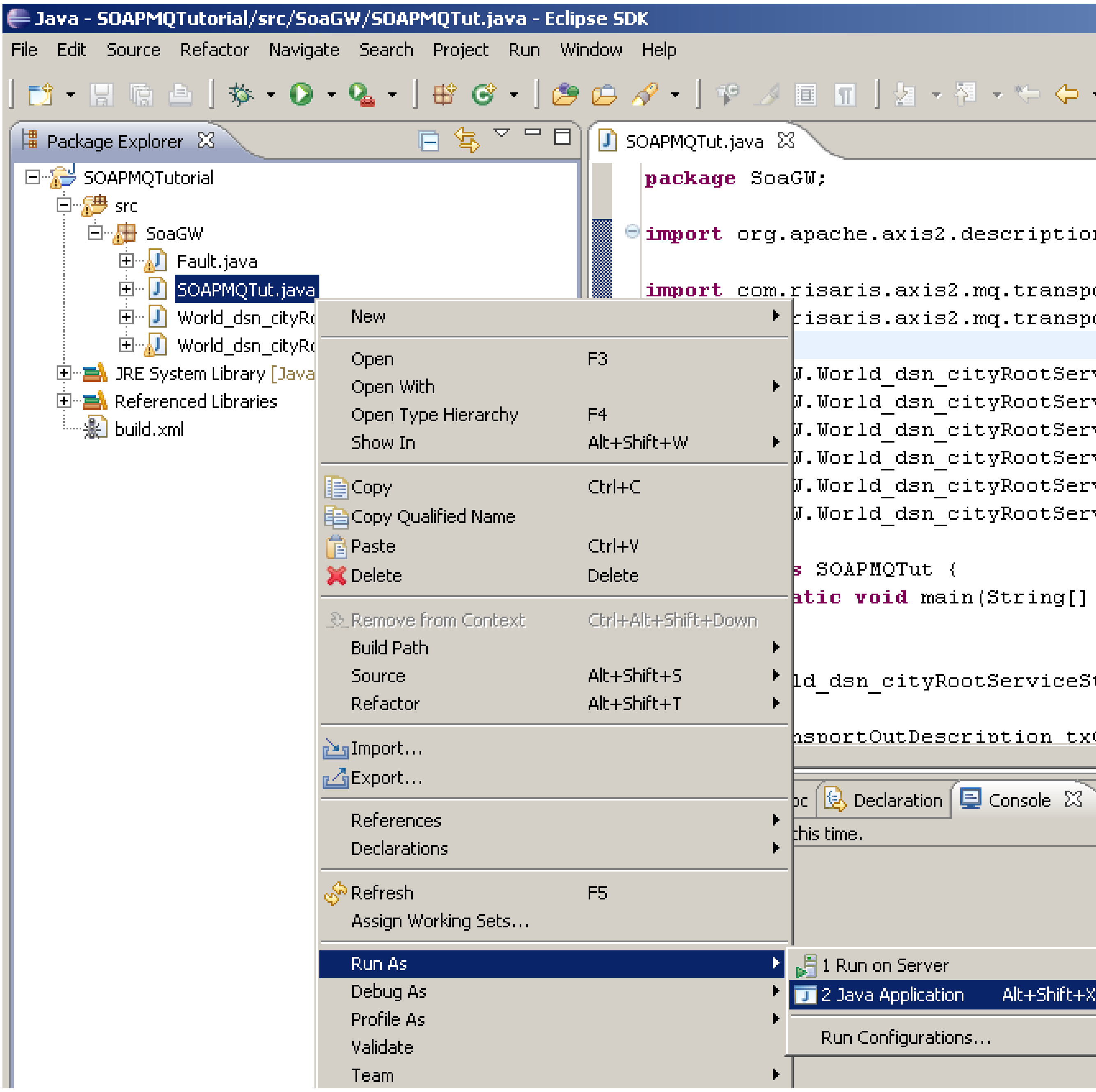
12 Get file *mqTransports-0.1.jar* and, when prompted, save it in a local directory.

13 Right-click on SOAPMQTutorial and select Build Path -> Add External Archives...



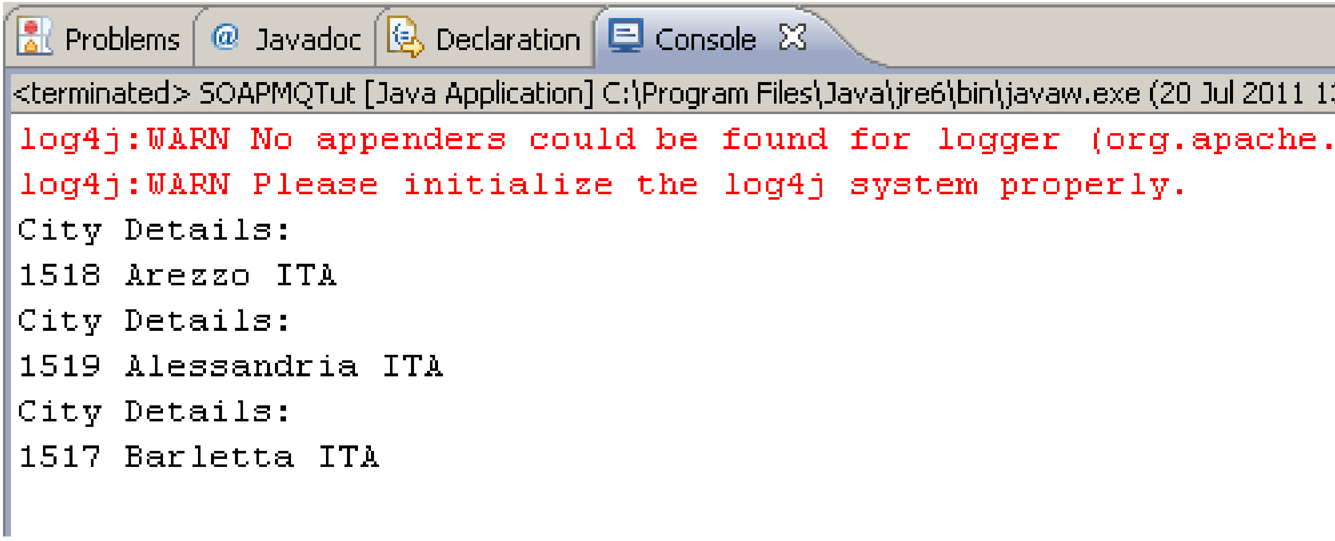
Navigate to where mqTransports-0.1.jar was saved and select Open to add it the project.

14 Right-click on SOAPMQTut.java and select Run As -> Java Application:



```
package SoaGW;

import org.apache.axis2.descriptio

import com.risaris.axis2.mq.transp
```

15 The output appears in the "Console" window:



```
<terminated> SOAPMQTut [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (20 Jul 2011 1
log4j:WARN No appenders could be found for logger (org.apache.
log4j:WARN Please initialize the log4j system properly.
City Details:
1518 Arezzo ITA
City Details:
1519 Alessandria ITA
City Details:
1517 Barletta ITA
```

Usage Governance Tutorials

Usage Governance can be reported in 3 ways and written to 3 output types.

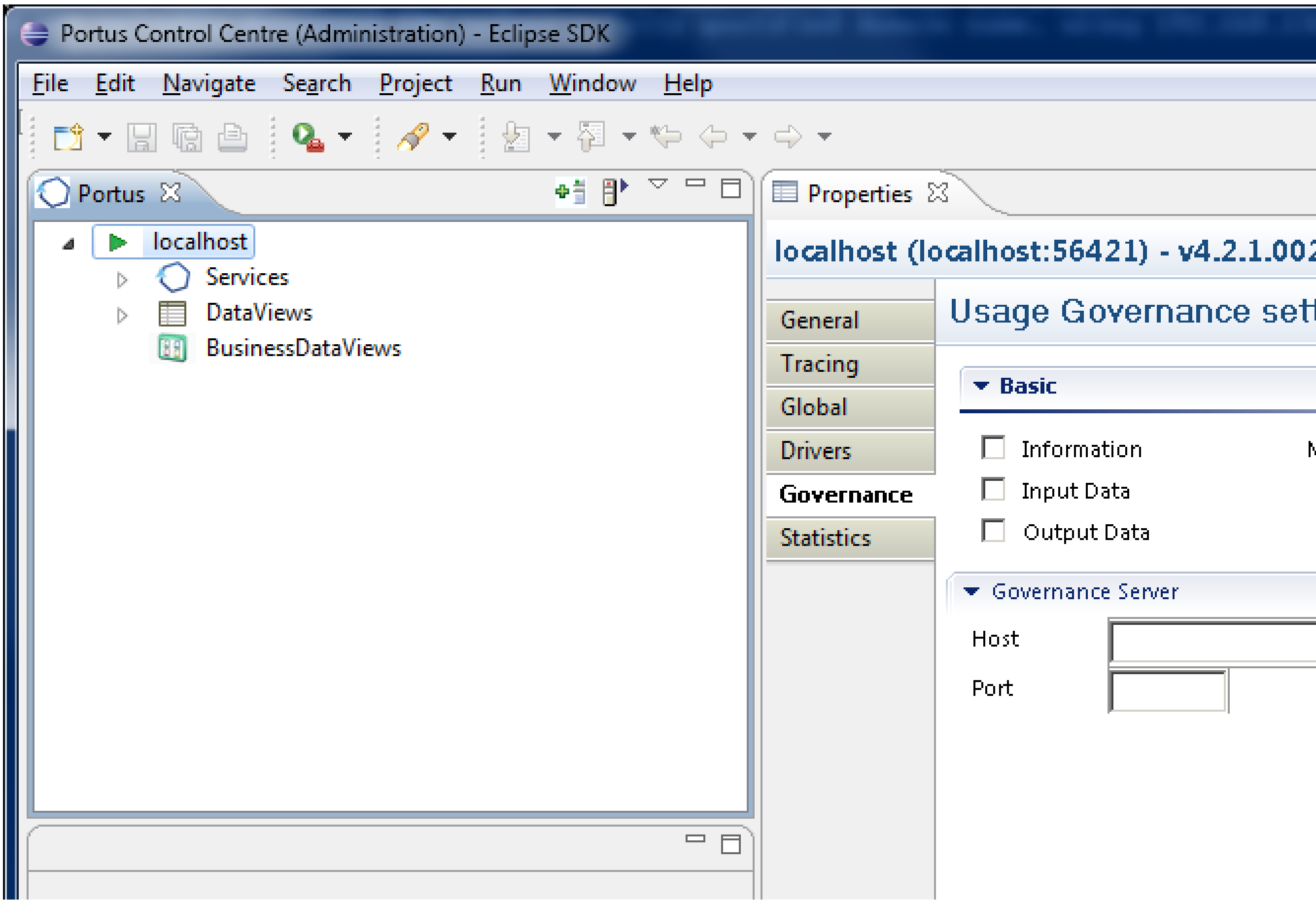
- *Using the local file system* on page 222
- *Using another Portus server* on page 223
- *Using MOM* on page 224

Using the local file system

Tutorial: Write Usage Governance data to the local file system

1. In the Control Centre select the server for which usage governance data will be generated.
2. Select the Governance tab in the Properties view.

   To turn data collection on the Information box must be selected. If required Input Data and Output Data may also be selected.



3. Click the Save button.
4. Stop and restart the server.See *here* on how to do this.
5. Issue a request to Portus e.g. a get request.
6. Go to Portus configuration directory.

   The default location of Portus configuration folder is [SERVER_INST]/Apache22/configuration replacing [SERVER_INST] with the location in which you have installed Portus.
7. A file should be present in the format soag_usage_governance_yyyy_mm_dd_hh_mm_ss_ms.txt. Note that it is not possible to view this file while Portus is running. Stopping the server will create an XML file with the same name.
8. Open the file to check its contents.
9. There should be 1 entry for the get request plus all the governance data collected as per the options selected in 2.

Using another Portus server

Tutorial: Write Usage Governance data to another Portus server

1 In the Control Centre select the server for which usage governance data will be generated.

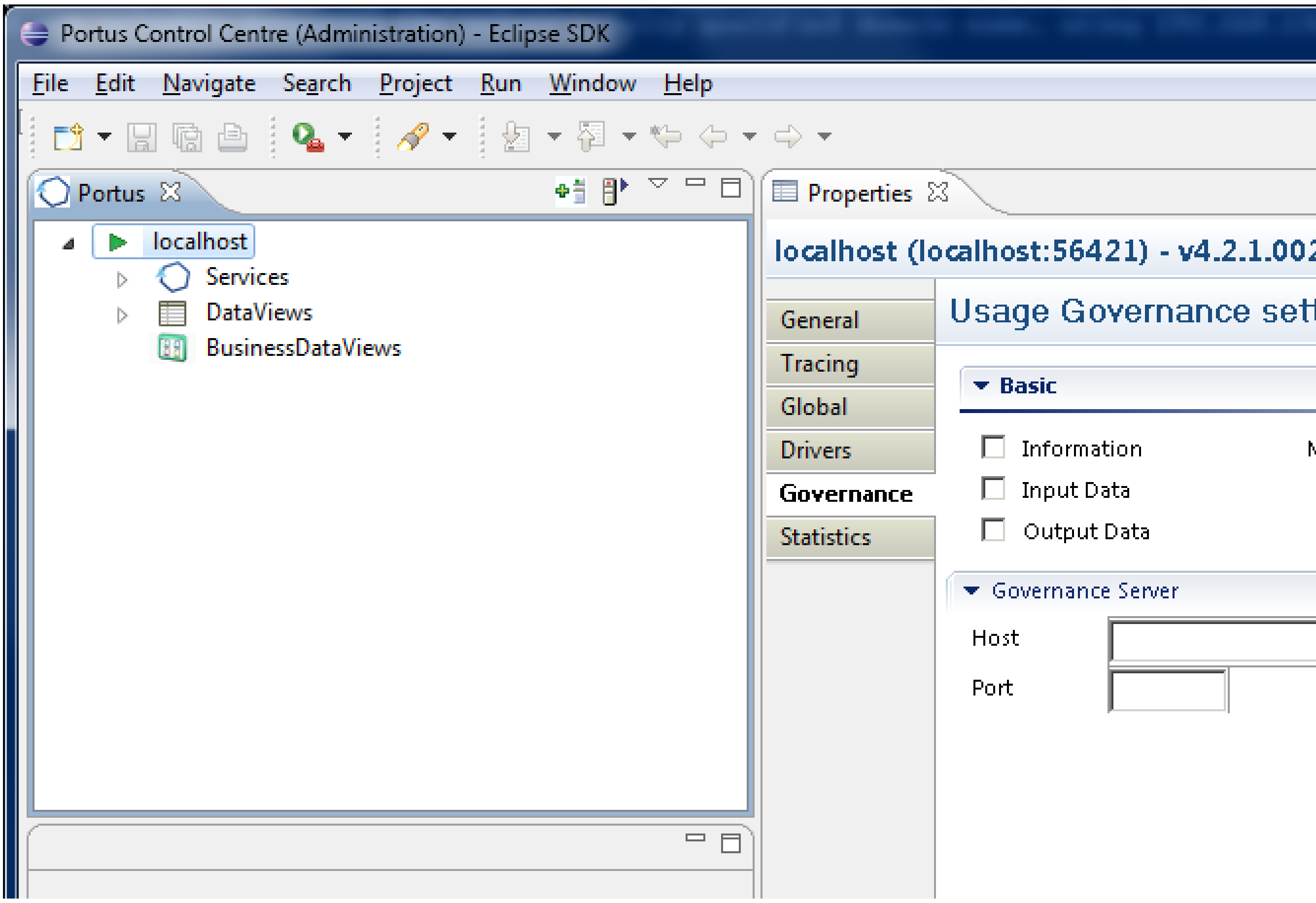
2 Select the Governance tab in the Properties view.

To turn data collection on the Information box must be selected. If required Input Data and Output Data may also be selected.

3 The Governance Server section is where we enter the details for another Portus.

- Host : the IP address for the server.
- Port : the port number of the server.

4 Enter these and save.



5 Stop the server.See *here* on how to do this.

6 Prior to restarting the server the following must be observed:

- The Governance Server is running i.e. that Portus with the host and port number entered has been started and is awaiting requests.
- The governance web service, usagegovernance, has been created successfully and is loaded.

  If this has not yet been done, following the instructions *here* to do this.

7 Start the server.See *here* on how to do this.

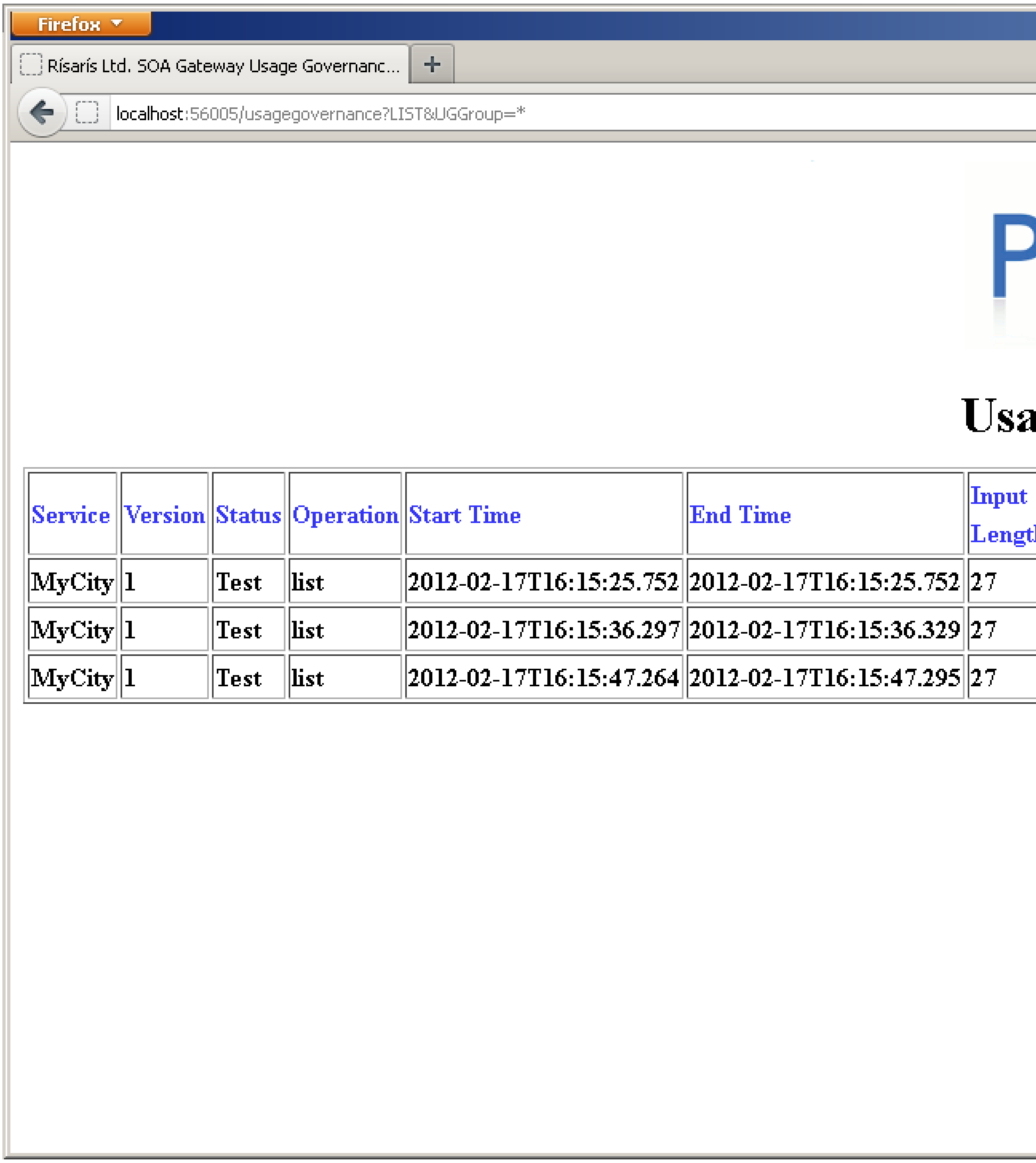
8 On startup an initial connection is made to the governance server to verify its details. If this is unsuccessful an appropriate error will be written to the error log so this should be checked now.

The default location of the Apache error_log is [SERVER_INST]/Apache22/logs/error_log replacing [SERVER_INST] with the location in which you have installed Portus.

9 Issue a request to Portus e.g. a list request.

10 There are a couple of ways to check that the data collection has reached its destination.

- Query the database table directly..
- Issue a list request on the usage governance web service e.g.


Firefox
Risaris Ltd. SOA Gateway Usage Governanc...
+
localhost:56005/usagegovernance?LIST&UGGroup=*
P

Usa

Service	Version	Status	Operation	Start Time	End Time	Input Lengt
MyCity	1	Test	list	2012-02-17T16:15:25.752	2012-02-17T16:15:25.752	27
MyCity	1	Test	list	2012-02-17T16:15:36.297	2012-02-17T16:15:36.329	27
MyCity	1	Test	list	2012-02-17T16:15:47.264	2012-02-17T16:15:47.295	27

Using MOM

Tutorial: Write Usage Governance data to a MOM system

Usage Governance data can be written to a WebSphere MQ queue or an Software AG EntireX server.

It is outside the scope of this tutorial to detail exactly how these are set up so it is assumed that the chosen Messaging system is configured correctly.

In this tutorial we will be using 2 Portus servers. One will direct usage governance data collected to a message queue and the other will read from this queue and process the input. The end result is that the target of the usage governance web service will receive an add request complete with the data.

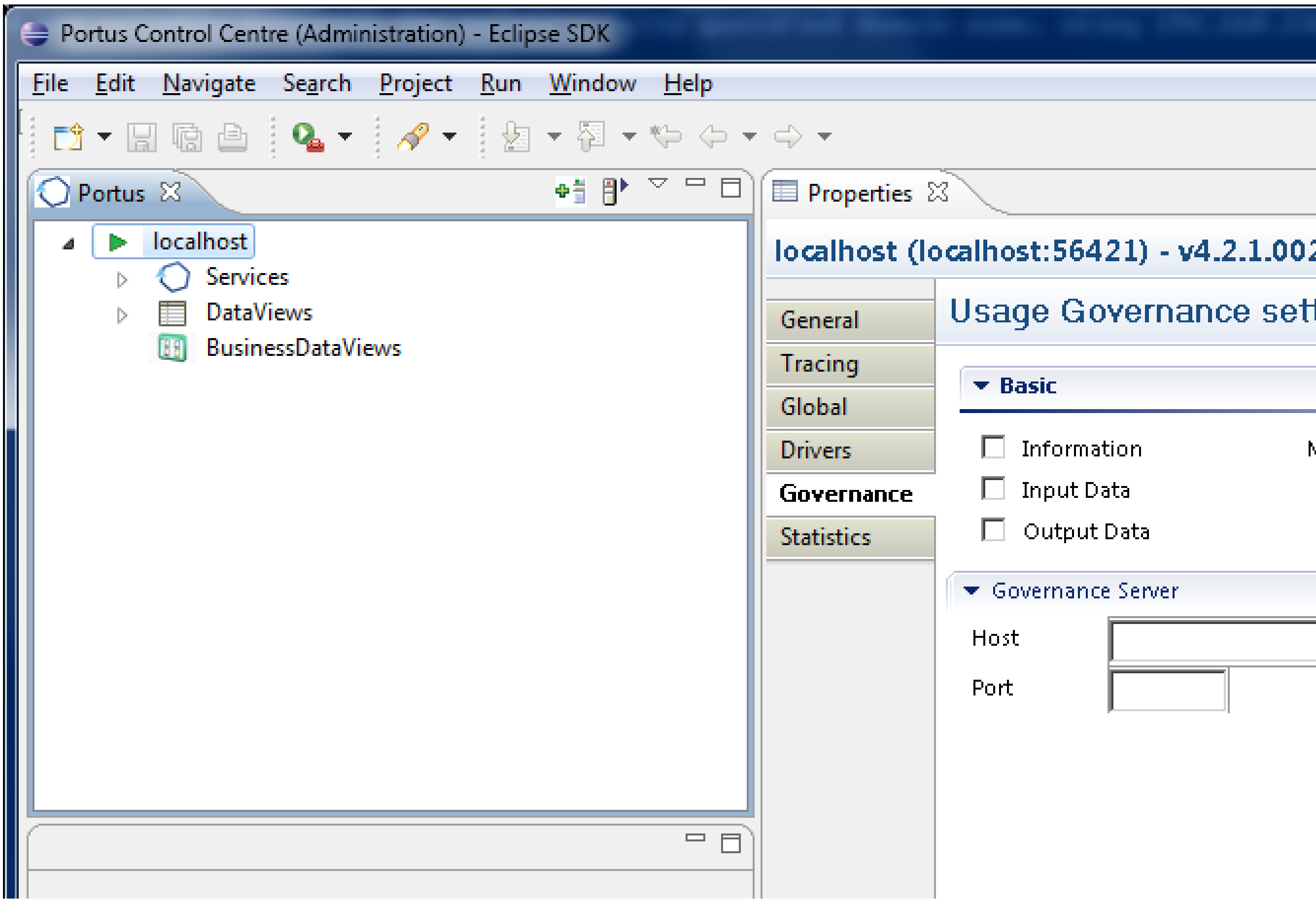
1 In the Control Centre select the server for which usage governance data will be generated.

2 Select the Governance tab in the Properties view.

To turn data collection on the Information box must be selected. If required Input Data and Output Data may also be selected.

3 The Messaging section allows one to enter the details required for None, MQ or EntireX :

Select the radio button of MQ or EntireX and fill in as appropriate for you MOM installation. Note that the queue will be opened for output and later on in this tutorial will be opened as an input queue.



□ Information

□ Input Data

□ Output Data

Host

Port

4 Click the Save button.

5 Stop and restart the server.See *here* on how to do this.

6 At this point Portus should have connected successfully to the manager and opened the output queue specified. Check the error log for any errors at this point.

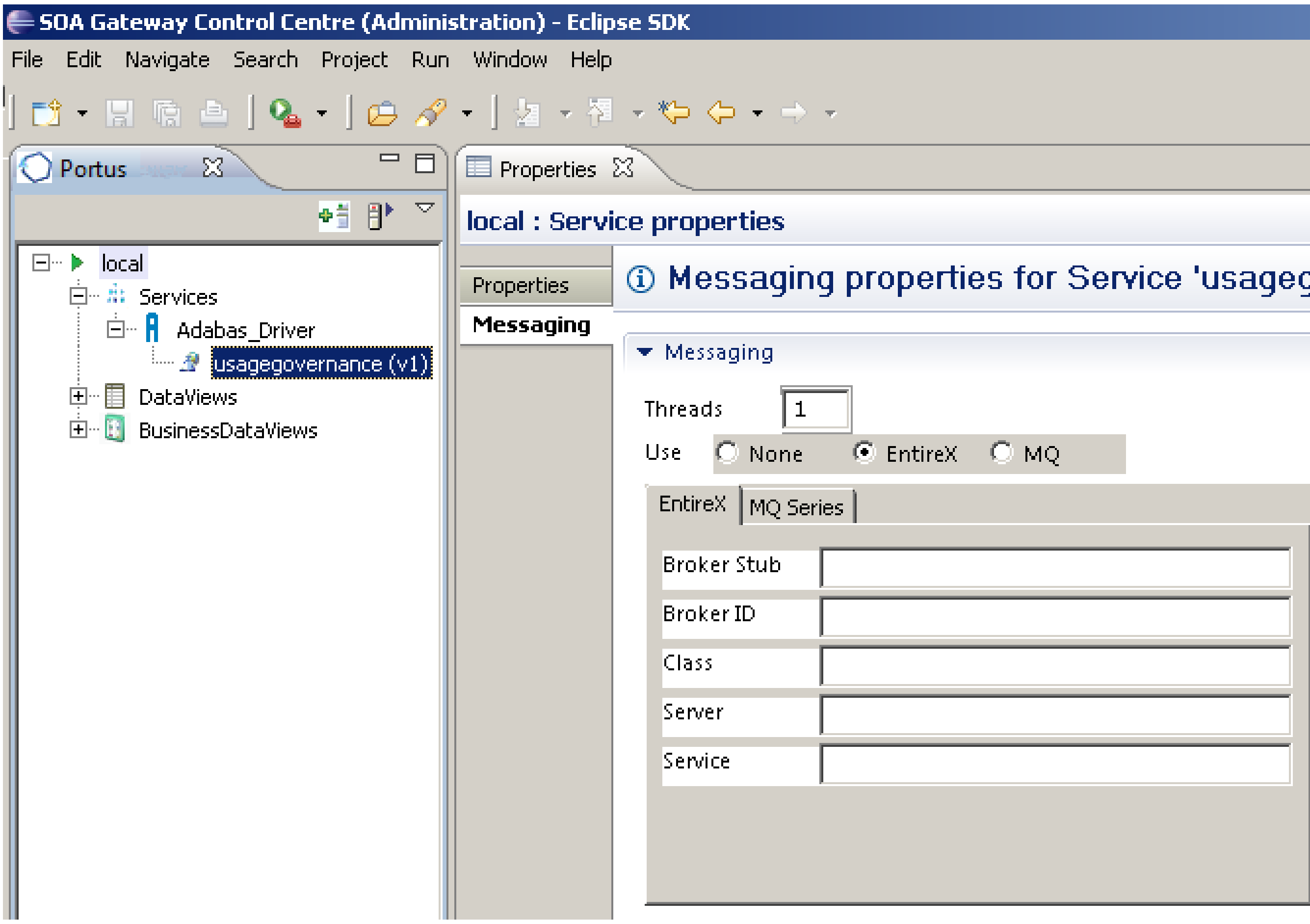
The default location of the Apache error_log is [SERVER_INST]/Apache22/logs/error_log replacing [SERVER_INST] with the location in which you have installed Portus.

7 Issue a request to Portus e.g. a get request.

8 An add request, containing usage governance data pertaining to this request, should appear as an entry in the output queue.

Use a search mechanism particular to your messaging system to check that this is the case. Note the syntax of this request is supported by the usage governance web service. If the latter has not yet been created, follow the steps outlined *here* to do so.

9 In the Control Centre, select the second Portus server. Open the Messaging tab in the Properties view for the usagegovernance web service:


SOA Gateway Control Centre (Administration) - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Portus
local
Services
Adabas_Driver
usagegovernance (v1)
DataViews
BusinessDataViews
Properties
local : Service properties
Properties
Messaging
Messaging properties for Service 'usageg
Messaging
EntireX
MQ Series

Threads 1

Use ○ None ● EntireX ○ MQ

Broker Stub

Broker ID

Class

Server

Service

10 Fill in the details for the messaging system you are using. Clearly, for MQ, the Input Q value has to be the value specified for Queue in step 3. For EntireX the details entered here should match those from 3.

Note: The Broker Stub value will depend on the system from which EntireX is being called i.e. broker32.dll for Windows, broker.so for Linux etc.

11 As per the hint, switch back to Service properties and select the Save button when complete.

12 Portus will now make contact with the appropriate messaging system. Check the error log for messages.

The default location of the Apache error_log is [SERVER_INST]/Apache22/logs/error_log replacing [SERVER_INST] with the location in which you have installed Portus.

13 As soon as successful connections have been made, the input queue/service will be read and the message(s) processed.

There are a couple of ways to check the status of this.

- Query the database table directly..
- Issue a list request on the usage governance web service.
- Check the Output Q (MQ) or the EntireX server for responses to the add request to the usage governance web service.

Natural Samples

The following Natural samples demonstrate the interface between Portus and the Natural programming language

- *ASGENVIN - Display natural environment* on page 227

- *ASGECHON - Echo input* on page 227
- *ASGCALCN - Simple calculator* on page 227
- *NSGNATI - Portus Natural interface driver* on page 228

ASGENVIN - Display natural environment

```
DEFINE DATA PARAMETER
1 #NATENVOUT (A80)
1 #OSENVOUT (A80)
1 #TIMEOUT  (A30)
END-DEFINE
*
COMPRESS 'NATURAL' *NATVERS 'PL' *PATCH-LEVEL
         'SRVTYPE =' *SERVER-TYPE ', UI =' *UI INTO #NATENVOUT
*
COMPRESS *OS *OSVERS
         'ON' *HARDWARE INTO #OSENVOUT
*
COMPRESS *DAT4I *TIME INTO #TIMEOUT
END
```

ASGECHON - Echo input

```
DEFINE DATA PARAMETER
1 echoIn  (a30)
1 echoOut (a30)
END-DEFINE
*
MOVE echoIn to echoOut
*
END
```

ASGCALCN - Simple calculator

```
DEFINE DATA
PARAMETER
  1 #OPERATION        (A1)
  1 #OPERAND-1        (I4)
  1 #OPERAND-2        (I4)
  1 #FUNCTION-RESULT  (I4)
LOCAL
  1 #WORK-RESULT      (I4)
END-DEFINE
*
  DECIDE ON FIRST VALUE OF #OPERATION
    VALUE '+'
      COMPUTE #FUNCTION-RESULT = #OPERAND-1 + #OPERAND-2
    VALUE '-'
      COMPUTE #FUNCTION-RESULT = #OPERAND-1 - #OPERAND-2
    VALUE '*'
      COMPUTE #FUNCTION-RESULT = #OPERAND-1 * #OPERAND-2
    VALUE '/'
      IF #OPERAND-2 NE 0 THEN
        COMPUTE #FUNCTION-RESULT = #OPERAND-1 / #OPERAND-2
      ELSE
        MOVE 0 TO #FUNCTION-RESULT
      END-IF
    VALUE '%'
      IF #OPERAND-2 NE 0 THEN
        DIVIDE #OPERAND-1 INTO #OPERAND-2 GIVING #WORK-RESULT
        REMAINDER #FUNCTION-RESULT
      ELSE
        MOVE 0 TO #FUNCTION-RESULT
```

```
       END-IF
      NONE VALUE
        MOVE 0 TO #FUNCTION-RESULT
    END-DECIDE
*
END
```

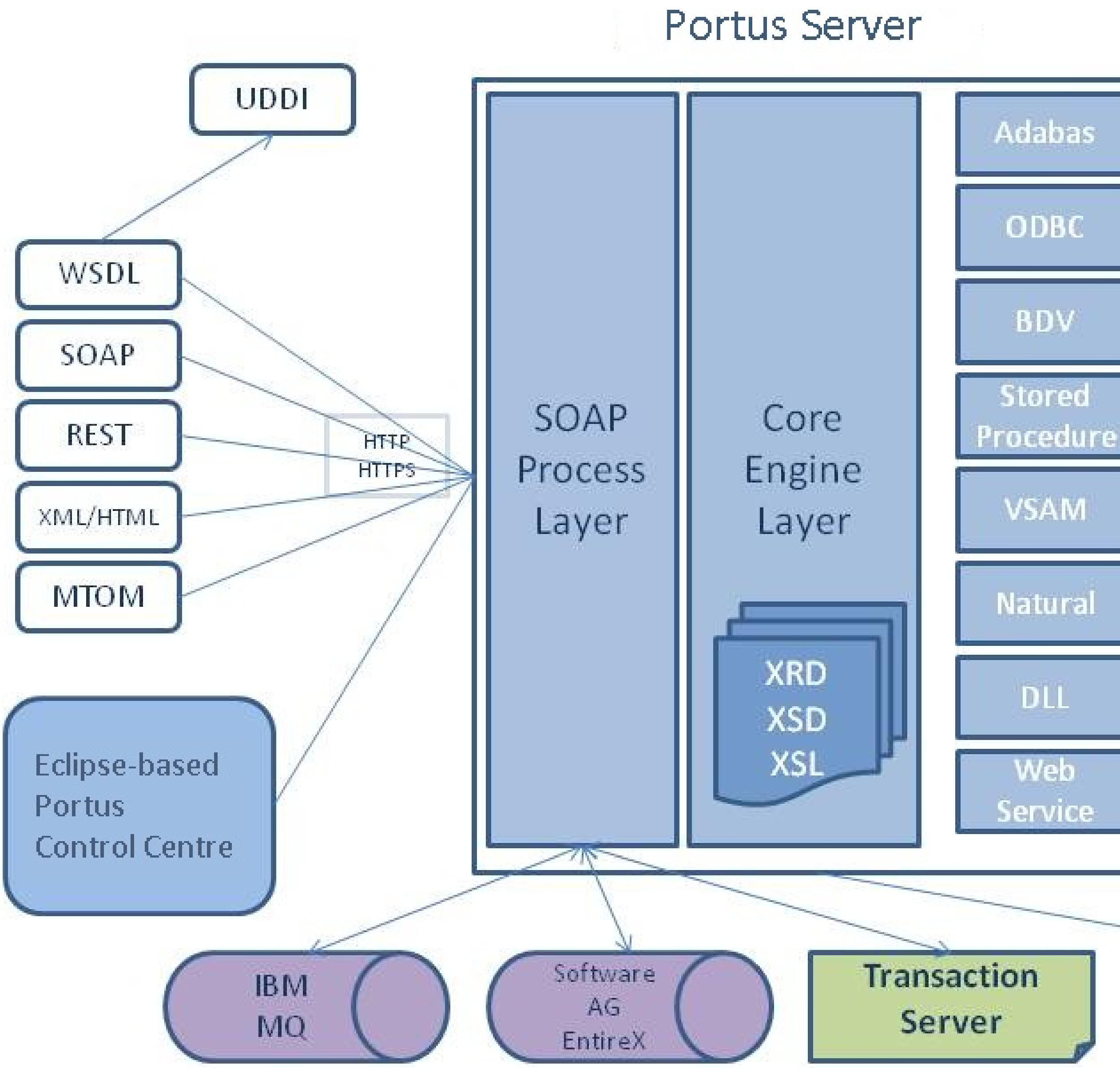
NSGNATI - Portus Natural interface driver

```
RESET #CODE (B4)
INPUT #CODE
CALL 'XMIDNATH' #CODE
END
```

SOA Resource Access

This section describes the operations exposed by Portus to access data sources. These operations are described in a WSDL which is defined for each web service. A client program connecting to a web service can read the WSDL to determine what operations are available on the server. These operations can be invoked using *SOAP* or *RESTful* query.

Portus also supports MTOM for binary data, raw XML, and HTML. WSDLs can be registered in a UDDI server for service look-up.



Supported protocol Versions

- *WSDL 1.1*
- *SOAP 1.1*
- *MTOM 1.0*
- *HTTP 1.1*
- *ODBC Version 3*

Further Reading

- *Prerequisites* on page 230
- *Retrieving the WSDL for your resource* on page 230
- *Portus Web Services Operations for Data Resources* on page 230
- *Specifying Key Data* on page 236

Prerequisites

At this point your Portus Server should be installed, configured and started.

If you have not yet configured any web services, please refer to the *Portus Control Centre* section and add at least one web service

Retrieving the WSDL for your resource

All web services defined will have a WSDL associated with them. This is the starting point for using the operations provided by Portus. The WSDL describes the operations that may be carried out, and how they are used. This includes a description of valid parameters, data and responses for each of the operations.

In order to get the WSDL for a particular resource you simply issue a standard http request, specifying:

- the server name or IP-address of the server where Portus is installed [and running]
- the TCP Port number that Portus is listening on (as provided in the installation)
- the name of the service. (this is the value of the "Name" field in the web service properties)

The following example shows the URL required to retrieve the WSDL for an Adabas "Employees" file.

Server Name	PortusGateway
Port Number	56000
Name	adabas_Employees
URL for WSDL	http://myGateway:56000/adabas_Employees?WSDL

Portus Web Services Operations for Data Resources

The operations provided by Portus for accessing data resources [files, databases, programs, etc.] are now explained.

Parameters are required unless otherwise stated.

Operation	list (SOAP) LIST (REST)
Description	The list operation returns a list of records or rows from your data source. The data returned can be limited or restricted by providing key information. This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL
Parameters	• Key data Key data must be entered for at least one of the fields defined as a key. Refer to the section *Specifying Key Data* for more information. • Options None
Result	SOAP The result will be either : • an XML document wrapped in a SOAP message and containing the requested data

- a SOAP fault message

REST

The result will be either :

- an XML document containing the requested data
- a fault message

Operation

select (SOAP)

SELECT (REST)

Description

The select operation returns a list of records or rows from your data source. The maximum number of rows/records returned can be set via the Portus Control Center.

This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL

The data returned can be limited or restricted by providing key information. The select operation extends the capability of the list operation by enabling searches on a larger set of criteria.

The key information for a select is wrapped in a condition block and can be repeated several times within that block. Each key entry represents an 'AND' condition. Condition blocks can also be repeated several times. Each condition block represents an 'OR' condition. The condition block accepts the following specifiers:

Conditions

- Less than a specific value (LT).
- Less than or equal to a specific value (LE).
- Equal to a specific value (EQ).
- Greater than a specific value (GT).
- Greater than or equal to a specific value (GE).
- Not equal to a specific value (NE).
- Starting with a specific value (START). Character based fields only.
- Ending with a specific value (ENDS). Character based fields only.
- Containing a specific value (CONTAINS). Character based fields only.

Example

SOAP

```
    <soapenv:Body>
       <nos:adabasEmployeeSelectElement>
          <!--1 or more repetitions:-->
          <condition>
             <!--Zero or more repetitions:-->
             <personnel_id Condition="GT">50012100</
personnel_id>
             <personnel_id Condition="LE">50012700</
personnel_id>
          </condition>
          <condition>
             <!--Zero or more repetitions:-->
             <personnel_id Condition="EQ">50012900</
personnel_id>
          </condition>
       </nos:adabasEmployeeSelectElement>
    </soapenv:Body>
```

REST

```
http://localhost:56005/adabas_Employees_9?
SELECT&condition[1].personnel_id>50012100
 &condition[1].personnel_id<=50012700&condition[2].personnel_id=500129
```

The example above specifies 2 condition blocks. This will return data where the (personnel_id > 50012100 and personnel_id <= 50012700) or personnel_id = 50012900

Parameters

- Key data

  Key data must be entered for at least one of the fields defined as a key.

  Refer to the section *Specifying Key Data* for more information.
- Options

  None

Result

SOAP

The result will be either :

- an XML document wrapped in a SOAP message and containing the requested data
- a SOAP fault message

REST

The result will be either :

- an XML document containing the requested data
- a fault message

Operation

selectNext

Description

The selectNext operation returns a list of records or rows from your data source. A selectNext operation may only be called following a select and subsequently other selectNext calls. For this functionality the initial select operation has to initiate a new Conversation. See *Conversational Processing.* The resultant conversation id must be passed in any associated selectNext calls. selectNext calls may be issued until end of file is reached or may be terminated by a selectEnd call. The maximum number of rows/records will be that set for set via the Portus Control Center.

This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL

There is no key information for a selectNext operation as this will have been passed in by the initiating select operation.

Parameters

- Key data

  None
- Options

  None

Result

SOAP

The result will be either :

- an XML document wrapped in a SOAP message and containing the requested data
- a SOAP fault message

REST

Not available

Operation

selectEnd

Description

The selectEnd operation terminates a sequence of select and/or selectNext calls with a conversation. A selectEnd operation may only be called following select or selectNext operations. For this functionality the select operation has to initiate a new Conversation. The resultant conversation id must be passed in the selectEnd call.

This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL

There is no key information for a selectEnd operation as this will have been passed by the initiating select operation.

Parameters

- Key data

  None
- Options

  None

Result

SOAP

The result will be either :

- an XML document wrapped in a SOAP message and containing the requested data
- a SOAP fault message

REST

Not available

Operation

selectCount

Description

The selectCount operation returns a count of the records or rows that match the criteria set in the condition block(s).

This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL

The selectCount operation is identical to that of the select operation in terms of its search capabilities.

The key information for a selectCount is wrapped in a condition block and can be repeated several times within that block. Each key entry represents an 'AND' condition. Condition blocks can also be repeated several times. Each condition block represents an 'OR' condition. The condition block accepts the following specifiers:

Conditions

- Less than a specific value (LT).
- Less than or equal to a specific value (LE).
- Equal to a specific value (EQ).
- Greater than a specific value (GT).
- Greater than or equal to a specific value (GE).
- Not equal to a specific value (NE).
- Starting with a specific value (START). Character based fields only.
- Ending with a specific value (ENDS). Character based fields only.
- Containing a specific value (CONTAINS). Character based fields only.

Parameters

- Key data

  Key data must be entered for at least one of the fields defined as a key.

  Refer to the section *Specifying Key Data* for more information.
- Options

None

Result

SOAP

The result will be either :

- an XML document wrapped in a SOAP message and containing the requested data
- a SOAP fault message

REST

The result will be either :

- an XML document containing the requested data
- a fault message

Operation

get (SOAP)

GET (REST)

Description

The get operation returns a single record or row from your data source.

The data returned is specified by providing unique key information identifying a single record / row.

This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL

Parameters

- Key Data

  Key data must be entered for at least one of the fields defined as a primary key field, so that a single record can be identified.

  Refer to the section *Specifying Key Data* for more information.
- Options

  None

Result

SOAP

The result will be either :

- an XML document wrapped in a SOAP message and containing the requested data
- a SOAP fault message

REST

The result will be either :

- an XML document containing the requested data
- a fault message

Operation

add (SOAP)

ADD (REST)

Description

The add operation adds a single record or row of data to your data source.

This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL

Parameters

- Add Data

  Provide values for each of the fields defined. These are the fields / columns of the data on your data source.

  You must add data for at least one of the fields specified as being primary key fields.

	Refer to the section *Specifying Key Data* for more information. • Options None
Result	SOAP The result will be either : • an XML document wrapped in a SOAP message and containing the requested data • a SOAP fault message REST The result will be either : • an XML document containing the requested data • a fault message

Operation	update (SOAP) UPDATE (REST)
Description	The update operation updates a single record or row of data. This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL
Parameters	• Update Data Provide values (NULL or otherwise) for each of the fields defined. These are the fields / columns of the data on your data source. Key data must be entered for at least one of the fields defined as a primary key field. Refer to the section *Specifying Key Data* for more information. Fields that have been left empty or NULL, e.g. `<data></data>` will be set to this value accordingly. • Options None
Result	SOAP The result will be either : • an XML document wrapped in a SOAP message and containing the requested data • a SOAP fault message REST The result will be either : • an XML document containing the requested data • a fault message

Operation	delete (SOAP) DELETE (REST)
Description	The delete operation deletes a single record or row of data from the data source.

	This operation will only be available when the web service is a "database" type, for example, Adabas or MySQL
Parameters	• Key Data Key data must be entered for at least one of the fields defined as a primary key field, so that a single record can be identified. Refer to the section *Specifying Key Data* for more information. • Options None
Result	SOAP The result will be either : • an XML document wrapped in a SOAP message and containing the requested data • a SOAP fault message REST The result will be either : • an XML document containing the requested data • a fault message

Operation	invoke (SOAP) INVOKE (REST)
Description	The invoke operation makes a call to a function or program. This operation will only be available when the web service is a "program" type, for example, NATURAL, COBOL or when the web service describes a database stored procedure.
Parameters	• Parameters Each of the fields defined can be input, output, or input+output. Provide values for each of the fields defined as input or input+output. These match the function/program parameters that are required on the call.
Result	SOAP The result will be either : • an XML document wrapped in a SOAP message and containing the requested data • a SOAP fault message REST The result will be either : • an XML document containing the requested data • a fault message

Specifying Key Data

Keys or Key Fields in Portus terms are simply fields or columns on your data source (file, table, etc) that may be used to narrow a search for a record or row of data.

Note: key fields may or may not be indexed at your data source. Indexing provides better response times from your data source. If you are unsure what, if any, fields are key fields (in Adabas terms: Descriptors) or allowed to be used for searching, contact your data source administrator (DBA, etc.).

Key fields are specified to Portus by setting the appropriate value for the attribute "key type" in the DataView.

Key Types

There are two types of key fields for data sources:

1. Primary Key Fields
   - Primary Key Fields are fields in a data source that must always contain unique values.
   - They are required on all data source operations except the list operation, where they are optional. Secondary keys may be used instead or in addition on a list request.
   - Their values will not be altered in the update operation.
   - Primary Keys cannot contain wild card symbols, except on the list operation.
2. Secondary Key Fields
   - Secondary Key Fields are fields that may be used in narrowing a search.
   - They are not required in any operation, and their values can be updated.
   - They may contain wild card symbols.

Using Wild card Symbols and other Generic Search criteria

Wild cards are used where you do not wish to specify an exact value in a key field, but use a generic specification that will match for a range of different values.

These search modifiers may only be used on the list operation.

Portus currently supports the following generic search criteria :

1. Wild card for one or more characters
   - The character "*" may be used as a wild card for all characters.
   - It may only be used on fields defined as "string" fields.
   - Where it appears, it will match any character or group of characters.

   Note: for Adabas resources, this wildcard may only appear at the end of the string data supplied.

   Example: "Ga*" would match "Gat", "Gate", "Gateway", etc.

Creating a Stylesheet for your Portus Data

XSL stands for EXtensible Stylesheet Language, and is a style sheet language for XML documents. XSLT stands for XSL Transformations.

XSLT can be applied to the response payload of Portus REST requests.

It is commonly used, but not restricted, to creating HTML pages based on the REST response.

When configured to use XSLT, a Portus service will (by default) return an XML stylesheet processing instruction embedded in the XML.

Therefore, the client consuming this data should be able to load the XSL link, and apply the XSLT to the payload.

In some cases, the client may not have the ability to understand XML processing instructions (browsers on mobile devices are a good example). In this case, Portus has the ability to apply the XSLT on the server side, and send the transformed results. In this case, providing the __xslTransform=server option on the REST request will tell Portus to apply the XSLT before sending the payload, and no XML stylesheet instruction will be included, e.g.

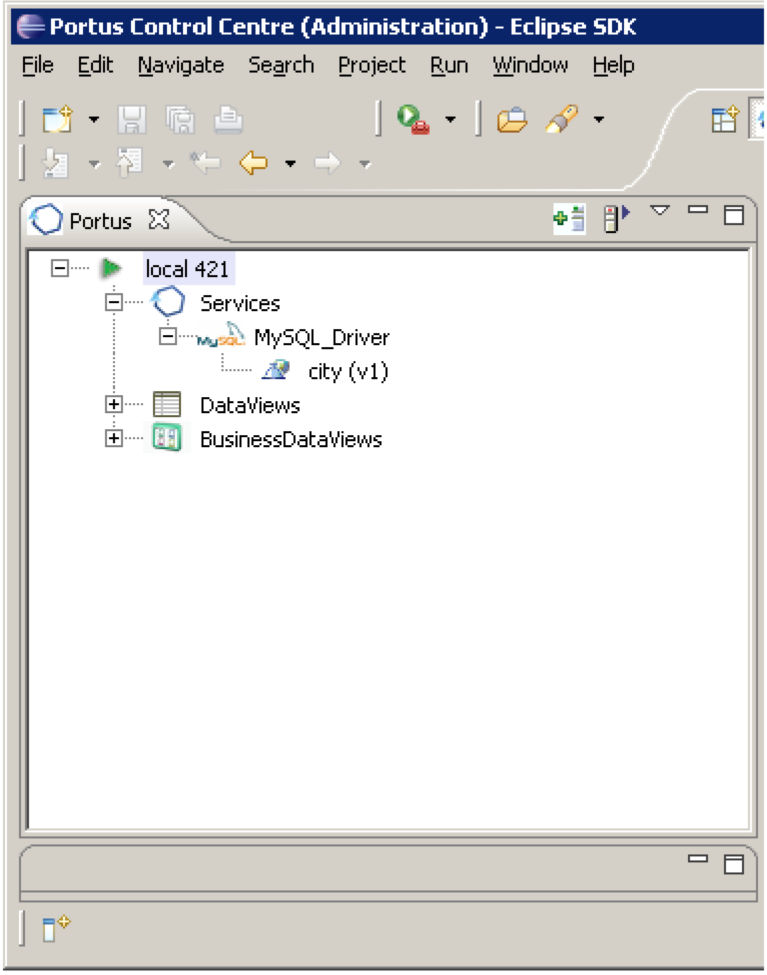
```
http://host/Service?LIST&ID=*&__xslTransform=server
```

Create HTML page from City XML

This tutorial will outline how to apply XSLT to a MySQL service. Without any transformation, the service will return raw XML, and we will apply an XSLT to display this in a HTML table.

- For the purpose of this tutorial we will be using the MySQL World database table, city. See here for the steps required to set this up. Please ensure that you select the city table when you get to the Discovery step.

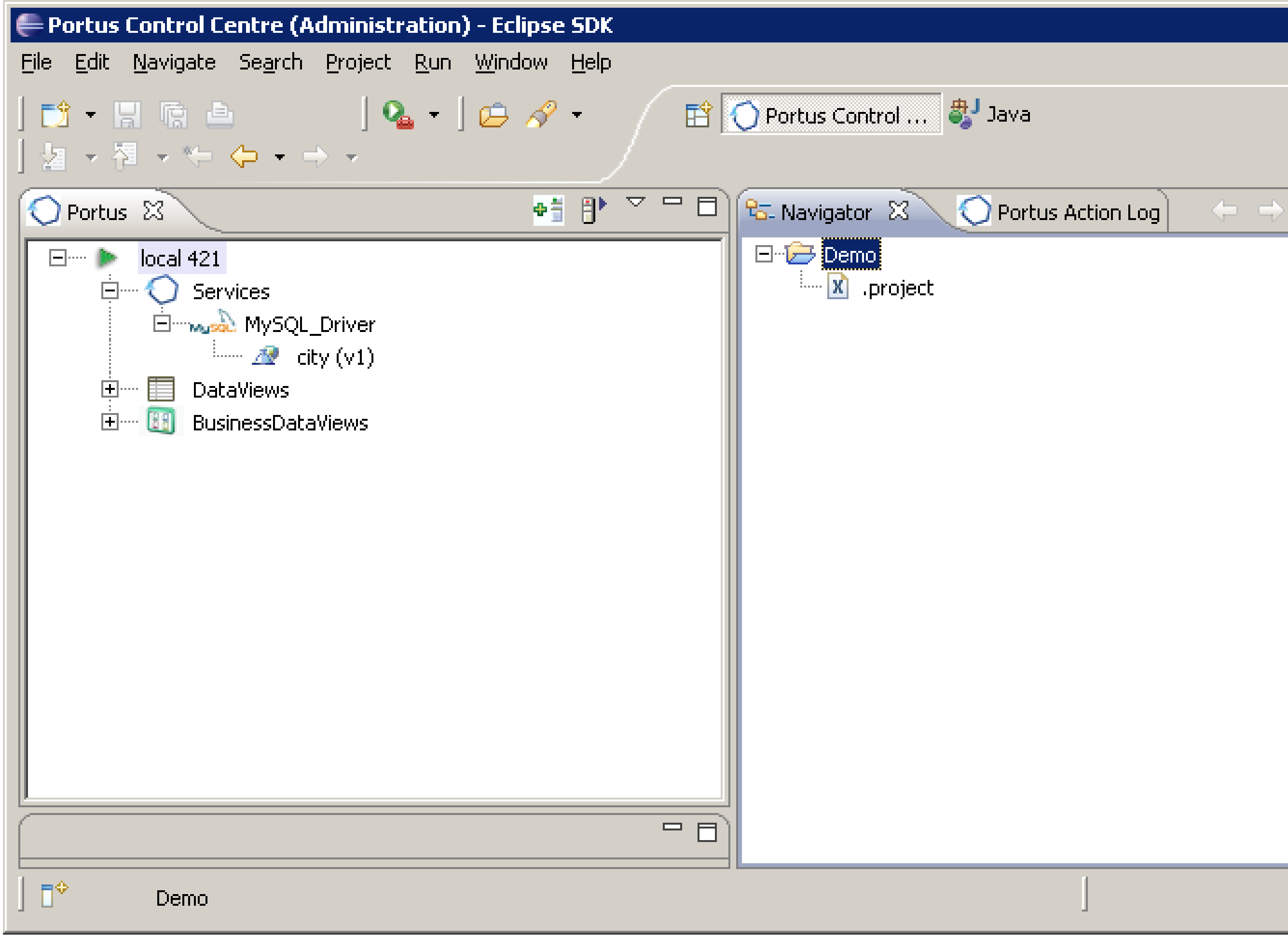
  When this step is completed you should have a Service similar to that shown below e.g. a Service capable of accessing the city table.

  
  Portus Control Centre (Administration) - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Portus
local 421
Services
MySQL_Driver
city (v1)
DataViews
BusinessDataViews

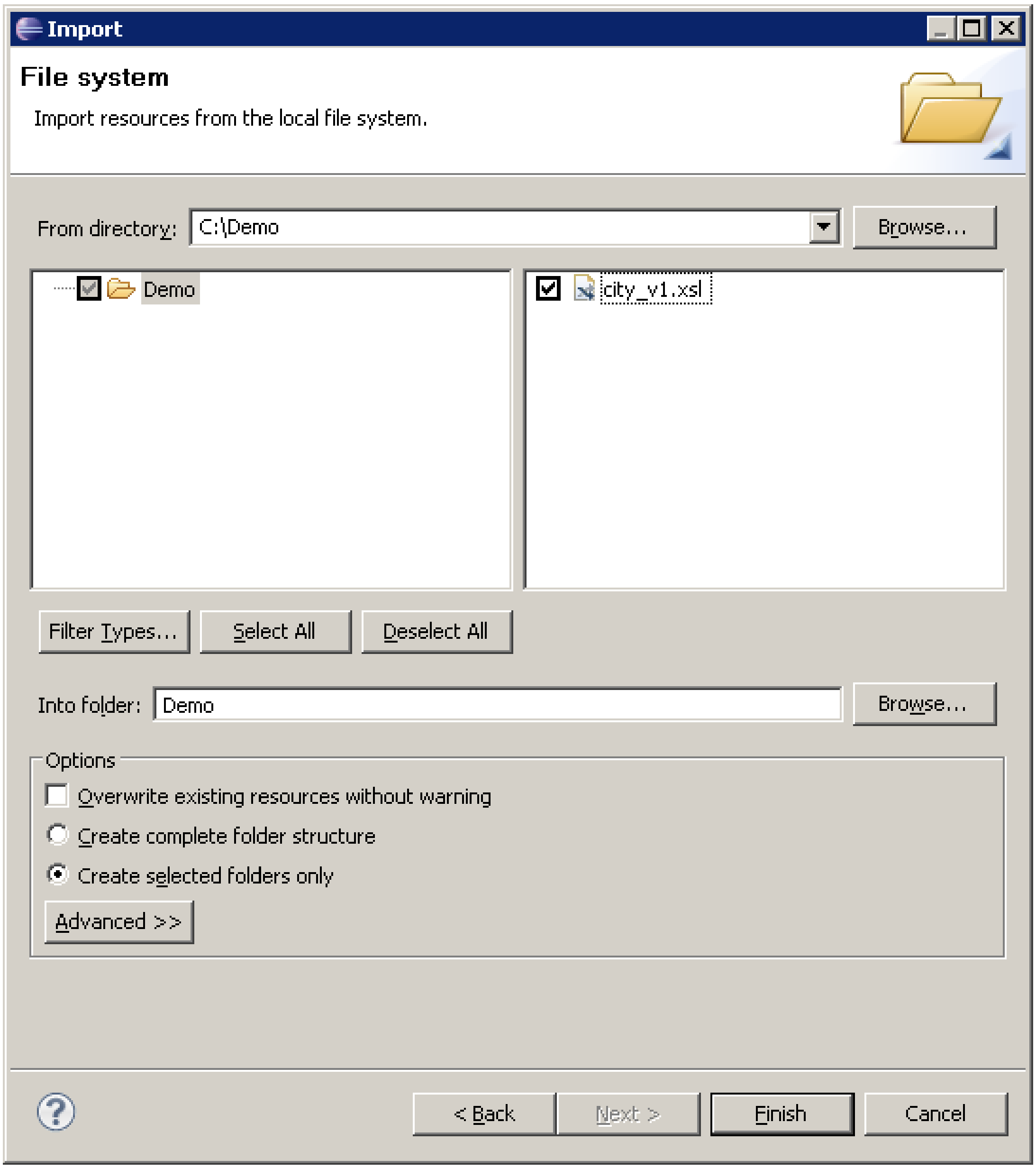
- A generic stylesheet is provided which can be modified to support your Service. Save *the following XSL file* to disk.

  Important: The name of the saved stylesheet has to be in the format 'ServiceName_version.xsl'. e.g. we will save our file as city_v1.xsl

- If you do not already have a project created *see here* on how to do so. In our case we have created a project named Demo. If this is not visible open Window -> Show View -> Navigator You should have a view similar to this:



- Right-click your project in the Navigator View and select **Import**
- Expand **General** and select **File System**. Click **Next**
- Click **Browse** and select the directory where you saved the above XSL. Check the XSL, and click **Finish** e.g.



Import

File system

Import resources from the local file system.

From directory: C:\Demo

☑ Demo

☑ city_v1.xsl

Into folder: Demo

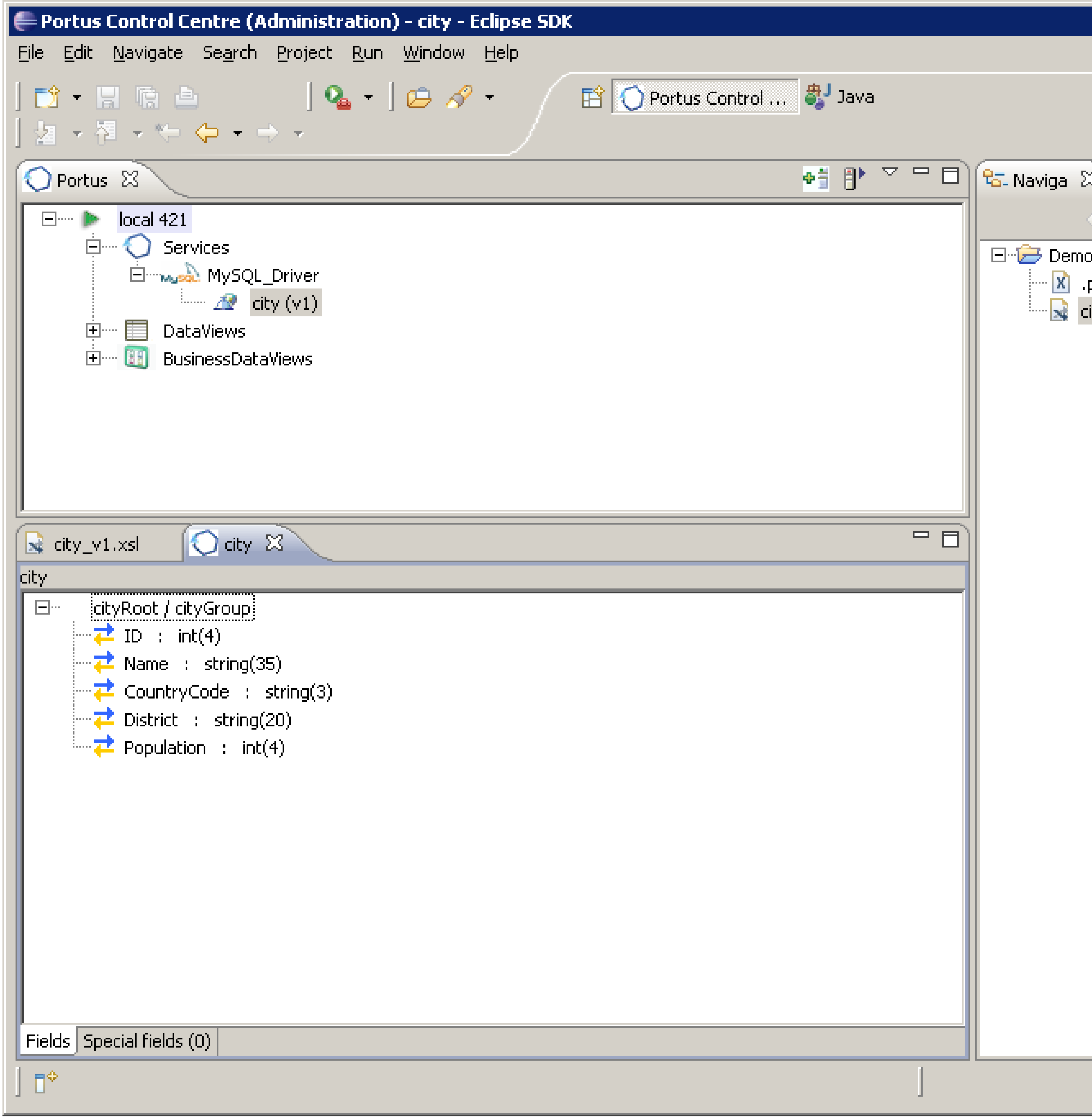
Options

□ Overwrite existing resources without warning

○ Create complete folder structure

● Create selected folders only

- Double-click your XSL file to open it for editing in a default editor or right-click on the file and select Open With... to choose your own editor.
- Right-click on your Service in the Servers View and select Edit DataView.
- You should now have a view similar to this:



- The DataView provides us with 3 important items which are required when we come to edit the stylesheet:

1 The ROOT name: e.g. cityRoot.
2 The GROUP name e.g. cityGroup.
3 The element names e.g. ID, Name, CountryCode, District and Population.

- Select the XSL tab to edit its contents.

1 Find the entry <xsl:template match="changeThisRoot"> and modify changeThisRoot to your root name.
2 Find the entry <xsl:template match="changeThisGroup"> and modify changeThisGroup to your group name.
3 Find the entry ColumnHeader1. It should be wrapped in TR tags as follows:

```
<TR>
<Td><font color="#3333FF">ColumnHeader1</font></Td>
</TR>
```

Create as many Td tags entries within the TR tag as there are elements in your DataView e.g. in this case 5.

Change each ColumnHeader1 value to the element names. For headers these do not have to match the element names but for simplicity we will do so here e.g.

```
<TR>
<Td><font color="#3333FF">ID</font></Td>
<Td><font color="#3333FF">Name</font></Td>
<Td><font color="#3333FF">CountryCode</font></Td>
<Td><font color="#3333FF">District</font></Td>
<Td><font color="#3333FF">Population</font></Td>
</TR>
```

4. Find the entry XRDElementName1. It should be wrapped in tr tags as follows:

```
<tr>
<td><xsl:value-of select="XRDElementName1" /></td>
</tr>
```

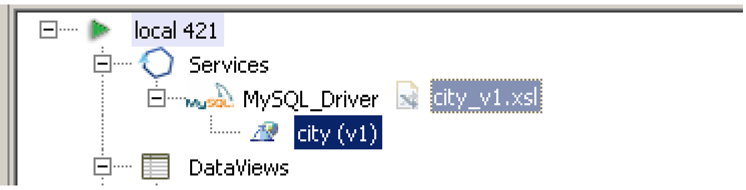
Create as many td tags entries within the tr tag as there are elements in your DataView e.g. in this case 5.

Change each XRDElementName1 value to the element names.

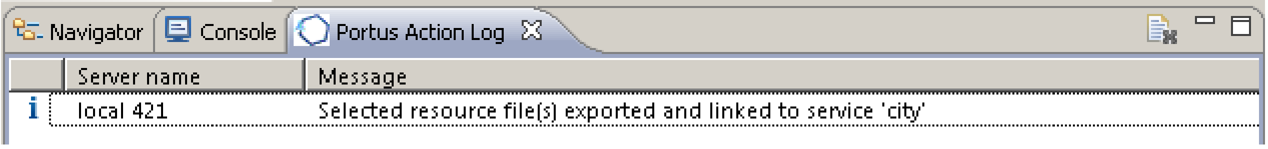
Important: These must match the element names exactly e.g.

```
<tr>
<td><xsl:value-of select="ID" /></td>
<td><xsl:value-of select="Name" /></td>
<td><xsl:value-of select="CountryCode" /></td>
<td><xsl:value-of select="District" /></td>
<td><xsl:value-of select="Population" /></td>
</tr>
```

5. Save the XSL (Ctrl+S).
6. In the Navigator View left-click the XSL file and, holding down the left button, drag and drop the file onto the Service as shown:.



7. You should see a message similar to this in the Portus Action Log:



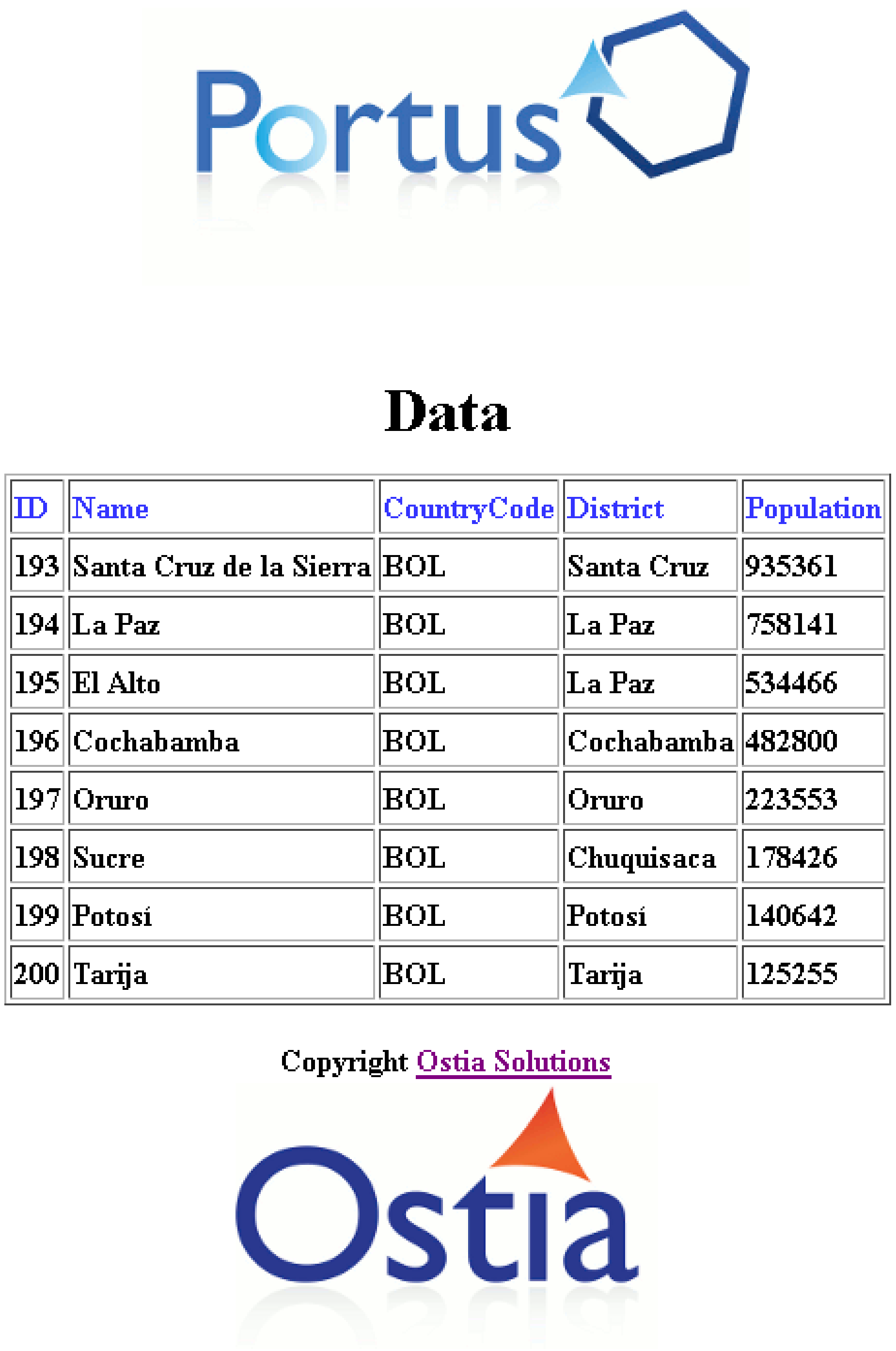
	Server name	Message
i	local 421	Selected resource file(s) exported and linked to service 'city'

8. Select the Service in the Servers View. If the Properties View is not open select Window -> Show View -> Properties.

Select the 'WSDL URL is ...' entry in the Properties View which should open a browser window.

In the browser window change ?WSDL to ?LIST&CountryCode=BO* e.g. http://localhost:56005/city?LIST&CountryCode=BO* and hit enter.

The results should be displayed as follows:



Data

ID	Name	CountryCode	District	Population
193	Santa Cruz de la Sierra	BOL	Santa Cruz	935361
194	La Paz	BOL	La Paz	758141
195	El Alto	BOL	La Paz	534466
196	Cochabamba	BOL	Cochabamba	482800
197	Oruro	BOL	Oruro	223553
198	Sucre	BOL	Chuquisaca	178426
199	Potosí	BOL	Potosí	140642
200	Tarija	BOL	Tarija	125255

Copyright Ostia Solutions

9 Congratulations! You have now created a stylesheet for your Service.

Data Masking

Data masking is the process of obscuring (masking) specific data elements within data stores. It ensures that sensitive data is replaced with realistic but not real data. The goal is that sensitive customer information is not available outside of the authorized environment.

Data masking is typically done while provisioning non-production environments so that copies created to support test and development processes are not exposing sensitive information and thus avoiding risks of leaking.

Portus provides functionality which allows you to select fields whose content you do not wish to divulge.

This masking takes place as the data flows through Portus, therefore is masked "in-flight". The data sitting on the back-end database is not changed in any way.

Almost 40 different algorithms are provided which you can use to maintain data integrity. The algorithms include generate a random email address, create a hash value, generate random credit card numbers, random dates and times, and look up a list of seed values in an external database table.

This allows users to query your data secure in the knowledge that its real content remains unexposed.

Example

Employee Data *without* masking

```
<EMPLOYEE_ID>100</EMPLOYEE_ID>
<FIRST_NAME>Steven</FIRST_NAME>
<LAST_NAME>King</LAST_NAME>
<EMAIL>SKING@SKING.COM</EMAIL>
<HIRE_DATE>2003-06-17</HIRE_DATE>
<SALARY>24000</SALARY>
```

Employee Data *with* masking

```
<EMPLOYEE_ID>100</EMPLOYEE_ID>
<FIRST_NAME>UKfaUAKuPFFAuPFap</FIRST_NAME>
<LAST_NAME>PPupUpaAfUfkpuFkpFkkFfFa</LAST_NAME>
<EMAIL>VIJAY@SMITHS.COM</EMAIL>
<HIRE_DATE>2002-08-16</HIRE_DATE>
<SALARY>823999</SALARY>
```

Algorithms used are random text for FIRST_NAME and LAST_NAME, random email for EMAIL, random date for HIRE_DATE and random number between 0 and 999999 for SALARY.

To find out more about this feature please contact your sales area representative.

Internationalization

Portus uses IBM's *International Components for Unicode* to support internationization (i18n). This supports text data conversion between almost any codepage.

- *Setting codepages* on page 244
- *Which codepage do I use ?* on page 244
- *SOAP versus REST differences* on page 245
- *Troubleshooting* on page 245

Setting codepages

The Single Byte Character Set (SBCS) or Multi Byte Character Set (MBCS) codepage can be set on the driver, for more information, see *here*.

The codepages can also be set on the on the web service itself, by simply left-clicking the web service and entering the codepage in the appropriate section of the web service Properties.

Note: The codepage set on the web service overrides the one set on the driver.

Which codepage do I use ?

This depends on what sort of information your service is going to return.

Generally the ASCII codepage is sufficient for the English language. The ISO-8859-1 (often called latin1) codepage should suffice for most languages of Western Europe. The windows-1251 codepage supports Cyrillic languages such as Russian and Bulgarian. The ISO-8859-8 codepage can be used for Hebrew script.

The ICU home page has provided a *useful web page* which displays the ICU internal name, and a list of the aliases that Portus will recognise. This page will also display the codepage map, which will allow you to choose the codepage best suited to your service.

SOAP versus REST differences

Generally when using WSDL and SOAP, once the correct codepage has been set, the payload should be recognised or returned correctly.

When using REST requests, things are slightly different. Non-ASCII characters entered on a URL bar of a browser will be escaped into their native hex value, of the form %XX. This native hex value differs depending on what codepage the browser recognises the character as. For example, a browser running in the latin1 codepage will recognise Á as %C1, but a browser running in the Cyrillic codepage will recognise # as %C1.

For this reason Portus allows users to provide an extra field on the REST request. This field is called __encoding. Thus users can indicate what codepage their browser is running in.

Note: By default, Portus assumes the escaped values are in the ISO-8859-1. The __encoding field is not required in this case.

Example 1

The browser escapes the Russian # into %C1. You need to tell Portus that this is the Cyrillic encoding.

The URL should be

```
http://host:port/Service?LIST&key=%C1&__encoding=windows-1251
```

Example 2

The browser escapes the Hebrew Shin (#) into %F9. You need to tell Portus that this is the Hebrew encoding

The URL should be

```
http://host:port/Service?LIST&key=%F9&__encoding=iso-8859-8
```

Troubleshooting

When Portus cannot display a character in the requested codepage, it writes a message to the error log, and continues to attempt to process the rest of the payload.

If you find your responses are missing some characters, check the error_log / error.log / XMIDCARD on *nix, Windows and z/OS respectively.

The error message to check should be something like this :

```
Unicode char 0xF1 is not representable in encoding ASCII.
```

Performance Hints

This section outlines some suggestions to improve the performance of Portus.

- **Remove XSD**

  This only applies for web services which use the invoke operation, e.g services built from the Natural, Cobol or DLL drivers.

  Portus will validate the incoming XML against an XSD. This ensures that the contents and structure of the payload are correct, and will catch potential errors early on in processing. But XML validation is a relatively expensive operation, so it is possible to turn this off if required.

  You may want to back up your existing XSD files before deletion. Use the "Import Service Definition" and ensure the XSD box is checked. See *here* for more information.

  To delete the XSD, use the "Delete Service Definition", and ensure the XSD box is checked. The XSD has now been deleted from the server and validation of the payload will not take place.

- **Turning off Access Logging**

  Each time Portus handles a request, it writes some logging information to the access log via Apache. By default, this file is access_log / access.log / DD:ACCESS based on the platform, *nix, Windows, z/OS respectively.

  To restrict this logging, see the following Apache directive *here*

  To remove this logging, remove the CustomLog directive from your httpd.conf / HTCONF. This can be accomplished by adding a # in front of the directive.

- **Use PFS caching**

  This only applies on z/OS or z/VSE.

  Edit your SYSPARM and ensure caching of the Portus filesystem has been turned on. The option is CACHESIZE=N option on the CDI_DRIVER directive, eg.

  ```
  CDI_DRIVER=('pfs,PAANPFS,CONTAINER=CIO://
  DD:PFS,CHARSET=ASCII,LRECL=4096,CACHESIZE=4096')
  ```

- **Enable/disable Streaming**

  By default, when a user issues a list request, with key data of "*", i.e. listing all records in the database, Portus will send back records in a "streamed" fashion.

  For example, as soon as one record is retrieved, it is immediately sent back to the client using the HTTP chunking protocol.

  It has been found that this is the most effective way of handling large amounts of data, but there is a small performance offset in doing this. There are a number of directives that affect how streaming is applied. These directives must be part of the Apache configuration file.

  - **SoaGatewayStreaming On** : This is the default setting. Responses will be streamed back to the client using the HTTP Chunking protocol when a list is requested that will retrieve every record in the database.
  - **SoaGatewayStreaming Off** : No streaming will ever take place. Use this option if you are concerned about performance, and will be listing every record in the database.
  - **SoaGatewayStreaming Force** : Portus will always to attempt to stream data back to the client. This is most effective if the SOA Gateway is running on a machine with low resources, and low memory usage is a priority.

- **Change MPM settings**

Portus uses the *Apache worker MPM* to handle requests. This can be modified to increase server threads, therefore allowing the server to serve more requests. See the Apache documentation for more information.

Note: Ensure that the ServerLimit of 1 is maintained at all times. Portus will not function correctly if more than server process is started.

Frequently Asked Questions

- *How do I active the Software AG sagenv file post Portus Installation ?* on page 247
- *How do I modify the machine identifier in the JESMSGLG?* on page 247

How do I active the Software AG sagenv file post Portus Installation ?

This issue only applies to Portus running on Linux, AIX or Solaris. If your license does not have any Adabas or Natural drivers enabled, this issue does not apply.

If the sagenv file is not available during the Portus install, Adabas or Natural drivers may not work correctly until this file is made available. For example, you have installed Portus before installing Adabas.

Follow these steps to enable a new sagenv file in an existing Portus installation.

1. Edit [INSTALL_HOME]/xmiddleEnv.sh
2. Append the following to the end of this file (assuming sagenv file is /opt/softwareAg/sagenv.new) :

```
if test -e "/opt/softwareAg/sagenv.new"; then
source /opt/softwareAg/sagenv.new &> /dev/null
fi
```

3. Save and close this file.
4. Stop and then Start the Portus server. See *here* for more information.
5. Modify [INSTALL_HOME]/apache2/conf/adabas_soa_gw.conf and change the User directive to use the "sag" user.
6. Change the file permissions of Portus files and directories

```
chown -R sag [INSTALL_HOME]
```

7. Review the error_log for errors/warnings.
8. Adabas or Natural drivers should now be added successfully. See *here* for more info.

How do I modify the machine identifier in the JESMSGLG?

During the FTP of Portus to z/OS, the machine identifier will be set to the hostnam or IP address of the FTP server.

This text will be displayed on messages appearing in the JESMSGLG.

To change this, modify the SYSPARM member of the CONF dataset and set this as required. It is recommended that this is set to the hostname or IP address of the z/OS machine.

Introduction to Eclipse

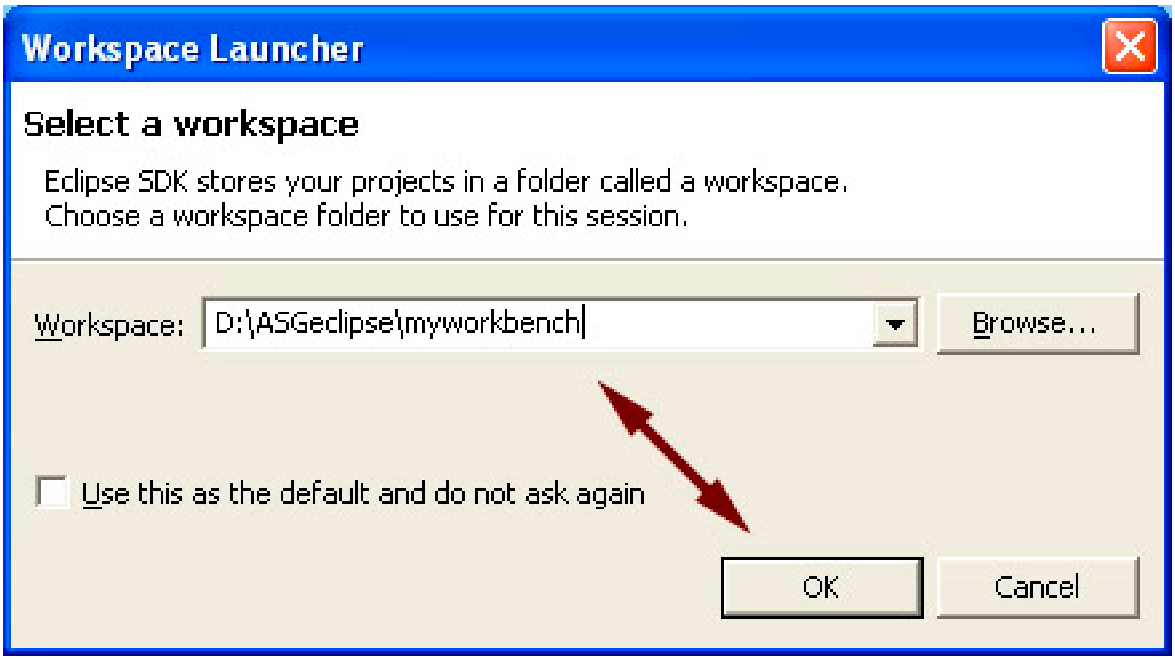
If you are new to Eclipse, please take the time to go through this introduction, and familiarize yourself with the basics of the Eclipse framework.

If you have worked with Eclipse before, you may skip this section.

Running Eclipse is as easy as launching the Eclipse executable - "eclipse" on Linux / Unix platforms, "eclipse.exe" on Windows - in the top-level Eclipse directory.

In case you are new to Eclipse and run into problems, try *the Eclipse FAQs*.

When the Eclipse Workbench is launched, the first thing you see is a dialog that allows you to select where the workspace should be located. The workspace is the directory where your work will be stored. When you start Eclipse for the very first time after installation, Eclipse will propose a workspace named "workspace" located in your "home" directory (depending on the operating system you are running Eclipse on). Either just click OK to pick the default location, or specify a path + filename of your choice, then click OK. (You can also check the checkbox to prevent this question from being asked again.)



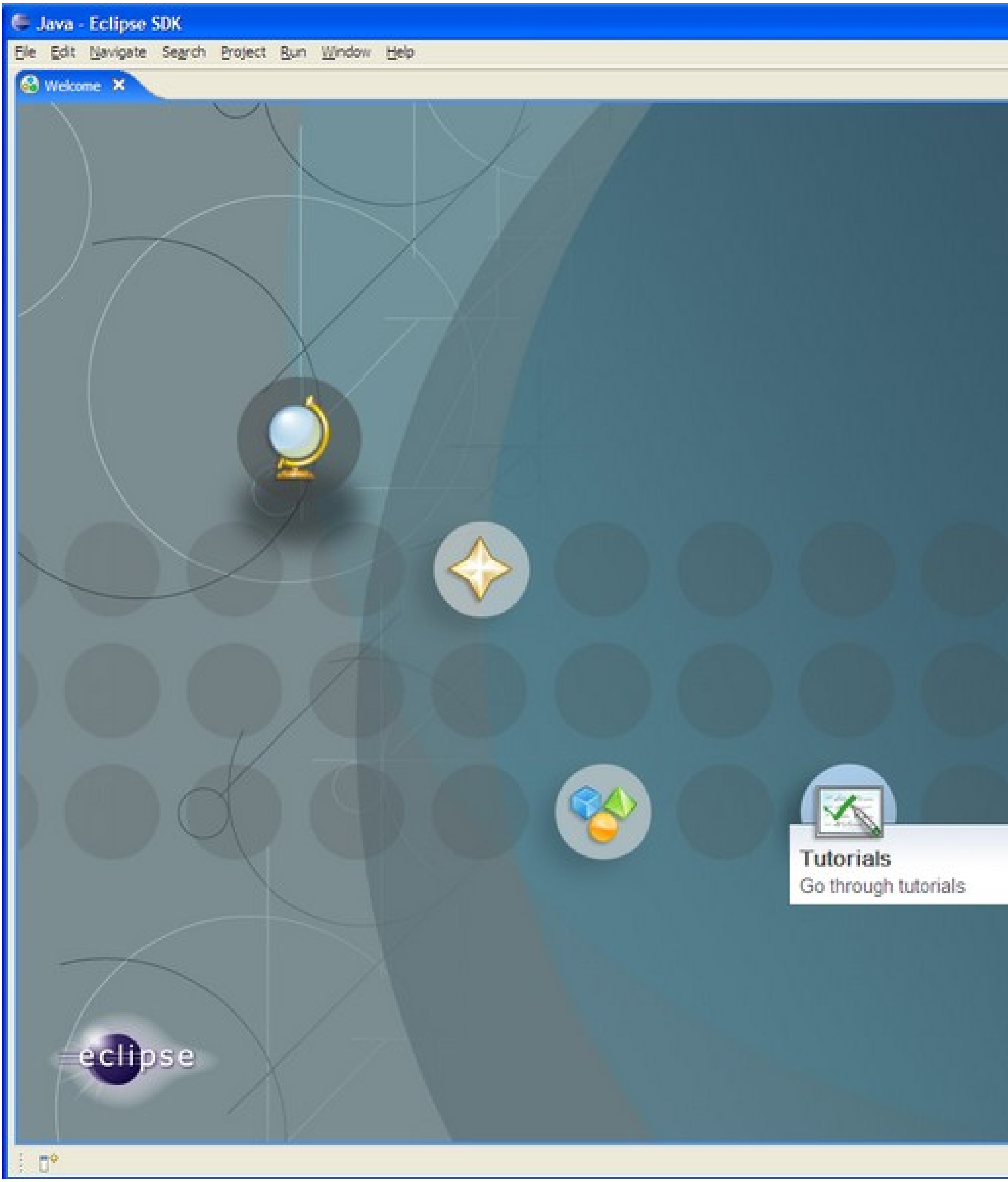
Select a workspace

Eclipse SDK stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: D:\ASGeclipse\myworkbench

□ Use this as the default and do not ask again

After the workspace location is chosen, the Eclipse Welcome Screen will be shown. You may want to start by going through some of the introductory tutorials offered by Eclipse, or just commence to the Workbench by clicking the bent arrow in the top right corner.

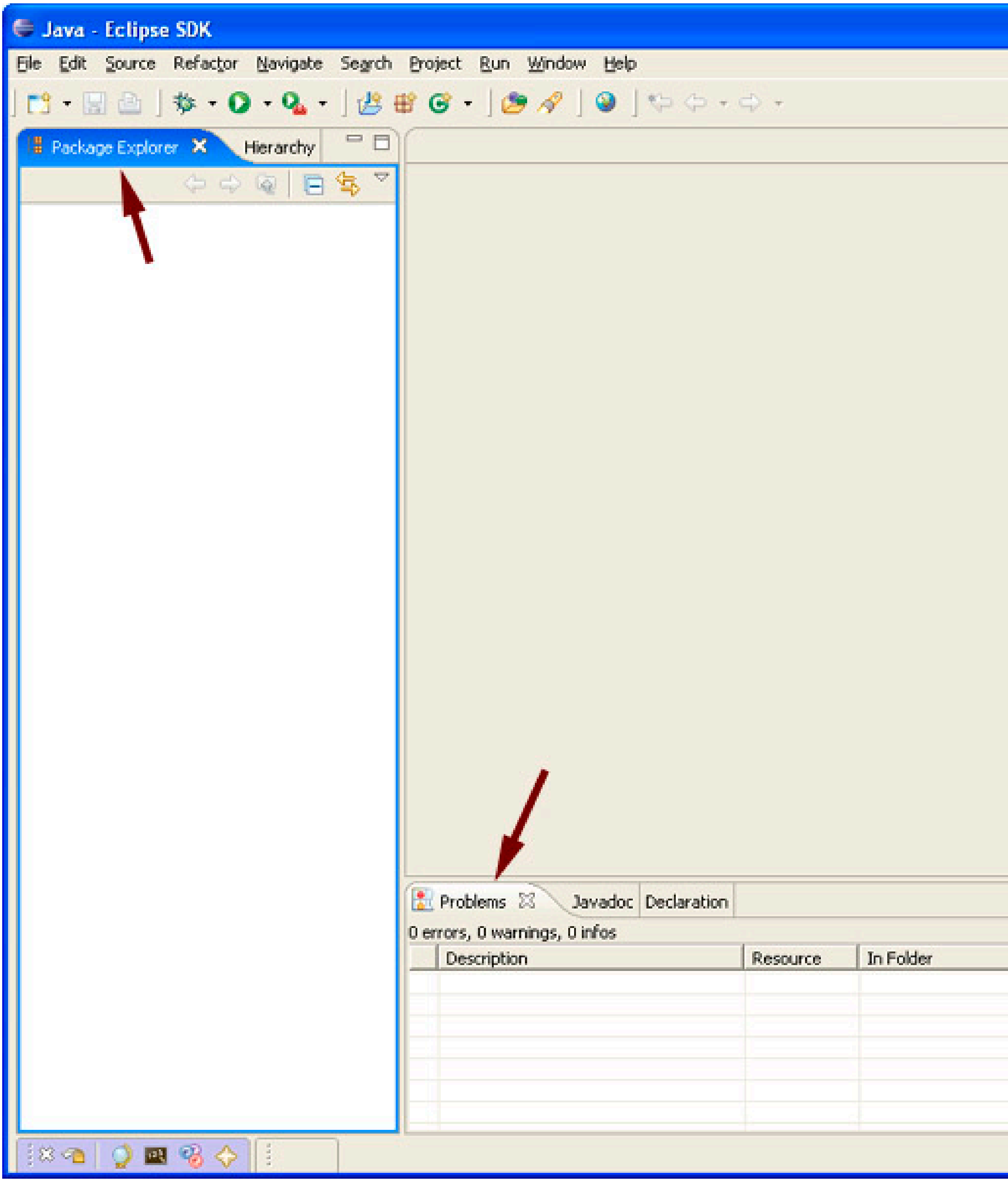


A single Workbench window is displayed. A Workbench window offers one or more perspectives. A perspective contains editors and views, such as the Package Explorer. Multiple Workbench windows can be opened simultaneously. Initially, in the first Workbench window that is opened, the Java perspective is displayed.

(You can get the Welcome view back at any time by selecting **Help** > **Welcome**.)

A shortcut bar appears in the top right corner of the window. This allows the user to open new perspectives and switch between ones already open. The name of the active perspective is shown in the title of the window and its item in the shortcut bar is highlighted.

The title bar of the Workbench window indicates which perspective is active. In this example, the Java perspective is in use. The Package Explorer, Problems, and Outline views are open along with an editor. The "editor" view is the one in the center of the screen and is currently "empty" because no resources have been opened for editing so far.



Elements of the Workbench

A Workbench consists of:

- perspectives
- views

- editors

Perspective

a group of views and editors in the Workbench window.

One or more perspectives can exist in a single Workbench window, each perspective contains one or more views and editors. Within a window, each perspective may have a different set of views but all perspectives share the same set of editors.

View

a visual component within the Workbench.

Typically used to navigate a hierarchy of information (e.g. the resources in the Workbench), open editors, or display properties for the active editor. Modifications made in a view are saved immediately. Only one instance of a particular type of view may exist within a Workbench window.

Editor

also a visual Workbench-component.

Used to edit or browse resources. Modifications made in an editor are only saved when explicitely requested. Multiple instances of an editor type may exist within a Workbench window concurrently.

Parts

as views and editors have a number of features in common, the term part is used to mean either a view or an editor.

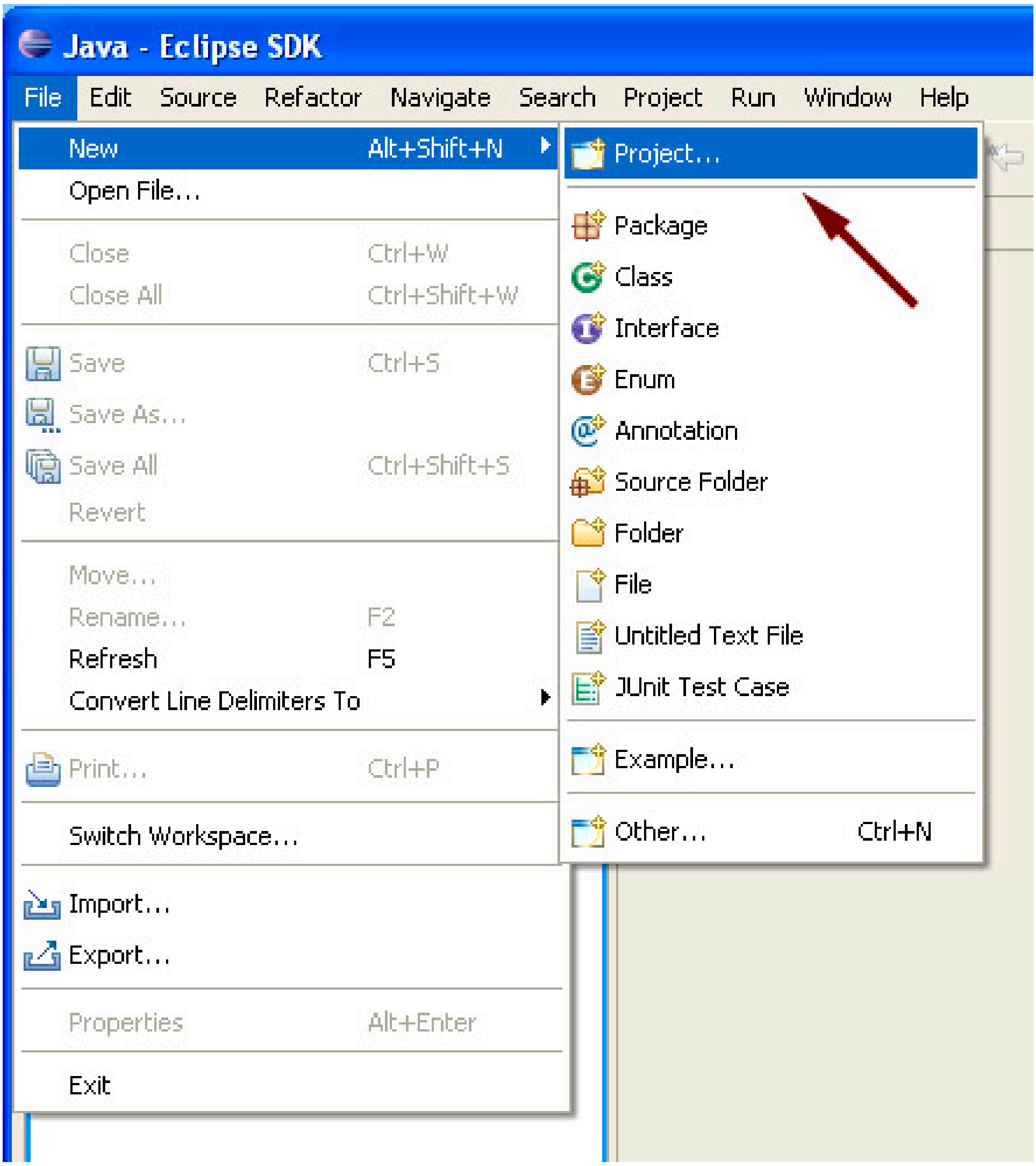
Parts can be active or inactive, but only one can be active at any one time. The active part is indicated by a highlighted title bar and is the target for common operations like cut & paste.

Creating Resources

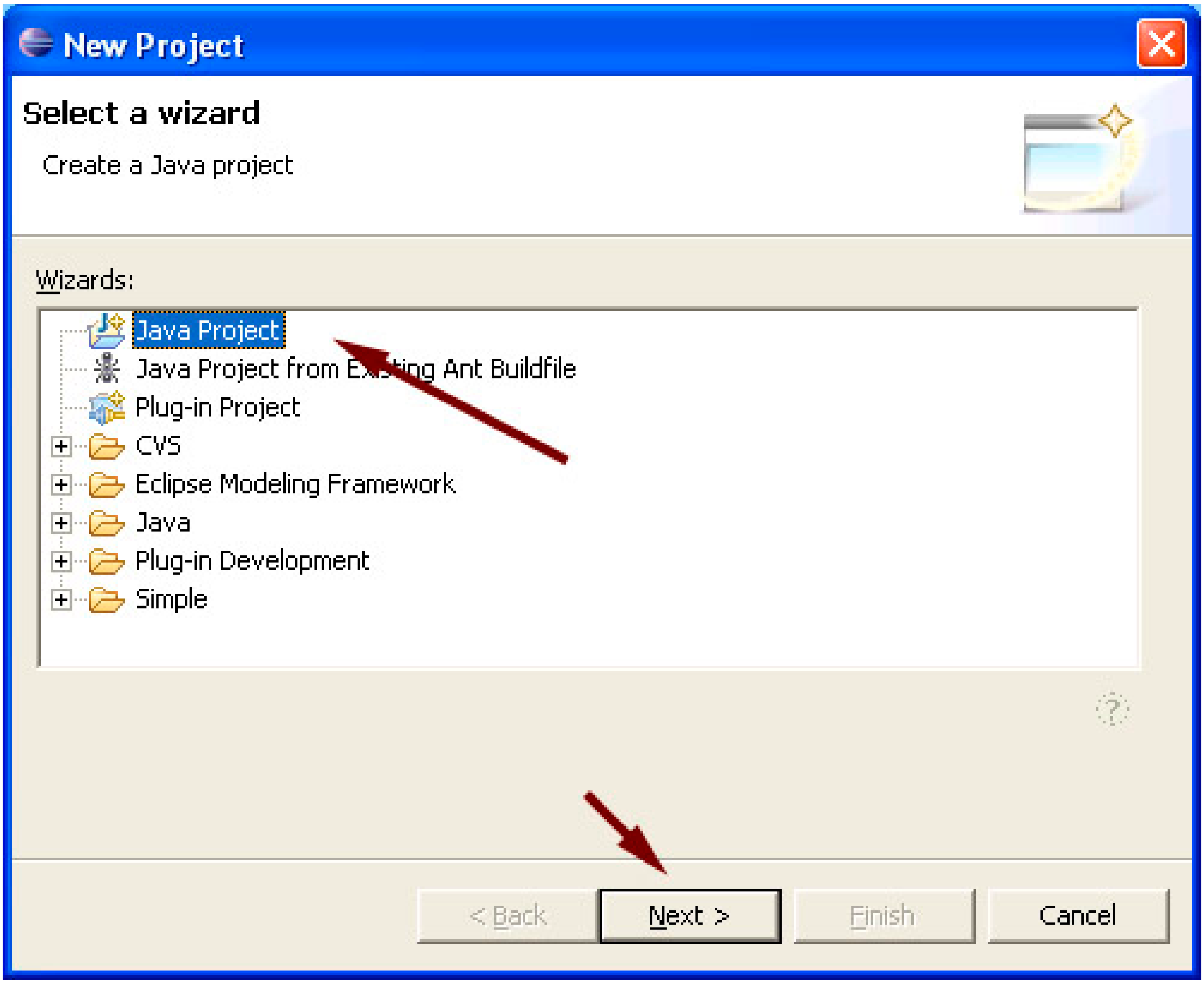
New resources (e.g. projects, folders, files) can be created using several different approaches:

1. **File** menu
2. Navigation view context menu
3. **New Wizard** button

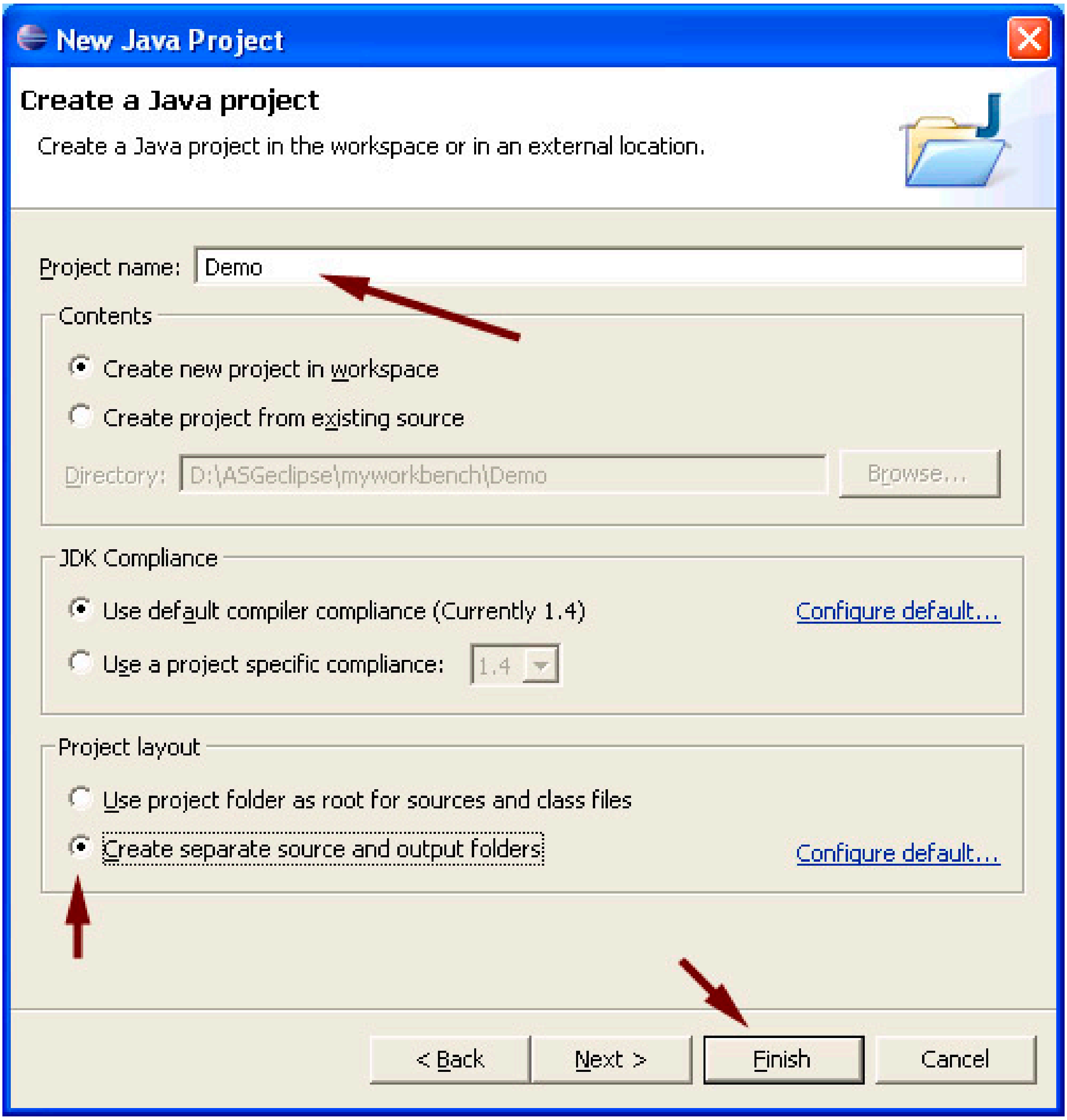
First of all, a *project* must be created using the **File** menu.



The "New Project Wizard" appears, select **Java Project** and click **Next**.



Enter a **Project name,** make sure to check the **Create separate source and output folders** selection, click **Finish** to create the project.



New Java Project

Create a Java project

Create a Java project in the workspace or in an external location.

Project name: Demo

Contents

● Create new project in workspace

○ Create project from existing source

Directory: D:\ASGeclipse\myworkbench\Demo

JDK Compliance

● Use default compiler compliance (Currently 1.4)

○ Use a project specific compliance: 1.4

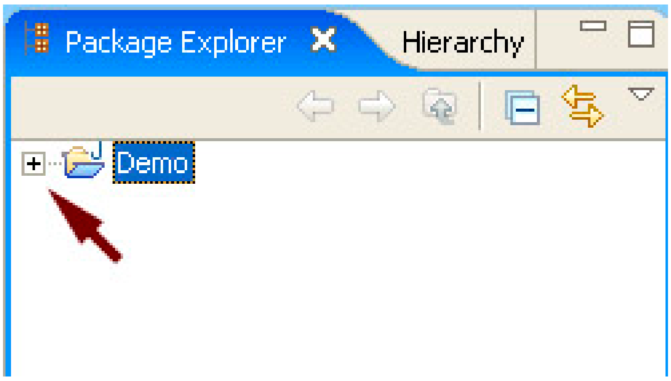
Project layout

○ Use project folder as root for sources and class files

● Create separate source and output folders

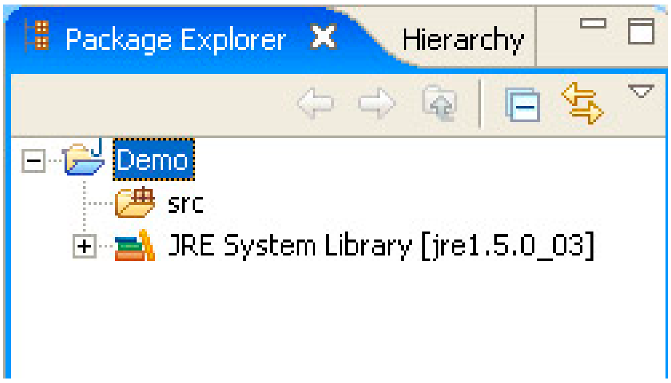
The **Package Explorer** now shows the project folder you just created, the "+" icon to the left of the project name indicates the folder is not empty, so click the "+" sign to see what's there.

Note: he symbol used to indicate a resource has contents and can be "unfold" is platform specific, this will be a "+" sign on Windows, an arrow on Linux.



The New (Java) Project Wizard did the following:

- create an empty src folder
- add the Java Runtime Environment (JRE) to the project's class-/buildpath



Chapter 6

Troubleshooting

Topics:

- *Portus Client side issues*
- *Portus Server side issues*
- *Portus Control Centre (Eclipse) issues*
- *Portus Installation issues*

This section provides troubleshooting information for all aspects of the system

Portus Client side issues

This section provides troubleshooting information for all aspects of the system

- **"adabasDriver::readLFdata() returned -300148, indicating Command 'LF' for dbid=1756, fnr=13: Rsp 148 (No active database was found, start the nucleus)**

  The Adabas database is not running, or cannot be accessed.

  Resolution:

  - Contact the administrator of your Adabas database to get it started.
  - Use the control centre to change the DBID that the Web Service is using.
  - Change the SVC that Portus is using to allow it to access the required DBID.
- **"adabasDriver::listGet() returned -300152, indicating Command 'L1' for dbid={dbid}, fnr={fnr}: Rsp 152 (The internal user buffer was not large enough to contain the user buffer areas) "**

  The Portus Adabas driver employs the Adabas "Multifetch" feature to reduce IPC (Inter-Process Communication) with the Adabas database server for maximum performance.

  For a LIST or SELECT request a default of 64 records to be "multifetched" is applied, an Adabas response code 152 is returned when the buffer areas required to return 64 records of the size determined by the Portus XRD for the WebService in question exceed the Adabas LU parameter.

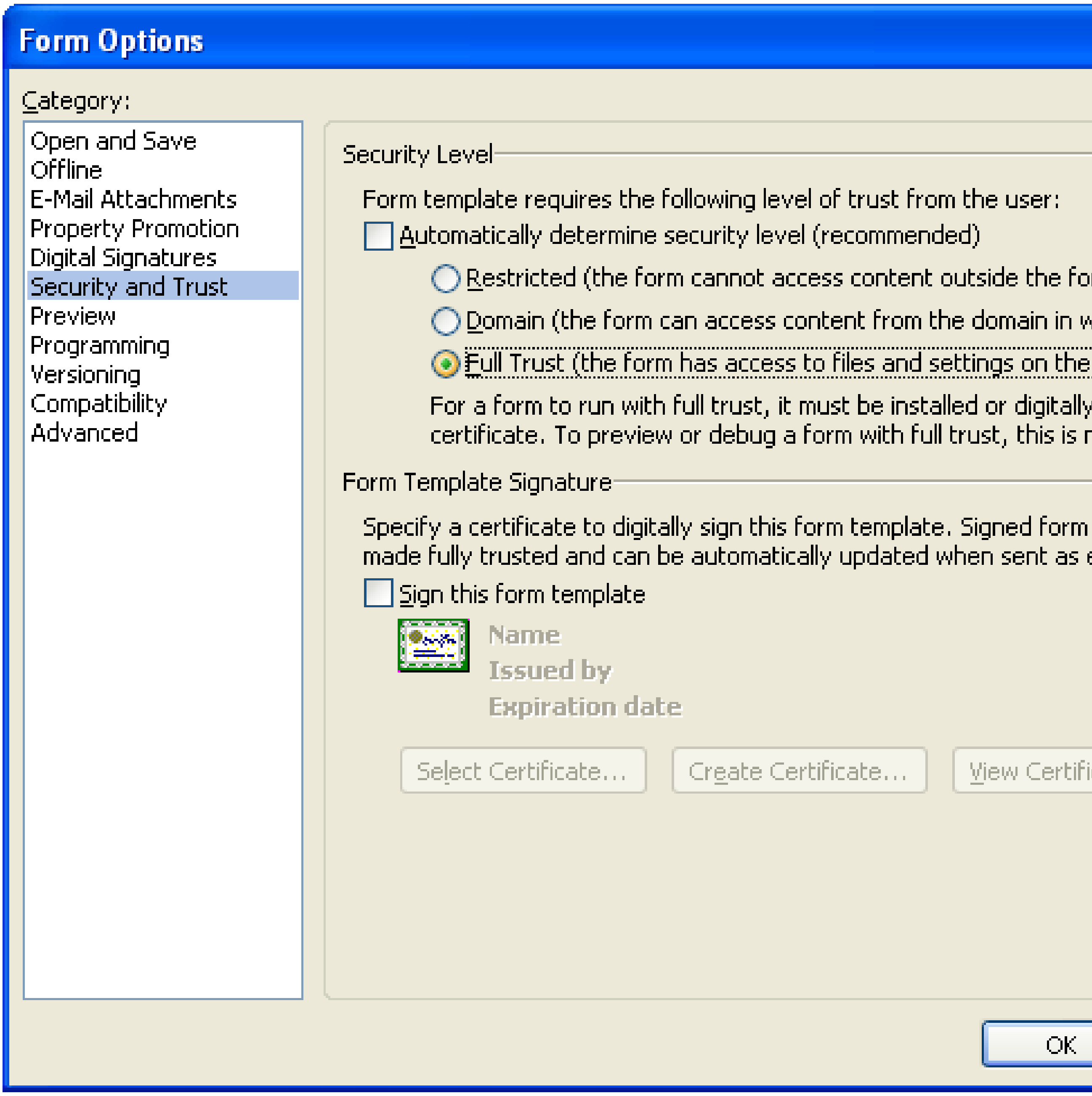
  Resolution:

  - Increase the Adabas LU parameter
  - Reduce the size of the data to be transferred per record, for example by lowering the "maxOccurs" setting for MU and/or PE fields within the XRD
  - Use the SOAP header option SOAGateway_Internal_Adabas_Multifetch_Limit to specify a value, lower than the default of 64, that results in a transfer buffer size small enough to not exhaust the LU parameter setting.
- **". . . nni_initialize rc=-2053 "**

  A -2016 means the Portus server can't access the Natural bufferpool "Unable to open buffer pool; contact the Natural system administrator." Is the bufferpool up ? Can you start a Natural session interactively ? If yes, it might be an authorization problem, i.e. the user assigned to the Portus Server not having the rights to talk to the Natural Bufferpool Server, or the Natural parameter module you are using specifying a bufferpool which isn't up.
- **Trust Issues with Microsoft Infopath**

  If you have "Problems connecting to data source" with *Microsoft Infopath* and your Portus web service, follow these instructions

  - Go **Tools, Options**, and ensure **Allow fully trusted forms to run on my computer** is checked
  - In your InfoPath form, go to **Tools, Form Options, Security and Trust** , uncheck **Automatically determine security level** and select **Full Trust**



Form Options

Category:

Security Level

Form template requires the following level of trust from the user:

□ Automatically determine security level (recommended)

○ Restricted (the form cannot access content outside the fo

○ Domain (the form can access content from the domain in w

● Full Trust (the form has access to files and settings on the

For a form to run with full trust, it must be installed or digitally certificate. To preview or debug a form with full trust, this is r

Form Template Signature

Specify a certificate to digitally sign this form template. Signed form made fully trusted and can be automatically updated when sent as e

□ Sign this form template

Name
Issued by
Expiration date

Portus Server side issues

This section provides troubleshooting information for Portus server side issues

- **Cannot load mod_xmiddle.so into server: failed to map segment from shared object: Permission denied.**

  This error can occur on Red Hat when SELinux is enabled. To disable it do the following:

  - As root, run system-config-securitylevel
  - Select SELinux Tab.
  - In the "Modify SELinux Policy" box, expand "HTTPD Service"
  - Tick the box "Disable SELinux protection for httpd daemon"
  - Click [OK]
  - Click [YES]
- **API module structure `xmiddle_module' in file XXX is garbled - perhaps this is not an Apache module DSO?**

Ensure that you have chosen the correct installation for your system. For example, this error can occur when SLES 9 files are installed on a SLES 10.

Ensure that the version of Apache you are installing on is either v2.0 or v2.2. Other versions are not supported.

- **Failed to bind to <port number>, port already in use**

  The port specified on the Listen directive in Apache's configuration file is already being used by another application. Change this to a port number not being used.
- **No error on command line, but Apache is not responding.**

  Check Apache's error_log file for error messages from the Portus server.

  The default location of the Apache error_log is [SOAG_INST]/apache2/logs/error_log replacing [SOAG_INST] with the location you installed Portus in.
- **A SOAP Fault is generated**

  Check the associated message in the <soap:faultString> element for error explanation. There may also be further messages available in the Apache's error_log.

  The default location of the Apache error_log is [SOAG_INST]/apache2/logs/error_log replacing [SOAG_INST] with the location you installed Portus in.
- **Abend S0C1 in module SAFASG**

  When the Portus server is running with Security Level = User or Password and is protected by Adabas SAF this abend indicates that SAFASG has not been linked correctly, please revisit the *Security Section* and relink SAFASG accordingly.
- **Command 'LF' for dbid=212, fnr=9, Rsp 17 (Invalid file number)**

  This error indicates you are accessing a file (number 9) that does not exist on the Adabas database (ID=212). For example, you may be running a installation verfication test, and the standard demo files ("Employees" and "Vehicles") delivered with Adabas have not been installed, or have not been installed in the standard locations.
- **Could not load natural ni library! Failed to open DLL, error 126: The specified module could not be found.**

  See setup instructions *here*.
- **Unicode char 0xF1 is not representable in encoding ASCII.**

  See information about i18n *here*
- **PRNG still contains insufficient entropy! Failed to generate temporary 512 bit RSA private key. Configuration Failed**

  Ensure the 2 following directives are enabled in your Apache configuration file (httpd.conf)

  ```
  SSLRandomSeed startup builtin
  SSLRandomSeed connect builtin
  ```

  If running on z/OS or z/VSE ensure the server is not active when submitting jobs which copy content to the Portus file system.

Portus Control Centre (Eclipse) issues

This section provides troubleshooting information for Portus server side issues

- **Eclipse unexpectedly terminates or throws "java.lang.OutOfMemoryError: PermGen space" errors**

Some memory limit defaults imposed by the Java VM might be too low, append the following parameters to (add at the end of) **eclipse.ini** (located in your top level Eclipse directory:

```
-XX:PermSize=128m
-XX:MaxPermSize=256m
```

- **Missing bundle "org.apache.axis" during Portus Control Centre install**

  Ensure the "Web, XML and Java EE Development" plugins have been installed from the Eclipse update site.

- **The Control Centre does not react as expected, functions appear to 'do nothing', but do not give any messages indicating the cause of the misbehaviour.**

  Usually at least a brief notice about any error is displayed in the Portus Log View.

  If the Log View does not give any information, or the information presented is not sufficient to identify the problem, consult the Eclipse Error Log which is accessible from the Eclipse main menu bar via:

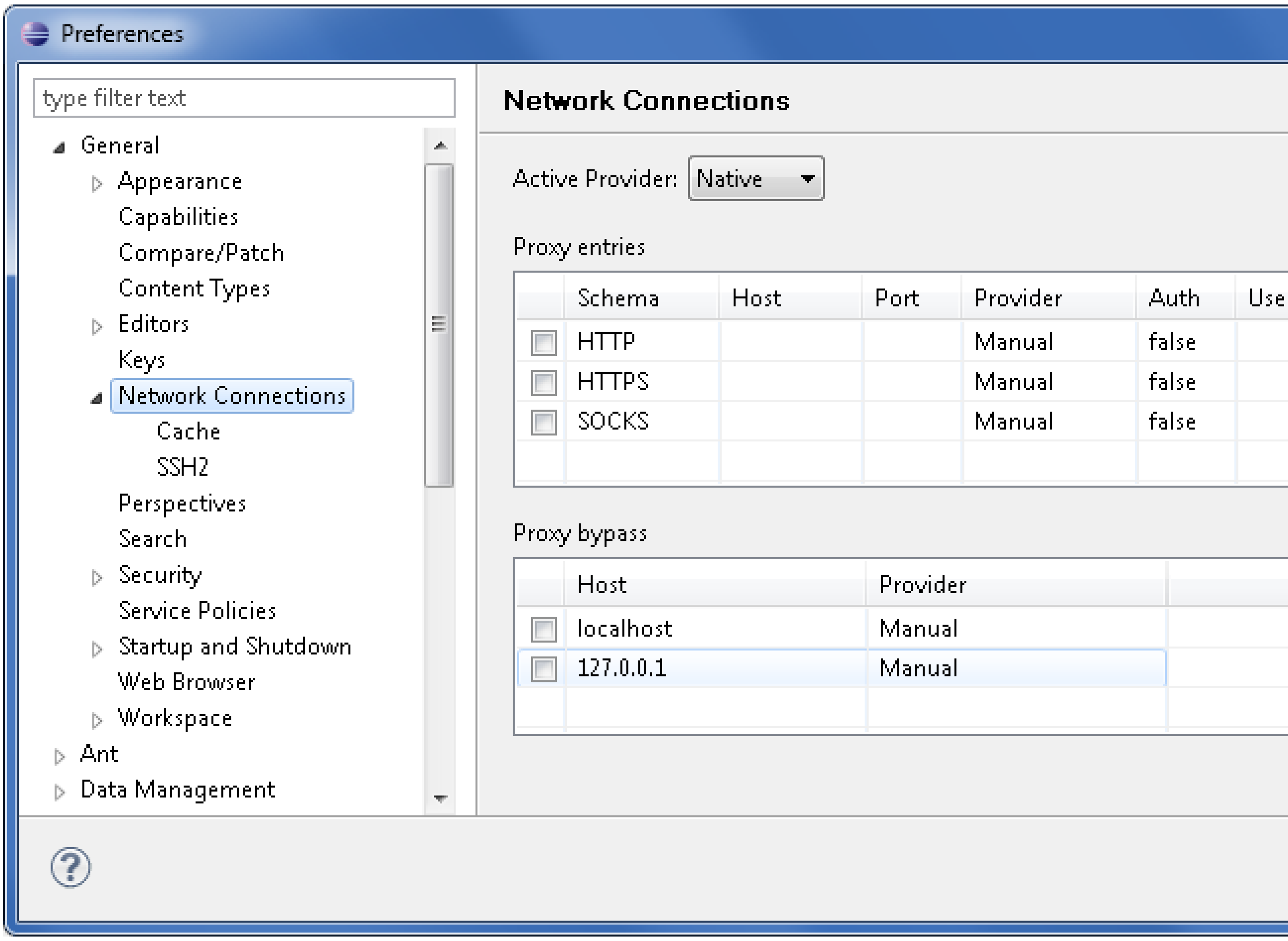
  - **Help**
  - **About Eclipse SDK** (this brings up a popup windows)
  - Click "Installation Details"
  - Select the Configuration tab
  - Click "View Error Log"
  - Scroll to the bottom of the error log to view information about the last error having occurred.

  Possible cause: please check if the JRE (Java Runtime Environment) in use it at least JRE 1.5, earlier levels may cause the behaviour described above !

- **The Eclipse update manager times out with network errors and does not get access to the update site(s)**

  When accessing the outside world from an intranet, it may be required to specify a HTTP proxy to Eclipse, this is done as follows:

  - Bring up the Preferences dialog by selecting Window -> Preferences from the Eclipse main menu.
  - Select the Network Connctions section, specify the Proxy settings as appropriate



Network Connections

Active Provider: Native

Proxy entries

	Schema	Host	Port	Provider	Auth	Use
☐	HTTP			Manual	false	
☐	HTTPS			Manual	false	
☐	SOCKS			Manual	false	

Proxy bypass

	Host	Provider
☐	localhost	Manual
☐	127.0.0.1	Manual

- Click OK to confirm the settings.

- **Font issues**

  The overall font of the Portus Control Centre can be modified by selecting

  Window, Preferences, General, Appearance, Colours and Fonts, and the **Portus** item from this. The Portus Control Centre must be restarted after modifications.

  Check Apache's error_log file for error messages from the Portus server.

  The default location of the Apache error_log is [SOAG_INST]/apache2/logs/error_log replacing [SOAG_INST] with the location you installed Portus in.

- **# A fatal error has been detected by the Java Runtime Environment**

```
# A fatal error has been detected by the Java Runtime
 Environment:
# SIGSEGV (0xb) at pc=0x9bbfc3e3, pid=13780, tid=3062007584
#
# JRE version: 6.0_29-b11
# Java VM: Java HotSpot(TM) Client VM (20.4-b02 mixed mode linux-
x86 )
# Problematic frame:
# C [libxul.so+0xb573e3] JSD_GetValueForObject+0x35319
#
```

  Include the following option in the eclipse.ini file

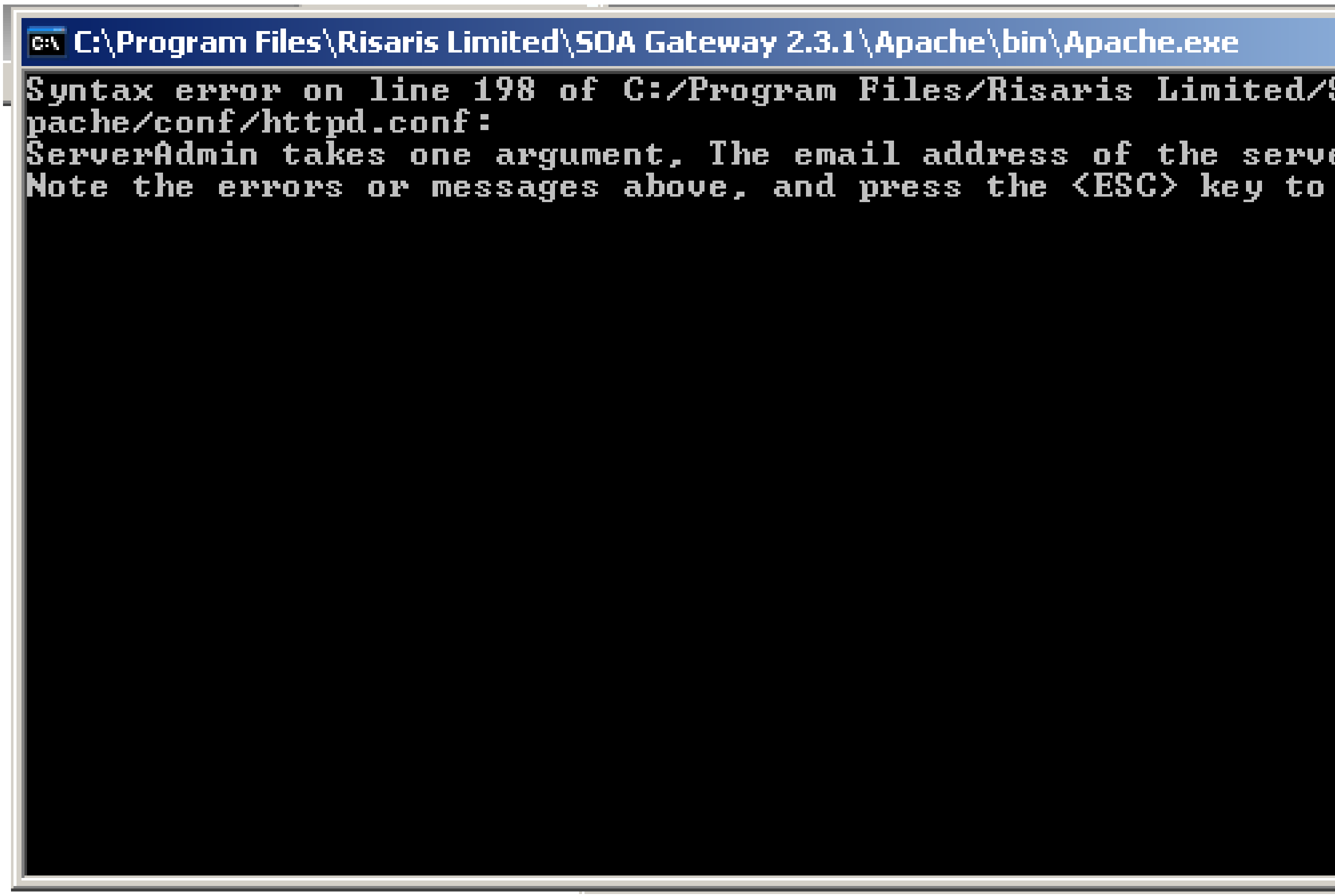
  -Dorg.eclipse.swt.browser.XULRunnerPath=/dev/null

Portus Installation issues

This section provides troubleshooting information for Portusinstallation related issues

- **Windows installation reports some files are marked to be renamed or deleted**

  Ensure that the directory C:\Program Files\Ostia\ does not exist, and restart your computer.
- **Syntax error with Server Admin**



C:\Program Files\Risaris Limited\SOA Gateway 2.3.1\Apache\bin\Apache.exe

```
Syntax error on line 198 of C:/Program Files/Risaris Limited/S
pache/conf/httpd.conf:
ServerAdmin takes one argument, The email address of the serve
Note the errors or messages above, and press the <ESC> key to
```

This error occurs when there is more than one entry in the server's DNS suffix list.

- Click **Start, Control Panel, Network Connections, Local Area Connection, Properties**
- Select **Internet Properties (TCP/IP)** and click **Properties**
- Click the **Advanced** button
- Select the **DNS** tab, and in the **Append these DNS Suffixes**, ensure there is only one entry.
- Click **OK**, **OK** and then **Close**
- In the **LAN Status** window select the **Support** tab, and click **Repair**
- Re-start the installation.

Chapter 7

Appendix

Topics:

- *Tut_02_List.java*
- *First steps (ex01_SoaGatewayFirst.php)*
- *List Employees (ex02_SoaGatewayEmpList.php)*
- *List Employees descending (ex02a_SoaGatewayEmpListDescending.php)*
- *List Employees sorted (ex02a_SoaGatewayEmpListSorted.php)*
- *List by Sub Descriptor (ex06_SoaGatewaySpecial_SubDescriptor.php)*
- *List by Super Descriptor (ex06_SoaGatewaySpecial_SuperDescriptor.php)*
- *Add an Employee (ex03_SoaGatewayEmpAdd.php)*
- *Get an Employee (ex04_SoaGatewayEmpGet.php)*
- *Delete an Employee (ex05_SoaGatewayEmpDel.php)*
- *A simple PHP form for accessing Adabas (ex10_SoaGatewaySimpleForm.php)*
- *All-in-one PHP form accessing the Employees file (ex15_SoaGatewayUpdateForm.php)*
- *empMiniList.rb (Ruby)*
- *ASGDemo.cs (C#)*
- *Get a record, save the LOB to a file (wsf_lobsGet.php)*
- *MySQL_City.cs (C#)*
- *SOAP over IBM MQ Series (SOAPEntireXTut.java)*

This appendix is a collection of various items referenced throughout the documentation, for example source files related to the tutorials etc.

Tut_02_List.java

```
import SoaG.*;
import SoaG.AdabasEmployeesMiniServiceStub.*;

/*
 *   The Portus LIST method will return all data matching the
 *   criteria at once, no matter how large the result set is.
 *
 *   So if you need (or want) to retrieve the result set in
 "chunks"
 *   of records, or your request requires complex search syntax,
 *   use the SELECT method.
 *
 *   Tut_03c_SelectConversational.java   provides an example for
 this.
 */

public class Tut_02_List {

  public static void main(String[] args) {

    try {
   AdabasEmployeesMiniServiceStub stub = new
 AdabasEmployeesMiniServiceStub();

   AdabasEmployeeKeyType keys = new AdabasEmployeeKeyType();
   keys.setPersonnel_id("50005*");

   AdabasEmployeeListElement listKey = new
 AdabasEmployeeListElement();
   listKey.setAdabasEmployeeListElement(keys);

   AdabasEmployeesMiniElement result = null;

   result = stub.list(listKey, null, null);

   AdabasEmployeesMiniElementType root =
 result.getAdabasEmployeesMiniElement();
   AdabasEmployeesMiniType group = root.getAdabasEmployeesMini();
   AdabasEmployeeType elements[] = group.getAdabasEmployee();

   System.out.println("Number of record read: " + elements.length);

   for (int i = 0; i < elements.length; i++) {
     AdabasEmployeeType r = elements[i];
     System.out.println("Record [" + i + "]"
                      + "Personnel Id=" + r.getPersonnel_id() + ",
 "
                      + "Name=" + r.getName() + ", "
                      + "First Name=" + r.getFirst_name() + ", "
                      + "City=" + r.getCity());
   }

    } catch (Exception ex) {
      ex.printStackTrace();
    }
  }

}
```

First steps (ex01_SoaGatewayFirst.php)

```
<?php

/*
 *  This set of examples demonstrate the simplicity in accessing
 Adabas
 *  data as "WebServices" from PHP, based on the "EmployeesMini"
 view
 *  representing a subset ("View") of the Adabas demo file
 "Employees".
 *
 *  This first example outlines the usage of the PHP SoapClient
 class,
 *  which provides the infrastructure for issuing SOAP requests.
 *
 *  All examples are based on an Adabas Portus server running on
 *  host "soagate", port 8082, these need to be adjusted to the
 actual
 *  server host/port used at your site.
 *
 */



/*
 *  Instantiate the PHP "SoapClient" class
 */
try {
 /*
  * The only required parameter when instantiating "SoapClient" is
 the URL
  * pointing to the WSDL for the WebService to be accessed.
  */
 $soapclient = new SoapClient("http://localhost:8022/
adabas_EmployeesMini?WSDL");
} catch (SoapFault $soapfault) {
 /*
  * In case of a SOAPFault being thrown, this will be caught and
 the fault
  * information printed. If a SOAPFault occurs outside of a "try/
catch"
  * structure, PHP will abend the script with a generic message.
  *
  * Here we print the SOAPFault information in a structured way,
 the "<pre>"
  * tags are required to format the object nicely.
  */
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Without any knowledge of, or looking at Adabas definitions, a
 "WebService programmer"
 *  can easily retrieve the signature of any exposed Adabas Portus
 Service and
 *  code based on it.
 *
 * First we use a method of the SoapClient class to retrieve the
 functions exposed
 * by the specific "WebService", and print it:
```

```
 */

echo "<pre><B>Functions:</B>\n\n";
print_r($soapclient->__getFunctions());

/*
 *  The function prototypes shown by the __getFunction() method
 also depict the required
 *  parameters and the return values. Their definitions can be
 printed as well:
 */
echo "\n
\n---------------------------------------------------------------------
\n\n<B>Type Definitions:</B>\n\n";
print_r($soapclient->__getTypes());
echo "</pre>";

/*
 *  Proceed to ex02_SoaGatewayEmpList.php - List and format
 Employees data
 */

?>
```

List Employees (ex02_SoaGatewayEmpList.php)

```
<?php

/*
 *  This example demonstrates usage of the "list" function exposed
 by any
 *  Adabas Portus "WebService" from PHP, retrieving selected
 records
 * from the Adabas demo file "Employees", and formatting it, in 4
 easy steps.
 */


/*
 *  Step 1: Instantiate the PHP "SoapClient" class
 */
try {
 $soapclient = new SoapClient("http://localhost:8022/
adabas_EmployeesMini?WSDL");
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Step 2: Build the key ("descriptor") array, due to parser
 requirements ALL
 *    key elements need to be specified, but elements may be left
 empty
 *    when unused.
 */
$listkey = array(
 'personnel_id' => '',
 'first_name' => '',
 'name' => '',
 'city' => 'CI*');

/*
 *  Step 3: Execute the "list" request, passing the key array as
 the only parameter,
 *    the response object will consist of an "adabasEmployees"
 element containing
 *    an array of "adabasEmployee" elements.
 */
try {
 $listresponse = $soapclient->list($listkey);
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Step 4: Format the Employee records nicely into a HTML table.
 */

echo "<table border=1 cellpadding=5>";
echo "<tr><th>Personnel Id</th><th>Name</th><th>First Name</
th><th>City</th><th width=200>Address Line</td>";

/*
 *  Loop through all "adabasEmployee" elements, creating a table
 row for every single one
 */

if ( isset($listresponse->adabasEmployees->adabasEmployee) )
{
 $Employees = $listresponse->adabasEmployees->adabasEmployee;
 if (!is_array($Employees))
  $Employees = $listresponse->adabasEmployees;

 foreach ($Employees as $Employee) {
  echo  "<tr><td>$Employee->personnel_id</td><td>$Employee->name</
td><td>",
   "$Employee->first_name</td><td>$Employee->city</td><td>";
  echo "<table>";

 /*
  *  Format the "address_line" element (a MU(ltiple value field))
 as a table within the table
  */
  if (!is_array($Employee->address_line)) {
   echo "<tr><td width=200>$Employee->address_line</td></tr>";
  } else {
   foreach ($Employee->address_line as $addr) {
    echo "<tr><td width=200>$addr</td></tr>";
   }
  }
 echo "</table>";
 echo "</td></tr>";
}
}
/*
 *  Proceed to ex03_SoaGatewayEmpAdd.php - Add a new employee
 record
 */
?>
```

List Employees descending (ex02a_SoaGatewayEmpListDescending.php)

```
<?php

/*
 *  This example demonstrates usage of the "list" function exposed
 by any
 *  Adabas Portus "WebService" from PHP, retrieving selected
 records
 * from the Adabas demo file "Employees", and formatting it, in 4
 easy steps.
 */


/*
 *  Step 1: Instantiate the PHP "SoapClient" class
 */
try {
 $soapclient = new SoapClient("http://localhost:8022/
adabas_EmployeesMini?WSDL");
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Step 2: Build the key ("descriptor") array, due to parser
 requirements ALL
 *    key elements need to be specified, but elements may be left
 empty
 *    when unused.
 */
$listkey = array(
 'personnel_id' => '50005*',
 'first_name' => '',
 'name' => '',
 'city' => '');

/*
 *  Step 3: Execute the "list" request, passing the key array as
 the only parameter,
 *    the response object will consist of an "adabasEmployees"
 element containing
 *    an array of "adabasEmployee" elements.
 */
try {
 $listresponse = $soapclient->list($listkey);
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

echo "List of Employees by personnel id - Ascending<br><br>";
formatResponse($listresponse);

/*
 *  Step 4: Build the SOAP Header structure to trigger a DESCENDING
 *    read instead of an ascending one.
 */

$headers = array(
 'SOAGateway_Internal_Adabas_Read_Direction' => "Descending"
 );

$header =  new SoapHeader("http://www.risaris.com/namespaces/
xmiddle",
       "adabasEmployeeHeader",
                            $headers, false);

$soapclient->__setSoapHeaders(array($header));

/*
 *  Step 5: Execute the "list" request, passing the key array as
 the only parameter,
 *    the response object will consist of an "adabasEmployees"
 element containing
 *    an array of "adabasEmployee" elements.
 */
try {
 $listresponse = $soapclient->list($listkey);
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

echo "<br><hr><br>List of Employees by personnel id -
 Descending<br><br>";
formatResponse($listresponse);

/*
 *  Sub: Format the Employee records nicely into a HTML table.
 */
function formatResponse($listresponse){

 echo "<table border=1 cellpadding=5>";
 echo "<tr><th>Personnel Id</th><th>Name</th><th>First Name</
th><th>City</th><th width=200>Address Line</td>";

 /*
  *  Loop through all "adabasEmployee" elements, creating a table
 row for every single one
  */
if ( isset($listresponse->adabasEmployees->adabasEmployee) )
{
 $Employees = $listresponse->adabasEmployees->adabasEmployee;
 if (!is_array($Employees))
  $Employees = $listresponse->adabasEmployees;

 foreach ($Employees as $Employee) {
  echo "<tr><td>$Employee->personnel_id</td><td>$Employee->name</
td><td>$Employee->first_name</td><td>$Employee->city</td><td>";
  echo "<table>";

  if (!is_array($Employee->address_line)) {
   echo "<tr><td width=200>$Employee->address_line</td></tr>";
  } else {
   foreach ($Employee->address_line as $addr) {
    echo "<tr><td width=200>$addr</td></tr>";
   }
  }
  echo "</table>";
  echo "</td></tr>";
 }

}

echo "</table>";
}
/*
 *  Proceed to ex03_SoaGatewayEmpAdd.php - Add a new employee
 record
 */
?>
```

List Employees sorted (ex02a_SoaGatewayEmpListSorted.php)

```
<?php

/*
 *  This example demonstrates usage of the "list" function exposed
 by any
 *  Adabas Portus "WebService" from PHP, retrieving selected
 records
 * from the Adabas demo file "Employees", and formatting it, in 4
 easy steps.
 */


/*
 *  Step 1: Instantiate the PHP "SoapClient" class
 */
try {
 $soapclient = new SoapClient("http://localhost:8022/
adabas_EmployeesMini?WSDL");
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Step 2: Build the key ("descriptor") array, due to parser
 requirements ALL
 *    key elements need to be specified, but elements may be left
 empty
 *    when unused.
 */
$listkey = array(
 'personnel_id' => '50005*',
 'first_name' => '',
 'name' => '',
 'city' => '');

/*
 *  Step 3: Execute the "list" request, passing the key array as
 the only parameter,
 *    the response object will consist of an "adabasEmployees"
 element containing
 *    an array of "adabasEmployee" elements.
 */
```

```
try {
 $listresponse = $soapclient->list($listkey);
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

echo "List of Employees by personnel id = 50005* - Default sort
 order<br><br>";
formatResponse($listresponse);

/*
 *  Step 4: Build the SOAP Header structure to trigger a DESCENDING
 *    read instead of an ascending one.
 */

$headers = array(
 'SOAGateway_Internal_Adabas_Sort_Order' => "city"
 );

$header =  new SoapHeader("http://www.risaris.com/namespaces/
xmiddle",
       "adabasEmployeeHeader",
                            $headers, false);

$soapclient->__setSoapHeaders(array($header));

/*
 *  Step 5: Execute the "list" request, passing the key array as
 the only parameter,
 *    the response object will consist of an "adabasEmployees"
 element containing
 *    an array of "adabasEmployee" elements.
 */
try {
 $listresponse = $soapclient->list($listkey);
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

echo "<br><hr><br>List of Employees by personnel id = 50005* - now
 sorted by City<br><br>";
formatResponse($listresponse);

/*
 *  Sub: Format the Employee records nicely into a HTML table.
 */
function formatResponse($listresponse){

 echo "<table border=1 cellpadding=5>";
 echo "<tr><th>Personnel Id</th><th>Name</th><th>First Name</
th><th>City</th><th width=200>Address Line</td>";

 /*
  *  Loop through all "adabasEmployee" elements, creating a table
 row for every single one
  */
if ( isset($listresponse->adabasEmployees->adabasEmployee) )
{
 $Employees = $listresponse->adabasEmployees->adabasEmployee;
 if (!is_array($Employees))
  $Employees = $listresponse->adabasEmployees;

 foreach ($Employees as $Employee) {
  echo "<tr><td>$Employee->personnel_id</td><td>$Employee->name</
td><td>$Employee->first_name</td><td>$Employee->city</td><td>";
  echo "<table>";

  if (!is_array($Employee->address_line)) {
   echo "<tr><td width=200>$Employee->address_line</td></tr>";
  } else {
   foreach ($Employee->address_line as $addr) {
    echo "<tr><td width=200>$addr</td></tr>";
   }
  }
  echo "</table>";
  echo "</td></tr>";
 }

}

echo "</table>";
}
/*
 *  Proceed to ex03_SoaGatewayEmpAdd.php - Add a new employee
 record
 */
?>
```

List by Sub Descriptor (ex06_SoaGatewaySpecial_SubDescriptor.php)

```
<?php

/*
 *  This example demonstrates usage of the "list" function exposed
 by any
 *  Adabas Portus "WebService" from PHP, retrieving selected
 records
 * from the Adabas demo file "Employees", and formatting it, in 4
 easy steps.
 */


/*
 *  Step 1: Instantiate the PHP "SoapClient" class
 */
try {
 $soapclient = new SoapClient("http://localhost:8022/
adabas_Employees_special?WSDL");
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Step 2: Build the key ("descriptor") array, due to parser
 requirements ALL
 *    key elements need to be specified, but elements may be left
 empty
 *    when unused. Here we are using the SUB-descriptor
 "department".
 */
$listkey = array(
        'personnel_id' => "",
        'first_name'    => "",
        'name'                  => "",
        'birthday'              => "",
        'city'                  => "",
        'dept'=> "",
        'job_title'             => "",
        'language_spoken'=>"",
        'leave_left'    => "",
        'dept_person'   =>  "",
        'currency_salary'=> "",
        'department'    => array('dept' => "MGMT"),
        'phonetic_name' => "");

/*
 *  Step 3: Execute the "list" request, passing the key array as
 the only parameter,
 *    the response object will consist of an "adabasEmployees"
 element containing
 *    an array of "adabasEmployee" elements.
 */
try {
 $listresponse = $soapclient->list($listkey);
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Step 4: Format the Employee records nicely into a HTML table.
 */

echo "Find all Employees in department group \"MGMT\" - using an
 Adabas SUB-Descriptor<br><br>";
echo "<table border=1 cellpadding=5>";
echo "<tr><th>Personnel Id</th><th>Name</th><th>First Name</
th><th>City</th><th>Department</td>";

/*
 *  Loop through all "adabasEmployee" elements, creating a table
 row for every single one
 */
foreach ($listresponse->adabasDemoEmployeesSpecial-
>demoEmployeeSpecial as $Employee) {
 echo  "<tr><td>$Employee->personnel_id</td><td>$Employee->name</
td><td>",
   "$Employee->first_name</td><td>$Employee->city</td><td>
$Employee->department_code</td>";
 echo "</td></tr>";
}

/*
 *  Proceed to ex03_SoaGatewayEmpAdd.php - Add a new employee
 record
 */
?>
```

List by Super Descriptor (ex06_SoaGatewaySpecial_SuperDescriptor.php)

```
<?php

/*
 *  This example demonstrates usage of the "list" function exposed
 by any
 *  Adabas Portus "WebService" from PHP, retrieving selected
 records
 * from the Adabas demo file "Employees", and formatting it, in 4
 easy steps.
 */


/*
 *  Step 1: Instantiate the PHP "SoapClient" class
 */
try {
 $soapclient = new SoapClient("http://localhost:8022/
adabas_Employees_special?WSDL");
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Step 2: Build the key ("descriptor") array, due to parser
 requirements ALL
 *    key elements need to be specified, but elements may be left
 empty
 *    when unused. Here we are using the SUB-descriptor
 "department".
 */
$listkey = array(
 'personnel_id' => "",
 'first_name' => "",
 'name'   => "",
 'birthday'  => "",
 'city'   => "",
 'dept'=> "",
 'job_title'  => "",
 'language_spoken'=>"",
 'leave_left' => "",
 'dept_person' => array('dept' => "MGMT10", 'name' => "K*"),
 'currency_salary'=> "",
 'department' => "",
 'phonetic_name' => "");

/*
 *  Step 3: Execute the "list" request, passing the key array as
 the only parameter,
 *    the response object will consist of an "adabasEmployees"
 element containing
 *    an array of "adabasEmployee" elements.
 */
try {
 $listresponse = $soapclient->list($listkey);
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Step 4: Format the Employee records nicely into a HTML table.
 */

echo "Find all Employees in department group 'MGMT10' whose names
 start with 'K' - using an Adabas SUPER-Descriptor<br><br>";
echo "<table border=1 cellpadding=5>";
echo "<tr><th>Personnel Id</th><th>Name</th><th>First Name</
th><th>City</th><th>Department</td>";

/*
 *  Loop through all "adabasEmployee" elements, creating a table
 row for every single one
 */
foreach ($listresponse->adabasDemoEmployeesSpecial-
>demoEmployeeSpecial as $Employee) {
 echo  "<tr><td>$Employee->personnel_id</td><td>$Employee->name</
td><td>",
   "$Employee->first_name</td><td>$Employee->city</td><td>
$Employee->department_code</td>";
 echo "</td></tr>";
}

/*
 *  Proceed to ex03_SoaGatewayEmpAdd.php - Add a new employee
 record
 */
?>
```

Add an Employee (ex03_SoaGatewayEmpAdd.php)

```
<?php

/*
 *  We now add a new Employee record, which is just as simple.
 */

try{
 $soapclient = new SoapClient("http://localhost:8022/
adabas_EmployeesMini?WSDL");
} catch (SoapFault $soapfault) {
 printSoapFault($soapclient, $soapfault);
 return;
}

/*
 *  Constructing the "data record" is similar to building the key
 array
 *  for a "list" operation, MUs are represented by an array within
 the
 *  array.
 */
$adabasEmployee = array (
 'personnel_id' => '99999999',
 'first_name'  => 'Kirk',
 'name'   => 'Newlyadded',
 'city'    => 'City',
 'address_line' => array ('route 66', 'from here', 'to there',
 'CA')
);

/*
 *  The expected structure is equivalent to the one returned by
 "list",
 *  thus we need to create an array of "adabasEmployee" elements,
 even
 *  though there is just one:
 */
$adabasEmployees = array($adabasEmployee);

/*
 *  Now add the Employee
 */
try {
 $Adabasresponse = $soapclient->add($adabasEmployees);
} catch (SoapFault $soapfault) {
 printSoapFault($soapclient, $soapfault);
 return;
}

/*
 *  An "add" results in a "short response", print the message:
 */
echo "<pre>result: $Adabasresponse->results</pre>";

/*
 *  The SOAPFault is handled in a function:
 */
function printSoapFault ($soapclient, $soapfault) {
 echo "<pre>";
 echo "\n\nSoap Fault occurred\n\nFaultCode..: ".$soapfault-
>faultcode."\nFaultString: ".$soapfault->faultstring;
 echo "</pre>";
}

/*
 *  Proceed to ex04_SoaGatewayEmpGet.php - Get and display the
 newlyadded employee record
 */
?>
```

Get an Employee (ex04_SoaGatewayEmpGet.php)

```
<?php

/*
 *  The "get" operation is even easier.
 */
try{
 $soapclient = new SoapClient("http://localhost:8022/
adabas_EmployeesMini?WSDL", array('trace' => 1));
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
}

/*
 *  Construct the "key array", which consists of just one element,
 *  the primary key for the employees file - "personnel_id";
 */
$primKey = array('personnel_id' => '99999999');

/*
 *  Get the record
 */
try {
 $Adabasresponse = $soapclient->get($primKey);
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Check if we actually got the record we are looking for
 */
if (is_null($Adabasresponse->adabasEmployees->adabasEmployee)) {
 echo "No Employee with personnel_id=".$primKey['personnel_id'];
 return;
}

/*
 *  Print the formatted response
 */
echo "<pre>";
print_r($Adabasresponse);
echo "</pre>";

/*
 *  Proceed to ex05_SoaGatewayEmpDel.php - Delete the record added
 in ex03_SoaGatewayEmpAdd.php
 */
?>
```

Delete an Employee (ex05_SoaGatewayEmpDel.php)

```
<?php

/*
 *  Finally we delete the Employee record with
 personnel_id=99999999 again
 */

try{
 $soapclient = new SoapClient("http://localhost:8022/
adabas_EmployeesMini?WSDL", array('trace' => 1));
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
}

/*
 *  All we need is the primary key
 */
$primKey = array('personnel_id' => '99999999');

/*
```

```
 *  Delete takes the key as the input and returns a "short
 response" (just a message)
 */
try {
 $Adabasresponse = $soapclient->delete($primKey);
} catch (SoapFault $soapfault) {
 echo "<pre>";
 print_r($soapfault);
 echo "</pre>";
 return;
}

/*
 *  Print the response
 */
echo "<pre>";
print_r($Adabasresponse);
echo "</pre>";
?>
```

A simple PHP form for accessing Adabas (ex10_SoaGatewaySimpleForm.php)

```
<?php
/*
 *  On entry to the form determine if the "List Employees" button
 has been pressed,
 *  if this is the case, retrieve the form field values.
 */
if (isset($_POST['submit'])) {
 $Fname  = $_POST["Fname"];
 $Lname  = $_POST["Lname"];
 $Persid = $_POST["Persid"];
 $City   = $_POST["City"];
}
?>
<html>
<head>
<title>Personnel Info</title>
</head>
<body>
<font face="courier">
<form method="post" action="<?php echo $PHP_SELF;?>">
Personnel Id: <input type="text" size="8" maxlength="8"
 name="Persid" value="<? echo $Persid; ?>"><br />
First Name..: <input type="text" size="20" maxlength="20"
 name="Fname"  value="<? echo $Fname; ?>"><br />
Last Name...: <input type="text" size="20" maxlength="20"
 name="Lname"  value="<? echo $Lname; ?>"><br />
City........: <input type="text" size="20" maxlength="20"
 name="City"   value="<? echo $City; ?>"><br />

<br/>
<input type="submit" value="List Employees" name="submit">
</form>
<?
/*
 *  If the "List Employees" button has been pressed, retrieve the
 Employees record(s) and format
 *  them into a HTML table. The code used here is, with the
 exception of the variables used for
 *  building the keys array, equivalent to
 ex02_SoaGatewayEmpList.php
 */
if (isset($_POST['submit'])) {

 echo "Selected: Personnel Id=''".$Persid."'', first_name=''".
$Fname."'',Name=''".$Lname;
 echo "'', City=''".$City."''<br/><br/>";

$soapclient = new SoapClient("http://localhost:8022/
adabas_EmployeesMini?WSDL");

$key = array(
 'personnel_id' => $Persid,
 'first_name'  => $Fname,
 'name'   => $Lname,
 'city'   => $City
);

$result = $soapclient->list($key);

echo "<table border=1 cellpadding=5>";
echo "<tr><th>Personnel Id</th><th>Name</th><th>first Name</
th><th>City</td><td width=200>Address</td></tr>";

if ( isset($result->adabasEmployees->adabasEmployee) )
{
 $Employees = $result->adabasEmployees->adabasEmployee;
 if (!is_array($Employees))
  $Employees = $result->adabasEmployees;

 foreach ($Employees as $Employee) {
  echo "<tr><td>$Employee->personnel_id</td><td>$Employee->name</
td><td>$Employee->first_name</td><td>$Employee->city</td><td>";
  echo "<table>";

  if (!is_array($Employee->address_line)) {
   echo "<tr><td width=200>$Employee->address_line</td></tr>";
  } else {
   foreach ($Employee->address_line as $addr) {
    echo "<tr><td width=200>$addr</td></tr>";
   }
  }
  echo "</table>";
  echo "</td></tr>";
 }

}

echo "</table>";

}
?>

</body></html>
```

All-in-one PHP form accessing the Employees file (ex15_SoaGatewayUpdateForm.php)

```
<?php
/*
 *  This form incorporates ALL Adabas Portus access methods,
 *     list, get, add, update, delete
 *
 *  and demonstrates how easily web applications can be implemented
 *     based on the Adabas Portus
 */

global $PHP_SELF;

$Persid = "";
$Fname = "";
$Lname = "";
$City = "";
$Addr[0] = "";
$Addr[1] = "";
$Addr[2] = "";
$Addr[3] = "";
$msg = "";

if (isset($_POST['submit'])) {

 if ($_POST['submit'] != "Reset") {
  $Fname  = $_POST["Fname"];
  $Lname  = $_POST["Lname"];
  $Persid = $_POST["Persid"];
  $City   = $_POST["City"];
  $Addr = $_POST["Addr"];

  try {
   $soapclient = new SoapClient(
    "http://localhost:8022/adabas_EmployeesMini?WSDL");
  } catch (SoapFault $soapfault) {
   echo "<pre>";
   print_r($soapfault);
   echo "</pre>";
   return;
  }
 }

 $msg = "";

  if ($_POST['submit'] == "Delete") {

 try {
  $Adabasresponse = $soapclient->delete(array('personnel_id' =>
 $Persid));
 } catch (SoapFault $soapfault) {
  echo "<pre>";

  print_r($soapfault);
  echo "</pre>";
  return;
 }
 $msg = $Adabasresponse->results;
  }

  if ($_POST['submit'] == "Get") {

 try {
  $Adabasresponse = $soapclient->get(array('personnel_id' =>
 $Persid));
 } catch (SoapFault $soapfault) {
  echo "<pre>";
  print_r($soapfault);
  echo "</pre>";
  return;
 }

 if (!isset($Adabasresponse->adabasEmployees->adabasEmployee)) {
  $msg = "No Employee with personnel_id=".$Persid;
  $Lname = "";
  $Fname = "";
  $City = "";
  $Addr = array("", "", "", "");
 } else {
  $Employee = $Adabasresponse->adabasEmployees->adabasEmployee;

  $Persid = $Employee->personnel_id;
  $Lname = $Employee->name;
  $Fname = $Employee->first_name;
  $City = $Employee->city;
  $Addr = $Employee->address_line;
 }
  }

  if (($_POST['submit'] == "Add") || ($_POST['submit'] ==
 "Update")) {

 $adabasEmployee = array (
  'personnel_id' => $Persid,
  'first_name'  => $Fname,
  'name'   => $Lname,
  'city'    => $City,
  'address_line' => $Addr
 );

 $adabasEmployees = array($adabasEmployee);

 if ($_POST['submit'] == "Add") {
  try {
   $Adabasresponse = $soapclient->add($adabasEmployees);
  } catch (SoapFault $soapfault) {
   echo "<pre>";
   print_r($soapfault);
   echo "</pre>";
  }
 } else {
  try {
   $Adabasresponse = $soapclient->update($adabasEmployees);
  } catch (SoapFault $soapfault) {
   echo "<pre>";
   print_r($soapfault);
   echo "</pre>";
  }
 }
  }
} else {
}
?>
<html>
<head>
<title>Personnel Info</title>

</head>
<body>
<font face="courier">
<form method="post" action="<?php echo $PHP_SELF;?>">
Personnel Id: <input type="text" size="8" maxlength="8"
 name="Persid" value="<? echo $Persid; ?>"><br />
```

```
First Name..: <input type="text" size="20" maxlength="20"
 name="Fname"  value="<? echo $Fname; ?>"><br />
Last Name...: <input type="text" size="20" maxlength="20"
 name="Lname"  value="<? echo $Lname; ?>"><br />
City........: <input type="text" size="20" maxlength="20"
 name="City"   value="<? echo $City; ?>"><br />
Address.....: <input type="text" size="20" maxlength="20"
 name="Addr[]"   value="<? echo $Addr[0]; ?>"><br />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;.....: <input type="text"
 size="20" maxlength="20" name="Addr[]"   value="<? echo
 $Addr[1]; ?>"><br />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;.....: <input type="text"
 size="20" maxlength="20" name="Addr[]"   value="<? echo
 $Addr[2]; ?>"><br />

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;.....: <input type="text"
 size="20" maxlength="20" name="Addr[]"   value="<? echo
 $Addr[3]; ?>"><br />
<br/>
<? echo $msg; ?><br><br>
<input type="submit" value="List" name="submit">
<input type="submit" value="Get"  name="submit">
<input type="submit" value="Add" name="submit">
<input type="submit" value="Update" name="submit">
<input type="submit" value="Delete" name="submit">
<input type="submit" value="Reset" name="submit">
</form>
<?
if (isset($_POST['submit'])) {

  if ($_POST['submit'] == 'List') {
 echo "Selected: Personnel Id=''".$Persid."'', first_name=''".
$Fname."'',Name=''".$Lname;
 echo "'', City=''".$City."''<br/><br/>";

 $listkey = array('personnel_id' => $Persid, 'first_name' =>
 $Fname, 'name' => $Lname, 'city' => $City);

 try {
  $listresponse = $soapclient->list($listkey);
 } catch (SoapFault $soapfault) {
  echo "<pre>";
  print_r($soapfault);
  echo "</pre>";
  return;
 }

 echo "<table border=1 cellpadding=5>";
 echo "<tr><th>Personnel Id</th><th>Name</th><th>First Name</
th><th>City</th><th width=200>Address Line</td>";

 if ( isset($listresponse->adabasEmployees->adabasEmployee) )
 {
  $Employees = $listresponse->adabasEmployees->adabasEmployee;
  if (!is_array($Employees))
   $Employees = $listresponse->adabasEmployees;

  foreach ($Employees as $Employee) {
   echo "<tr><td>$Employee->personnel_id</td><td>$Employee->name</
td><td>$Employee->first_name</td><td>$Employee->city</td><td>";
   echo "<table>";
   if (!is_array($Employee->address_line)) {
    echo "<tr><td width=200>$Employee->address_line</td></tr>";
   } else {
    foreach ($Employee->address_line as $addr) {
     echo "<tr><td width=200>$addr</td></tr>";
    }
   }
   echo "</table>";
   echo "</td></tr>";
  }

 }
  }

}
?>

</body></html>
```

empMiniList.rb (Ruby)

```
require 'soap/wsdlDriver'

wsdl_url = "http://soagate:8023/adabas_EmployeesMini?WSDL"

soap = SOAP::WSDLDriverFactory.new( wsdl_url ).create_rpc_driver

soap.wiredump_file_base = "soapresult"

param = {"personnel_id" => "50005*", "name" => "", "city" => ""}

result = soap.list( param )

print( "\nNumber of Employees is ",
 result.adabasEmployees.adabasEmployee.length, "\n\n")

for emp in (result.adabasEmployees.adabasEmployee)

  print(emp.personnel_id, ", ", emp.name, " ", emp.first_name,
 "\n")

end
```

ASGDemo.cs (C#)

```
using System;
using System.Collections.Generic;
using System.Text;

namespace EmployeesList
{
    class Program
    {
        static void Main(string[] args)
        {
            adabasEmployeesService emplService;
            adabasEmployeeKeyType keys;
            adabasEmployeesElementType result;
            adabasEmployeeType[] empl;

            keys = new adabasEmployeeKeyType();
            keys.personnel_id = "300000*";
            emplService = new adabasEmployeesService();

            try
            {
                result = emplService.list(keys);
            }
            catch (SystemException ex)
            {
                Console.WriteLine("exception: " + ex.Message);
                return;
            }

            empl = result.adabasEmployees;

            if (empl.Length > 0)
            {
                Console.WriteLine("Number of Employees returned: "
 + empl.Length);
            }

            for (int i = 0; i < empl.Length; i++)
            {
                Console.WriteLine("Record [" + i + "],
 Personnel_Id=" + empl[i].personnel_id +
                    ", Name=" + empl[i].name + ", First_Name=" +
 empl[i].first_name);
            }

        }
    }
}
```

Get a record, save the LOB to a file (wsf_lobsGet.php)

Get a record from the Adabas demo file and save the LOB to a file, the extension is determined from the mime-type stored on the file,

```
<?php

$id = $_GET['id'];

$reqPayloadString = <<<XML
<emp:EmployeePhotoGetElement xmlns:emp="com.SOAGateway/
EmployeePhoto">
<personnel_id>$id</personnel_id>
</emp:EmployeePhotoGetElement
> XML;

try {

$client = new WSClient(
array("to"=>"http://localhost:56000/adabas_blobs",
"useMTOM"=>TRUE,
"responseXOP"=>TRUE));

$reqMessage = new WSMessage($reqPayloadString);
$resMessage = new WSMessage('');

$resMessage = $client->request($reqMessage);

printf("Response = %s \n\n", $resMessage->str);

$cid2stringMap = $resMessage->attachments;
$cid2contentMap = $resMessage->cid2contentType;
$imageName;

if($cid2stringMap && $cid2contentMap){
foreach($cid2stringMap as $i=>$value){
$f = $cid2stringMap[$i];
$contentType = $cid2contentMap[$i];

if(strcmp($contentType,"image/pjpeg") ==0){
$imageName = "C:\\TEMP\\".$i."."."jpg";
file_put_contents($imageName, $f);
echo "<pre>File saved as ".$imageName."</pre>";
}

if(strcmp($contentType,"image/gif") ==0){
$imageName = "C:\\TEMP\\".$i."."."gif";
file_put_contents($imageName, $f);
echo "<pre>File saved as ".$imageName."</pre>";
}

if(strcmp($contentType,"audio/mpeg") ==0){
$imageName = "C:\\TEMP\\".$i."."."mp3";
file_put_contents($imageName, $f);
echo "<pre>File saved as ".$imageName."</pre>";
}

}

}else{
printf("attachments were not found ");
}

} catch (Exception $e) {
if ($e instanceof WSFault) {
printf("Soap Fault: %s\n", $e->Reason);
} else {
printf("Message = %s\n",$e->getMessage());
}


echo "<pre>";
print_r($e);
print_r($reqMessage);
print_r($resMessage);
echo "</pre>";
}
```

MySQL_City.cs (C#)

```
using System;
using System.Collections.Generic;
using System.Text;

namespace CityList
{
    class Program
    {
        static void Main(string[] args)
        {
            RootElementNameService cityService;
            GroupElementNameKeyType key;
            RootElementNameElementType result;
            GroupElementNameType[] aCity;

            key = new GroupElementNameKeyType();
            key.ID = "305*";

            cityService = new RootElementNameService();

            try
            {
                result = cityService.list(key);
            }
            catch (SystemException ex)
            {
                Console.WriteLine("exception: " + ex.Message);
                return;
            }

            aCity = result.RootElementName;

            if (aCity.Length > 0)
            {
                Console.WriteLine("Number of Cities returned: " +
 aCity.Length);
            }

            for (int i = 0; i < aCity.Length; i++)
            {
                Console.WriteLine("City [" + i + "], ID=" +
 aCity[i].ID
                    + " Name = " + aCity[i].Name
                    + " CountryCode = " + aCity[i].CountryCode
                    + " District = " + aCity[i].District
                    + " Population = " + aCity[i].Population );
            }
        }
    }
}
```

SOAP over IBM MQ Series (SOAPEntireXTut.java)

```
package SoaGW;

import org.apache.axis2.description.TransportOutDescription;

import com.risaris.axis2.entirex.transports.ExxTransportReceiver;
import com.risaris.axis2.entirex.transports.ExxTransportSender;

import SoaGW.AdabasEmployeesServiceStub.AdabasEmployeeKeyType;
import
 SoaGW.AdabasEmployeesServiceStub.AdabasEmployeeListReceiveElement;
import
 SoaGW.AdabasEmployeesServiceStub.AdabasEmployeeListSendElement;
import SoaGW.AdabasEmployeesServiceStub.AdabasEmployeeType;
import SoaGW.AdabasEmployeesServiceStub.AdabasEmployeesElement;


public class SOAPEntireXTut {

public static void main(String[] args) throws Exception, Fault {

AdabasEmployeesServiceStub stub = new AdabasEmployeesServiceStub();
```

```
TransportOutDescription txOut = new
 TransportOutDescription("http");
ExxTransportSender sender = new ExxTransportSender();
txOut.setSender(sender);
stub._getServiceClient().getOptions().setTransportOut(txOut);

AdabasEmployeeKeyType key = new AdabasEmployeeKeyType();

key.setPersonnel_id("11100107");
key.setName("");
key.setCity("");

AdabasEmployeeListSendElement lkey = new
 AdabasEmployeeListSendElement();
lkey.setAdabasEmployeeListSendElement( key );

stub.list_send( lkey , null, null);

Object convState1 = sender.getConvState();

key.setPersonnel_id("11100108");
key.setName("");
key.setCity("");

lkey.setAdabasEmployeeListSendElement( key );

stub.list_send( lkey , null, null);

Object convState2 = sender.getConvState();

key.setPersonnel_id("11100109");
key.setName("");
key.setCity("");

lkey.setAdabasEmployeeListSendElement( key );

stub.list_send( lkey , null, null);

Object convState3 = sender.getConvState();

AdabasEmployeeListReceiveElement input = new
 AdabasEmployeeListReceiveElement();

TransportOutDescription txOutR = new
 TransportOutDescription("http");
ExxTransportReceiver receiver = new ExxTransportReceiver();
receiver.setService( sender.getService() ); // This sets up the
 receiver to use the same logon and service as the sender
txOutR.setSender( receiver );
stub._getServiceClient().getOptions().setTransportOut(txOutR);

//
// Set the conversation state which we retrieved from the sender.
 We wish
// to get the response from our 2nd request i.e. 11100108 so use
// convState2.
//

receiver.setConvState(convState2);

AdabasEmployeesElement results = stub.list_receive(input, null,
 null);
```

```
AdabasEmployeeType [] res =
 results.getAdabasEmployeesElement().getAdabasEmployees().getAdabasEmployee()

for (AdabasEmployeeType emp : res)
{
System.out.println( "Employee: " + emp.getPersonnel_id() + " " +
 emp.getFirst_name() + " " + emp.getCity() );
}

//
// Set the conversation state which we retrieved from the sender.
 We wish
// to get the response from our 3rd request i.e. 11100109 so use
// convState3.
//

receiver.setConvState(convState3);

results = stub.list_receive(input, null, null);

res =
 results.getAdabasEmployeesElement().getAdabasEmployees().getAdabasEmployee()

for (AdabasEmployeeType emp : res)
{
System.out.println( "Employee: " + emp.getPersonnel_id() + " " +
 emp.getFirst_name() + " " + emp.getCity() );
}

//
// Set the conversation state which we retrieved from the sender.
 We wish
// to get the response from our 1st request i.e. 11100107 so use
// convState1.
//

receiver.setConvState(convState1);

results = stub.list_receive(input, null, null);

res =
 results.getAdabasEmployeesElement().getAdabasEmployees().getAdabasEmployee()

for (AdabasEmployeeType emp : res)
{
System.out.println( "Employee: " + emp.getPersonnel_id() + " " +
 emp.getFirst_name() + " " + emp.getCity() );
}

sender.logoff();
}
}
```